



ເອກສານປະກອບກາຮັດ
ຮາຍວິຊາ ຕະໂຄໂຄນໄຕ ເຄຣີ່ຂໍ້ມູນຄອມພິວເຕອົງ
(Computer Networks)

ທຽມຖ້ວີ ກິຕີຄຣີວິເວັບພັນຈຸ
ວ.ສ.ປ., ວ.ສ.ມ., ປ.ດ.(ວິສວກຮມຄອມພິວເຕອົງ)

ເອກສານປະກອບກາຮັດນີ້ສໍາຫັກກາຮັດເສັນອອກກຳນົດຕຳແໜ່ງທາງວິຊາກາຮັດ
ສາຂາວິຊາວິສວກຮມຄອມພິວເຕອົງ
ມາຮວິທາລ້ຽນຄຣົມ

ປັບປຸງລ່າສຸດ ແລ້ວ



ເອກສາຣປະກອບກາຮສອນ
ຮາຍວິຊາ ຕະໂຄໂຄນໄຈ ເຄື່ອງຂໍ້າຍຄອມພິວເຕອີ່
(Computer Networks)

ທຽງຖາວີ່ ກິຕີຄຣີວຽກພັນຈຸ່
ວ.ສ.ປ., ວ.ສ.ມ., ປ.ດ.(ວິສວກຮມຄອມພິວເຕອີ່)

ເອກສາຣປະກອບກາຮສອນນີ້ສໍາຫັກກາຮເສັນອອກກຳນົດຕຳແໜ່ງທາງວິຊາກາຮ
ສາຂາວິຊາວິສວກຮມຄອມພິວເຕອີ່
ມາວິທຍາລ້ຽນຄຣົມ

ປັບປຸງລ່າສຸດ ແລ້ວ

คำนำ

เอกสารฉบับนี้ เป็นเอกสารประกอบการสอนเพื่อใช้ในการเตรียมและวางแผนการสอน รายวิชา เครื่องข่ายคอมพิวเตอร์(๓๑๑๓๓๑) สำหรับสอนนักศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ โดยมีเนื้อหาสาระเกี่ยวกับ

ระบบเครื่องข่ายคอมพิวเตอร์เบื้องต้น แบบจำลองโอลอสไอ ชั้นโปรแกรมประยุกต์ เอชทีพี เอสเอ็มทีพี ดีอี็นเอส ชั้นขนส่ง ทีซีพี ยูดีพี ชั้นเครือข่าย ไอพีรุ่น๔ ไอพีรุ่น๖ เน็ตมาส มาส ชั้นตอนวิธีเลือกเส้นทาง เลือกเส้นทางที่สั้นที่สุด เลือกเส้นทางแบบระยะทางเด็กเตอร์ การค้นหาเส้นทางแบบบอร์ดคาสและมัลติคาส ชั้นการเชื่อมต่อ การส่งข้อมูลแบบวงจร การส่งข้อมูลแบบแพคเกจ อีเทอร์เน็ต และไวร์ลีย์ เครื่องข่ายไวร์ลีย์ประเภทประยุกต์ พลังงาน การตรวจสอบความผิดพลาด ซีอาร์ซี ชั้นภาษาภาพ การออกแบบเครื่องข่าย

ทั้งนี้ผู้เขียนได้สอนรายวิชาเครื่องข่ายคอมพิวเตอร์นับเป็นเวลาสิบปีตั้งแต่ พ.ศ.๒๕๕๓ และได้จัดบันทึกปรับปรุงเนื้อหาให้ทันสมัย ได้มีการปรับปรุงแก้ไขจุดบกพร่องตลอดมา โดยเอกสารฉบับนี้ใช้สอนกับนักศึกษาสาขาวิชาวิศวกรรมคอมพิวเตอร์นับตั้งแต่ภาคเรียนที่ ๑ ปีการศึกษา ๒๕๖๓

ผู้เขียนได้รวบรวมจัดทำเกี่ยวกับหลักสูตรรายวิชา ประกอบด้วย ลักษณะวิชา การแบ่งบทเรียน หัวข้อ จุดประสงค์การสอนและการประเมินผลรายวิชา พร้อมทั้งได้จัดทำหนังสือสอนในเบื้องต้น รายสัปดาห์ตลอดทั้ง ๑๕ สัปดาห์ ซึ่งประกอบด้วย จุดประสงค์การสอน เนื้อหาสาระที่สอน วิธีการสอน เอกสารและสื่อประกอบการสอนเป็นต้น ทั้งนี้ ผู้เขียนคาดหวังว่า เอกสารประกอบการสอนฉบับนี้จะเป็นเอกสารคู่มือของอาจารย์ใช้ประกอบการสอนที่ได้มีการเตรียมและวางแผนการสอนไว้อย่างรอบคอบ ซึ่งจะส่งผลให้การเรียนการสอนรายวิชานี้มีประสิทธิภาพและมีคุณภาพยิ่งขึ้นต่อไป

.....
(นายทรงฤทธิ์ กิติศรีวราพันธุ์)

(มีนาคม ๒๕๖๓)

สารบัญ

เรื่อง	หน้า
คำนำ	ii
สารบัญ	iii
รายการตาราง.....	v
รายการภาพประกอบ.....	vi
วัตถุประสงค์ของหลักสูตร.....	1
ลักษณะรายวิชา	2
การแบ่งบทเรียน/หัวข้อ.....	2
จุดประสงค์การสอน.....	6
ตารางกำหนดน้ำหนักคะแนน	10
กำหนดการสอน.....	11
บทที่ 1 บทนำ	13
1.1 พัฒนาการการสื่อสารของมนุษย์.....	13
1.2 เทคโนโลยีเครือข่ายคอมพิวเตอร์	14
1.3 สถาปัตยกรรม	20
1.4 ค่าประสิทธิภาพ.....	25
บทที่ 2 การเชื่อมต่อโดยตรง	29
2.1 เทคโนโลยีขั้นกายภาพ.....	29
2.2 มุมมองเทคโนโลยีภาพกว้าง	32
2.3 เอ็นโค้ดดิ้ง.....	35
2.4 เฟรมมิ่ง.....	36
2.5 ตรวจจับข้อผิดพลาด	38
2.6 การมีเสถียรภาพในการส่งข้อมูล	43
2.7 Multi-Access Networks	51
2.8 เครือข่ายแลนรีสไทร.....	55
2.9 Access Networks.....	60
บทที่ 3 โพรโท콜ขั้นเครือข่าย	63
3.1 พื้นฐานเนตเวิร์กสวิตช์	64
3.2 อีเทอร์เนตสวิตช์	73
3.3 อินเทอร์เน็ตโพรโทคอล (IP)	80
3.4 ขั้นตอนวิธีเลือกเส้นทาง	97
3.5 โพรโทคอลไอพีรุ่นที่ ๖	106
บทที่ 4 โพรโทคอล แบบ End-to-End.....	111
4.1 โพรโทคอล UDP	112
4.2 โพรโทคอล TCP	113

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 5 การควบคุมความคับคั่ง.....	128
5.1 ประเด็นการจัดสรรทรัพยากรเครือข่าย	129
5.2 รูปแบบคิว.....	136
5.3 TCP Congestion Control.....	140
บทที่ 6 ความปลอดภัยทางเครือข่าย	144
6.1 ความไว้วางใจ และ ภัยคุกคาม	145
6.2 กระบวนการเข้ารหัส	145
6.3 ขั้นตอนแลกเปลี่ยนกุญแจรหัสลับ	152
6.4 โพรโทคอลพิสูจน์ตัวจริง	158
6.5 ตัวอย่างการใช้งานของพ็อตเวอร์ด้านความปลอดภัย	164
บทที่ 7 ชั้นแอปพลิเคชัน.....	171
7.1 แอปพลิเคชันดังเดิม	171
7.2 แอปพลิเคชันประเภทโครงสร้างพื้นฐาน	183
7.3 เทคโนโลยีคอนเทนเนอร์	190
บทที่ 8 เครือข่ายไร้สายประเพณีประยัดพลังงาน	191
8.1 บลูทูธ (IEEE802.15.1).....	191
8.2 เทคโนโลยีอรา	192
บทที่ 9 การออกแบบเครือข่าย.....	202
9.1 การออกแบบเครือข่าย.....	202
9.2 วิเคราะห์เป้าหมายของการทำธุรกิจและข้อจำกัด	202
9.3 การวิเคราะห์ด้านเทคนิคและข้อจำกัด	208
9.4 ทำความเข้าใจเครือข่ายเดิมของลูกค้า	210
9.5 Physical Network Design	210
9.6 Logical Network Design	214
9.7 การออกแบบเส้นทางสำรองทางเครือข่าย	214
บรรณานุกรม	216

รายการตาราง

ตาราง	หน้า
ตารางที่ 2.1 การจับคู่แรงดันไฟฟ้ากับเลขใบนารี	30
ตารางที่ 2.2 ประเภทสายและอัตราเร็วบริการอินเทอร์เน็ตบ้าน.....	35
ตารางที่ 2.3 เอ็นโคเด็ดดิ้งแบบ 4B/5B	37
ตารางที่ 2.4 แปลง Hello, world เป็นรหัสแอสกี	38
ตารางที่ 2.5 ลำดับข้อมูลเรียงตามบิต	39
ตารางที่ 2.6 เกิดข้อผิดพลาดระดับบิต ของตัวอักษร 0 และ r	39
ตารางที่ 2.7 การตรวจสอบข้อผิดพลาดระดับบิตด้วยวิธีเช็คชั้ม	40
ตารางที่ 3.1 ตารางฟอร์เวิร์ดของเนตเวิร์กสวิตซ์ 1	65
ตารางที่ 3.2 ตารางฟอร์เวิร์ดของเนตเวิร์กสวิตซ์ 2	67
ตารางที่ 3.3 ตัวรับบุuginรเมื่อน สวิตซ์-1	68
ตารางที่ 3.4 ตัวรับบุuginรเมื่อน สวิตซ์-2	69
ตารางที่ 3.5 ตัวรับบุuginรเมื่อน สวิตซ์-3	70
ตารางที่ 3.6 ฟอร์เวิร์ดดิงเทเบิล ภายใต้บริดจ์	73
ตารางที่ 3.7 รหัสระบุประเภทความคับคั่ง	85
ตารางที่ 3.8 ตารางฟอร์เวิร์ด R2	89
ตารางที่ 3.9 ตารางฟอร์เวิร์ดสมมูลรูปของ R2	89
ตารางที่ 3.10 ฟอร์เวิร์ดดิงเทเบิลที่แบ่งชั้บเน็ต	91
ตารางที่ 3.11 ฟอร์เวิร์ดดิงเทเบิลของเร้าเตอร์ R1	97
ตารางที่ 3.12 ตัวอย่างตารางเราท์ติ้ง	98
ตารางที่ 3.13 ตัวอย่างฟอร์เวิร์ดดิงเทเบิล	98
ตารางที่ 3.14 ข้อมูลเริ่มต้นในแต่ละโหนดสำหรับคำนวน DV	99
ตารางที่ 3.15 ข้อมูลตารางเราท์ติ้งที่โหนด A	100
ตารางที่ 3.16 ข้อมูลตารางเราท์ติ้งที่โหนด A	100
ตารางที่ 3.17 ข้อมูลเมื่อเลือกเส้นทางเสร็จสิน	101
ตารางที่ 3.18 ลำดับการสร้างตารางเราท์ติ้งสำหรับโหนด D	107
ตารางที่ 3.19 หมายเลขพีร์ฟิคของไอพีรุ่น ๖	109
ตารางที่ 4.1 หน่วยเวลาขนาด 32-บิต	121
ตารางที่ 4.2 ต้องการ Window Size สำหรับ 100-ms รวมถึงทริปไทม์	122
ตารางที่ 7.1 คำสั่งร้องขอ ของโพรโทคอล HTTP	178
ตารางที่ 7.2 คำสั่งร้องตอบกลับ ของโพรโทคอล HTTP	179
ตารางที่ 8.1 ค่าพารามิเตอร์ของลอร่า	196
ตารางที่ 9.1 แบบสำรวจโปรแกรม	204
ตารางที่ 9.2 ตัวอย่างแบบฟอร์มสำรวจด้านเทคนิคของการใช้งานแอปพลิเคชัน	210

รายการภาพประกอบ

รูป	หน้า
รูปที่ 1.1 ภาพสัตว์ประภูบันแห่นพินในถ้ำโซเวต.....	13
รูปที่ 1.2 การใช้แอพพลิเคชันในการประชุมออนไลน์	15
รูปที่ 1.3 ไดเรกซิลิก (a) point-to-point (b) multiple-access	17
รูปที่ 1.4 เครือข่ายสิวิตช์	18
รูปที่ 1.5 การเชื่อมต่อ helyalayเครือข่ายเข้าด้วยกันผ่านตารางเราท์ดิ้ง	18
รูปที่ 1.6 การทำการรวมสัญญาณจากไฮสต์ helyalayเครือข่ายเข้าด้วยกันร่วมกัน	19
รูปที่ 1.7 การใช้แพ็กเก็ตสวิตซ์สำหรับใช้สายสัญญาณร่วมกัน	19
รูปที่ 1.8 อธิบายกระบวนการสื่อสารโดยใช้แนวคิดนามธรรม	20
รูปที่ 1.9 ตัวอย่างเลเยอร์ระบบเครือข่าย	21
รูปที่ 1.10 ระบบเลเยอร์ที่ระดับเดียวกันมี helyalayทางเลือก	21
รูปที่ 1.11 ส่วนบริการการเชื่อมต่อและการสื่อสารภายในเลเยอร์	21
รูปที่ 1.12 ตัวอย่างแผนภาพ协议โพรโทคอล	22
รูปที่ 1.13 Jersey Telecom telephone operator at switchboard 1975	23
รูปที่ 1.14 ข้อความที่ส่งไปเลเยอร์สูงขึ้น มีขั้นตอนแพ็คข้อมูล	23
รูปที่ 1.15 ตัวแบบ OSI 7-layer	24
รูปที่ 1.16 แผนภาพอินเทอร์เน็ตโพรโทคอล	25
รูปที่ 1.17 มุ่งมองอธิบายเลเยอร์โดยกำหนดโพรโทคอล ในส่วน Subnetwork ก่อนหน้านี้อยู่ในขั้นเครือข่าย แต่บ่อยครั้งถูกจัดให้อยู่ในชั้นลิ่งค์ “Layer 2” (ตามการอ้างอิง OSI 7-Layer)	25
รูปที่ 1.18 การส่งข้อมูลระดับบิตภายในชั้นลิ่งค์ (a)ส่งข้อมูลบิตด้วยอัตราเร็ว 1Mbps (หนึ่งบิตใช้เวลาหนึ่งไมโครวินาที) (b)ส่งข้อมูลบิตด้วยความเร็ว 2Mbps (หนึ่งบิตใช้เวลา 0.5ไมโครวินาที)	26
รูปที่ 1.19 ค่าประสิทธิภาพ ค่าดีเลย์แฟง และ RTT	27
รูปที่ 1.20 มุ่งมองเครือข่ายเป็นท่อส่งข้อมูล	27
รูปที่ 1.21 ความสมมั่นใจระหว่างแบบดิจิทและดิจิล	28
รูปที่ 1.22 การเกิดจิทเทอร์ในเครือข่าย	28
รูปที่ 2.1 Pulse Code Modulation (Aqueegg commonswiki, 2014)	29
รูปที่ 2.2 การสื่อสารในชั้นกายภาพ (Forouzan, 2012 , p.93)	31
รูปที่ 2.3 เปรียบเทียบข้อมูลแอนะล็อก และ สัญญาณดิจิทัล (Forouzan, 2012 , p.94)	32
รูปที่ 2.4 การเชื่อมต่ออินเทอร์เน็ตของผู้ใช้งาน	33
รูปที่ 2.5 スペกตรัม ของแม่เหล็กไฟฟ้า	34
รูปที่ 2.6 สัญญาณเดินทางผ่านตัวนำสัญญาณเป็นสัญญาณดิจิทัลสิ่งครั้งละบิต	35
รูปที่ 2.7 การเอ็นโคดดิ้งแบบไม่กลับไปเป็นศูนย์ลับสัญญาณตรงกลางบิต	36
รูปที่ 2.8 การเอ็นโคดดิ้งแบบ ไม่กลับไปเป็นศูนย์ ไม่กลับไปเป็นศูนย์กลับหัว และแม่นเซสເຕວົງ	36
รูปที่ 2.9 ບິຕໍລາເລີຍຜ່ານອະແດປເຕວົງຕັນທາງສ່າງຜ່ານຊ່ອງສັນຍາໄປຄິງຂະແດປເຕວົງປາຍທາງ	37

รายการภาพประกอบ (ต่อ)

รูป	หน้า
รูปที่ 2.10 Odd parity bit ขนาด (7+1)-bit 7บิตข้อมูล และ 1 parity bit	41
รูปที่ 2.11 การหารายาระบบการคำนวณCRC.....	43
รูปที่ 2.12 การควบคุมอัตราเร็วข้อมูลด้วยวิธี Stop-and-wait.....	44
รูปที่ 2.13 การเพิ่มหมายเลขอ้างโน้ตเจนท์ เพื่อป้องกันการส่งข้อความที่เดินทางกลับซ้ำ.....	44
รูปที่ 2.14 การควบคุมอัตราเร็วข้อมูลด้วยวิธี Sliding Window.....	46
รูปที่ 2.15 ตัวแปรสำหรับคำนวณการปรับขนาดวินโดว์ของเครื่องส่ง	47
รูปที่ 2.16 ตัวแปรสำหรับคำนวณการปรับขนาดวินโดว์ของเครื่องรับ	48
รูปที่ 2.17 ทรานซิฟเวอร์และอะแดปเตอร์ ในเครือข่ายอีเทอร์เน็ต	52
รูปที่ 2.18 การใช้รีพีทเตอร์ในเครือข่ายอีเทอร์เน็ตขนาดใหญ่.....	53
รูปที่ 2.19 การใช้สายสัญญาณร่วมกันของไฮสต์ทั้งหมดหากเครื่อง.....	53
รูปที่ 2.20 โครงสร้างเฟรมระบบอีเทอร์เน็ต	54
รูปที่ 2.21 การส่งเฟรมผ่านสายนำสัญญาณ.....	54
รูปที่ 2.22 การเข้ารหัสด้วยเทคโนโลยี DSSS	56
รูปที่ 2.23 ลูกข่ายแลนไร้สายเชื่อมเครือข่ายผ่านสถานีฐาน	56
รูปที่ 2.24 รูปแบบการเชื่อมต่อแบบ Ad-hoc.....	57
รูปที่ 2.25 โหนด A และ โหนด C เกิดปัญหา hidden node ระหว่างกัน	58
รูปที่ 2.26 โหนด A และ D ไม่ทราบการมีอยู่ของกันและกัน เกิดปัญหา expose node	58
รูปที่ 2.27 กรณีการเชื่อมต่อแลนไร้สายตามจุดต่างๆทุกโหนดสามารถสื่อสารกันได้ผ่านทาง DS	59
รูปที่ 2.28 โหนดมีการเคลื่อนที่เปลี่ยนออกเซพอยต์	59
รูปที่ 2.29 เฟรมมาตรฐาน IEEE 802.11	59
รูปที่ 2.30 ตัวอย่างการเชื่อม PON กับ OLT ภายในสำนักงานก่อนส่งออกทาง BNG	60
รูปที่ 2.31 โทรศัพท์เคลื่อนที่ส่งคลื่นวิทยุผ่าน RAN ไป BBU และส่งเข้าสำนักงานกลาง ก่อนส่งออก อินเทอร์เน็ต	61
รูปที่ 2.32 วิธีจัดสรรทรัพยากรแบบ OFDMA.....	62
รูปที่ 3.1 กราฟบริบูรณ์มีโหนดจำนวน 8 โหนดต้องการ 28 เส้นเชื่อม	63
รูปที่ 3.2 กราฟบริบูรณ์มีโหนดจำนวน 8 โหนดต้องการ 8 เส้นเชื่อม	64
รูปที่ 3.3 การเชื่อมสวิตช์แบบโครงข่ายสถาาร์	65
รูปที่ 3.4 ตัวอย่างรูปแบบการเชื่อมต่อโดยใช้เนตเวิร์กสวิตช์.....	66
รูปที่ 3.5 วงจรเส้นก้อนของการเชื่อม ไฮสต์ A กับ ไฮสต์ B.....	68
รูปที่ 3.6 ตัวอย่างแพ็กเก็ตกำหนดเส้นทางด้วยวิธีวงจรเส้นก้อน.....	69
รูปที่ 3.7 แพ็กเก็ตส่งผ่านวงจรเส้นก้อน.....	70
รูปที่ 3.8 การกำหนดเส้นทางแบบต้นทางกำหนดเส้นทาง(สวิตช์อ่านเลขขวาสุดเสมอ).....	72
รูปที่ 3.9 วิธีต้นทางกำหนดเส้นทางทั้งสามแบบ	72
รูปที่ 3.10 บริดจ์เชื่อมการสื่อสารระหว่างสองเครือข่าย	75

รายการภาพประกอบ (ต่อ)

รูป	หน้า
รูปที่ 3.11 อีเทอร์เน็ตสวิตช์ที่มีลูป.....	76
รูปที่ 3.12 เครือข่ายปกติและเครือข่ายแบบตันไม้แบบทดสอบข้าม	76
รูปที่ 3.13 เรเดีย พีร์ลแม่น้ำของสิทธิบัตรตันไม้แบบทดสอบข้ามໂพรໂทคอล.....	76
รูปที่ 3.14 เครือข่ายเชื่อมแบบบริดจ์ในมุมมองລອຈິກັດ	78
รูปที่ 3.15 เครือข่ายเครือข่ายแลนນີສອນເຄຣູອຂ່າຍທີ່ເຫັນສົມບົນກັນ	79
รูปที่ 3.16 802.1Q VLAN tag ແທກຂ້ອງມູລໃນເຂດເດອຮມາຕຽບຮູນອື່ເທອ້ຣັນ(802.1).....	80
รูปที่ 3.17 ອິນເທອ້ຣັນເຕີເວີຣັກທີ່ພັບໂດຍຕາມປົກຕິ H ແທນໂໂສຕົ້ລແລະ R ແທນຮ້າເຕົວໆ	81
รูปที่ 3.18 ອິນເທອ້ຣັນເຕີເວີຣັກຍ່າງຈ່າຍ ມີການເຂື່ອມຕ່ອກກັນຮະຫວ່າງ H5 ແລະ H5 ຜ່ານໂພຣໂທຄອລ ແລ້ນໄວ້ສາຍແລະອື່ເທອ້ຣັນເນື້ດ	82
รูปที่ 3.19 ເຂດເດອຮ້ອຂອງໄວີ່ຢູ່ນ ۴.....	83
รูปที่ 3.20 ຮහສີບານາຮີກຳນົດໂພຣໂທຄອລໄວີ່ຢູ່ນ ۴.....	84
รูปที่ 3.21 ຮහສີບານາຮີກຳນົດໂພຣໂທຄອລໄວີ່ຢູ່ນ ۶.....	84
รูปที่ 3.22 ຕ້າວຍ່າງຮ້າສີບານາຮີກຳນົດເຊື້ດເດອຮ້ມື້ນາດ 20 ໄບຕໍ່	84
รูปที่ 3.23 ກາຣດີນທາງຂອງອິນເທອ້ຣັນໂພຣໂທຄອລເດຕາແກຣມຜ່ານເທັກໂນໂລຢີເຄຣູອຂ່າຍໜາຍເທັກໂນໂລຢີໄດຍ້ທີ່ R2 ເຂື່ອມກັບ R3 ແບບ PPP ມີເວັ້ມທີ່ຢູ່ຕໍ່ທີ່ສຸດ	86
รูปที่ 3.24 ແພັກເກີດເຂົດເດອຮ້ອຂອງເດຕາແກຣມທີ່ເກີດກາຮ້ັນແພັກເກີດ	86
รูปที่ 3.25 ອິນເທອ້ຣັນໂພຣໂທຄອລແດວເດຣສ: (a) ຄລາສ A; (b) ຄລາສ B; (c) ຄລາສ C	87
รูปที่ 3.26 ຊັບເນື້ຕມາສັກໄວີ່ປົກລາສ B	91
รูปที่ 3.27 ເຄຣູອຂ່າຍມີເຮົາເຕົວໆຈໍານານສອນເຄຣູອງມີສາມເຄຣູອຂ່າຍ	92
รูปที่ 3.28 ກາຣດີນ່ຳມື່ນ່ຳ /21 ເປັນ /24	93
รูปที่ 3.29 ໂຄງສ້າງໂພຣໂທຄອລ ARP	94
รูปที่ 3.30 ໂໂສຕົ້ລສ່າງບຽດຄາສົ່ຕເຂົດເຄຣູອຂ່າຍເພື່ອຂອ້ຂ່າຍເລີ້ມໄວີ່	95
รูปที่ 3.31 ໂຄງສ້າງໂພຣໂທຄອລ DHCP	95
รูปที่ 3.32 ຕ້າວຍ່າງເຄຣູອຂ່າຍເຄຣູອຂ່າຍສ່ວນຕ້າວເສີມອື່ນ: (a) ສອນເຄຣູອຂ່າຍແຍກຈາກກັນທາງກາຍກາພ (b) ເຄຣູອຂ່າຍໃໝ່ສາຍສັນຍານຮ່ວມກັນແຕ່ແຍກກັນໂດຍໃໝ່ຈະເສີມອື່ນ	96
รูปที่ 3.33 ກາຣດີນ້ຳມື່ນ້ຳທີ່ພັນເນັນຜ່ານເຄຣູອຂ່າຍ 18.5.0.1 ທຳໃຫ້ R1 ແລະ R2 ເຂື່ອມຕ່ອກກັນໄດ້ໂດຍຕຽນຜ່ານເຄຣູອຂ່າຍ 2.x	97
รูปที่ 3.34 ເຂົ້າໝາຍການເຂື່ອມເຄຣູອຂ່າຍດ້ວຍການ	98
รูปที่ 3.35 Distance-vector ກຳນົດເສັນທາງ: ຕ້າວຍ່າງເຄຣູອຂ່າຍ	99
รูปที่ 3.36 ເຄຣູອຂ່າຍທີ່ມີສິ່ງເຮົາເຕົວໆກົກເນີຕເວີຣັກ	103
รูปที่ 3.37 ໂຄງສ້າງເຟຣິມ RIP	104
รูปที่ 3.38 ກາຣດີນໃນ LS: (a) LSP ເຂົ້າໂທນັດ X; (b) X ພັດ LSP ໄປ A ແລະ C; (c) A ແລະ C ພັດໄປ B ແຕ່ມີສິ່ງໄປ X; (d) ພັດຄຽບທຸກໂທນັດ	105
รูปที่ 3.39 ເຄຣູອຂ່າຍຕ້າວຍ່າງການການກົດລັບລັບການແນບ LS	106

รายการภาพประกอบ (ต่อ)

รูป	หน้า
รูปที่ 3.40 รูปแบบเซดเดอร์ OSPF	106
รูปที่ 3.41 รูปแบบ advertisement ภายในแพ็กเก็ต OSPF	107
รูปที่ 3.42 การเชื่อมต่อระหว่างเครือข่ายใช้หมายเลขไอพีสำหรับตารางเราท์ติ้ง	108
รูปที่ 3.43 โครงสร้างเซดเดอร์ของไอพีรุ่น ๖	110
รูปที่ 4.1 โครงสร้างเซดเดอร์ UDP	112
รูปที่ 4.2 เมื่อ halfway แอปพลิเคชันต้องการใช้ UDP ส่งข้อมูลจะใช้หมายเลขพอร์ตแตกต่างกัน	113
รูปที่ 4.3 การบริหารจัดการข้อมูลของโปรโตคอล TCP	114
รูปที่ 4.4 เซดเดอร์โปรโตคอล TCP.....	114
รูปที่ 4.5 TCP มีส่ง SequenceNum และเครื่องรับได้ตอบกลับ	115
รูปที่ 4.6 ขั้นตอน Three-way Handshake ในระบบ TCP	116
รูปที่ 4.7 TCP state-transition diagram	117
รูปที่ 4.8 ความสัมพันธ์ระหว่างบัฟเฟอร์ เครื่องส่ง (a) และ บัฟเฟอร์เครื่องรับ (b).....	118
รูปที่ 4.9 ปัญหา Silly window syndrome.....	123
รูปที่ 4.10 การทำงานกับ แอ็กโนเล็ตจเมนท์: (a) การทำงานปกติ (b) การส่ง ส่งข้อมูลอีกรัง	124
รูปที่ 4.11 การทดสอบระบบ: ลินักซ์สองเครื่องเชื่อมต่อกันโดยใช้การดึงเครื่อข่ายเครื่องละสองใบ	126
รูปที่ 4.12 การทดลองวัดค่าทรูพุตโดยมีแพ็กเก็ตหลายขนาด	127
รูปที่ 5.1 การเกิดปัญหาค่าขวดกับเร้าเตอร์.....	131
รูปที่ 5.2 โฟล์ผ่านร้าเตอร์จำนวนหลายโฟล์	132
รูปที่ 5.3 จุดดีที่สุดอยู่ตรงจุดที่ทำให้ทรูพุตภัยดีเลย์มีโหลดสูงสุด	134
รูปที่ 5.4 ใช้เปรียบเทียบความเท่าเทียม หนึ่งโฟล์ต้องการสื่อสารระหว่างทางเครือข่าย กับ สามโฟล์ต้องการโฟล์ละหนึ่งระยะห่างทางเครือข่าย	135
รูปที่ 5.5 คิวนิด FIFO (a) การตัดคิวแบบรอบป้ายแควร ของ FIFO(b).....	136
รูปที่ 5.6 บริการ Round-robin จากทั้งหมดสี่โฟล์เข้าเร้าเตอร์.....	137
รูปที่ 5.7 ตัวอย่าง fair queueing : (a)แพ็กเก็ตที่ส่งเสร็จเร็วกว่าจะได้รับสิทธิส่งก่อน (b) Flow 2 กำลังส่งและมี Flow 1 เข้ามาแทรก.....	139
รูปที่ 5.8 jacobi ฟัน อูเทรคต์ ผู้เสนอใช้ Congestion Control สำหรับ TCP	140
รูปที่ 5.9 แพ็กเก็ตขณะเดินทางมีการเพิ่มโฟล์ขึ้นเรื่อยๆ	141
รูปที่ 5.10 การทำงาน TCP มีทรูพุตเป็นฟันเลื่อย	142
รูปที่ 5.11 แพ็กเก็ตทำงานแบบ slow start.....	143
รูปที่ 5.12 การทำงานของระบบ ควบคุมความคับคั่ง สำหรับ TCP	143
รูปที่ 6.1 ใช้กุญแจรหัสลับในการเข้ารหัสและถอดรหัส	146
รูปที่ 6.2 การแบ่งข้อมูลเป็นท่อนก่อนเข้ารหัส	147
รูปที่ 6.3 การเข้ารหัสด้วยกุญแจสาธารณะ	149
รูปที่ 6.4 การพิสูจน์ตัวจริงโดยใช้กุญแจส่วนตัว	150

รายการภาพประกอบ (ต่อ)

รูป	หน้า
รูปที่ 6.5 การประมวลผล MAC (a) เทียบกับการประมวลผล HMAC (b).....	151
รูปที่ 6.6 โครงสร้างต้นไม้ของระบบในรับรองอิเล็กทรอนิกส์.....	154
รูปที่ 6.7 การโจมตีแบบ man-in-the-middle.....	158
รูปที่ 6.8 โพรโทคอล challenge-response.....	159
รูปที่ 6.9 การใช้ public-key พิสูจน์ตัวจริงที่ต้องการ synchronization	160
รูปที่ 6.10 การใช้ public-key พิสูจน์ตัวจริงแบบไม่ต้องการ synchronization	161
รูปที่ 6.11 พิสูจน์ตัวจริงโพรโทคอลแบบ Needham-Schroeder.....	162
รูปที่ 6.12 ระบบพิสูจน์ตัวจริงแบบเคอร์เบรส.....	163
รูปที่ 6.13 การใช้ลายเซ็นดิจิทัล ด้วยวิธี PGP	165
รูปที่ 6.14 ใช้ SSH สร้างทันเนลที่ปลอดภัยในการสื่อสาร	166
รูปที่ 6.15 โพรโทคอล secure transport layer เพิ่มตระกากระหว่างชั้นแอปพลิเคชันกับ TCP.....	167
รูปที่ 6.16 กระบวนการ Handshake ของ TLS.....	168
รูปที่ 6.17 ใช้ Authentication server สำหรับเครือข่ายแลนไร้สาย	169
รูปที่ 6.18 ไฟร์วอลล์ทำหน้าที่กรองแพ็กเก็ต	169
รูปที่ 7.1 ลำดับการส่งอีเมลใช้หลักการ store-and-forward.....	173
รูปที่ 7.2 การเปลี่ยนสถานะของโพรโทคอล IMAP	176
รูปที่ 7.3 เว็บเบราว์เซอร์ Safari.....	177
รูปที่ 7.4 HTTP Non-persistent อ่านข้อมูลครั้งละชุด	181
รูปที่ 7.5 ปรับปรุง HTTP ด้วยวิธี persistent connections	181
รูปที่ 7.6 การแปลงชื่อโฮสต์เป็นที่อยู่โฮสต์ ตั้งแต่ลำดับ 1 ถึง 5	184
รูปที่ 7.7 การแบ่งลำดับชั้นของระบบโดยเมนเเนม.....	185
รูปที่ 7.8 การแบ่งเเนมสเปซออกเป็นโซน	186
รูปที่ 7.9 ลำดับชั้นของเเนมเชิร์ฟเวอร์	186
รูปที่ 7.10 Root เนมเชิร์ฟเวอร์	189
รูปที่ 7.11 ขั้นตอนการสอบถามหมายเลขไอพีในระบบ DNS	189
รูปที่ 8.1 เครือข่ายพีดีเน็ตเทคโนโลยีบลูทู�	192
รูปที่ 8.2 สัญญาณคลื่นวิทยุเทคโนโลยีเชิร์ฟ ส่วนด้านบนอธิบายด้วยสเปกตรัม ส่วนด้านล่างอธิบายด้วย waterfall	194
รูปที่ 8.3 ส่วน preamble ของ เอิร์พสเปกตรัม	195
รูปที่ 8.4 ส่วน preamble และส่วนข้อมูล.....	195
รูปที่ 8.5 ความสัมพันธ์ของสเปรดเฟกเตอร์ กับแบบต์วิด์ช และ symbol duration	197
รูปที่ 8.6 ตัวอย่างสัญญาณลอรา จำนวน 1 down-chirp	198
รูปที่ 8.7 ตัวอย่างสัญญาณลอรา จากเครื่องส่งและเครื่องรับ	199
รูปที่ 8.8 โครงข่ายล้อราแวน เป็นแบบ Star Topology.....	200

รายการภาพประกอบ (ต่อ)

รูป	หน้า
รูปที่ 8.9 โครงข่ายลอราแวน เป็นแบบ Star Topology	201
รูปที่ 9.1 การออกแบบเครือข่ายและวิธีการดำเนินการ	203
รูปที่ 9.2 เนตเวิร์กโดยรวมของผู้ประกอบการด้านผลิตชั้นส่วนอิเล็กทรอนิกส์	211
รูปที่ 9.3 ภาพเน็ตเวิร์กโดยรวมแบบ Modular Block Diagram	212
รูปที่ 9.4 การแบ่งชั้บเน็ต	212
รูปที่ 9.5 โภคภัณฑ์การเขียนลายสำหรับออกแบบ	213
รูปที่ 9.6 Hierarchical Network Design	215

วัตถุประสงค์ของหลักสูตร

หลักสูตรระดับ ปริญญาตรี สาขาวิชาวิศวกรรมคอมพิวเตอร์

1. มีคุณธรรมจริยธรรมและยึดมั่นในจรรยาบรรณในการประกอบวิชาชีวิศวกรรม คอมพิวเตอร์
2. มีความรู้พื้นฐานและเข้าใจในศาสตร์วิศวกรรมคอมพิวเตอร์สำหรับการประยุกต์ใช้ ปัญญาประดิษฐ์ใน การต่อยอดและสร้างนวัตกรรมได้
3. สามารถวิเคราะห์และแก้ไขปัญหางานด้านวิศวกรรมคอมพิวเตอร์ได้
4. มีทักษะด้านภาษา ความเข้าใจด้านสังคมวัฒนธรรมและสามารถทำงานร่วมกับผู้อื่นได้
5. มีความรู้ในศาสตร์ที่เกี่ยวข้องทั้งทางทฤษฎีและปฏิบัติสำหรับนำไปประยุกต์ใช้ในการประกอบอาชีพ ทางวิศวกรรมคอมพิวเตอร์
6. มีทักษะการเรียนรู้ตลอดชีวิต รู้เท่าทันการเปลี่ยนแปลงและแนวโน้มที่จะเกิดขึ้นของเทคโนโลยีด้าน คอมพิวเตอร์สมัยใหม่

ลักษณะรายวิชา

หลักสูตรระดับ ปริญญาตรี สาขาวิชาวิศวกรรมคอมพิวเตอร์

- | | |
|-----------------------|---|
| 1. รหัสและชื่อวิชา | เครือข่ายคอมพิวเตอร์(๓๑๑๐๓๑๔) |
| 2. สภาพวิชา | วิชาชีพบังคับหลักสูตร วิศวกรรมศาสตรบัณฑิต |
| 3. ระดับรายวิชา | ภาคการศึกษาที่ 1 ชั้นปีที่ 2 |
| 4. รายวิชาพื้นฐาน | หมวดวิชาเฉพาะ |
| 5. เวลาศึกษา | ทฤษฎี 3 ชั่วโมง ปฏิบัติ 0 ชั่วโมง รวมทั้งสิ้น 3 ชั่วโมง และนักศึกษาจะต้องใช้เวลาศึกษาค้นคว้านอกเวลา 6 ชั่วโมงต่อสัปดาห์ ตลอด 15 สัปดาห์ |
| 6. จำนวนหน่วยกิต | 3 |
| 7. จุดมุ่งหมายรายวิชา | |

1. รู้วิวัฒนาการการสื่อสารคอมพิวเตอร์
 2. เข้าใจวิธีการเชื่อมต่อโดยตรง
 3. วิเคราะห์วิธีการทันหน้าเล่นทางเครือข่ายคอมพิวเตอร์
 4. เข้าใจโทรศัพท์แบบเครือข่าย และ นำวิธีคำนวนໄ้อฟ์ไปใช้ออกแบบเครือข่ายได้
 5. เข้าใจโทรศัพท์แบบ End-to-End และสังเคราะห์ประเด็นการควบคุมความคับคั่ง
 6. เข้าใจลักษณะปัญหาด้านความปลอดภัยทางเครือข่าย
 7. เข้าใจการทำงานของแอปพลิเคชัน
 8. เข้าใจการทำงานเครือข่ายไร้สายประเภทหนึ่งที่มีอยู่ในปัจจุบัน
 9. สังเคราะห์การออกแบบเครือข่ายได้
 10. เห็นคุณค่าของการแลกเปลี่ยนความรู้

8. คำอธิบายรายวิชา

ระบบเครือข่ายคอมพิวเตอร์เบื้องต้น แบบจำลองโวเอสไอ ชั้นโปรแกรมประยุกต์ เอชที ทีพี เอสเอ็มทีพี ดีอี็นเอส ชั้นขนส่ง ทีซีพี ยูดีพี ชั้นเครือข่าย ไอพีรุ่น๔ ไอพีรุ่น๖ เน็ต มาส มาส ชั้นตอนวิธีเลือกเส้นทาง เลือกเส้นทางที่สั้นที่สุด เลือกเส้นทางแบบระยะทาง เวลาเตอร์ การค้นหาเส้นทางแบบบอร์ดคาสและมลติคาส ชั้นการเชื่อมต่อ การส่งข้อมูล แบบวงจร การส่งข้อมูลแบบแพคเกจ อีเทอร์เน็ต และไร้สาย เครือข่ายไร้สายประเภท ประยุกต์ พลังงาน การตรวจสอบความผิดพลาด ซีอาร์ซี ชั้นกายภาพ การออกแบบเครือข่าย

การแบ่งบทเรียน/หัวข้อ

บทเรียนที่	รายการ	เวลา(ชั่วโมง)	
		ท	ป
1	1. บทนำ 1.1 พัฒนาการการสื่อสารของมนุษย์ 1.2 เทคโนโลยีเครือข่ายคอมพิวเตอร์ 1.3 สถาปัตยกรรม	3	0
2	1. บทนำ 1.4 ค่าประสิทธิภาพ 2. การซื้อมต่อโดยตรง 2.1 มุมมองเทคโนโลยีภาพกว้าง 2.2 เอ็นโคడดิ้ง 2.3 เพرمมิ่ง 2.4 ตรวจจับข้อผิดพลาด	3	0
3	2. การซื้อมต่อโดยตรง 2.5 การมีเสถียรภาพในการส่งข้อมูล 2.6 Multi-Access Networks 2.7 Wireless Networks 2.8 Access Networks	3	0
4	3. โพรโทคอลเครือข่าย 3.1 พื้นฐานเนตเวิร์กสวิตช์ 3.2 อีเทอร์เนตสวิตช์ 3.3 อินเทอร์เน็ตโพรโทคอล (IP)	3	0
5	3. โพรโทคอลเครือข่าย 3.4 Routing 3.5 โพรโทคอลไอโอพีรุ่น ๖	3	0
6	4. โพรโทคอล แบบ End-to-End 4.1 โพรโทคอล UDP 4.2 โพรโทคอล TCP	3	0

7	5. การควบคุมความคับคั่ง 5.1 ประเด็นการจัดสรรที่พยากรณ์เครือข่าย 5.2 รูปแบบคิว 5.2 TCP Congestion Control	3	0
8	สอดคล้องภาค		
9	6. ความปลอดภัยทางเครือข่าย 6.1 ความไว้วางใจ และ ภัยคุกคาม 6.2 กระบวนการเข้ารหัส	3	0
10	6. ความปลอดภัยทางเครือข่าย 6.3 ขั้นตอนแลกเปลี่ยนกุญแจรหัสลับ	3	0
11	6. ความปลอดภัยทางเครือข่าย 6.4 ໂໂຣໄທຄອລີສູຈົນຕ້ວຈິງ 6.5 ຕ້າວອຍ່າງການໃຊ້ຈານຂອງຟັ່ງແວ່ງດ້ານຄວາມປລອດວັນ	3	0
12	7. ชັ້ນແອປພລິເຄື່ນ 7.1 ແອປພລິເຄື່ນດັ່ງເດີມ 7.2 ແອປພລິເຄື່ນປະເກດໂຄຮງສຽງພື້ນຖານ 7.3 ເທັນໂລຢີຄອນເທັນເນອົງ	3	0
13	8. ເຄື່ອງໝາຍໄຮສາຍປະເກດປະຫຼັດພັບງານ 8.1 ບລຸຫຼຸກ (IEEE 802.15.1) 8.2 ເທັນໂລຢີລອກຮາ	3	0
14	9. ກາຣອອກແບບເຄື່ອງໝາຍ 9.1 ກາຣອອກແບບເຄື່ອງໝາຍ 9.2 ວິເຄຣາະທີ່ເປົ້າໝາຍຂອງກາຣທຳຊຸກົງຈີແລະຂໍ້ຈຳກັດ 9.3 ກາຣວິເຄຣາະທີ່ດ້ານເທັນນິກີດແລະຂໍ້ຈຳກັດ	3	0
15	9. ກາຣອອກແບບເຄື່ອງໝາຍ 9.4 ທຳການເຂົ້າໃຈເຄື່ອງໝາຍເດີມລູກຄ້າ 9.5 Physical Network Design 9.6 Logical Network Design	3	0

16	9. การออกแบบเครื่อข่าย 9.7 การออกแบบเส้นทางสำรองทางเครื่อข่าย สรุปวิชา	3	0
17	สอบปลายภาค		

จุดประสงค์การสอน

บทเรียนที่	รายการ	เวลา(ชั่วโมง)	
		ท	ป
1	<p>พุทธิพิสัย</p> <p>1. รู้วัฒนาการการสื่อสารคอมพิวเตอร์</p> <p> 1.1 บอกพัฒนาการการสื่อสารของมนุษย์ได้</p> <p> 1.2 สรุปเทคโนโลยีเครือข่ายคอมพิวเตอร์ได้</p> <p> 1.3 ระบุประเภทสถานปัตยกรรมได้</p> <p>ด้านจิตพิสัย</p> <p> 1.1 แสดงความเห็นประเด็นต่าง ๆ ในบทเรียนได้</p>	3	0
2	<p>พุทธิพิสัย</p> <p>1. รู้วัฒนาการการสื่อสารคอมพิวเตอร์</p> <p> 1.4 ให้ความหมายของค่าประสิทธิภาพได้</p> <p>2. เข้าใจวิธีการเข้มต่อโดยตรง</p> <p> 2.1 อธิบายมุมมองเทคโนโลยีเครือข่ายในภาพกว้างได้</p> <p> 2.2 อธิบายกระบวนการอิเน็โนడีดีดีได้</p> <p> 2.3 อธิบายขั้นตอนทำเพรเมมได้</p> <p> 2.4 อธิบายวิธีการคำนวณการตรวจจับข้อผิดพลาดได้</p> <p>ด้านจิตพิสัย</p> <p> 2.1 ตั้งใจศึกษาบทเรียน</p>	3	0
3	<p>พุทธิพิสัย</p> <p>2. เข้าใจวิธีการเข้มต่อโดยตรง</p> <p> 2.5 บอกความหมายของการมีเสถียรภาพในการส่งข้อมูลได้</p> <p> 2.6 อธิบายความสำคัญของ Multi-Access Networks ได้</p> <p> 2.7 จำแนกประเภท Wireless Networks ได้</p> <p> 2.8 จำแนกประเภท Access Networks ได้</p> <p>ด้านจิตพิสัย</p> <p> 3.1 ตอบคำถามใน课堂เรียน</p>	3	0
4	พุทธิพิสัย	3	0

	<p>3. วิเคราะห์วิธีการค้นหาเส้นทางเครือข่ายคอมพิวเตอร์ และ เข้าใจโปรโตคอลเครือข่าย</p> <p>3.1 จำแนกคุณสมบัติพื้นฐานเนตเวิร์กสวิตช์ได้</p> <p>3.2 จำแนกการทำงานอีเทอร์เนตสวิตช์ได้</p> <p>3.3 ใช้สูตรการคำนวนอินเทอร์เน็ตໂປຣໂຫຄອລ (IP) ได้</p> <p>ด้านจิตพิสัย</p> <p>4.1 ตอบคำถามในภาคเรียน</p>		
5	<p>พุทธิพิสัย</p> <p>3. เข้าใจโปรโตคอลเครือข่าย</p> <p>3.4 อธิบายประเภท Routing ได้</p> <p>3.5 อธิบายโปรโตคอลไอพีรุ่น ๖ ได้</p> <p>ด้านจิตพิสัย</p> <p>5.1 ตอบคำถามในภาคเรียน</p>	3	0
6	<p>พุทธิพิสัย</p> <p>4. เข้าใจโปรโตคอลแบบ End-to-End</p> <p>4.1 ยกตัวอย่างแอปพลิเคชันที่หมายจะกับโปรโตคอล UDP ได้</p> <p>4.2 ยกตัวอย่างแอปพลิเคชันที่หมายจะกับโปรโตคอล TCP ได้</p> <p>ด้านจิตพิสัย</p> <p>6.1 ตอบคำถามในภาคเรียน</p>	3	0
7	<p>พุทธิพิสัย</p> <p>5. สังเคราะห์ประเด็นการควบคุมความคับคั่งได้</p> <p>5.1 อภิปรายประเด็นปัญหาการจัดสรรทรัพยากรเครือข่ายได้</p> <p>5.2 เสนอแนะวิธีใช้คิวสำหรับงานบางประเภทได้</p> <p>5.2 สรุปใจความสำคัญของ TCP Congestion Control ได้</p> <p>ด้านจิตพิสัย</p> <p>7.1 ตอบคำถามในภาคเรียน</p>	3	0
8	สอบกลางภาค		
9	<p>พุทธิพิสัย</p> <p>6. เข้าใจลักษณะปัญหาด้านความปลอดภัยทางเครือข่าย</p> <p>6.1 อธิบายความแตกต่างระหว่าง ความไว้วางใจ และ ภัยคุกคาม ได้</p>	3	0

	6.2 ยกตัวอย่างกระบวนการเข้ารหัสได้ ด้านจิตพิสัย 9.1 ตอบคำถามในคาบเรียน		
10	พุทธิพิสัย 6. เข้าใจลักษณะปัญหาด้านความปลอดภัยทางเครือข่าย 6.3 อธิบายขั้นตอนแลกเปลี่ยนกุญแจรหัสลับได้ ด้านจิตพิสัย 10.1 ตอบคำถามในคาบเรียน	3	0
11	พุทธิพิสัย 6. เข้าใจลักษณะปัญหาด้านความปลอดภัยทางเครือข่าย 6.4 อธิบายขั้นตอนพิสูจน์ตัวจริงได้ 6.5 ยกตัวอย่างการใช้งานซอฟต์แวร์ด้านความปลอดภัยได้ ด้านจิตพิสัย 11.1 ตอบคำถามในคาบเรียน	3	0
12	พุทธิพิสัย 7. เข้าใจการทำงานของแอปพลิเคชัน 7.1 ยกตัวอย่างแอปพลิเคชันดังเดิมได้ 7.2 อธิบายความสำคัญของแอปพลิเคชันประเภทโครงสร้างพื้นฐานได้ 7.3 อธิบายประโยชน์คุณแทนเนอร์ของ ด้านจิตพิสัย 12.1 ตอบคำถามในคาบเรียน	3	0
13	พุทธิพิสัย 8. รู้หลักการทำงานเครือข่ายไว้สายประเภทประกายดัดพลังงาน 8.1 บอกความถี่ที่บลูทูธ (IEEE 802.15.1) ใช้งานได้ 8.1 บอกความถี่ที่เทคโนโลยีโลหะไร้สายใช้ได้ ด้านจิตพิสัย 13.1 ตอบคำถามในคาบเรียน	3	0
14	พุทธิพิสัย	3	0

	<p>9. วิเคราะห์การออกแบบเครือข่ายได้</p> <p>9.1 จำแนกข้อแตกต่างของการออกแบบเครือข่ายได้</p> <p>9.2 เปรียบเทียบวิธีการวิเคราะห์เป้าหมายของการทำธุรกิจและข้อจำกัดได้</p> <p>9.3 เที่ยนโถะแกรมวิธีการวิเคราะห์ด้านเทคนิคและข้อจำกัดได้</p> <p>ด้านจิตพิสัย</p> <p>14.1 ตอบคำถามในภาคเรียน</p>		
15	<p>พุทธิพิสัย</p> <p>9. สังเคราะห์การออกแบบเครือข่ายได้</p> <p>9.4 สรุปแนวคิดวิธีเก็บข้อมูลเครือข่ายเดิมลูกค้าได้</p> <p>9.5 ออกแบบ Physical Network Design ได้</p> <p>9.6 ออกแบบ Logical Network Design ได้</p> <p>ด้านจิตพิสัย</p> <p>15.1 ตอบคำถามในภาคเรียน</p>	3	0
16	<p>พุทธิพิสัย</p> <p>9. ประเมินการออกแบบเครือข่ายได้</p> <p>9.7 อภิปรายการออกแบบเส้นทางสำรองทางเครือข่ายได้</p> <p>ด้านจิตพิสัย</p> <p>16.1 ตอบคำถามในภาคเรียน</p>	3	0
17	สอบปลายภาค		

ตารางกำหนดน้ำหนักคะแนน

เลขที่แบบรุ่น	คะแนนรายบทเรียนและน้ำหนักคะแนน ชื่อบทเรียน	ระดับคะแนน	น้ำหนักคะแนน				หัวข้อพิเศษ	
			พุทธิพิสัย					
			บุญธรรม-รู้-เคารพ	ความเข้าใจ	การนำไปใช้			
1	บทนำ	5	3	2				
2	การเข้มต่อโดยตรง	10	5	5				
3	โพรโทคอลเครือข่าย	20	2	5	3	10		
4	โพรโทคอล แบบ End-to-End	20	10	10				
5	การควบคุมความคับคั่ง	10		5		15		
6	ความปลอดภัยทางเครือข่าย	10		10				
7	ชั้นแอปพลิเคชัน	10		10				
8	เครือข่ายไร้สายประเภทประยุกต์พลังงาน	5	5					
9	การออกแบบเครือข่าย	10			5	5		
ก	คะแนนภาควิชาการ	100	25	47	8	30		
ข	คะแนนภาคผลงาน	0						
ค	คะแนนจิตพิสัย							
	รวมทั้งสิ้น	100						

กำหนดการสอน

สัปดาห์ที่	วัน / เดือน	คาบที่	รายการสอน	หมายเหตุ
1	31/10/2019	1	บทนำ	
2	7/11/2019	2	การเข้มต่อโดยตรง	
3	12/11/2019	3	การเข้มต่อโดยตรง	
4	28/11/2019	4	โพรโทคอลเครือข่าย	
5	12/12/2019	5	โพรโทคอลเครือข่าย	
6	16/01/2020	6	โพรโทคอล แบบ End-to-End	
7	23/01/2020	7	การควบคุมความคับคั่ง	
8	-		สอบกลางภาค	
9	27/01/2020	8	ความปลอดภัยทางเครือข่าย	
10	06/02/2020	9	ความปลอดภัยทางเครือข่าย	
11	13/02/2020	10	ความปลอดภัยทางเครือข่าย	
12	20/02/2020	11	ชั้นแอปพลิเคชัน	
13	27/02/2020	12	เครือข่ายไร้สายประเทียดพลังงาน	
14	05/03/2020	13	การออกแบบเครือข่าย	
15	12/03/2020	14	การออกแบบเครือข่าย	
16	19/03/2020	15	การออกแบบเครือข่าย	
17	-		สอบปลายภาค	

ສັນຍາອນຸມາຕ

ເນື້ອຫາແລະກາພທີ່ປ່ຽກງົງໃນເອກສາຮນີ້ ໄດ້ຮັບອນຸມາຕໃຫ້ຜລິຕໍ່ສຳຫວັງວ່າໄດ້ ກາຍໃຕ້ເງື່ອນໄຂສັນຍາອນຸມາຕ
CC BY 4.0(Creative Commons Attribution 4.0 International Public License)

Title: Computer Networks: A Systems Approach

Authors: Larry Peterson and Bruce Davie

Copyright: Elsevier, 2012

Source: <https://github.com/SystemsApproach>

License: CC BY 4.0

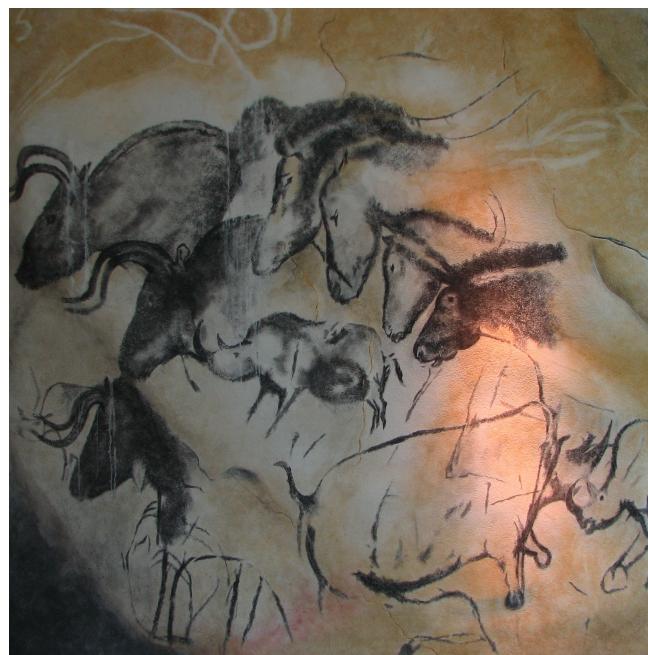
บทที่ 1

บทนำ

1.1 พัฒนาการการสื่อสารของมนุษย์

หลังเกิดการระบาดใหญ่ของไวรัสโควิด-19 ปี ปลายปี พ.ศ. 2562 ส่งผลให้เกิดการเปลี่ยนแปลงทางสังคม ผู้คนยอมกับการปฏิบัติงานผ่านเครือข่ายมากขึ้น เครือข่ายคอมพิวเตอร์กล้ายเป็นเทคโนโลยีพื้นฐานสำคัญสำหรับการดำเนินชีวิตวิถีใหม่ (New normal) มีการใช้งานคอมพิวเตอร์ผ่านระบบเครือข่ายคอมพิวเตอร์มากขึ้นเป็นประวัติการณ์ ผลสำรวจประชากรอเมริกัน โดย Center (2021) พบกว่า 90% ยอมรับว่า อินเทอร์เน็ต (internet) มีความสำคัญต่อการดำรงชีพ ส่งผลกระทบต่อการเปลี่ยนแปลงวิถีชีวิต เช่นการทำงานออนไลน์ การเรียนออนไลน์ การเปลี่ยนแปลงเหล่านี้ส่งผลต่อภาพรวมของโลก ซึ่งการสื่อสารผ่านเครือข่ายอินเทอร์เน็ตอย่างมีประสิทธิภาพมีพื้นฐานจากเทคโนโลยีการสื่อสารผ่านเครือข่ายคอมพิวเตอร์

การติดต่อสื่อสารระหว่างมนุษย์ด้วยกันเองมีมาตั้งแต่ก่อนประวัติศาสตร์ มนุษย์มีวิวัฒนาการสื่อสารมาโดยตลอด มีหลักฐานภาพวาดปรา加ภูบันแห่นหินในถ้ำโชเวต์ (Chauvet-Pont-d'Arc Cave) ประเทศฝรั่งเศส ตัวอย่างปรา加ภูในรูปที่ 1.1



รูปที่ 1.1: ภาพสัตว์ปรา加ภูบันแห่นหินในถ้ำโชเวต์

ลิขสิทธิ์ภาพ Public Domain แหล่งที่มา [https://commons.wikimedia.org/wiki/File:Paintings_from_the_Chauvet_cave_\(museum_replica\).jpg](https://commons.wikimedia.org/wiki/File:Paintings_from_the_Chauvet_cave_(museum_replica).jpg)

นอกจากนั้นยังพบหลักฐานรูปภาพที่พับในถ้ำลากอและถ้ำแม่โกร์ ทำให้เชื่อได้ว่ามนุษย์อาศัยในบริเวณดังกล่าว ตั้งแต่ 17,000 ปีที่ผ่านมา ได้สื่อสารโดยใช้รูปสัญลักษณ์เป็นตัวแทนการสื่อสารด้วยเสียงซึ่งช่วย

ลดความผิดพลาดจากการออกเสียงไม่เหมือนกันของมนุษย์ โดยใช้รูปวดบนจำเพงแสดงถึงการสื่อสารภาษาด้วยภาพ เช่น รูปสัตว์ รูปมนุษย์ รวมถึงรูปสัญลักษณ์อื่นๆ และมีการแลกเปลี่ยนความคิดเห็นกันอย่างมีรูปแบบซึ่งต่อมา มีการติดต่อสื่อสารโดยใช้ธาราที่หลากหลายรูปแบบมากขึ้น เช่น

- การใช้เสียง เสียงจากการเปล่งเสียงของผู้ส่ง เสียงจากเครื่องมือ เช่น เคาะไม้ เป่ากลิ้ง และเสียงอื่น ๆ
- การใช้ทศนสัญญาณ ควันไฟ กระจาดหัวอนแสง สัญญาณร่องต่าง ๆ
- การใช้ตัวหนังสือ การเขียนข้อความบนหนังสัตว์ ส่งทางม้าเร็ว ใช้คนพิราบ เป็นต้น

1.1.1 โทรคมนาคม: นิยามและความหมาย

โทรคมนาคม “หมายถึงการส่ง หรือการรับเครื่องหมายสัญญาณ ตัวหนังสือ หรือทางระบบแม่เหล็กไฟฟ้าอื่นๆ จากจุดหนึ่งไปยังอีกจุดหนึ่ง ที่อยู่ห่างไกลกันโดยกรรมวิธีทางวิศวกรรมสื่อสาร ([สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ \(2017\)](#))”

โทรคมนาคมมาจากคำว่า โทร และ คมนาคม “โทร” หมายถึง ไกล ขณะที่ “คมนาคม” หมายถึง การติดต่อสื่อสาร จึงสรุปความหมายของโทรคมนาคมคือ การติดต่อสื่อสารระยะไกล หรือการส่งรับข่าวสารทางไกล

1.1.2 พื้นฐานเทคนิคทั่วไป

การสื่อสารสามารถจัดกลุ่มตามรูปแบบสื่อสารได้แก่ สื่อมีตัวนำ และ สื่อไม่มีตัวนำ

สื่อมีตัวนำ เป็นการส่งสัญญาณผ่านตัวนำซึ่งอาจมีสายสัญญาณหรือไม่มีสายสัญญาณก็ได้ การสื่อสารผ่านสายสัญญาณเช่น การสื่อสารข้อมูลโดยใช้สายนำสัญญาณเช่น ทองแดง ในการนำสัญญาณจากต้นทางไปปลายทาง การสื่อสารไม่มีสายสัญญาณเช่นการสื่อสารด้วยคลื่นสั่นสะเทือนโดยใช้อากาศหรือน้ำ เช่นการส่งสัญญาณคลื่นเหนือเสียง เป็นต้น

สื่อแบบไม่มีตัวนำ ในที่นี้หมายถึงการสื่อสารผ่านคลื่นแม่เหล็กไฟฟ้าซึ่งไม่ต้องการตัวนำสัญญาณ ซึ่งไม่ต้องการสายนำสัญญาณ ท่อน้ำสัญญาณ หรือบรรยายกาศในการนำสัญญาณ แต่ใช้การกระจายผ่านโดยคลื่นแม่เหล็กไฟฟ้าทำให้สามารถสื่อสารในอากาศได้

1.2 เทคโนโลยีเครือข่ายคอมพิวเตอร์

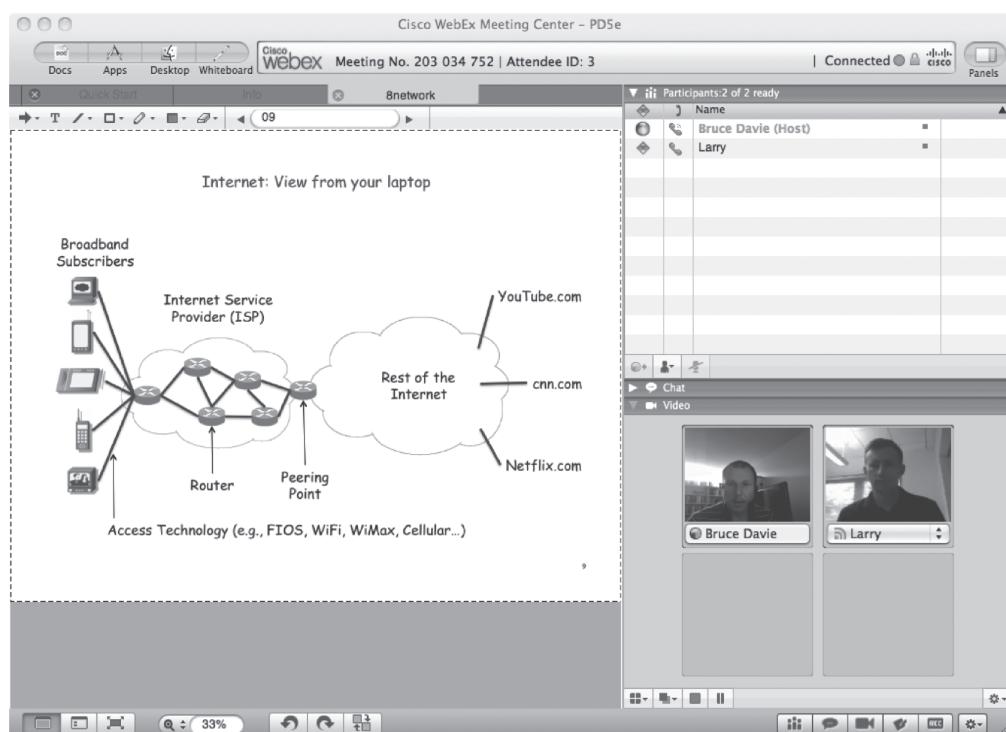
ผู้พัฒนาทฤษฎีแกรคอล [Kleinrock \(2010\)](#) ซึ่งเป็นพื้นฐานสำคัญให้เกิดระบบ อินเทอร์เน็ต ได้กล่าวไว้ว่า ช่วงเวลาเริ่มต้นของเครือข่ายอินเทอร์เน็ตไม่ปรากฏแน่นชัด แต่เครือข่ายอินเทอร์เน็ตมีการใช้งานจริงเมื่อมหาวิทยาลัยในสหรัฐอเมริกา ต้องการแลกเปลี่ยนข้อมูล ภายในแต่ละมหาวิทยาลัยสามารถแลกเปลี่ยนกันภายในมหาวิทยาลัยมาก่อน แล้วจึงมีความต้องการให้มีระบบสื่อสารที่เชื่อมโยงระหว่างกันโดยมีรูปแบบใกล้เคียงกัน

ระบบเครือข่ายคอมพิวเตอร์ กล่าวถึงระบบที่นำพาข้อมูลในรูปแบบดิจิทัลสื่อสารผ่านระยะทางไกล และไกลได้ ความสามารถในการสื่อสารได้เนี้ยบคงมีวิวัฒนาการต่อเนื่อง ซึ่งปัจจุบันรูปแบบสื่อสารที่กว้างใหญ่

ที่สุดเท่าที่โลกเคยมีมาเป็นการสื่อสารอ่านเครือข่ายอินเทอร์เน็ต ถึงแม้ไม่อาจระบุวันที่แน่ชัดของต้นกำเนิดที่นำมาสู่เทคโนโลยีอินเทอร์ได้ ในหัวข้อนี้ได้เรียบเรียงนับจากวิวัฒนาการสื่อสารของมนุษย์จนถึงปัจจุบัน

1.2.1 แอปพลิเคชัน

ปัจจุบันนี้คำว่า แอปพลิเคชัน เป็นคำที่คุ้นหูกันทั่วไปมากเสียกว่าคำว่าอินเทอร์เน็ต ผู้คนใช้แอปพลิเคชันโดยไม่ต้องทราบเทคโนโลยีเบื้องหลัง ซึ่งแอปพลิเคชันใช้เรียก ซอฟต์แวร์(software) ที่ใช้บริการอินเทอร์เน็ตหรือไม่ก็ได้ แต่ส่วนใหญ่แอปพลิเคชันที่ติดตั้งบนสมาร์ทโฟน เป็นแอปพลิเคชันใช้ประโยชน์จากการเชื่อมต่อทางอินเทอร์เน็ต ด้วยอย่างเช่น แอปพลิเคชันประเภทสังคมออนไลน์ เช่น เฟชบุ๊ก(Facebook) ไลน์(LINE) ยูทูบ(Youtube) เป็นต้น



รูปที่ 1.2: การใช้แอปพลิเคชันในการประชุมออนไลน์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

มีการประยุกต์ใช้เครือข่ายอินเทอร์เน็ตสำหรับใช้ประชุมทางไกล (วีดีโອคอนเฟอเรนซ์) ตั้งแต่ปี ค.ศ. 1990 เป็นต้นมา ผู้ใช้งานเพียงล็อกอินเข้าเว็บไซต์(website)สำหรับประชุมทางไกลจะเกิดการติดตั้งแอปพลิเคชันโดยอัตโนมัติใช้เวลาเพียงไม่กี่วินาที ผู้ใช้งานจะสามารถประชุมทางไกลได้แม้ไม่เคยประชุมทางไกลด้วยแอปพลิเคชันนี้มาก่อนจากรูปที่ 1.2 แสดงตัวอย่างการใช้แอปพลิเคชันประชุมทางไกลซึ่งมีข้อมูลแผนภูมิและภาพเคลื่อนไหวของผู้ร่วมประชุม

1.2.2 ความต้องการของผู้ใช้เครือข่าย

เป้าหมายของการเรียนนี้มีเพื่อวัตถุประสงค์ของการพัฒนาเทคโนโลยีอินเทอร์เน็ตเพื่อให้เข้าใจที่มาการสื่อสาร ผ่านเกิดขึ้นได้อย่างไรหรือแนวทางที่จะเริ่มต้นจากการบูรณาissanของระบบและตอบคำถามในมุมมองผู้ใช้ ซึ่งเป็น คำถามประเภท why (ทำไม ?) ก่อนอธิบายการทำงานทางเทคนิค ซึ่งตอบตามประเภท how (อย่างไร ?) ใน ลำดับต่อไป

องค์ประกอบที่เกี่ยวข้องกับอินเทอร์เน็ต

ในการอธิบาย อินเทอร์เน็ต ผ่านมุมมองผู้ใช้ เกี่ยวข้องกับการทำให้อินเทอร์เน็ตทำงานได้ อินเทอร์เน็ต กระจาย ได้ กว้าง ขวาง ไม่ได้เกิดจากการเชื่อมต่ออินเทอร์เน็ตแต่เกิดจาก จุดเริ่มที่ผู้ใช้งานเชื่อมเครือข่ายผ่าน โมเด็ม (modulator-demodulator) ไป ระบบกระดาษขาว(bulletin board system) เกิดจากมหาวิทยาลัย เชื่อมต่อ เครือข่ายภายใน และต้องการเชื่อมต่อไปมหาวิทยาลัยภายนอก ทำให้เกิดการขยายจากกลุ่มเล็กเพิ่มจำนวนขึ้น ตามลำดับ

องค์ประกอบทำให้เกิดการเชื่อมต่อผ่านอินเทอร์เน็ตนั้นเกี่ยวข้องกับส่วนประกอบหลักส่วน บทนี้ได้ อธิบายถึง ส่วนประกอบหลักสำคัญที่ทำให้อินเทอร์เน็ตสามารถเชื่อมต่อกันได้ซึ่งองค์ประกอบนี้ประกอบไปด้วย ข้อมูลที่เป็นภาษาภาพ และข้อมูลที่เป็นนามธรรม

สำหรับหนังสือเล่มนี้มีองค์ประกอบเครือข่ายสามกลุ่มได้แก่ นักพัฒนาซอฟต์แวร์ที่ใช้งานเครือข่าย นัก ออกแบบเครือข่าย และ ผู้ให้บริการเครือข่าย

- นักพัฒนาซอฟต์แวร์ที่ใช้งานเครือข่าย มีความต้องการให้เครือข่ายสามารถให้บริการได้หลากหลาย รองรับการรันซอฟต์แวร์หลายรูปแบบผ่านเครือข่ายที่มี
- นักออกแบบเครือข่าย มีความต้องการออกแบบเครือข่ายให้คุ้มค่าต่อการลงทุน ใช้ทรัพยากรน้อยได้ ผลลัพธ์มาก
- ผู้ให้บริการเครือข่าย สนใจการบริหารจัดการเครือข่ายที่สอดคล้องกับตัวตน ต้นทุนต่ำ

ในหัวข้อต่อไปจะกล่าวลงในรายละเอียด ที่ผู้ใช้งานเครือข่ายมีความต้องการแตกต่างกัน เป็นกุญแจสู่ การออกแบบเครือข่ายเพื่อตอบสนองความต้องการ และแรงจูงใจต่างๆ ที่ทำให้มีการศึกษาเรียนรู้ในวิชาเครือ ข่ายคอมพิวเตอร์

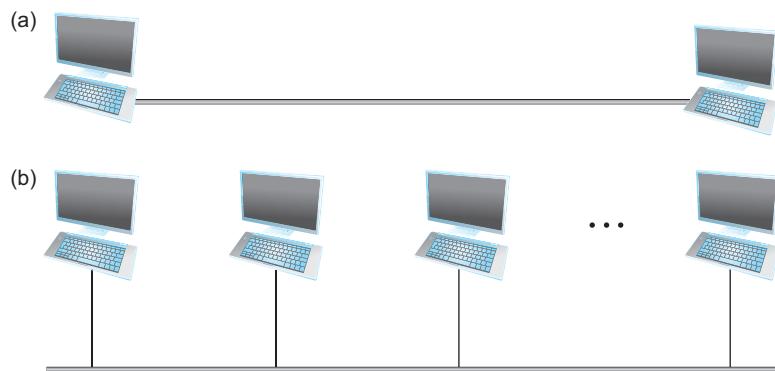
ความสามารถในการขยายตัว(scalability)

หนึ่งในความสามารถที่ประسังค์ของเครือข่ายได้แก่ สามารถขยายพื้นที่ให้บริการการสื่อสารได้ในอนาคต เรียก ความสามารถนี้ว่า “การขยายตัว” วิวัฒนาการทางเทคโนโลยีเครือข่ายที่ผ่านมา มีหลายช่วงเวลาที่มีการคิดค้น เทคโนโลยีด้านการสื่อสารที่มีประสิทธิภาพแต่ไม่ได้รับการยอมรับแพร่หลาย สาเหตุหนึ่งเกิดจากมีเทคโนโลยี ด้านสื่อสารอื่นที่มีประสิทธิภาพเหนือกว่า ปัจจัยสำคัญได้แก่ คุณสมบัติประสิทธิภาพด้านการขยายตัว เมื่อต้อง การขยายความสามารถให้บริการแต่ต้นทุนในการขยายพื้นที่สูงกว่าเทคโนโลยีอื่น

การสื่อสารผ่านระบบเครือข่ายยุคเริ่มต้นนั้นมีวัตถุประสงค์เพื่อให้คุณสื่อสารสามารถสื่อสารระหว่างกันได้โดยตรง แล้วจึงขยายความต้องการเป็นสื่อสารได้หลายคน และให้หลายคนสามารถสื่อสารระหว่างกันได้ เพื่อทำความเข้าใจที่มารูปแบบการสื่อสารในปัจจุบัน เนื้อหาส่วนนี้จะเริ่มทำความเข้าใจจากการเชื่อมต่อเครือข่าย จากรูปที่ 1.3(a) เป็นการเชื่อมโดยตรงระหว่างคู่สื่อสารและรูปที่ 1.3(b) เป็นการนำคู่สื่อสารที่เกิดจากการเชื่อมต่อโดยตรงมาเชื่อมกันหลายเครื่องเรียกว่า การเข้าถึงแบบหลายจุด(multiple-access)

การเชื่อมต่อโดยตรงเป็นพื้นฐานการเชื่อมต่อที่ทำให้คุณสื่อสารตันทางและปลายทางสามารถสื่อสารกันได้ แต่เมื่อเป็นการสื่อสารระยะไกลที่ต้นทางมีผู้ต้องการสื่อสารจำนวนเพิ่มขึ้น เช่นเดียวกับปลายทาง แต่มีข้อจำกัดจากสายสื่อสารที่ไม่อาจสร้างได้เท่ากับจำนวนการเชื่อมต่อ จึงเกิดการพัฒนาเทคโนโลยีที่สามารถแข่งข่ายสื่อสารร่วมกันดังรูปที่ 1.3(b)

การเข้าถึงแบบหลายจุดจากที่แสดงในรูปที่ 1.3(b) ประยุกต์ใช้ในอินเทอร์เน็ตจากการออกแบบให้ส่งข้อมูลแต่ละครั้งส่งเป็นชุดข้อมูลเรียกว่า แพ็กเก็ต โดยแพ็กเก็ตนี้จะถูกส่งต่อผ่านอุปกรณ์ที่สามารถเก็บข้อมูลได้ ชั่วคราวและสามารถส่งต่อได้มีอัลลงเวลา วิธีนี้เรียกว่า สโตร์แอนฟอร์เวิร์ด(store-and-forward) กระบวนการส่งต่อแพ็กเก็ต แบบ สโตร์แอนฟอร์เวิร์ด เป็นวิธีการรอจังหวะเวลาที่สามารถส่งข้อมูลได้ เรียกว่า แพ็กเก็ตสวิตชิ่ง



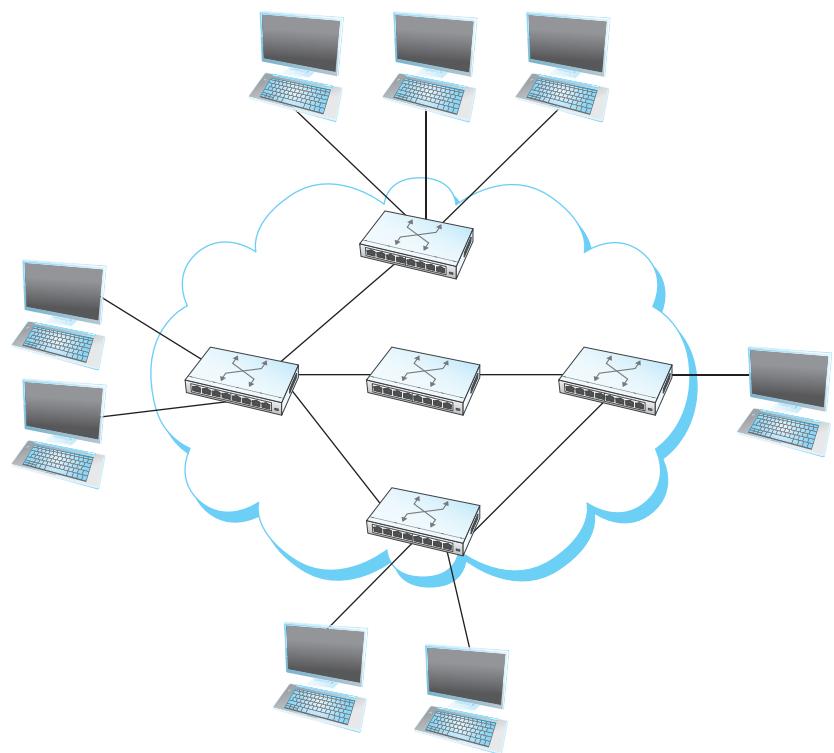
รูปที่ 1.3: ไดเรกสิงค์ (a) point-to-point (b) multiple-access
ลิขสิทธิภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

อุปกรณ์ประเภทแพ็กเก็ตสวิตชิ่ง เชื่อมต่อตามรูปที่ 1.4 ทำหน้าที่รับข้อมูลจากไฮสต์ที่ต่อโดยตรงแล้ว รอจังหวะที่สามารถส่งข้อมูลได้ แล้วส่งต่อไปอุปกรณ์ต่อไปจนถึงปลายทาง

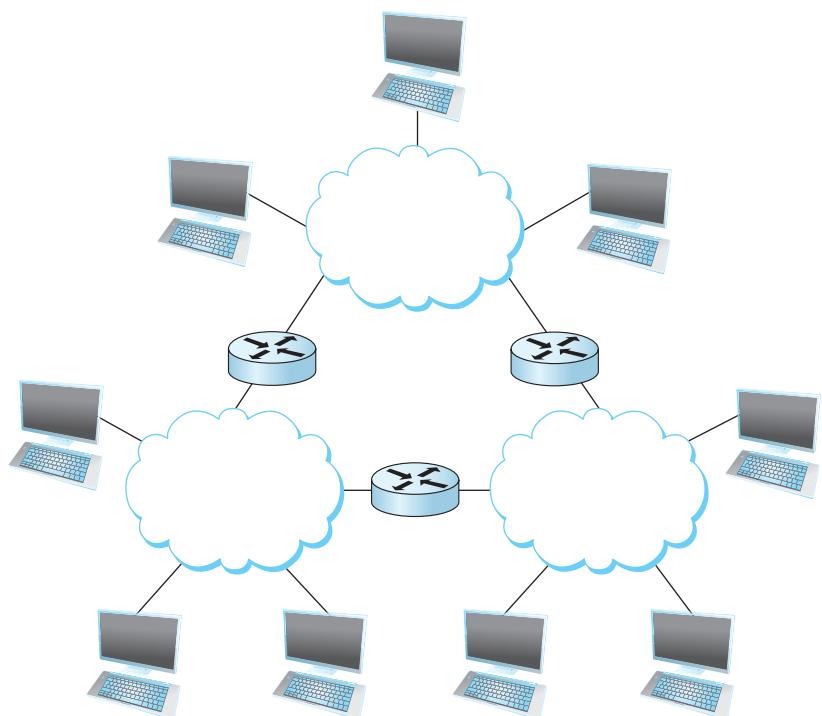
การสื่อสารผ่านเครือข่ายเริ่มจากการเชื่อมต่อโดยตรง เมื่อมีผู้ใช้งานจำนวนมากต่อมานี้การเชื่อมต่อแบบการเข้าถึงแบบหลายจุด ผ่านอุปกรณ์แพ็กเก็ตสวิตชิ่ง ทำให้เครือข่ายภายในที่สามารถเชื่อมต่อไฮสต์จำนวนมาก ต่อมามีเครือข่ายนี้ต้องการเชื่อมต่อไปยังเครือข่ายภายนอกจะทำได้โดยการเชื่อมระหว่างเครือข่ายตามรูปที่ 1.5 เกิดเป็นอินเทอร์เน็ตในปัจจุบัน

Cost-Effective Resource Sharing

การใช้ช่องสัญญาณซึ่งมีจำนวนจำกัด ให้ใช้งานได้อย่างมีประสิทธิภาพ ใช้วิธีการแบ่งช่วงเวลาให้แต่ละข้อมูลสามารถใช้ได้ตามเวลาที่ตนมี วิธีสำคัญสำหรับการแบ่งการทำงานนี้เรียกว่า การรวมสัญญาณ การทำงานการ

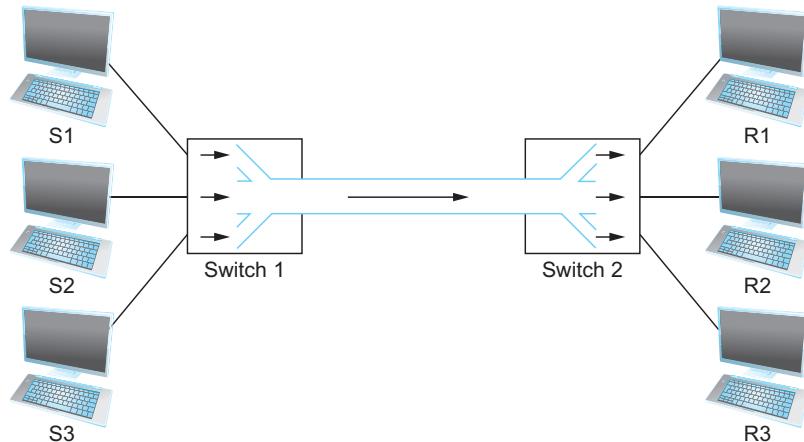


รูปที่ 1.4: เครือข่ายสวิตซ์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>



รูปที่ 1.5: การเชื่อมต่อ helyo เครือข่ายเข้าด้วยกันผ่านตารางเรขาตั้ง
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

รวมสัญญาณเป็นไปตามรูปที่ 1.6 เมื่อจะส่งข้อมูลเข้าช่องสัญญาณที่ใช้ร่วมกันเรียกว่า การรวมสัญญาณ และ เมื่อต้องการถอดข้อมูลออกจากช่องสัญญาณเรียกว่า การถอดสัญญาณ(demultiplex)



รูปที่ 1.6: การทำการรวมสัญญาณจากไฮสต์ทลายเครือข่ายใช้ช่องสัญญาณร่วมกัน
ลิขสิทธิภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

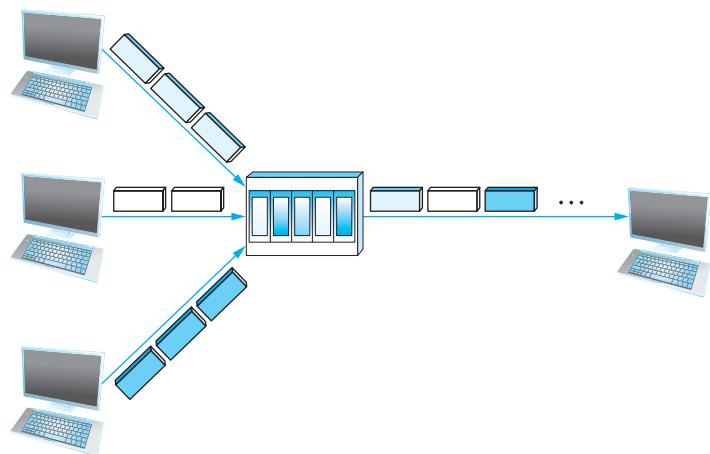
วิธีการสลับช่วงเวลาเพื่อให้ไฮสต์ใช้ช่องสัญญาณร่วมกันสามารถสรุปได้ 2 แนวทาง ได้แก่ การรวมสัญญาณทางเวลาแบบเข้าจังหวะ(synchronous time-division multiplexing) และ การรวมสัญญาณทางความถี่(frequency-division multiplexing) แนวคิด STDM(การรวมสัญญาณทางเวลาแบบเข้าจังหวะ) เป็นวิธีจัดสรรเวลาให้แก่แต่ละไฮสต์ให้ได้จำนวนเวลาเท่ากัน วิธีที่นิยมได้แก่ 輪流權限(round-robin) โดยกำหนดให้ไฮสต์มีเวลาในการส่งข้อมูลด้วยเวลาเท่ากัน เรียกว่า คوانต้า(quanta) หรือเรียกว่า 1 คوانตัม(quantum) จากรูปที่ 1.6 เมื่อ S1 ต้องการส่งข้อมูลไป R1 ได้ส่งข้อมูลในคوانตัมที่ 1 สำหรับคوانตัมที่ 2 กำหนดให้เป็นช่วงเวลาที่ S2 ได้ส่งไป R2 ใน คwanตัมที่ 3 เป็นช่วงเวลาที่ S3 ส่งไป R3 จากจุดนี้ หากเส้นทางแตก (S1 ไป R1) ต้องการส่งข้อมูลอีกครั้งจะต้องรอให้ไฮสต์อื่นส่งต่อครบรอบ ทำให้เป็นที่มาของการแบ่งข้อมูลแบบ輪流權限(RR)

อีกแนวทางใช้ FDM(การรวมสัญญาณทางความถี่) ใช้วิธีกำหนดความถี่ของข้อมูลให้แตกต่างกันทำให้ใช้ช่องร่วมกันได้และไม่รบกวนกัน แนวคิดนี้เป็นวิธีเดียวกับการส่งสัญญาณโทรทัศน์ที่แต่ละช่องส่งความถี่แตกต่างกัน ผู้ใช้สามารถเลือกช่องได้ครั้งละ 1 ช่อง

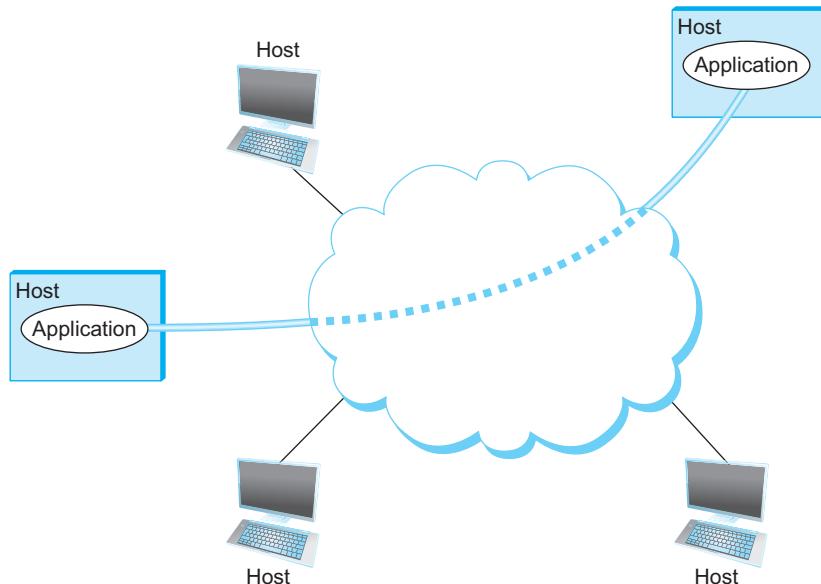
จากรูปที่ 1.7 เป็นวิธีกำหนดให้มีแพ็กเก็ตเป็นก้อน ส่งครั้งละก้อน โดยมีสวิตซ์ทำหน้าที่ สโตร์แอนฟอร์เวิร์ดโดยเรียงข้อมูลลงบนสายสัญญาณเส้นเดียว

การรองรับบริการสำหรับใช้ร่วมกัน

เครือข่ายอินเทอร์เน็ตเป็นโครงข่ายที่เชื่อมสายสัญญาณร่วมกันขนาดใหญ่จากรูปที่ 1.8 แสดงตัวอย่างการใช้สายสัญญาณร่วมกันของเครือข่ายอินเทอร์เน็ตที่ส่งจากไฮสต์ต้นทางไปไฮสต์ปลายทางผ่านแอปพลิเคชันใดๆ



รูปที่ 1.7: การใช้แพ็กเกจสวิตซ์สำหรับใช้สายสัญญาณร่วมกัน
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>



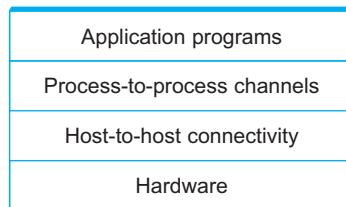
รูปที่ 1.8: อธิบายกระบวนการสื่อสารโดยใช้แนวคิดนามธรรม
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

1.3 สถาปัตยกรรม

จากที่กล่าวมาได้กล่าวถึงข้อมูลที่ทำให้เกิดการเชื่อมต่อเครือข่ายและเทคนิคไวริที่เกี่ยวข้องในการใช้ช่องสัญญาณร่วมกัน สำหรับการออกแบบวิธีการสื่อสารนั้นมีความเกี่ยวข้องกับองค์ประกอบอื่นๆ ซึ่งการทำความเข้าใจให้ง่าย จึงแบ่งการติดต่อสื่อสารเป็นลำดับชั้นเรียกว่า เลเยอร์(layer) สำหรับการเรียนรู้ลำดับการทำงานเครือข่ายคอมพิวเตอร์ได้กำหนดตัวแบบ(model)มาตรฐานชื่อว่า OSI 7-layer เพื่อใช้เป็นตัวแบบกลางในการกล่าวถึงสถาปัตยกรรมการสื่อสารทางคอมพิวเตอร์ อย่างไรก็ตามรูปแบบการกำหนดสถาปัตยกรรมที่ใช้งานจริงยังเป็นไปตามแนวคิดของผู้ออกแบบ ซึ่งการสื่อสารผ่านเครือข่ายอินเทอร์เน็ตมีการใช้งาน OSI 7-layer เพียงบางส่วน

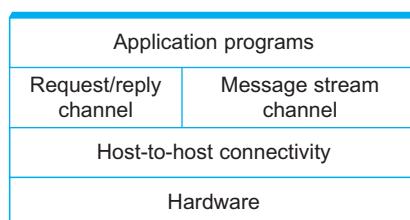
1.3.1 เลเยอร์ริง และ พอร์โทคอล

การทำความเข้าใจระบบเลเยอร์ใช้แนวคิดแบบนามธรรม และช่องข้อมูลที่ซับซ้อนไว้เบื้องหลังโดยแบ่งข้อมูลซับซ้อนออกเป็นลำดับชั้นและมีส่วนติดต่ออย่างสำหรับเรียกใช้งาน จากรูปที่ 1.9 เป็นรูปที่นิยมใช้อธิบายการทำงานเป็นเลเยอร์จากรูปมีทั้งหมด 4 เลเยอร์ ได้แก่ Application program, Process-to-process channels, Host-to-host connectivity และ Hardware การแบ่งส่วนออกเป็น 4 เลเยอร์ทำให้การออกแบบทำได้ง่ายขึ้น โดยช่องความซับซ้อนไว้เบื้องหลังของแต่ละเลเยอร์ ซึ่งในเลเยอร์ชั้นเดียวกันอาจจะแบบเป็นกลุ่มย่อยได้ตามรูปที่ 1.10



รูปที่ 1.9: ตัวอย่างระบบเครือข่าย
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

จากรูปที่ 1.10 ในชั้น Request/replay channel อยู่ชั้นเดียวกันกับ Message stream channel

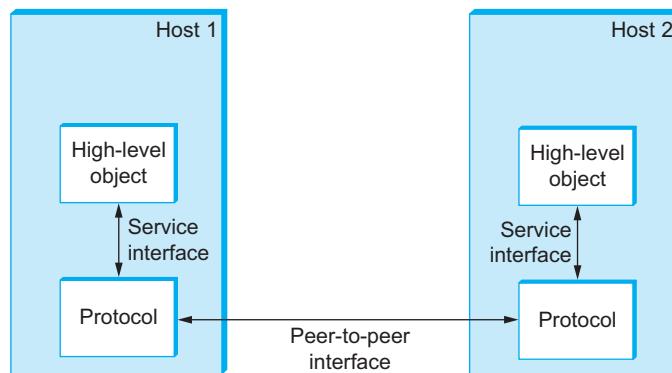


รูปที่ 1.10: ระบบเลเยอร์ที่ระดับเดียวกันมีหลายทางเลือก
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

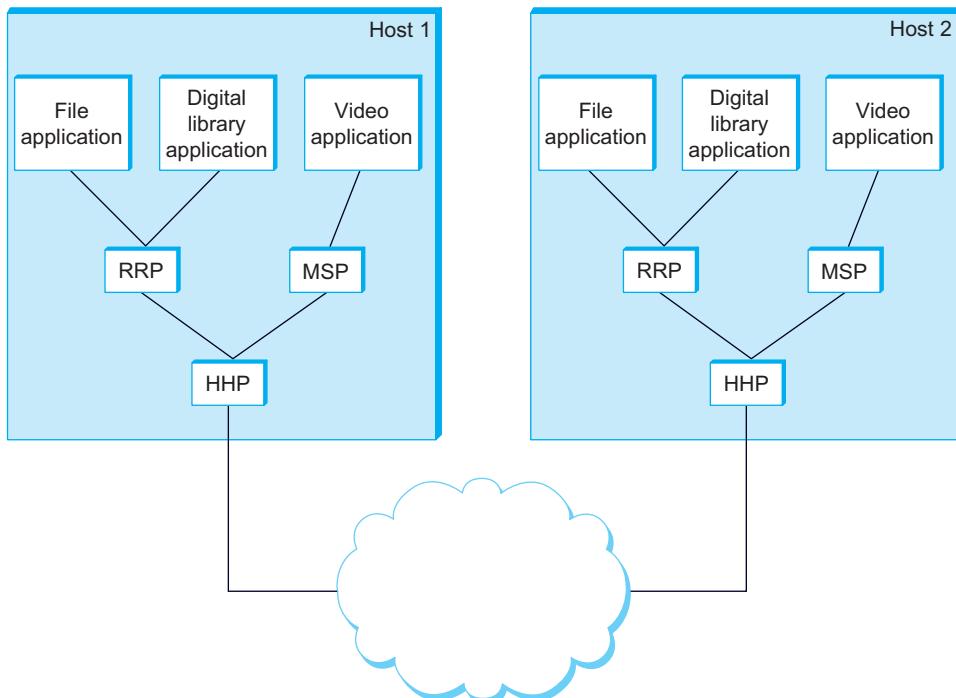
ในการออกแบบเลเยอร์จากที่กล่าวมาเป็นการกำหนดชั้นสำหรับติดต่อสื่อสาร ซึ่งแต่ละชั้นที่ต้องการการสื่อสารจะกำหนดรูปแบบวิธีที่ใช้ในการสื่อสาร รูปแบบวิธีนั้นเรียกว่า พอร์โทคอล

ตัวอย่างเช่นรูปที่ 1.11 กำหนดให้ Host 1 สื่อสารกับ Host 2 ซึ่งลำดับการสื่อสารนี้ต้องการเลเยอร์ 2 ชั้น ได้แก่ High-level object และ Protocol เมื่อ Host 1 ส่งข้อมูลไป Host 2 จะทำผ่านพอร์โทคอล โดยข้อมูลตั้งต้นนั้นเกิดที่ชั้น High-level object

รูปที่ 1.12 อธิบายพอร์โทคอลที่พบรูปในการสื่อสารซึ่งพอร์โทคอลหนึ่งสามารถเชื่อมต่อกับพอร์โทคอลอื่นได้ผ่านข้อตกลงที่เป็นนามธรรม ก่อนเกิดการเชื่อมต่อไปโโซต์อื่นผ่าน HHP(Host-to-Host Protocol) การอธิบายการทำงานแต่ละพอร์โทคอลใช้วิธีแนบข้อมูลของพอร์โทคอลไปกับแพ็คเก็ต เรียบกระบวนการแนบข้อมูลนี้ว่า เอ็นแคปชูเลชัน(encapsulation)



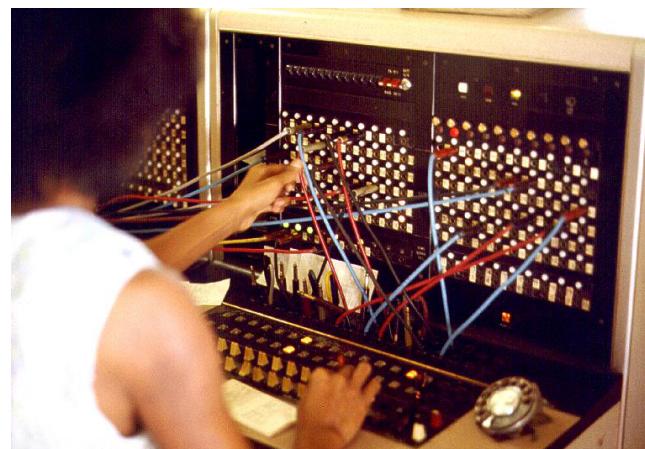
รูปที่ 1.11: ส่วนบริการการเชื่อมต่อและการสื่อสารภายในเลเยอร์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>



รูปที่ 1.12: ตัวอย่างแผนภาพprotocol
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

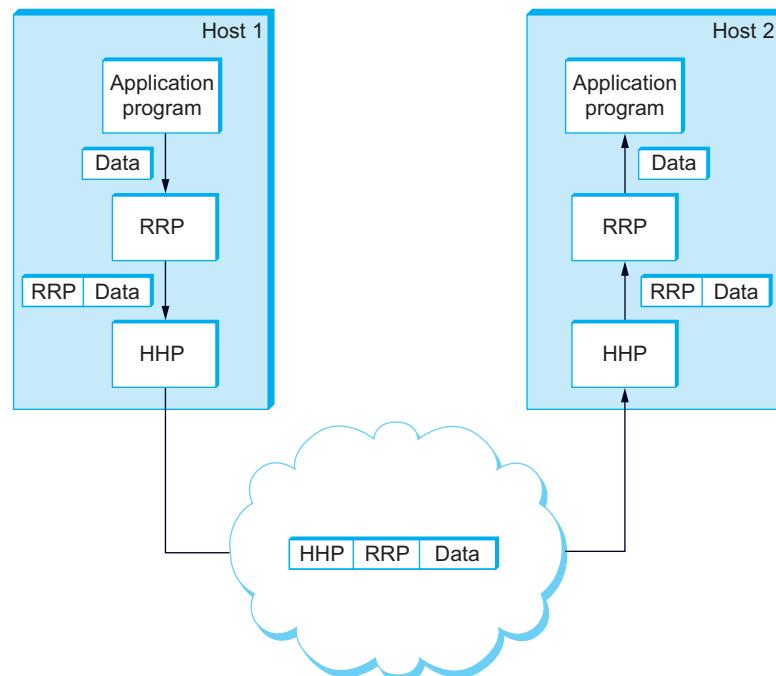
1.3.2 การรวมสัญญาณและการถอดสัญญาณ

นับตั้งแต่มีการสื่อสารผ่านสัญญาณทางไฟฟ้าทำให้สามารถสื่อสารได้ด้วยความเร็วสูงกว่าการสื่อสารรูปแบบเก่าที่ผ่าน เทคโนโลยีพื้นฐานที่สำคัญทำให้การสื่อสารสัญญาณไฟฟ้าสามารถสื่อสารได้หลายเครื่อง คือการสลับสัญญาณ การสลับสัญญาณที่มีอินพุทเข้ามากมากแต่ส่งออกเพียงหนึ่งเรียกว่า การรวมสัญญาณ และถ้าอินพุทเข้ามาหนึ่งแต่ออกรายเส้นทางเรียกว่า การถอดสัญญาณ แนวคิด การรวมสัญญาณ และ การถอดสัญญาณ นำมาใช้ในการสลับสัญญาณโทรศัพท์แบบสายมาตั้งแต่อดีตโดยมีมุนุชย์เป็นผู้สลับสัญญาณดังภาพที่ 1.13



รูปที่ 1.13: Jersey Telecom telephone operator at switchboard 1975

พื้นฐานของการใช้ช่องสัญญาณร่วมกันเกิดจากเทคนิคการรวมสัญญาณและการถอดสัญญาณ ได้อธิบายในรูปที่ 1.12 ซึ่ง RRP(Request/Reply Protocol) ทำหน้าที่การรวมสัญญาณและการถอดสัญญาณ เมื่อข้อมูลส่งมาจากชั้น File application หรือ Digital library application จะเข้าชั้นตอนการรวมสัญญาณ ซึ่งใช้โพรโทคอล RRP ลำดับการสื่อสารเป็นไปตามรูปที่ 1.14 จากรูปข้างต้นการอ่านแคปชูเลชันขั้นสุดท้าย ก่อนส่งข้อมูลอุปกรณ์จะเป็นชั้นตอน HHP



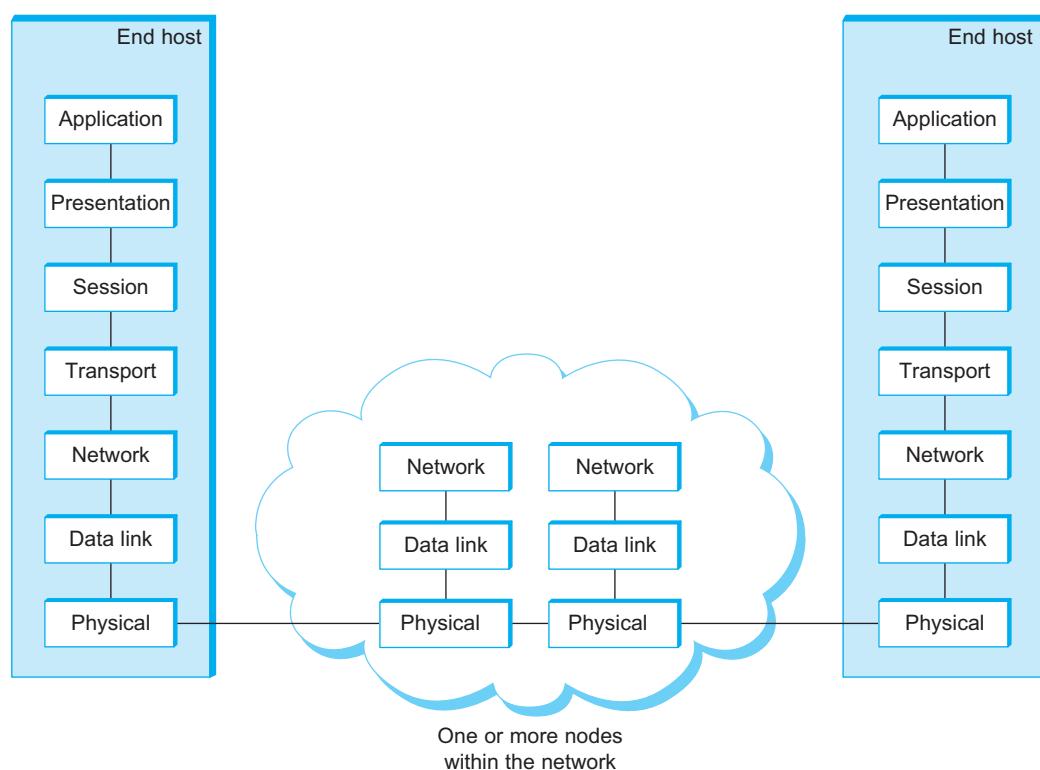
รูปที่ 1.14: ข้อความที่ส่งไปแลຍอร์สูงขึ้นเมื่อชั้นตอนแพ็คข้อมูล
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

แนวทางในการส่งต่อสัญญาณจากต้นทางไปถึงปลายทางผ่านตัวกลาง แบ่งเทคโนโลยีออกเป็นสองรูปแบบได้แก่ วงจรสวิตซ์(circuit switching) และ แพ็กเก็ตสวิตซ์ การสลับสัญญาณด้วยแนวคิด วงจรสวิตซ์

เริ่มต้นใช้กับสัญญาณแอนะล็อกและสามารถใช้ได้กับสัญญาณประเภทดิจิทัล โดยเป็นการสลับโดยใช้คุณสมบัติทางไฟฟ้ามีประโยชน์กับระบบที่ต้องการควบคุมเวลา ขณะที่แพ็กเก็ตสวิตซ์ใช้ได้กับสัญญาณประเภทดิจิทัล ด้วยการอ่านข้อมูลดิจิทัลและแปลความหมายได้เป็นชุดก่อนคัดเลือกเส้นทางการส่งผ่านระบบคอมพิวเตอร์ เทคโนโลยีปัจุบันส่วนใหญ่ถูกเปลี่ยนผ่านจาก วงจรสวิตซ์ มาเป็น แพ็กเก็ตสวิตซ์ เกือบทั้งหมด

1.3.3 OSI 7-layer Model

การโน้มเดลแบบ OSI 7-layer เป็นไปตามรูปที่ 1.15 ประกอบด้วย Application Presentation Session Transport Network DataLink และ Physical เป็นโปรโทคอลทำงานภายในโอดีต์เมื่อโอดีต์ส่งข้อมูลออกภายนอกจะใช้โปรโทคอลเพียงสามชั้นได้แก่ Network DataLink และ Physical

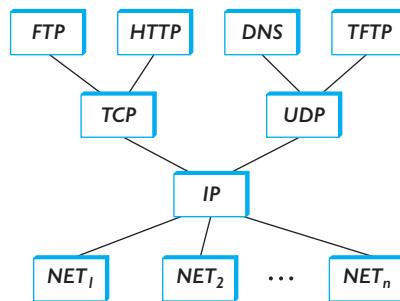


รูปที่ 1.15: ตัวแบบ OSI 7-layer
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

1.3.4 สถาปัตยกรรมอินเทอร์เน็ต

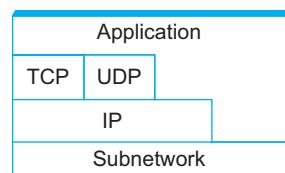
สถาปัตยกรรมการสื่อสารของอินเทอร์เน็ตเกิดจากการทำงานร่วมกันของหลายโปรโทคอลสำหรับโปรโทคอลหลักอธิบายในรูปที่ 1.16 โดยมีการใช้ร่วมการทำงานจาก OSI 7-layer ให้เหลือเพียง 4 ชั้น

เรียกรูปที่ 1.17 ว่าเป็น Internet protocol stack ประกอบด้วยโปรโทคอลสำคัญสำหรับการสื่อสาร ดังนี้ Application TCP, UDP IP และ Subnetwork



รูปที่ 1.16: แผนภาพอินเทอร์เน็ตโพรโทคอล
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

เมื่อ Application คือซอฟต์แวร์ที่ใช้งานการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต และ TCP หรือ UDP ใช้ในการขนส่งข้อมูลโดย TCP ใช้ส่งข้อมูลประเภทต้องการการยืนยัน ขณะที่ UDP ไม่ต้องการการยืนยัน ไอพีใช้สำหรับกำหนดที่ตั้งของไอพีและสุดท้าย Subnetwork แทนการเชื่อมต่อ กันโดยตรง



รูปที่ 1.17: มุมมองอัจฉริยะโดยกำหนดโพรโทคอล ในส่วน Subnetwork ก่อนหน้านี้อยู่ในชั้นเครือข่าย
แต่บ่อยครั้งถูกจัดให้อยู่ในชั้นลิ�ก “Layer 2” (ตามการอ้างอิง OSI 7-Layer)
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

End host, ไคลเอนต์(client) และ เซิร์ฟเวอร์

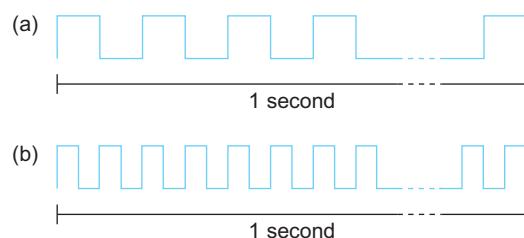
เครือข่ายคอมพิวเตอร์เป็นการเชื่อมของโครงข่ายตั้งแต่ต้นทางไปถึงปลายทาง ซึ่งส่วนใหญ่ไม่เป็นการเชื่อมโดยตรงจะมีการเชื่อมต่อผ่านอุปกรณ์ตัวกลางและส่งไปเรื่อยๆ จนถึงปลายทาง เรียกว่าเครือข่ายอินเทอร์เน็ต จุดเริ่มต้นของการเชื่อมต่อเป็นคอมพิวเตอร์ต่อเข้าเครือข่าย ก่อนที่ภายในเครือข่ายจะเชื่อมต่อกับผู้ให้บริการ จนถึงปลายทางที่คอมพิวเตอร์ปลายสุดเชื่อมต่อผู้ให้บริการ เรียกคอมพิวเตอร์ต้นทางและปลายทางว่า “End host” สาเหตุที่เรียกว่า End host เพราะจุดต้นและจุดปลายของการเชื่อมต่ออินเทอร์เน็ต จากรูปที่ 1.15 End host ของเครือข่ายอินเทอร์เน็ต ประกอบไปด้วยเครื่องคอมพิวเตอร์ อาจเป็นเครื่องระบบปฏิบัติการ Windows Mac Linux เครื่องเซิร์ฟเวอร์ รวมถึงโทรศัพท์เคลื่อนที่ จากรูปเห็นได้ว่า End host ใช้อัจฉริยะกล่าวถึงภายหลังของ อุปกรณ์ที่เชื่อมต่อกัน แต่ไม่ได้แบ่งว่าเครื่องใดเป็นเครื่องขอใช้บริการและเครื่องให้บริการ ในหนังสือเล่มนี้ใช้ คำว่า ไคลเอนต์ อัจฉริยะ end host ที่ขอใช้บริการ และ เซิร์ฟเวอร์ หมายถึง end host ที่เป็นเครื่องให้บริการ

1.4 ค่าประสิทธิภาพ

การวัดประสิทธิภาพเครือข่ายเป็นพื้นฐานที่มักกล่าวถึงถึง เมื่อกับคอมพิวเตอร์ทั่วไปที่สนใจว่าจะคำนวณได้เร็ว จะเก็บข้อมูลได้มากแค่ไหน สำหรับค่าประสิทธิเครือข่ายจะเป็นตัวบ่งบอกถึงประสิทธิของการทำงานซึ่งเป็นประโยชน์ต่อการออกแบบการทำงานของเครือข่าย

1.4.1 Bandwidth และ ค่าดีเลย์แฟรง

แบบดิจิตร์สำหรับเครือข่ายคอมพิวเตอร์ใช้เรียก จำนวนข้อมูลที่ส่งจากต้นทางไปจนถึงปลายทางเสร็จภายในหนึ่งวินาที เช่นสามารถส่งข้อมูลจำนวน 1000บิตได้เสร็จภายในหนึ่งวินาทีเรียกว่ามีแบนด์วิดธ์ 1000 bps (bit per second) หรือบ่อยครั้งเรียกว่า ทรูพุต จากรูปที่ 1.18 ยกตัวอย่างการส่งข้อมูลที่แบนด์วิดธ์ 1Mbps เทียบกับแบนด์วิดธ์ 2Mbps ขณะที่ดีเลย์หมายถึงจำนวนเวลาที่ต้องรอนับจากเริ่มต้นส่งข้อมูลจะทำเกิดช่วงเวลาในการเดินทางข้อมูลไปถึงปลายทาง ช่วงเวลาดังกล่าวเรียกว่า ดีเลย์ มีหน่วยเป็นเวลา เช่นเครือข่ายมีดีเลย์ 1 วินาที



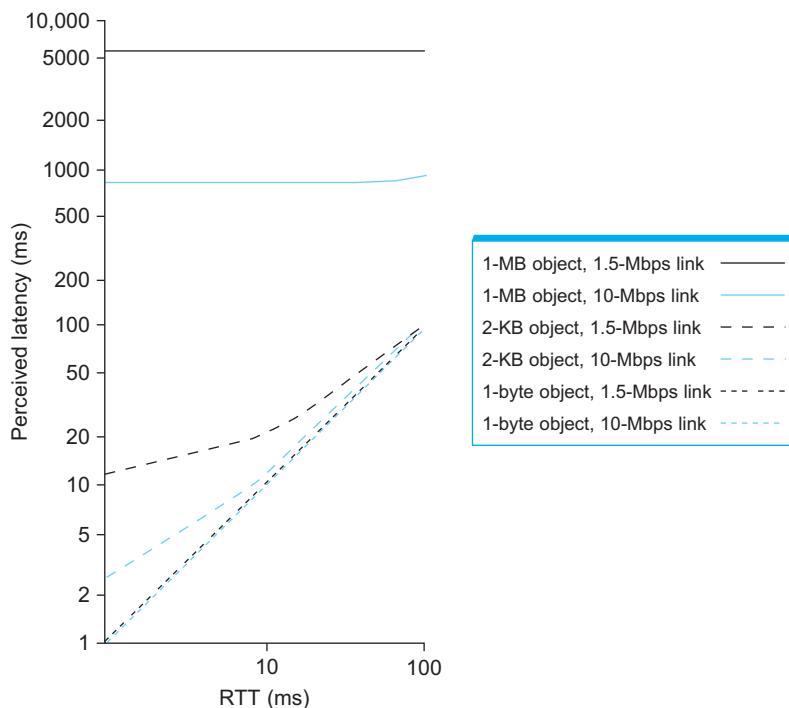
รูปที่ 1.18: การส่งข้อมูลระดับบิตภายในหนึ่งวินาที (a) ส่งข้อมูลบิตตัวอัตราเร็ว 1Mbps (หนึ่งบิตใช้เวลาหนึ่งไมโครวินาที) (b) ส่งข้อมูลบิตตัวความเร็ว 2Mbps (หนึ่งบิตใช้เวลา 0.5 ไมโครวินาที)
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

$$\text{latency} = \text{Propagation} + \text{Transmit} + \text{Queue}$$

$$\text{Propagation} = \text{Distance}/\text{SpeedOfLight}$$

$$\text{Transmit} = \text{Size}/\text{Bandwidth}$$

ค่าประสิทธิภาพต่อมานำเสนอในกระบวนการส่งข้อมูลออกจากต้นทางไปถึงปลายทางแล้ว ปลายทางตอบกลับ เรียกว่าราวน์ดทริปไทม์ ค่าประสิทธิภาพนี้เป็นประโยชน์ต่อการทดสอบประสิทธิภาพที่ไม่สามารถทดสอบค่าประสิทธิภาพของอุปกรณ์เครือข่ายตลอดระยะทาง เช่นที่ 1.19 จากรูปเมื่อการเชื่อมต่อ มีแบนด์วิดธ์แตกต่างกันจะทำให้ความเร็วในการเดินทางข้อมูลแตกต่างกัน ความสัมพันธ์ระหว่าง ค่าดีเลย์แฟรง และ ราวน์ดทริปไทม์ จะเปลี่ยนแปลงตามประเภทของลิงก์(link) จากรูปที่มีลิงก์มีขนาดแบนด์วิดธ์ 1.5-Mbps เมื่อส่งข้อมูลขนาด 1MB มี ค่าดีเลย์แฟรง คงที่ ขณะที่มีราวน์ดทริปไทม์เพิ่มขึ้น



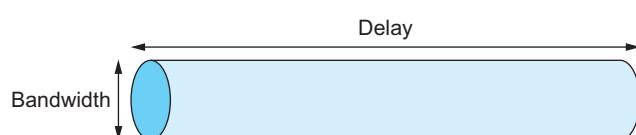
รูปที่ 1.19: ค่าประสิทธิภาพ ค่าดีเลย์แฟง และ RTT
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

1.4.2 การคูณดีเลย์ × แบนด์วิดธ์

ค่าการคูณ $\text{delay} \times \text{bandwidth}$ ใช้คำนวณความจุเครือข่ายนั้นสามารถอัดข้อมูลลงช่องสัญญาณแน่นที่สุดได้เท่าใด หากแน่นเต็มที่แล้วจะไม่สามารถอัดเพิ่มได้ สังเกตในรูปที่ 1.20 เมื่อลิงก์มีแบนด์วิดธ์ 45Mbps และมีดีเลย์ 50ms คำนวนค่า $\text{Delay} \times \text{Bandwidth}$ product ได้ดังนี้

$$50 \times 10^{-3} \text{ sec} \times 45 \times 10^6 \text{ bit/sec} = 2.25 \times 10^6 \text{ bit}$$

หรือค่าประมาณ 280KB ซึ่งมุ่งมองการคำนวนนี้เทียบกับรูปที่ 1.20

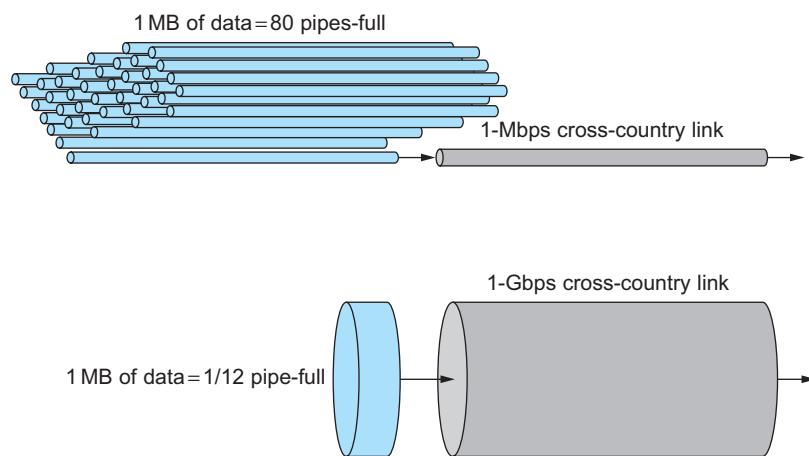


รูปที่ 1.20: นูมมองเครือข่ายเป็นท่อส่งข้อมูล
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ค่า $\text{delay} \times \text{bandwidth}$ ใช้ในการทดสอบประสิทธิภาพเครือข่าย เมื่อข้อมูลเดินทางผ่านท่อเครือข่าย ใช้เวลาช่วงหนึ่งในการเดินทางจากทางเข้าท่อจนกระทั่งออกถึงปลายท่อ และหากข้อมูลมีอัตราเร็วสูงมากเกินกว่าท่อส่งได้ทันจะทำให้เกิดข้อมูลล้นได้

1.4.3 เครือข่ายความเร็วสูง

ในส่วนนี้ก่อตัวถึงเครือข่ายความเร็วสูงซึ่งเกิดได้จากการนำเครือข่ายมาสั่งร่วมกันหลายเส้น ตามอธิบายในรูปที่ 1.22 มีจุดสังเกตของการเพิ่มแบบวิดร์น์จากเปลี่ยนแปลง ค่าดีเลย์แฟง ได้ ตัวอย่างเช่นการเพิ่มความเร็ว 1Mbps เป็น 80Mbps ด้วยการเพิ่มลิงก์จำนวน 80เส้น แต่ไม่สามารถลด ค่าดีเลย์แฟง ของเครือข่ายได้



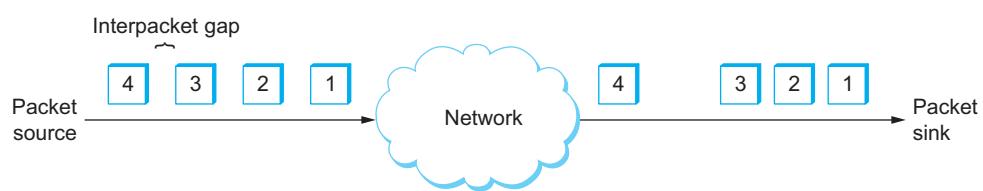
รูปที่ 1.21: ความสัมพันธ์ระหว่างแบบวิดร์และดีเลย์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

1.4.4 ค่าประสิทธิภาพของแอปพลิเคชัน

มีบางแอปพลิเคชันที่ต้องการการใช้งานเครือข่ายลักษณะเฉพาะ เช่นการส่งข้อมูลแบบ การไลฟ์สดผ่านอินเทอร์เน็ต(live-streaming) ต้องการความต่อเนื่องของภาพและเสียง หรือการสื่อสารผ่านการโทรศัพท์ต้องการความต่อเนื่องของเสียง ค่าประสิทธิภาพที่เกี่ยวข้องกับความต้องกันนี้เรียกว่า จิตเหอร์ โดยอธิบายในรูปที่ 1.22 จากรูปเมื่อข้อมูลออกเดินทางจากต้นทางผ่านเครือข่ายอินเทอร์เน็ตซึ่งมีการส่งต่อผ่านอุปกรณ์จำนวนหนึ่ง อุปกรณ์เหล่านี้ทำงานแบบสโตร์แอนด์ฟอร์เวิร์ดทำให้ความต่อเนื่องของแพ็กเกจเปลี่ยนไป ซึ่งการเปลี่ยนของดีเลย์ระหว่างแต่ละแพ็กเกจจะถูกกำหนดเป็นรูปแบบการกระจายของดีเลย์เป็นแบบสุ่ม เรียกว่าค่า จิตเหอร์

จากที่กล่าวมาข้างต้น ความเร็วในการส่งข้อมูลเกิดจากความสัมพันธ์ของ แบบวิดร์ และ ค่าดีเลย์แฟง เพื่อทำความเข้าใจให้ดี หากกำหนดให้ แบบวิดร์ มีได้ไม่จำกัด จะทำให้เห็นได้ความเร็วเครือข่ายยังคงมีค่าคงที่อยู่กับ ค่าดีเลย์แฟง ดังนั้น การส่งข้อมูลด้วยความเร็วสูง (high speed) ไม่ได้ปรับปรุงค่าดีเลย์แฟง เครือข่าย ได้หากส่งข้อมูลด้วยแบบวิดร์เท่ากัน ยกตัวอย่างเช่น บิตเดินทางไปกลับในเครือข่าย 1-Gbps ใช้เวลา 100ms ใช้เวลาเท่ากันกับ เครือข่ายที่มีแบบวิดร์ 1-Mbps

แต่ระยะเวลาในการส่งข้อมูลจะเริ่มมีผลแตกต่างกันเมื่อเพิ่มขนาดข้อมูล 1-MB ส่งผ่านเครือข่ายที่มีแบบวิดร์ 1-Mbps เทียบกับเครือข่ายมีแบบวิดร์ 1-Gbps โดยมีค่าดีเลย์แฟง 100ms เท่ากัน ในกรณีลิงค์แบบวิดร์ 1-Mbps



รูปที่ 1.22: การเกิดจิทเทอร์ในเครือข่าย
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

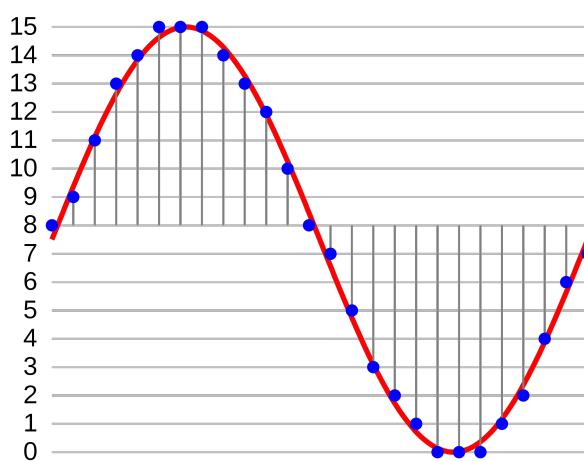
บทที่ 2

การเข้ามต่อโดยตรง

2.1 เทคโนโลยีขั้นกายภาพ

หนึ่งในคุณสมบัติสำคัญของขั้นกายภาพคือการย้ายข้อมูลที่อยู่ในรูปสัญญาณทางไฟฟ้าผ่านตัวนำสัญญาณ หากนึกถึงวิธีการส่งข้อมูลนับตั้งแต่มีบันทึก มนุษย์ใช้เครื่องมืออื่นที่ไม่ใช้ไฟฟ้าในการส่ง เช่น การใช้ม้าเป็นสื่อนำเอกสาร แต่ในพื้นที่ไม่มีถนนหรือข้ามแม่น้ำ หรือขณะอยู่บนเรือ การขนส่งด้วย yanพาหนะ หรือสัตว์นั้นไม่สะดวกนัก จึงมีการประยุกต์ใช้ควันเพื่อเป็นตัวแทนสัญญาณอย่างง่าย เริ่มมีการแปลงจากภาษาบ้านเป็นรหัสบันตั้งแต่นั้น

สิ่งที่สำคัญในการแปลงสัญญาณ ให้คำนึงถึงการแปลงจากสัญญาณทางกายภาพเป็นข้อมูลดิจิทัล ตัวอย่างที่นิยมใช้สัญญาณไนน์เป็นตัวแบบเสียง นำสัญญาณไนน์มาแบ่งตามระดับความสูงของสัญญาณ(แอมเพลจูด amplitude) ที่มาของสัญญาณเริ่มจากการใช้อุปกรณ์รับเสียงที่มีลักษณะเป็นไดอะเฟรม รับการสั่นสะเทือนของคลื่น โดยไดอะเฟรมมีขดลวดสองด้านด้านหนึ่งติดกับไดอะเฟรม อีกด้านต่อ กับขั้วสายสัญญาณ เมื่อไดอะเฟรมเคลื่อนที่จะทำให้ขดลวดมีการเคลื่อนที่ ทำให้เกิดการตัดกันของขดลวด เกิดกระแสไฟฟ้า ความเร็วในการตัดขดลวดทำให้เกิดความถี่ไฟฟ้าต่างกับการตัดขดลวด ซึ่งหมายถึงความถี่เท่ากับความถี่เสียง เมื่อถึงขั้นตอนนี้จะทำให้ได้สัญญาณไฟฟ้าประเภทแอนะล็อก ขั้นตอนต่อไปเป็นการแปลงสัญญาณแอนะล็อกเป็นดิจิทัล ด้วยการแบ่งส่วนของความแรงสัญญาณให้แทนด้วยข้อมูลบิต ดังรูปที่ 2.1 กำหนดให้มีพื้นที่การแปลงสัญญาณแทนด้วยเลขไบนารี(binary)จำนวน 4-บิต ทำให้มีค่าที่เป็นไปได้ทั้งหมด $2^4 = 16$ ค่า แต่ละค่าใช้แทนระดับความแรงสัญญาณด้านบวก 8 ค่า (รวม 0) และด้านลบ 8 ค่า เป็นข้อมูลดิจิทัลสำหรับอ้างอิงในจุดสินាเงิน



รูปที่ 2.1: Pulse Code Modulation ([Aquegg commonswiki, 2014](https://github.com/SystemsApproach/book))
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

เมื่อต้องการแปลงสัญญาณกลับเป็นสัญญาณและล็อกทำได้โดยวิธีการประมาณค่าของสัญญาณ หากจำนวนบิตมากจะทำให้ได้สัญญาณใกล้เคียงกับต้นฉบับ ตารางที่ 2.1 แสดงตัวอย่างการจับคู่ค่าความแรงสัญญาณกับบิตจำนวน 4-บิต

ตารางที่ 2.1: การจับคู่แรงดันไฟฟ้ากับเลขไบนารี

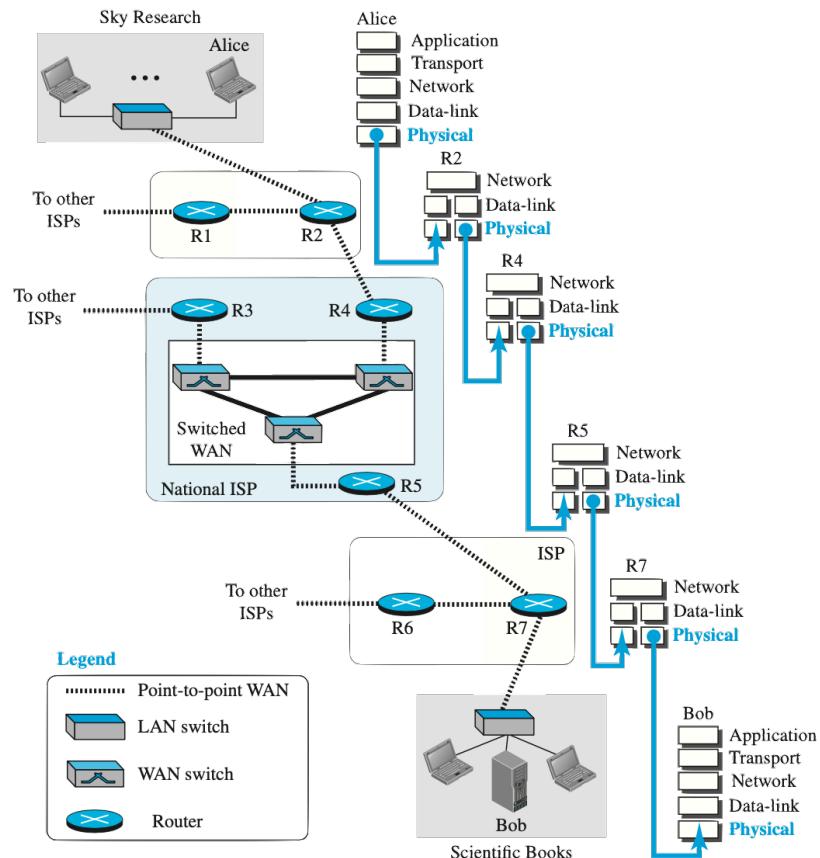
ระดับแรงดัน	บิต
15	0111
14	0110
13	0101
12	0100
11	0011
10	0010
9	0001
8	0000
7	1000
6	1001
5	1010
4	1011
3	1100
2	1101
1	1110
0	1111

2.1.1 เดต้า(data) และ สัญญาณ(signal)

รูปที่ 2.2 ใช้อธิบายตัวอย่างการสั่งซื้อหนังของอลิช(Alice) ซึ่งทำงานในบริษัทวิจัยชื่อ Sky Research ต้องการซื้อหนังสือกับร้านหนังสือออนไลน์ชื่อ Scientific Books ผ่านระบบออนไลน์ที่มีบอน(Bob)เป็นเจ้าของ

จากรูปที่ 2.2 ใช้วิธีมองรูปด้วยการจัดกลุ่มการเขื่อมต่อออกเป็นหาระดับ มี ชั้นกายภาพ(Application) ชั้นขนส่ง(Transport) ชั้นเครือข่าย(Network) ชั้นเขื่อมต่อโดยตรง(Data-link) เป็นการเขื่อมต่อในมุมมองโลจิคัล โดยที่ ชั้นกายภาพ(Physical) เป็นการเขื่อมต่อโดยตรงระหว่างอุปกรณ์ เพื่อทำความเข้าใจได้ง่ายขึ้นอีก รูปที่ 2.2 มองได้เป็นการเชื่อมต่อระหว่าง host-to-router, router-to-router และ router-to-host ในรูปมี Switched ถือเป็นการทำงานทำงานในชั้นกายภาพ คำสั่งซื้อระหว่างอลิชและบ๊อบถือเป็นสิ่งที่แลกเปลี่ยนกัน ผ่านเครือข่าย เเรียกว่าเป็น “เดต้า” สิ่งที่เป็นข้อมูล เช่น ชื่อหนังสือ จำนวนเล่มที่สั่ง ราคา รวมถึงที่อยู่สำหรับจัดส่ง ข้อมูลนี้ถูกป้อนในชั้นกายภาพ และส่งต่อลงมา ชั้นขนส่ง ชั้นเครือข่าย ชั้นเขื่อมต่อโดยตรง จนสุดท้ายถึงชั้นกายภาพ ซึ่งข้อมูลจะถูกแปลงเป็นสัญญาณไฟฟ้าก่อนส่งออกสายนำสัญญาณ เเรียกข้อมูลหลังจากการแปลงเป็นสัญญาณไฟฟ้าว่า “สัญญาณ”

ข้อมูลที่อลิชและบ๊อบแลกเปลี่ยนระหว่างกันคือ เดต้า นั้น เมื่อถึงชั้นกายภาพจะแปลงเป็นสัญญาณ ซึ่งสัญญาณสามารถมาในรูปของ อะนะล็อก หรือ ดิจิทัล ก็ได้



รูปที่ 2.2: การสื่อสารในชั้นกายภาพ (Forouzan, 2012, p.93)

ข้อมูลแอนะล็อกและข้อมูลดิจิทัล(digital data)

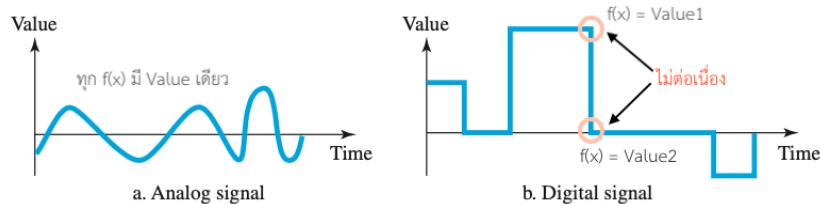
เด็กอาจแปลงมาจากแอนะล็อก หรือ ดิจิทัล ก็ได้ เมื่อกล่าวถึง “ข้อมูลแอนะล็อก” หมายถึงประเภทสัญญาณที่มีความต่อเนื่องเป็นเส้น และ “ข้อมูลดิจิทัล” หมายถึงข้อมูลที่มีการเปลี่ยนแปลงไม่เป็นเส้น หรือเรียกว่า *discrete state* ตัวอย่างเช่นนาฬิกาแอนะล็อก มีหน่วยเป็นชั่วโมงนาทีและวินาทีต่อเนื่องกัน แต่นาฬิกาดิจิทัล มีการแบ่งชั่วโมง นาที และวินาทีเป็นเลขจำนวนเต็ม เช่น 8:06n.

ตัวอย่างข้อมูลประเภทข้อมูลแอนะล็อกอีกตัวอย่าง เช่น สัญญาณเสียงพูดที่มีความต่อเนื่อง เมื่อมีคนพูดจะเกิดคลื่นแอนะล็อกเดินทางผ่านอากาศ ซึ่งจะรับคลื่นผ่านไมโครโฟน(Microphone)และแปลงเป็นสัญญาณแอนะล็อก หรือใช้กระบวนการการสุ่มเลือก(Sampling)เพื่อให้ได้สัญญาณดิจิทัลดังรูปที่ 2.1

สัญญาณแอนะล็อกและสัญญาณดิจิทัล

ตามที่ได้กล่าวมาข้างต้นสัญญาณอาจมามีรูปแบบแอนะล็อกหรือดิจิทัล สัญญาณแอนะล็อกจะมีระดับการแบ่งระดับสัญญาณไม่สิ้นสุด ดังอธิบายในรูปที่ 2.3(a) สัญญาณแอนะล็อกที่เปลี่ยนแปลงจากระดับ A ไประดับ B จะมีความต่อเนื่องมีลักษณะเป็นฟังก์ชัน ขณะที่สัญญาณดิจิทัลในรูปที่ 2.3(b) ไม่คุณสมบัติเป็นฟังก์ชันสังเกตจาก

$f(x)$ มี Value มากกว่า 1 ค่า จากรูปที่เห็นได้ว่าค่า x ทำให้ได้ $f(x)=\text{Value1}$ และ $f(x)=\text{Value2}$ ซึ่ง $\text{Value1} \neq \text{Value2}$



รูปที่ 2.3: เปรียบเทียบข้อมูลแอนะล็อก และ สัญญาณดิจิทัล (Forouzan, 2012, p.94)

2.2 มุมมองเทคโนโลยีภาพกว้าง

ในบทที่ 1 ได้กล่าวถึง องค์ประกอบที่ทำให้ โหนดสามารถเชื่อมต่อระหว่างกันได้ ซึ่งหนึ่งในคำาณพื้นฐานสำคัญ ก็คือการเชื่อมต่อคือ จะทำให้โหนดสองโหนดเชื่อมต่อกันได้อย่างไร ซึ่งบทที่ 1 กล่าวเพียงแนวคิดที่ทำให้ เกิดการเชื่อมต่อ ยังไม่ลงรายละเอียดที่อยู่เบื้องหลังแนวคิดนั้น สำหรับบทนี้จะกล่าวถึงรายละเอียดในส่วนที่ซับ ซ้อนขึ้น เป็นส่วนที่ทำให้เกิดการเชื่อมต่อระหว่างโหนดสู่โหนดโดยตรง

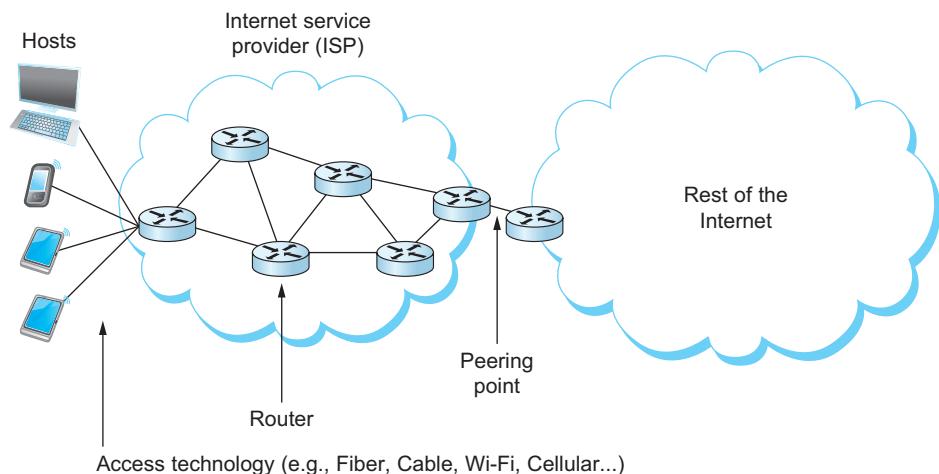
บทนี้จะกล่าวถึงการเชื่อมต่อที่มีเพียงโหนดสองโหนดและเชื่อมต่อกันโดยตรง การเชื่อมต่อตรงนี้จะมี ปัญหาที่เกี่ยวข้องทางด้านวิศวกรรมอยู่หลายส่วน ตัวอย่างเช่น ปัญหาที่เกิดจากคุณสมบัติทางกายภาพของสาย สัญญาณหรือของช่องสัญญาณ และคุณสมบัติของโปรดักโคลที่นำมาใช้

การเชื่อมระหว่างโหนดสองโหนดเป็นเพียงจุดเริ่มต้นของการเชื่อมเครือข่ายอินเทอร์เน็ต มีปัญหา สำคัญ 5 ปัญหาที่จะกล่าวถึงในบทนี้ หากเทียบกับ OSI 7-layer การเชื่อมต่อโดยตรงจัดอยู่ใน การเชื่อมต่อ Layer-2 (L2)

อันดับแรกข้อมูลจะถูกแปลงจากสัญญาณใดๆ เป็นสัญญาณไฟฟ้าและแปลงเป็นสัญญาณดิจิทัล กระบวนการแปลงนี้เรียกว่า เอ็นโคเดดดิ้ง แล้วจึงส่งผ่านช่องสัญญาณไปสู่ปลายทาง

วิธีทำความเข้าใจระบบการสื่อสารผ่านเครือข่ายคอมพิวเตอร์ ในบทเรียนนี้ใช้การนำเสนอด้วยการ แบ่งข้อมูลขนาดใหญ่เป็นส่วนเล็กทั้งหมด 7 ส่วน (ในแบบ OSI Layer) และ 5 ส่วนสำหรับ Internet protocol suite

รูปที่ 2.4 ประกอบด้วยอุปกรณ์เครือข่ายเชื่อมต่อกันหลายรูปแบบผ่านโครงข่ายอินเทอร์เน็ตซึ่งเป็น เทคโนโลยีในปัจจุบัน สร้างจากด้านซ้ายมือเห็นได้ว่าเป็นอุปกรณ์ที่เกิดจากการเชื่อมต่อโดยผู้ใช้งาน อุปกรณ์ ที่เชื่อมต่อโดยผู้ใช้งานอาจเป็นคอมพิวเตอร์โทรศัพท์เคลื่อนที่และแท็บเล็ต เรียกการเชื่อมต่อนี้ว่าเป็น access technology เทคโนโลยีที่อยู่ในชั้นนี้จะเป็นการเชื่อมต่อโดยตรง เช่น สายใยแก้วนำแสง สายทองแดง และไฟ สาย โทรศัพท์มือถือ เป็นต้น เมื่ออุปกรณ์ผ่าน access technology จะเชื่อมโยงภายในโครงข่ายผู้ให้บริการ เรียกว่า ไอเอสพี(Internet service provider) และ ไอเอสพี ส่งต่อไปยัง ไอเอสพี ปลายทาง และส่งถึง access technology ปลายทาง



รูปที่ 2.4: การเชื่อมต่ออินเทอร์เน็ตของผู้ใช้งาน
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

การเดินทางของสัญญาณไฟฟ้าเป็นการเคลื่อนที่ของอิเล็กตรอน(electron)ผ่านสายนำสัญญาณ การเดินทางผ่านสายนำสัญญาณเดินทางได้ช้ากว่าความเร็วแสง สมมติให้ความเร็วการเดินทางอิเล็กตรอนเท่ากับ $\frac{2}{3}$ ของความเร็วแสง หรือเท่ากับ $\frac{2}{3} \times 3 \times 10^8$ เมตรต่อวินาที(meter per second)

$$\text{wavelength} = \text{Speed Of Light In Copper} / \text{Frequency}$$

$$\text{wavelength} = \text{Speed Of Light In Copper} / \text{Frequency}$$

$$= 2/3 \times 3 \times 10^8 / 300 \quad (2.1)$$

$$= 667 \times 10^3 \text{ meters}$$

คุณสมบัติ ที่สำคัญอีกประการหนึ่งของการสื่อสารได้แก่ ความถี่(frequency) ความถี่มีหน่วยเฮิรตซ์ (hertz) หนึ่งหน่วยเฮิรตซ์หมายถึงการครบรอบสัญญาณจำนวนหนึ่งรอบในหนึ่งวินาที ระยะเวลาที่สัญญาณเดินทางครบรอบจะเท่ากับความเร็วในการแพร่สัญญาณของชนิดคลื่นนั้น เช่น คลื่นแม่เหล็กไฟฟ้ามีความเร็ว $c = 3 \times 10^8$ เมตรต่อวินาที ที่ความถี่ 10 เฮิรตซ์ความยาวของคลื่นจะคำนวณได้เป็น

$$\text{Distance}(s) = \text{Speed}(v) \times \text{Time } (t)$$

เมื่อ t แทนระยะเวลาที่คลื่นครบรอบ จากตัวอย่าง 10 Hz ทำให้ $t = 1/10 = 0.1$ ดังนั้นค่าระยะทางของคลื่นเท่ากับ

$$3 \times 10^8 \times 0.1 = 0.3 \times 10^8 \text{ m/s}$$

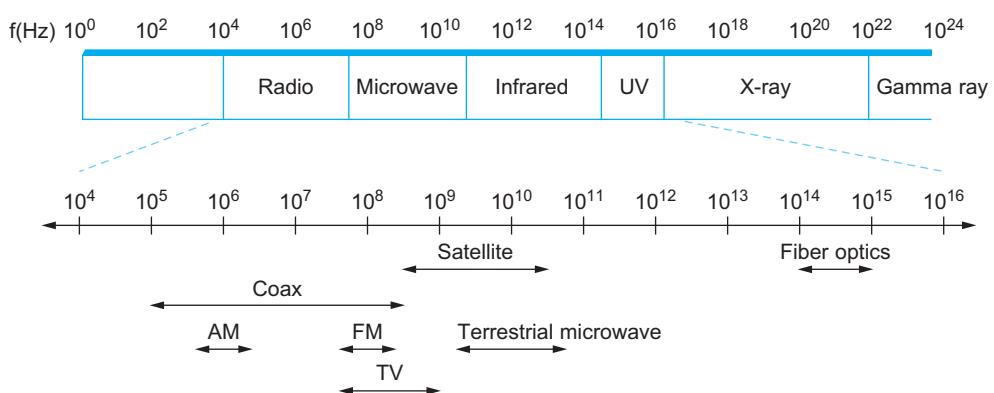
กำหนดให้ λ แทนความยาวของคลื่น มีหน่วยเมตร(meter) สมการความยาวคลื่นเป็นไปตามสมการต่อไปนี้

$$\lambda = \frac{c}{f} \quad (2.2)$$

เมื่อ λ แทนค่าความยาวคลื่น c แทนความเร็วแสง และ f แทนความถี่

สัญญาณไฟฟ้ากระแสสลับ(alternating current) ให้ผลผ่านตัวนำทำให้เกิด เกิดสนามไฟฟ้า(electric field) และ สนามแม่เหล็ก(magnetic field) เรียกว่าส่องรวมกันว่า แม่เหล็กไฟฟ้า มนุษย์ใช้การสื่อสารผ่าน แม่เหล็กไฟฟ้านั้นบดังแต่การค้นพบของ เจมส์ เคลิริก แมกซ์เวลล์(James Clerk Maxwell) Maxwell (1890) สามารถจำลองความสัมพันธ์ของกระแสไฟฟ้าสับกับสนามไฟฟ้า และ สนามแม่เหล็ก ได้ ซึ่งคุณสมบัติแม่เหล็กไฟฟ้าสามารถเดินทางได้โดยไม่ต้องการตัวนำพาสัญญาณ ทำให้มีการแพร่กระจายสัญญาณได้เร็วกว่า การสื่อสารด้วยเสียงหรือการสื่อสารผ่านสายนำสัญญาณ ทั้งยังมีความสะดวกด้านการใช้งานที่ไม่ต้องใช้สายสัญญาณ

การใช้งานแม่เหล็กไฟฟ้านั้นขึ้นกับการสื่อสารผ่านสายนำสัญญาณ ปัจจุบันนี้คือการกำหนด ช่วงความถี่ของสัญญาณ คือความถี่แม่เหล็กไฟฟ้าที่ใช้ในการสื่อสารผ่านโทรศัพท์เคลื่อนที่แล้วความถี่แม่เหล็กไฟฟ้ายังถูกนำไปใช้ในงานประเพณีด้วย ดังรูปที่ 2.5 ได้อธิบายช่วงความถี่แม่เหล็กไฟฟ้าที่ใช้ในงานต่าง โดยความถี่ที่เหมาะสมในการสื่อสารอยู่ในช่วงความถี่ในช่วง $10^4 - 10^{16}$ MHz ซึ่งได้แบ่งตามคุณสมบัติของตัวนำสัญญาณ ความถี่สำหรับการใช้งานด้านคลื่นวิทยุ(Radio) และไมโครเวฟ(Mircowave) อยู่ในช่วง $10^8 - 10^{10}$ และ การใช้งาน Infrared ขึ้นไปเริ่มทำงานตัวนำที่ส่งผ่านแสงได้ ตัวอย่าง เช่น สายใยแก้วนำแสง และความถี่ช่วง UV และ X-ray ใช้ในงานด้านการแพทย์



รูปที่ 2.5: สเปกตรัม ของแม่เหล็กไฟฟ้า
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

มีการใช้เทคโนโลยีหลายรูปแบบในการให้บริการ access networks เช่น เครือข่ายให้บริการ access networks แบบใช้สายสัญญาณ โดยมี ไอเอสพี เป็นผู้ให้บริการ ยกตัวอย่างตามตารางที่ 2.2 มีสายสัญญาณ 2

¹ ความถี่ที่มีความยาวคลื่นน้อยกว่า มิลลิเมตร(milli meter)

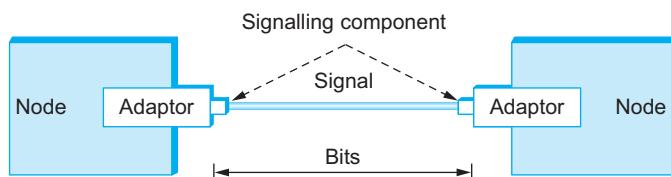
ประเภทที่ได้แก่ ทองแดงและสายใยแก้วนำแสง และใช้สามเทคโนโลยีประกอบด้วย DSL G.Fast และ PON ทั้งสามเทคโนโลยีมีความเร็วสูงสุดแตกต่างกัน

ตารางที่ 2.2: ประเภทสายและอัตราเร็วบริการอินเทอร์เน็ตบ้าน

Service	แบบดิจิทัล
DSL (สายทองแดง)	สูงสุด 100 Mbps
G.Fast (สายทองแดง)	สูงสุด 1 Gbps
PON(ใยแก้วนำแสง)	สูงสุด 10 Gbps

2.3 เอ็นโคเดดิ้ง

เมื่อมีข้อมูลต้องการส่งออกภายนอกจะเริ่มต้นจากแปลงข้อมูลเป็นสัญญาณดิจิทัลและนำมาแปลงเป็นข้อมูลที่มีรูปแบบตรงกับเครื่องปลายทาง จากรูปที่ 2.6 ได้ยกตัวอย่างการส่งข้อมูลระหว่างสองโนนด เมื่อข้อมูลออกจากโหนดหนึ่งจะทำการแปลงให้มีรูปแบบที่มีรูปแบบตรงกับปลายทางผ่านอุปกรณ์ชื่อว่า อะแดปเตอร์ ซึ่งอะแดปเตอร์ทำหน้าที่ในการแปลงข้อมูลให้อยู่ในรูปแบบเดียวกับอะแดปเตอร์ปลายทาง ในที่นี้ใช้ในการสื่อสารผ่านเครือข่าย เรียกว่า เนตเวิร์กอะแดปเตอร์



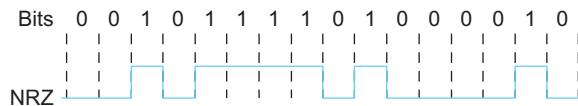
รูปที่ 2.6: สัญญาณเดินทางผ่านตัวนำสัญญาณเป็นสัญญาณดิจิทัลส่งครั้งละบิต
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ข้อมูลที่ใช้สื่อสารปัจจุบันใช้การอ้างอิงสัญญาณด้วยระดับความต่างของสัญญาณเพียงสองระดับ ตามที่เรียกว่า “สัญญาณดิจิทัล” อย่างไรก็ตามข้อมูลไม่ได้ถูกส่งเป็นสัญญาณดิจิทัลโดยตรง สาเหตุจากปัญหาการแยกความแตกต่างของข้อมูลที่ส่งต่อเนื่องกัน เช่น 0 กับ 00 หรือ 000 จะตรวจสอบความแตกต่างยาก วิธีในการส่งสัญญาณเป็นได้สองรูปแบบได้แก่ ชิงโครนัส(synchronous) และ อะชิงโครนัส(asynchronous) การส่งสัญญาณแบบชิงโครนัสต้องการใช้สัญญาณนาฬิกา(clock)เพื่อเข้าจังหวะสัญญาณที่เครื่องส่งและเครื่องรับขณะที่การส่งแบบชิงโครนัสไม่ต้องใช้สัญญาณนาฬิกา

มีวิธีแปลงสัญญาณให้สามารถระบุความแตกต่างได้หลายวิธี เรียกวิธีแปลงสัญญาณนั้นว่า เอ็นโคเดดิ้ง ตัวอย่างการเอ็นโคเดดิ้งแบบ ไม่กลับไปเป็นศูนย์ การเอ็นโคเดดิ้งแบบไม่กลับไปเป็นศูนย์เป็นการส่งแบบชิงโครนัส มีการทำงานดังรูปที่ 2.7 การไม่กลับไปเป็นศูนย์กำหนดให้มีสัญญาณดิจิทัลที่เป็นไปได้ 2 กรณีคือ บวก(+) หรือ ลบ(-) ซึ่งให้เลี้ยงปัญหาที่จาก DC component ได้ เมื่อกำหนดให้สถานะของสัญญาณเป็น + หรือ เป็น - เมื่อสัญญาณเป็น + กำหนดให้แทนบิต 1 เมื่อสัญญาณเป็น - แทนบิต 0

0010111101000010

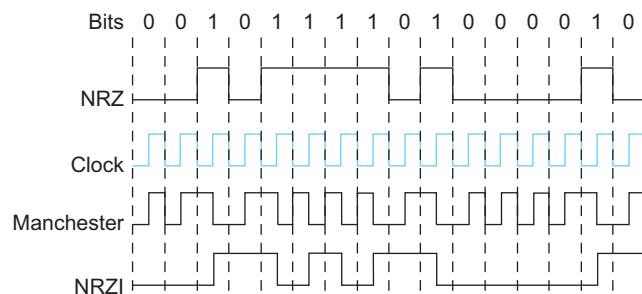
เมื่อเอ็นโค้ดดิ้งด้วย ไม่กลับไปเป็นศูนย์ ใช้วิธีกำหนดให้สัญญาณ - แทน 0 ดังนั้นจึงมีสัญญาณเป็น - จำนวนสองช่อง แต่มาสัญญาณเปลี่ยนเป็น 1 ทำให้สัญญาเปลี่ยนเป็น +



รูปที่ 2.7: การเอ็นโค้ดดิ้งแบบไม่กลับไปเป็นศูนย์สลับสัญญาณทรงกลางบิต
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

นอกจากการสลับสัญญาณแบบไม่กลับไปเป็นศูนย์แล้วยังมีการสลับสัญญาณในรูปแบบต่างๆด้วย เช่น ไม่กลับไปเป็นศูนย์กลับหัว และ แมนเชสเตอร์

การเอ็นโค้ดดิ้งแบบไม่กลับไปเป็นศูนย์กลับหัว เป็นการแปลงตรงข้ามกับไม่กลับไปเป็นศูนย์ โดยกำหนดให้ + แทน 0 และ - แทน 1 ตามรูปที่ 2.8 สัญญาณจะเปลี่ยนไปตามสัญญาณนาฬิกาที่เปลี่ยนจาก Low ไป High การเอ็นโค้ดดิ้งแบบแมนเชสเตอร์ ใช้วิธีอ่านการเปลี่ยนแปลงสัญญาณ เมื่อเปลี่ยนจาก High ไป Low หมายถึงบิต 0 และเมื่อเปลี่ยนสัญญาณจาก Low ไป High หมายถึงบิต 1 และใช้วิธีสลับสัญญาณทุกครั้งที่มีบิตซ้ำกัน เช่น 00 และ 11 และจะยังคงเป็นสัญญาณตามระดับปกติ



รูปที่ 2.8: การเอ็นโค้ดดิ้งแบบไม่กลับไปเป็นศูนย์ ไม่กลับไปเป็นศูนย์กลับหัว และ แมนเชสเตอร์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ตารางที่ 2.3 กำหนดให้การเอ็นโค้ดดิ้ง 5-bit ใช้อ้างถึงข้อมูลขนาด 4-bit มีรูปแบบตัวเลขเป็นได้ 16 แบบ ($2^4 = 16$) ซึ่งเพียงพอในการใช้อ้างอิง เมื่อจับคู่กับ 5-bit ซึ่งอ้างอิงตัวเลขได้ 32 แบบ ส่วนรหัสอื่นที่ไม่ถูกใช้เอ็นโค้ดดิ้ง จะนำไปใช้ในการจัดการ เช่น 11111 ใช้มีอย่างว่า และ 00000 แทนรหัสการวางแผน และ 00100

2.4 เฟรมมิ่ง

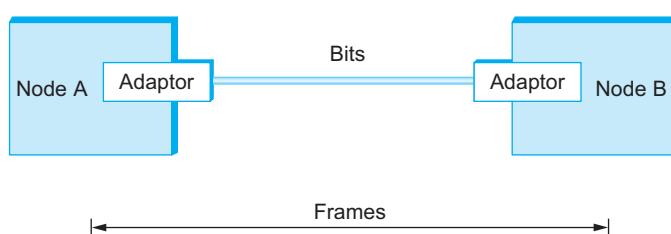
เฟรมมิ่ง คือการจัดรูปแบบบิตให้เรียงตัวเป็นชุดข้อมูลที่เป็นประโยชน์ต่อการสื่อสาร ซึ่งการสื่อสารข้อมูลนั้นจะมีตัวข้อมูล(เพย์โหลด(payload)) ที่ต้องการส่งและข้อมูลส่วน帧เดอร์ ข้อมูลส่วน帧เดอร์ใช้สำหรับช่วยเหลือให้ข้อมูลเดินทางถึงปลายทาง เฟรมมิ่ง ทำหน้าที่กำหนดรูปแบบข้อมูลที่เกิดจากการต่อกันของ เ帧เดอร์ และ เพย์โหลด

ตารางที่ 2.3: เอ็นโคดดิ้งแบบ 4B/5B

4-bit Data Symbol	5-bit Code
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

frame = header + payload

รูปที่ 2.9 อธิบายข้อมูลบิตจะถูกส่งไปทีละครั้งซึ่งหนึ่งครั้งจะมีจำนวนบิตขึ้นอยู่กับโปรโตคอลที่ใช้ สื่อสาร เเรียงกกลุ่มข้อมูลบิตนี้ว่าเฟรม และเรียกวาร่วมกกลุ่มบิตว่าเฟรมมิ่ง การแบ่งกกลุ่มข้อมูลนั้นจะเรียงตาม โครงสร้างที่กำหนดในโปรโตคอล



รูปที่ 2.9: บิตสำามีอย่างผ่านอะแดปเตอร์ต้นทางส่งผ่านช่องสัญญาณไปถึงอะแดปเตอร์ปลายทาง
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

การกำหนดให้ส่งข้อมูลเป็นเฟรมทำให้ออกแบบโปรโตคอลได้หลากหลาย โดยไม่ต้องเปลี่ยนแปลง สายสัญญาณ ตัวอย่างการสื่อสารแบบ PPP Ethernet และ WiFi จะมีรูปแบบเฟรมแตกต่างกัน แต่ยังคงสื่อสาร ระหว่างกันได้ผ่านการแปลงข้อมูลของอะแดปเตอร์ซึ่งอะแดปเตอร์จะอ่านข้อมูลตามรูปแบบเฟรมที่เข้ามา

2.5 ตรวจจับข้อผิดพลาด

จากที่ได้กล่าวถึงในบทที่ 1 การเกิด ข้อผิดพลาดระดับบิต อาจเกิดขึ้นภายในเฟรม ปัญหานี้เกิดขึ้นได้ง่าย ตัวอย่างเช่น มีการรับกวนสัญญาณภายในสายสัญญาณเพียงจุดเดียวทำให้การเปลี่ยนแปลงของสัญญาณไฟฟ้า กลับข้ามได้ แต่ปัญหานี้จะไม่เกิดขึ้นกับสายสัญญาณประเภทสายใยแก้วนำแสง ซึ่งสายใยแก้วนำแสงจะพบ ปัญหาข้อผิดพลาดระดับบิตในรูปแบบต่างๆ ออกไป

ในการออกแบบ วิธีตรวจสอบข้อผิดพลาดระดับบิต ที่ทำความเข้าใจง่ายและพบในการบันทึกข้อมูล ที่ต้องการตรวจสอบความสมบูรณ์ เช่นกับการเขียนข้อมูลหาร์ดิสก์ ใช้วิธีออกแบบให้ระบบมีฮาร์ดดิสก์สองลูก โดยเทียบข้อมูลเดียวกันลงในฮาร์ดดิสก์ทั้งคู่ เมื่อต้องการตรวจสอบความถูกต้องข้อมูล ทำได้โดยอ่านข้อมูลจาก ฮาร์ดดิสก์ทั้งสอง แล้วเปรียบเทียบกันหากพบความแตกต่างจะสามารถตรวจสอบได้ว่าเกิดข้อผิดพลาดระดับบิต อย่างไรก็ตามวิธีนี้ไม่สามารถตรวจสอบได้ว่าบิตใดเป็นข้อผิดพลาดระดับบิต

การส่งข้อมูลข้ามสองรอบเพื่อใช้ในการตรวจสอบข้อผิดพลาดระดับบิตทำได้ดังนี้ เมื่อต้องการส่งข้อมูล “Hello, world!” เรือจะแปลงข้อมูลจากภาษาอังกฤษที่อยู่ในตารางรหัสแอลกอริทึม หรือการทำงานโดยใช้ข้อมูล ในตารางที่ 2.4

ตารางที่ 2.4: แปลง Hello, world เป็นรหัสแอลกอริทึม

ตัวอักษร	เลขฐานสิบ	เลขไบนารี
H	72	01001000
e	101	01100101
l	108	01101100
l	108	01101100
o	111	01101111
,	44	00101100
	32	00100000
w	119	01110111
o	111	01110010
r	114	01110010
l	108	01101100
d	100	01100100
!	33	00100001

ข้อมูลจัดส่งถึงปลายทางแบ่งเป็นสองชุดที่เป็นข้อมูลสำคัญและนำข้อมูลทั้งสองชุดนั้นมาเรียงตามรูป แบบตารางที่ 2.5 ในการตรวจสอบนั้นทำได้โดยนำข้อมูลแต่ละบิตมาเปรียบเทียบกันซึ่งตัวดำเนินการที่นำมาใช้ เปรียบเทียบสามารถใช้ตัวดำเนินการทางลอจิก XOR สำหรับเปรียบเทียบบิตได้

สมมติเกิด Error ขึ้นที่ตัวอักษร ‘o’ และ ‘r’ เกิดเปลี่ยนข้อมูลเป็น ‘n’ และ ‘s’ ตามลำดับ ระบบจะ ตรวจสอบพบรูปแบบที่ไม่ตรงกัน ดังตารางที่ 2.7

ตารางที่ 2.5: ลำดับข้อมูลเรียงตามบิต

H	e	l	l	o	,	w	o	r	l	d	!
1001000	01100101	1101100	01101100	01101111	00101100	00100000	01110111	01101111	01110010	01101100	01100100
1001000	01100101	1101100	01101100	01101111	00101100	00100000	01110111	01101111	01110010	01101100	01100100

ตารางที่ 2.6: เกิดข้อผิดพลาดระดับบิต ของตัวอักษร o และ r

H	e	l	l	o	,	w	o	r	l	d	!
01001000	01100101	01101100	01101100	01101111	00101100	00100000	01110111	01101111	01110010	01101100	01100100
01001000	01100101	01101100	01101100	01101111	00101100	00100000	01110111	01101111	01100110	01101100	01100100
H	e	l	l	o	,	w	o	r	l	d	!
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000001	00000001	00000000	00000000	00000000

วิธีส่งข้อมูลซ้ำอาจนำมาประยุกต์ใช้ในการสื่อสารทางเครือข่ายคอมพิวเตอร์ได้ แต่ยังขาดประสิทธิภาพในด้านการสื้นเปลืองทรัพยากรเครือข่ายจากการเสียเวลาไปกับการส่งข้อมูลซ้ำ

ค่าประสิทธิภาพการใช้ทรัพยากรเครือข่ายเมื่อเทียบกับวิธีส่งข้อมูลซ้ำ ทำให้มีการใช้ข้อมูลเป็น 2 เท่าของข้อมูลจริง เช่นต้องการส่งข้อมูลขนาด 10 ไบต์ จะต้องส่งข้อมูล 10+10 ไบต์ เพื่อเป็นข้อมูลสำหรับตรวจสอบทำให้ต้องส่งข้อมูล 20 ไบต์ เป็นการใช้ทรัพยากรเครือข่ายในการตรวจสอบความผิดพลาด 50% ยกตัวอย่างการใช้งานจริง เช่น เมื่อระบบเครือข่ายมีแบบดิจิตอล 1Mbps หมายถึงส่งข้อมูลขนาด 1Mbit ได้เสร็จภายใน 1 วินาที สมมติมีข้อมูล 1Mbit ส่งแบบไม่ตรวจสอบข้อผิดพลาดระดับบิตจะใช้เวลา 1 วินาที แต่เมื่อส่งแบบตรวจสอบข้อผิดพลาดระดับบิตด้วยวิธีส่งซ้ำ จะต้องใช้เวลาเพิ่มขึ้นจากเดิมอีก 1 วินาทีสำหรับส่งข้อมูลซ้ำ รวมเป็นใช้เวลาส่ง 2 วินาที

ในการสื่อสารชั้นลึกไม่ว่าประเภทใดจะเลี่ยงการเกิดข้อผิดพลาดระดับบิตได้ยาก ปัญหานี้พบมาตั้งแต่เริ่มมีการสื่อสาร มีวิธีตรวจสอบข้อผิดพลาดระดับบิตที่มีประสิทธิภาพอยู่หลายวิธี อาทิ เช่น รหัสแฮมมิง(Hamming code)(Hamming, 1950) และ รหัสเรด-โซโลมอน(Reed-Solomon code)(Reed และ Solomon, 1960) ซึ่งประยุกต์ใช้ในการตรวจสอบข้อผิดพลาดระดับบิตสำหรับอุปกรณ์ประเภทแถบแม่เหล็ก เป็นต้น พื้นฐานการตรวจสอบข้อผิดพลาดระดับบิต กล่าวถึงในหัวข้อต่อไป

2.5.1 Internet Checksum Algorithm

สำหรับหัวข้อนี้จะกล่าวถึงการตรวจสอบข้อผิดพลาดระดับบิตที่ใช้ในการสื่อสารเป็นหลัก การตรวจสอบข้อผิดพลาดระดับบิต มีประโยชน์ให้คุณสื่อสารสามารถตรวจสอบความถูกต้องของข้อมูลได้ เมื่อพบข้อผิดพลาดระดับบิตจะส่งแพ็กเก็ตไปแจ้งต้นทางให้ส่งใหม่อีกรัง นอกจากแนวทางการตรวจสอบข้อผิดพลาดระดับบิตแล้วยังมีการต่อยอดให้สามารถแก้ไขข้อผิดพลาดระดับบิตได้ที่ปลายทาง วิธีนี้เรียกว่า รหัสแก้ความผิดพลาด(error-correcting codes) โพรโทคอลที่ได้รับความนิยมสำหรับรหัสแก้ความผิดพลาดคือ CRC

2.5.2 Checksum

จากปัญหาของการส่งข้อมูลซ้ำทำให้ขาดประสิทธิภาพด้านการใช้ทรัพยากรเครือข่าย วิธีเช็คชั่มเป็นวิธีที่มีประสิทธิภาพสูงขึ้น ด้วยการนำข้อมูลมาหักโดยคำนวณทีละเฟรม เมื่อแปลงข้อมูล Hello, world! เป็นรหัสรหัสแอกสกี ได้ตามตารางที่ 2.4 การเช็คชั่มเป็นกระบวนการนำข้อมูลมาหักกัน เช่นหักครั้งละ 8-bit เมื่อข้อมูลมีความยาวกินจะนำบิตส่วนเกินกลับมาหาก ตามตารางที่ 2.7

ตารางที่ 2.7: การตรวจสอบข้อผิดพลาดระดับบิตด้วยวิธีเช็คชั่ม

ตัวอักษร	เลขฐานสิบ	เลขไบนารี
H	72	01001000
e	101	01100101
l	108	01101100
l	108	01101100
o	111	01101111
,	44	00101100
	32	00100000
w	119	01110111
o	111	01110010
r	114	01110010
l	108	01101100
d	100	01100100
!	33	00100001
checksum	1161	10010001001
checksum	1161	10001001 +100 10001101

วิธีเช็คชั่มมีประสิทธิภาพดี เมื่อร่วมข้อมูลที่ต้องการส่งแล้วแนบເheadsเดอร์ เช็คชั่ม ในขั้นตอนสุดท้าย จากตารางที่ 2.7 มีข้อมูลขนาด 13 ใบต์ และ checksum 1 ใบต์ รวมเป็นข้อมูลที่ต้องการส่ง 13+1 ใบต์และเมื่อ มีข้อมูลขนาดใหญ่มากจะทำให้ประยัดทรัพยากรเครือข่ายได้มากตามขึ้นไปด้วย สิ่งที่จะต้องกลับนั้นอาจไม่ได้ เสนอออกไป เมื่อข้อมูลมีขนาดใหญ่ขึ้นจะทำให้มีโอกาสเกิดข้อผิดพลาดระดับบิตเพิ่มขึ้น ส่งผลให้เกิดการส่งข้อมูล ใหม่ทั้งหมด

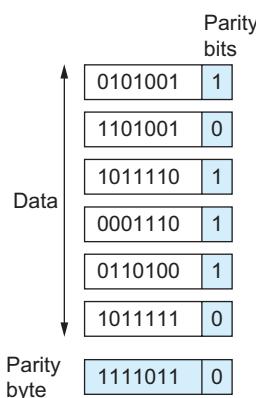
2.5.3 Parity bit check

การตรวจสอบข้อผิดพลาดระดับบิต วิธีออกแบบง่ายและมีประสิทธิภาพดีกว่าการส่งข้อมูลซ้ำ พบรการใช้งานในวงจร อิเล็กทรอนิกส์ เป็นการตรวจสอบนับจำนวนบิตมีจำนวนคู่หรือมีจำนวนคี่ แนวคิดทำความเข้าใจง่ายเพียง นับ จำนวนบิตที่มีค่า ‘1’ มีจำนวนเป็นคู่กี่จำนวน เรียกวิธีนี้ว่า even-parity bit หากให้วิธีนับจำนวนบิตที่มีค่า ‘1’ จำนวนคือคู่กี่จำนวนเรียกว่า odd-parity bit แนวทางนี้ทั้งหมด 4 รูปแบบ

1. เลือกนับบิต 0 ตรวจจำนวนคู่
2. เลือกนับบิต 0 ตรวจจำนวนคี่
3. เลือกนับบิต 1 ตรวจจำนวนคู่
4. เลือกนับบิต 1 ตรวจจำนวนคี่

ในการเลือกว่าเป็นบิต 0 หรือ 1 หรือเลือก คู่หรือคี่ จะขึ้นอยู่กับการออกแบบวงจรอิเล็กทรอนิกส์ เช่น วงจรที่มีบิต 1 ผิดพลาดน้อยกว่าบิต 0 จะเลือกใช้บิต 0 เป็นข้อมูลสำหรับตรวจสอบ parity เป็นต้น

ตัวอย่างการคำนวณ parity bit รูปที่ 2.10 ตัวอย่างการตรวจสอบ odd parity bit ของบิต=1 หาก บิต=1 มีจำนวนเป็นคี่จะมี parity bit = 1 และเป็น 0 เมื่อจำนวนบิตเป็นคู่



รูปที่ 2.10: Odd parity bit ขนาด (7+1)-bit 7บิตข้อมูล และ 1 parity bit
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

จากรูปที่ 2.10 ข้อมูลถูกส่งเป็นชุด ชุดละ 8-บิต รวม parity bit เมื่อข้อมูลส่งถึงปลายทางจะตรวจสอบ parity bit หากพบว่า parity bit ไม่ตรงกันจะให้ส่งข้อมูลใหม่เฉพาะเฟรมที่ผิดพลาด

ค่าสำหรับใช้วัดประสิทธิภาพการทำงาน ใช้การเปรียบเทียบสัดส่วนของจำนวนบิตที่ใช้ส่งข้อมูลกับจำนวนบิตที่ใช้สำหรับตรวจสอบข้อผิดพลาดระดับบิต เช่นการตรวจสอบด้วยวิธี parity bit ใช้ 1บิตสำหรับตรวจสอบข้อมูล ค่าประสิทธิภาพจึงเท่ากับ 8/1

2.5.4 Cyclic Redundancy Check

CRCเป็นกระบวนการตรวจสอบข้อผิดพลาดระดับบิตที่ใช้งานแพร่หลายในระบบสื่อสารและระบบบันทึกข้อมูล เป้าหมายเพื่อให้ข้อมูลมีความถูกต้องครอบคลุมจำนวนบิตมากที่สุดโดยเพิ่มส่วนເຂົດເວົ້ວທີ່ທໍານັກທີ່ເປັນ redundant bit น้อยที่สุด ซึ่งวิธี CRC ใช้กระบวนการทางคณิตศาสตร์สำหรับตรวจสอบความผิดพลาดได้อย่างเหมาะสม สำหรับ CRC ขนาด 32-bit สามารถตรวจสอบข้อผิดพลาดระดับบิตได้ครอบคลุมหนึ่งพันໄບຕໍ່ ຖານ໌ທີ່ທີ່ໃຫ້CRCตรวจสอบໄດ້ຈຳນວນບົດມາກ ມາຈາກຄณิตศาสตร์ແໜ່ງທີ່ອ່າວ່າ *finite fields* ອີຍາຍການທຳນານ CRC ດັ່ງນີ້

การคำนวน CRC เกี่ยวข้องกับตัวแปร 3 ตัวได้แก่ M แทนข้อมูลใบหนารีที่ต้องการส่ง G แทนเลขใบหนารีที่จะใช้เป็นตัวหาร โดยที่เป็นค่าเดียวกันในเครื่องส่งและเครื่องรับ และสุดท้าย R แทนเศษ(Remainder)ของ การหาร ตัวอย่างเช่น 8/5 เหลือเศษ 3 เป็นต้น

การคำนวน CRC

เลขจำนวนใดๆ สามารถเขียนอยู่ในรูปพหุนาม(polynomial)ได้ ตัวอย่างเช่น 123 เขียนในรูปแบบพหุนาม สังเกตสมการที่(2.3) อ้างอิงในเลขฐาน 10 ทำให้ตัวยกกำลังมีค่าเป็น 10 และจำนวนดิจิตของ 123 เท่ากับ 3 ทำให้มีค่าเลขยกกำลังสูงสุดเป็น $3-1 = 2$ เรียกว่าเป็น

$$a_1 \times b^2 + a_2 \times 10^1 + a_3 \times 10^0$$

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 \quad (2.3)$$

วิธีนี้เข้ากับอ้างอิงเลขใบหนารีได้ ตัวอย่างเช่น $(1011)_2$ เขียนในรูปพหุนาม ได้ในสมการที่(2.4)

$$(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \quad (2.4)$$

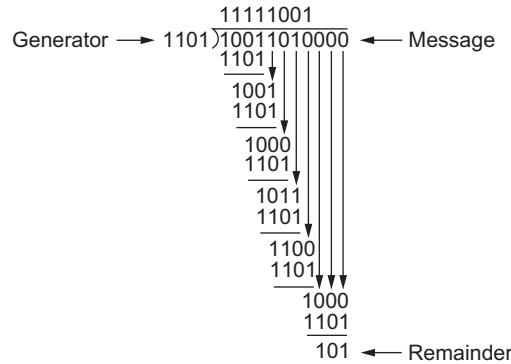
ข้อตกลงของโพรโทคอลCRC กำหนดให้ ถ้า M มีขนาด 8-บิต จะต้องใช้ G ที่มีขนาดน้อยกว่า 8-บิต ยกตัวอย่างเช่น M=10011010 มีจำนวน 8-บิต เลือกใช้ G=1101 ซึ่งมีจำนวน 4-บิต ทำให้เศษเหลือจากการหารจะเหลือ 3-บิต

การเตรียมข้อมูลสำหรับคำนวนเริ่มต้นมีข้อมูลขนาด 8-บิต โดยเลือก G จำนวน 4-บิต ทำให้ได้ผลลัพธ์ เป็นจำนวนที่เกิดจากเศษการหาร แทนด้วย R ที่มีขนาด 4-1 = 3-บิต ซึ่งตอนแรกรายไม้ได้คำนวนกำหนดให้ R=000 สำหรับการคำนวนจะนำเลขใบหนารี M มาต่อ กับ R ยกตัวอย่างเช่น M=10011010 และ R=000 ทำให้ได้ ข้อมูลที่ส่งออกเครือข่ายเท่ากับ 8+3=11-บิต

M R = 10011010000

สำหรับการคำนวน R เป็นไปตามรูปที่ 2.11 ทำได้โดยการตั้งหารยาวจนเหลือเศษที่หารไม่ได้ (จำนวน บิตน้อยกว่าตัวหาร)

เศษจากการหารในที่นี้ได้ R=101 ทำให้ได้ข้อมูลประกอบการส่งเป็น M|R=10011010101 เมื่อข้อมูลเดินทางถึงปลายทาง เครื่องปลายทางได้รับข้อมูล 10011010101 และเครื่องปลายทางทราบอยู่แล้วว่าได้ ตกลงกันใช้ G=101 เมื่อได้รับข้อมูล 11-บิต ตัดส่วน R ออก 3-บิต เหลือข้อมูล 8-บิต เป็นข้อมูลที่ต้องการตรวจสอบความสมบูรณ์ เครื่องรับใช้วิธีหารเข่นเดียวกับที่เครื่องส่งทำ และตรวจสอบเศษเหลือจากการหารว่าตรงกับ 3-บิตที่ได้รับหรือไม่ จากตัวอย่างนี้ผลลัพธ์ที่ได้จากการหารจะมีค่าเท่ากับ 101 ถือว่าข้อมูลถูกต้อง



รูปที่ 2.11: การหารายวิจัยการคำนวณCRC
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

2.6 การมีเสถียรภาพในการส่งข้อมูล

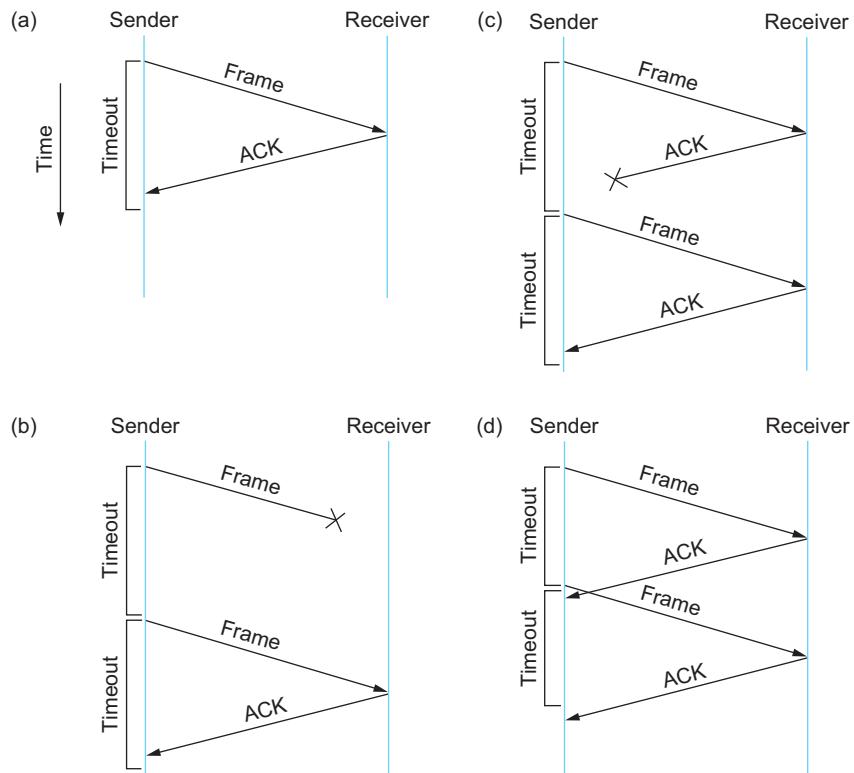
ปัญหานี้ของการส่งข้อมูลขั้น ลิงก์ ได้แก่การปรับอัตราเร็วข้อมูลระหว่างเครื่องส่งและรับข้อมูลในอัตราเดียวกัน เครื่องส่งจะไม่ส่งเร็วเกินกว่าเครื่องรับจะรับได้ ปัญหานี้เกิดขึ้นเมื่อเครื่องส่ง ไม่ทราบว่าเครื่องรับสามารถรับข้อมูลด้วยอัตราเร็วเท่าใด ทำให้มีโอกาสส่งข้อมูลด้วยอัตราเร็วสูงกว่าที่เครื่องรับจะรับได้ ทำให้เกิดข้อมูลลับที่ปลายทาง แต่หากส่งข้อมูลด้วยอัตราเร็วน้อยเกินไปก็จะทำให้ใช้แบนด์วิธไม่เต็มประสิทธิภาพ วิธีแก้ปัญหาการส่งข้อมูลเร็วเกินไปทำได้โดยควบคุมความเร็วที่เครื่องส่ง แต่การจะรู้ว่าต้องส่งด้วยความเร็วเท่าใดนั้นจะต้องรู้ความสามารถในการรับข้อมูลของเครื่องรับจึงจะทำให้การส่งข้อมูลทำได้เต็มประสิทธิภาพ

2.6.1 Stop-and-Wait

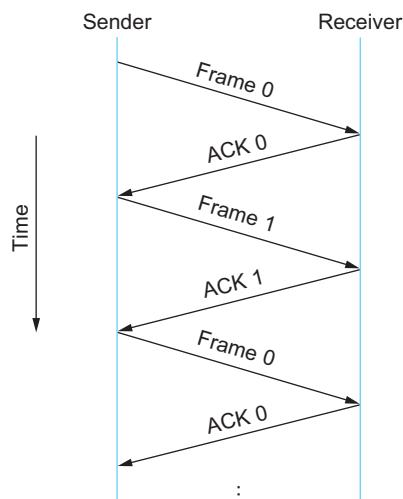
การควบคุมอัตราเร็วข้อมูลด้วยวิธี Stop-and-Wait ใช้การควบคุมอัตราเร็วในการส่งข้อมูลจากน้อยไปมาก จนว่าปลายทางจะรับข้อมูลไม่ไหว วิธีตรวจสอบปลายทางรับไม่ไหวทำได้โดย ทุกครั้งที่ส่งข้อมูลไปเครื่องส่งจะคอยรับการตอบกลับจากปลายทาง หากปลายทางรับข้อมูลไม่ไหวจะทำให้ไม่สามารถส่งข้อมูลตอบกลับมาเครื่องส่งได้ ทำให้เครื่องรู้ว่าข้อความที่ส่งไปถูกต้อง

การทำงาน Stop-and-Wait ตามรูปที่ 2.12(a) ใช้วิธีส่งข้อมูลไปก่อนแล้วรอการตอบกลับ โดยกำหนดค่า Timeout เอาไว้ เมื่อข้อมูลส่งไปแล้วยังไม่ส่งข้อมูลตอบกลับแอ็คโนเล็จเมนท์ ตามรูปที่ 2.12(b) จะถือว่า มีการส่งด้วยอัตราเร็วสูงเกินไป เมื่อกรณีที่ส่งข้อมูลออกไปแล้วไม่ได้รับแอ็คโนเล็จเมนท์ภายในระยะเวลา timeout จะทำให้เครื่องส่งเข้าใจได้ว่ามีการส่งข้อมูลด้วยอัตราเร็วเกินไปทำให้ข้อมูลลับ และไม่สามารถตอบกลับได้ จึงปรับอัตราเร็วข้อมูลลง เป็นไปตามรูปที่ 2.12(c)

จากรูปที่ 2.12(d) อาจเกิดกรณีที่แอ็คโนเล็จเมนท์เดินทางมาช้าทำให้เครื่องส่งได้ส่งข้อมูลเข้าไปยังปลายทาง แต่สามารถป้องกันปัญหาดังกล่าว ได้โดยเครื่องส่งจะกำหนดหมายเลขแอ็คโนเล็จเมนท์ ให้แต่ละเฟรม เมื่อมีแอ็คโนเล็จเมนท์ เดินทางกลับมาจะตรวจสอบหมายเลข(ack) เพื่อไม่ให้เกิดการส่งซ้ำทุกครั้งถือว่า มีการควบคุมอัตราเร็วของทั้งสองฝ่ายเท่ากัน เป็นไปตามรูปที่ 2.13



รูปที่ 2.12: การควบคุมอัตราเร็วข้อมูลด้วยวิธี Stop-and-wait



รูปที่ 2.13: การเพิ่มหมายเลขแอ็กโนเเล็จเมนท์ เพื่อป้องกันการส่งซ้ำของข้อมูลที่เดินทางกลับซ้ำ
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

วิธี stop-and-wait เป็นการส่งทีละเฟรมแล้วรอจนกว่าจะได้รับแอ็กโนเเล็จเมนท์ แล้วจึงส่งเฟรมถัดไป

ตัวอย่าง 2.1 เครือข่ายมีความเร็ว 1.5 Mbps ใช้ระยะเวลาเดินทางไปกลับ 4.5 ms ลิงค์นี้จะได้มี $\text{bandwidth} \times \text{delay}$ product เท่ากับ 67.5 Kb หรือประมาณ 8 KB จากที่เครื่องล่าสุดจะล่าสุดได้ครั้งละ 1 เฟรม อัตราเร็วสูงสุดของ

เครือข่ายนี้จะเท่ากับ

$$\text{Bits-PerFrame}/\text{Time-Per-Frame} = 1024 \times 8 / 0.045 = 182 \text{ kbps}$$

จากการคำนวณตัวอย่างที่ 2.1 เห็นได้ว่าความเร็วเครือข่ายมี 1.5Mbps แต่สามารถใช้งานได้เพียง 1/8 ส่วนเท่านั้น

วิธีควบคุมอัตราเร็วด้วย Stop-and-wait นี้เป็นวิธีที่ยังขาดประสิทธิภาพ เพราะทำให้เครื่องส่งเสียเวลารอคอยในช่วง timeout จึงเกิดการปรับปรุงประสิทธิภาพด้วยการใช้วิธี Sliding window

2.6.2 Sliding Window

วิธี sliding window ปรับปรุงประสิทธิภาพของ Stop-and-Wait ด้วยการส่งเฟรมต่อเนื่องเพื่อไม่ให้เสียช่วงเวลาการอคoyer ทำให้ส่งข้อมูลได้ไกล์เดียว $delay \times bandwidth$ ตามรูปที่ 2.14 ซึ่งยังคงใช้แนวคิดการทำงานร่วมกันระหว่าง แอ็อกโนเหล็จเมนท์ และ timeout เครื่องส่งจะตรวจพบข้อมูลสูญหายได้จากหมายเลขแอ็อกโนเหล็จเมนท์ โดยรอไม่เกินเวลา timeout

จากตัวอย่างที่ 2.1 ได้แสดงให้เห็นว่าวิธี stop-and-wait ทำให้มีการใช้ความสามารถของเครือข่ายได้เพียง 1 ใน 8 ตามค่าความจุเครือข่าย $delay \times bandwidth$ product สำหรับหัวข้อนี้จะปรับปรุงแนวคิด stop-and-wait โดยเปลี่ยนวิธีควบคุมอัตราเร็วเป็น sliding window หลักการทำงานของ sliding window มีดังนี้ จากข้อจำกัด $delay \times bandwidth = 8KB$ หากส่งข้อมูลครั้งละ 1 KB จะส่งได้ 8 เฟรม ก่อน จะได้รับแอ็อกโนเหล็จเมนท์ สังเกตตามรูปที่ 2.14 ปรับปรุงให้มีการส่งต่อเนื่องโดยไม่ต้องรอแอ็อกโนเหล็จเมนท์ จำนวนเฟรมที่ส่งไปจนกว่าจะเต็มความจุของเครือข่าย และเมื่อได้รับแอ็อกโนเหล็จเมนท์จึงเตรียมเฟรมในเครือข่าย เรียกวิธีนี้ว่า “sliding window”

Sliding Window Algorithm

ทำความเข้าใจการทำงาน sliding window ได้ดังนี้ เมื่อมีข้อมูลส่งผ่านเครือข่าย ระบบสื่อสารจะเริ่มต้นจากสร้างแพ็กเก็ตในขั้นตอนพลิกเครื่นส่งต่อลงมาถึงอะแดปเตอร์ และเขียนข้อมูลลงหน่วยความจำของอะแดปเตอร์ เมื่ออะแดปเตอร์พบรข้อมูลในหน่วยจำจะอ่านข้อมูลจากหน่วยความจำนั้น และส่งออกไปปลายทาง เมื่อปลายทางได้รับข้อมูลจะเก็บข้อมูลลงหน่วยความจำของอะแดปเตอร์เครื่องปลายทาง ก่อนอ่านข้อมูลเพื่อส่งต่อขึ้นไปเลเยอร์ด้านบน

กำหนดให้อะแดปเตอร์มีหน่วยความจำเรียกว่าบัฟเฟอร์ บัฟเฟอร์ที่เครื่องส่งแทนด้วย SendBuf และบัฟเฟอร์ที่อะแดปเตอร์เครื่องรับแทนด้วย RecvBuf ตัวแปรทั้งสองนี้ใช้เก็บข้อมูลก่อนนำไปประมวลผล

Sliding Window ยอมให้ส่งเฟรมต่อเนื่องได้แต่ต้องไม่เกินความจุของลิงค์ ถึงแม้เป้าหมายต้องการส่งให้เร็วที่สุดแต่ก็ยังมีข้อจำกัดจากความสามารถของลิงค์และความสามารถของเครื่องรับ ดังนั้นจึงต้องการวิธีการควบคุมอัตราเร็วของเฟรม เพื่อใช้จำกัดการส่งของเครื่องส่งไม่ให้ข้อมูลล้นเครือข่าย

การควบคุมอัตราเร็วการส่งข้อมูลแบบ sliding window เป็นการควบคุมเครื่องส่งกำหนดขนาด SendBuf ให้ตรงกับ RecvBuf โดยมีเงื่อนไขว่าเครื่องส่งไม่สามารถถือ RecvBuf ของเครื่องรับได้โดยตรง และส่งข้อมูลได้ไม่เกินขนาดบัฟเฟอร์สูงสุดที่พร็อกโคลกำหนดไว้ กำหนดให้มีขนาดเป็น SWS(send window size) หน่วยไบต์ และ ข้อจำกัดของบัฟเฟอร์เครื่องรับแทนด้วย RWS(receive window size) หน่วยไบต์

ค่า SWS ไบต์ ที่เครื่องส่ง และ RWS ไบต์ ที่เครื่องรับเป็นค่าคงที่ไม่จำเป็นต้องมีขนาดเท่ากัน หน้าที่ของอะแดปเตอร์ทั้งสองทางจะทำหน้าที่ควบคุมอัตราเร็วของการส่งรับข้อมูลไม่ให้มีเฟรมที่ยังส่งไม่เสร็จ(ยังไม่ได้รับ ACK) มีมากกว่า SWS หรือ RWS ที่เครื่องส่งและรับตามลำดับ

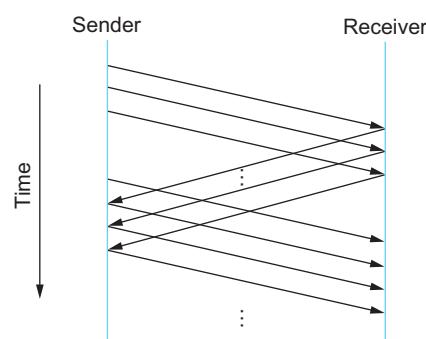
ดังนั้นบัฟเฟอร์ที่อะแดปเตอร์จะมีการเปลี่ยนแปลงตลอดเวลา ตามภาวะเครือข่าย เช่น การทำงานของเลเยอร์ที่สูงขึ้นในทั้งสองเครื่องและการส่งเฟรมซ้ำเมื่อไม่ได้รับแอ็กโนเหล็จเมนท์ แต่กำหนดขนาดสูงสุดสำหรับเครื่องส่งได้ไม่เกิน SWS และเครื่องรับได้ไม่เกิน RWS

บัฟเฟอร์ ที่เครื่องส่ง ณ เวลาใดๆ คำนวนโดยใช้ความต่าง SeqNum² ของเฟรมที่ส่งกับเฟรมที่ตอบแอ็กโนเหล็จเมนท์ เป็นตามสมการที่(2.5) เครื่องส่งจะควบคุม LFS ที่ทำให้ LFS - LAR ยังคงน้อยกว่าหรือเท่ากับ SWS

$$LFS - LAR \leq SWS \quad (2.5)$$

LFS(last frame sent) แทน SeqNum ล่าสุดที่กำลังจะส่ง และ LAR(last acknowledge received) แทน SeqNum ที่ได้รับจาก แอ็กโนเหล็จเมนท์

ที่จุดเริ่มต้นของการส่งข้อมูล เครื่องส่งจะส่งเฟรมต่อเนื่องไปจนมีจำนวนข้อมูลเต็มขนาด SWS เช่น ลิงค์มีความจุ $delay \times bandwidth = 8KB$ เฟรมมีขนาด 1KB เครื่องส่งจะส่งเฟรมต่อเนื่องจำนวน 8 เฟรม ทำให้ข้อมูลเต็มความจุ 8KB แล้วจึงหยุด พร้อมกับร่องกว่าจะมี(แอ็กโนเหล็จเมนท์) จากเครื่องรับตอบกลับมา หากตอบกลับจะทำให้มีพื้นที่ buffer เหลือจากที่ส่งสำเร็จ เครื่องส่งจะส่งเฟรมเพิ่มเท่ากับจำนวน buffer ที่ได้คืนมา



รูปที่ 2.14: การควบคุมอัตราเร็วข้อมูลด้วยวิธี Sliding Window
ลิขสิทธิภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

² ในที่นี้กำหนดให้เฟรมมีขนาด 1 ไบต์ ซึ่งหากว่าแต่ละเฟรมมีขนาดเป็นอย่างอื่นสามารถใช้ ความยาวเฟรม (Frame Length) ในการคำนวณได้

ตัวแปรที่เกี่ยวข้องกับการคำนวนที่เครื่องส่งได้แก่ LFS LAR SWS และ timeout อธิบายการประมวลผลที่เครื่องส่งได้ดังนี้

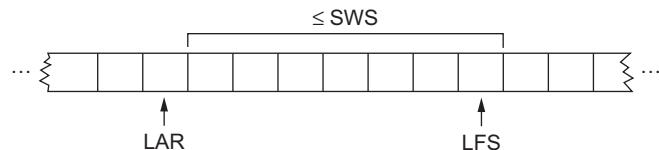
ที่เครื่องส่ง สมมติหนึ่งเฟรมมีขนาด 1-ไบต์(byte) เริ่มส่งเฟรมไปเครื่องรับครั้งละเฟรมต่อเนื่องไปเรื่อยๆ โดยกำหนดค่าเริ่มนั้น seqnum=1 พร้อมกับกำหนด LFS เท่ากับ seqnum ในที่นี้เป็นเฟรมแรกยังไม่ได้รับ แอ็คโนเหล็จเม้นท์ จึงกำหนด LAR=0 สังเกตได้ว่า LAR จะคงที่ตราบเท่าที่ยังไม่ได้รับแอ็คโนเเหล็จเม้นท์ ขณะที่ LFS เพิ่มจำนวนขึ้นจนกว่า LFS กับ LAR จะมีความต่างกันมากกว่า SWS จึงหยุดส่ง

การส่งแบบ sliding window จะส่งไปเรื่อยๆจนกว่าจะส่งเต็ม window ซึ่ง window มีขนาด SWS เครื่องส่งจะหยุดส่งเมื่อ LFS-LAR > SWS โดยที่ LFS-LAR คือค่า sliding window (WS)

$$WS = LFS-LAR \quad (2.6)$$

จะเห็นได้ว่า LFS เพิ่มขึ้นตามจำนวนเฟรมที่ส่งออก ทำให้ขนาด WS เพิ่มขึ้นเรื่อยๆ แต่จะไปหยุดที่ $WS > SWS$

แต่เมื่อเครื่องส่งได้รับแอ็คโนเเหล็จเม้นท์ จะเพิ่ม LAR ไปทางขวาซึ่งเป็นการยืนยันว่าข้อมูลที่ส่งไปแล้วเดินทางถึงปลายทาง เห็นได้ว่าเมื่อ LAR เพิ่มขึ้นทำให้ WS มีค่าน้อยกว่า SWS จะทำให้เครื่องส่งสามารถส่งข้อมูลได้อีกหลังจากหยุดรอ การทำงานเป็นตามรูปที่ 2.15



รูปที่ 2.15: ตัวแปรสำหรับคำนวนการปรับขนาดวินโดว์ของเครื่องส่ง
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ทางฝั่งเครื่องรับ เมื่อได้รับข้อมูล seqnum=1 จะอ่านข้อมูลและประมวลผลพร้อมกับตอบ แอ็คโนเเหล็จเม้นท์ทันที ในการรับข้อมูลครั้งแรก buffer ยังว่างทำให้เครื่องรับยังคงรับข้อมูลได้ต่อเนื่อง

ในกรณีที่เฟรมเดินทางมาเร็วเกินไป เครื่องรับมีการตรวจสอบโดยกำหนดความสามารถในการรับข้อมูลได้ไม่เกิน RWS และใช้แนวทางแบบเดียวกับเครื่องส่งดังนี้

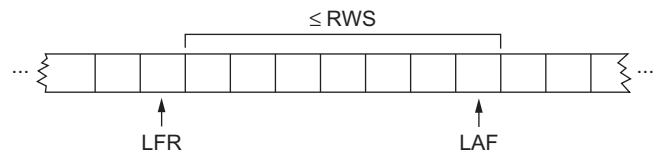
รายละเอียดการทำงานภายใต้เครื่องรับมีดังนี้ เมื่อเฟรมเดินทางมาถึงมี โดย SeqNum อยู่นอกพื้นที่รับ (outside receiver's window) ดังนี้ $\text{SeqNum} \leq \text{LFR}$ และ $\text{SeqNum} > \text{LAF}$ จะปฏิเสธเฟรม แต่ถ้า $\text{LFR} < \text{SeqNum} \leq \text{LAF}$ จะรับเฟรมเข้าระบบ ในลำดับต่อไปเครื่องรับจะพิจารณาว่าจะตอบ แอ็คโนเเหล็จเม้นท์กลับไปเครื่องส่งหรือไม่ กำหนดให้ SeqNumToAck แทนค่า SeqNum ค่ามาสุดที่ยังไม่เคย แอ็คโนเเหล็จเม้นท์ เมื่อเครื่องรับพร้อมส่งแอ็คโนเเหล็จเม้นท์จะใช้ SeqNumToAck เป็นค่า Acknowledge number ทำให้ $\text{LFR} = \text{SeqNumToAck}$ แล้วปรับค่า $\text{LAF} = \text{LFS} + \text{RWS}$

ตัวอย่างเช่น กำหนดให้ $\text{LFR}=5$ และ กำหนดให้ $\text{RWS}=4$ ซึ่งหมายความได้ว่า $\text{LAF}=\text{LFR}+\text{RWS} = 5+4 = 9$ ลำดับต่อมามีเฟรม SeqNum = 7 และ 8 เดินทางมาถึง จะถูกเก็บเข้า buffer เพราะ 7 และ 8 อยู่ใน

ช่วง $[LFR, LAF] = [5, 9]$ แต่เฟรม 7 และ 8 มาโดยไม่มีเฟรม 6 มา ก่อน อาจจะเกิดอะไรขึ้นกับเฟรม 6 ในที่นี่เครื่องรับยังไม่ส่ง แอ็กโนเหล็จเม้นท์เพราเพرم SeqNum = 6 ยังเดินทางมาไม่ถึง ระหว่างนี้ค่า LFR=5 และมี LAF=9 ต่อมาเฟรม 6 เดินทางมาถึงพบว่า $[LFR, LAF] = [5, 9]$ เครื่องรับพบว่า SeqNum=6 อยู่ในช่วง Received Window จึงรับ ในระบบจะมีเฟรม 6 7 และ 8 ที่กำลังเตรียมตัว แอ็กโนเหล็จเม้นท์ สำหรับเฟรม ที่มี SeqNum มากที่สุดและยังไม่ตอบแอ็กโนเหล็จเม้นท์ กำหนดให้ชื่อ SeqNumToAck=8 เครื่องรับจึงเริ่มส่ง แอ็กโนเหล็จเม้นท์ ที่มี SeqNum=6, 7 และ 8 ตามลำดับ โดยปรับค่า LFR และ LAF ดังนี้เมื่อตอบเฟรม 6 จึง เชต LFF=6, LAF=10 ต่อมา แอ็กโนเหล็จเม้นท์ เพرم 7 จะเชตค่า LFR=7, LAF=11 ต่อมา แอ็กโนเหล็จเม้นท์ เพرم 8 และเชตค่าเพิ่ม LFR=8 พร้อมกับปรับค่า LAF = 8+4 = 12 ทำให้มี Received Window ในช่วง [8, 12]

เมื่อข้อมูลเดินทางมาเรือยา เครื่องรับจะรับข้อมูลเมื่อ SeqNum อยู่ในช่วง Received Window สำหรับเฟรมลำดับต่อไปจะเพิ่มค่า LAF ไปเรือยา ซึ่งระหว่างนี้เครื่องรับจะประมวลผลตรวจสอบความสมบูรณ์ ลงนำไปใช้งานหากทุกอย่างปกติจะส่งแอ็กโนเหล็จเม้นท์กลับไปเครื่องส่ง โดยที่ยังคงรับเฟรมที่เดินทางมาจาก เครื่องส่งเรือยา หากอัตราเร็วเครื่องส่งสูงว่าที่เครื่องรับได้รับ จะทำให้ LAF เพิ่มค่าไปเรือยาจน LAF มากกว่า $LFR + RWS$ ถือเป็นขีดจำกัดที่เครื่องรับจะรับได้ การทำงานเป็นตามรูปที่ 2.16

เมื่อข้อมูลถูกส่งมาจนเครื่องรับไม่สามารถรับไหว จะทำให้มีแอ็กโนเหล็จเม้นท์ ส่งกลับไปเครื่องส่ง หากเครื่องส่งรอจนถึงเวลา timeout ยังไม่ได้รับแอ็กโนเหล็จเม้นท์ จะเริ่มส่งเฟรมที่มี seqnum ที่ไม่มีแอ็กโน เหล็จเม้นท์ นั้นใหม่ ถือเป็นสิ่นสุดขั้นตอนทำงาน sliding window



รูปที่ 2.16: ตัวแปรสำหรับคำนวนการปรับขนาดวินโดว์ของเครื่องรับ
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

กลับมาที่เครื่องส่ง เมื่อเครื่องส่งได้รับแอ็กโนเหล็จเม้นท์ที่มีหมายเลข SeqNum ล่าสุดที่เครื่องรับตอบกลับ จะเริ่มคืนหัวร่ายความจำโดยนำค่า SeqNum ที่อ่านได้แอ็กโนเหล็จเม้นท์ มาลบออกจากจำนวนที่ได้ส่งไป แล้ว เช่น มี แอ็กโนเหล็จเม้นท์ ของ SeqNum=5 เครื่องส่งจะหยิบ จาก LAR=5 เป็น LAR=6 ซึ่งทำให้ LFS-LAR มีพื้นที่เหลือมากขึ้นหนึ่งเฟรม ทำให้เครื่องส่งสามารถส่งเฟรมเพิ่มได้อีกหนึ่งเฟรม

Finite Sequence Numbers and Sliding Window

$$SWS < (MaxSeqNum + 1)/2 \quad (2.7)$$

เก้าดการทำงานของ Sliding Window

```
typedef u_char SwpSeqno;
```

```

typedef struct {
    SwpSeqno SeqNum; /* sequence number of this frame */
    SwpSeqno AckNum; /* ack of received frame */
    u_char Flags; /* up to 8 bits worth of flags */
} SwpHdr;

typedef struct {
    /* sender side state: */
    SwpSeqno LAR; /* seqno of last ACK received */
    SwpSeqno LFS; /* last frame sent */
    Semaphore sendWindowNotFull;
    SwpHdr hdr; /* pre-initialized header */
    struct sendQ_slot {
        Event timeout; /* event associated with send-timeout */
        Msg msg;
    } sendQ[SWS];
}

/* receiver side state: */
SwpSeqno NFE; /* seqno of next frame expected */
struct recvQ_slot {
    int received; /* is msg valid? */
    Msg msg;
} recvQ[RWS];
} SwpState;
}

static int
sendSWP(SwpState *state, Msg *frame)
{
    struct sendQ_slot *slot;
    hbuf[HLEN];

    /* wait for send window to open */
    semWait(&state->sendWindowNotFull);
    state->hdr.SeqNum = ++state->LFS;
    slot = &state->sendQ[state->hdr.SeqNum % SWS];
    store_swp_hdr(state->hdr, hbuf);
    msgAddHdr(frame, hbuf, HLEN);
    msgSaveCopy(&slot->msg, frame);
    slot->timeout = evSchedule(swpTimeout, slot, SWP_SEND_TIMEOUT);
}

```

```

        return send(LINK, frame);
    }

static int
deliverSWP(SwpState state, Msg *frame)
{
    SwpHdr    hdr;
    char      *hbuf;

    hbuf = msgStripHdr(frame, HLEN);
    load_swp_hdr(&hdr, hbuf)
    if (hdr->Flags & FLAG_ACK_VALID)
    {
        /* received an -acknowledgmentdo SENDER side */
        if (swpInWindow(hdr.AckNum, state->LAR + 1, state->LFS))
        {
            do
            {
                struct sendQ_slot *slot;

                slot = &state->sendQ[++state->LAR % SWS];
                evCancel(slot->timeout);
                msgDestroy(&slot->msg);
                semSignal(&state->sendWindowNotFull);
            } while (state->LAR != hdr.AckNum);
        }
    }

    if (hdr.Flags & FLAG_HAS_DATA)
    {
        struct recvQ_slot *slot;

        /* received data -packetdo RECEIVER side */
        slot = &state->recvQ[hdr.SeqNum % RWS];
        if (!swpInWindow(hdr.SeqNum, state->NFE, state->NFE + RWS - 1))
        {
            /* drop the message */
            return SUCCESS;
        }
    }
}

```

```

    }

    msgSaveCopy(&slot->msg, frame);
    slot->received = TRUE;
    if (hdr.SeqNum == state->NFE)
    {
        Msg m;

        while (slot->received)
        {
            deliver(HLP, &slot->msg);
            msgDestroy(&slot->msg);
            slot->received = FALSE;
            slot = &state->recvQ[++state->NFE % RWS];
        }
        /* send ACK: */
        prepare_ack(&m, state->NFE - 1);
        send(LINK, &m);
        msgDestroy(&m);
    }
    return SUCCESS;
}

static bool
swpInWindow(SwpSeqno seqno, SwpSeqno min, SwpSeqno max)
{
    SwpSeqno pos, maxpos;

    pos      = seqno - min;          /* pos *should* be in range [0..MAX] */
    maxpos = max - min + 1;         /* maxpos is in range [0..MAX] */
    return pos < maxpos;
}

```

2.7 Multi-Access Networks

จากการพัฒนาวิจัยของ PARC(Xerox Palo Alto Research Center) ทำให้เกิดระบบ อีเทอร์เน็ต ส่งผลให้มี การเปลี่ยนแปลงวิธีสื่อสารจากเดิมโดยสิ้นเชิง ปัจจุบันเทคโนโลยีอีเทอร์เน็ตพัฒนาต่ออยอดเป็นเทคโนโลยีแลนไวร์

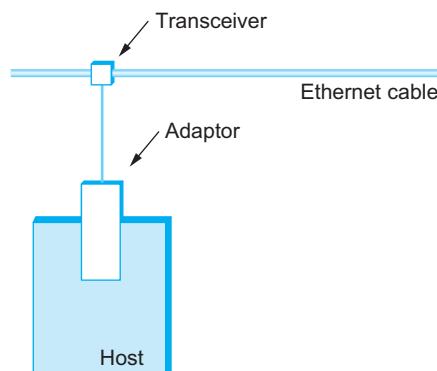
สาย เป็นสองเทคโนโลยีที่มีการใช้งานแพร่หลายในโลกนี้ พื้นฐานการเข้าใช้ช่องสัญญาณของอีเทอร์เน็ต คือการใช้วิธี CSMA/CD(Carrier Sense, Multiple Access with Collision Detect)

โดยที่ CSMA(Carrier Sense, Multiple Access) มาจากการระบบอีเทอร์เน็ตจะตรวจสอบความว่างของช่องสัญญาณก่อนที่จะส่งข้อมูล วิธีการตรวจสอบความว่างของช่องสัญญาณเรียกว่า “carrier sense” โดยที่ carrier คือสัญญาณไฟฟ้าในสายสัญญาณ และ sense คือการเช็คสายสัญญาณ

การทำงานของ carrier sense คือตรวจสอบแรงดันไฟฟ้าในสายสัญญาณ หากมีแรงดันไฟฟ้าหมายถึงมีเครื่องไดเครื่องหนึ่งกำลังส่งข้อมูล หากวัดแล้วแรงดันไฟฟ้าเป็น 0V หมายถึงช่องสัญญาณว่าง

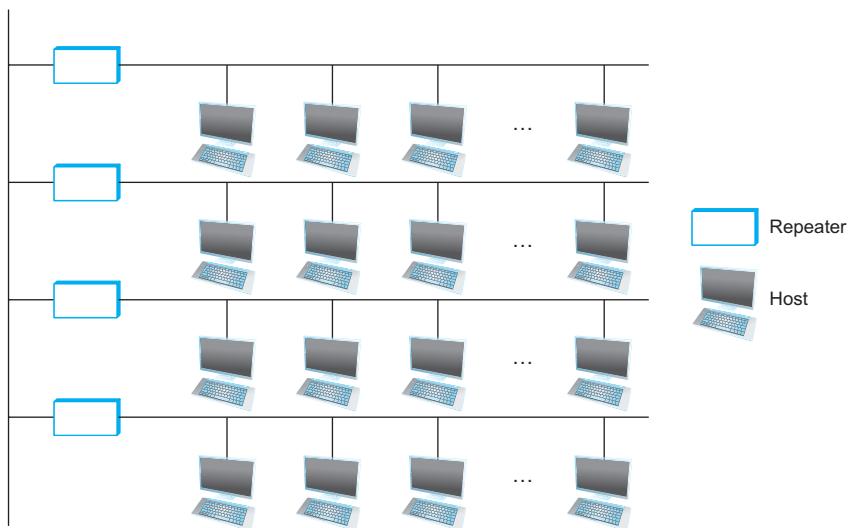
2.7.1 Physical Properties

การสื่อสารด้วยผมเริ่มต้นจากการใช้สายนำสัญญาณชนิด โคแอกซิ얼(coaxial cable) มีความยาว 500 เมตรสายนำสัญญาณของอินเทอร์เน็ตสมัยใหม่ใช้สายทองแดงตีเกรียวโดยไม่ต้องมีฉนวนหุ้มรู้วัจกันในชื่อ “Category 5” หรือมีการใช้สายใหญ่กว้างนำเสนอ สายชนิดโคแอกซิ얼นี้เป็นประเภทเดียวกับสายสัญญาณที่ใช้กับ โทรทัศน์ (television) โพสต์ที่เชื่อมผ่านเครือข่ายอินเทอร์เน็ตจะนำอุปกรณ์ไปหนีบกับสายโคแอกซิ얼 ตามรูปที่ 2.17



รูปที่ 2.17: ทรานซีฟเวอร์และอะแดปเตอร์ ในเครือข่ายอีเทอร์เน็ต
ลิขสิทธิภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

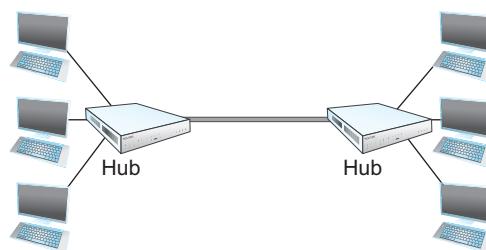
อุปกรณ์ที่หนีบกับสาย โคแอกซิ얼 เรียกว่า transceiver ภายในเครือข่ายจะมีท่านสิเออร์หลายตัวโดยการแบ่งกลุ่มของเครือข่ายจะเรียกว่าเป็น อีเทอร์เน็ตเซกเมนต์(segment) เป็นการใช้เครือข่ายภายในหนึ่งห้องในห้องถัดไปจะมีอีเทอร์เน็ตเซกเมนต์ใหม่ การเชื่อมระหว่างแต่ละ เซกเมนต์ใช้อุปกรณ์ที่ชื่อว่ารีพิทเตอร์ ทำหน้าที่ขยายสัญญาณเบรียบเสมือน วงจรขยายสัญญาณ(amplifier)โดยไม่เข้าใจข้อมูลและจัดบิตหรือเฟรม การใช้รีพิทเตอร์ขยายสัญญาณได้ไม่เกินสีเครื่อง ทำให้ในอดีตสามารถส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ตมีระยะเพียง 2500 เมตร จากรูปที่ 2.18 เป็นการใช้รีพิทเตอร์จำนวนสองตัวขยายสัญญาณ โดยแต่ละแควรเมื่อตอนอยู่ภายในตึกเดียวกันแต่คุณจะเห็น สาเหตุของข้อจำกัดรีพิทเตอร์ที่ไม่สามารถเชื่อมต่อกันได้เกินสีเครื่องนั้นเกิดจาก การแพร่สัญญาณ(propagation) ดีเลย์ซิงส์ผลต่อการทำงานของ CSMA/CD



รูปที่ 2.18: การใช้รีพีตเตอร์ในเครือข่ายอีเทอร์เน็ตขนาดใหญ่
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

2.7.2 Access Protocol

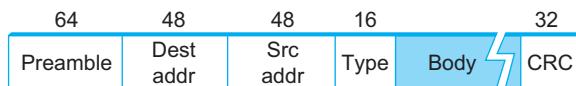
หัวข้อนี้จะกล่าวถึง อัลกอริทึม(algorithm)ในการแชร์ช่องสัญญาณของเครือข่ายอินเทอร์เน็ต เรียกโดยรวมว่า media access control (MAC) อัลกอริทึมนี้เป็นซอฟต์แวร์ติดตั้งภายในการ์ดเครือข่าย การใช้ช่องสัญญาณร่วมกันเป็นไปตามรูปที่ 2.19 มีไฮสต์ ทั้งหมดหากเครื่องต้องการใช้สายสัญญาณร่วมกัน



รูปที่ 2.19: การใช้สายสัญญาณร่วมกันของไฮสต์ทั้งหมดหากเครื่อง
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

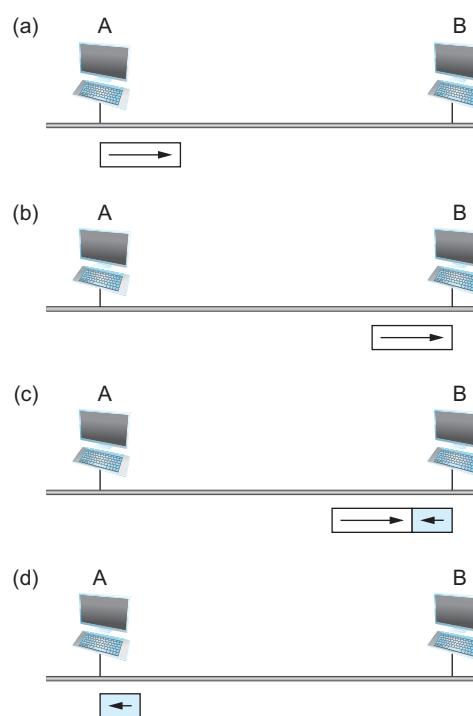
Frame Format

ระบบอีเทอร์เน็ตจะส่งข้อมูลครั้งละเฟรม โดยเฟรมนั้นประกอบด้วยข้อมูลส่วน เอดเดอร์ และ เพย์โหลด อธิบายเพร姆ส่วนເຍດເດອຣ์ตามรูปที่ 2.20 ในส่วนເອສເທອຣປະກອບໄປດ້ວຍຂໍ້ມູນແບ່ງເປັນພິວທັນໜົດ 5 ສ່ວນ ສ່ວນແຮກໄດ້ແກ່ preamble ມີນາດ 64-ບີຕ ກາຍໃນບຣຈຸບີຕ 0 ແລະ 1 ສລັບກັນ ສ່ວນຕ່ອມາ Dest addr ແລະ Src addr ມີນາດ 48-ບີຕ ໃຊ້ຮະບຸໝາຍເລີຂອງຄຸປຣົນປລາຍທາງແລະຕັ້ນທາງ ພຶລດ໌(field)ຕ່ອມາ type ນາດ 16-ບີຕໃ້ກຳນົດໂພຣໂທຄອລທີ່ຈະໃຊ້ໃນລຳດັບຂຶ້ນໄປ ຕ່ອມາເປັນສ່ວນເພີ້ໂຫດ ມີຄວາມຍາວຂຶ້ນອູ້ກັບນາດຂອງຂໍ້ມູນ ແລະ ສຸດທ້າຍເປັນຟຶລດ໌ CRC ມີນາດ 32-ບີຕ



รูปที่ 2.20: โครงสร้างเฟรมระบบอีเทอร์เน็ต
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

รูปที่ 2.21(a) แสดงถึงการส่งข้อมูลโดยโไฮสต์ A เข้าสู่เครือข่ายโดยเครื่อข่ายเชื่อมผ่านสายสัญญาณไปยังโไฮสต์ B ข้อมูลที่ส่งเข้าสู่สายนำสัญญาณนั้นมีการเรียงบิตในรูปแบบเฟรมที่กำหนดในพร็อกโคล เมื่อข้อมูลเดินทางถึงโไฮสต์ B ตามรูปที่ 2.21(b) ที่โไฮสต์ B จะเริ่มต้น kod เพื่อให้พร็อกโคลที่จะตอบโต้ได้กำหนดไว้ร่วมกัน ไว้และสามารถสื่อสารกลับไปยังโไฮสต์ A โดยใช้เฟรมรูปแบบเดียวกัน ตาม 2.21(c)-(d)



รูปที่ 2.21: การส่งเฟรมผ่านสายนำสัญญาณ
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

2.7.3 ข้อสังเกตสิ่งที่ทำให้อีเทอร์เน็ตไม่ต่อกยุค

เทคโนโลยีอีเทอร์เน็ต เป็นเทคโนโลยีที่มีการใช้งานเริ่มตั้งแต่การเชื่อมต่อกันโดยตรงต่อกันมาตั้งแต่ 30 ปีที่ผ่านมาที่มีการใช้งานเริ่มตั้งแต่การเชื่อมต่อ กับเครือข่ายท้องแวดวงชนิด โคแอกซ์ และพัฒนาความเร็วตั้งแต่ 10Mbps จนปัจจุบันสามารถสื่อสารได้ด้วยความเร็ว 1-10Gbps การที่ได้รับการใช้งานมาโดยตลอด 30 ปีและพัฒนามาตรฐานต่อเนื่องนั้นมีจุดสังเกตจาก มาตรฐานการสื่อสารทำงานเข้าใจได้ง่าย ในแนวคิดการเปลี่ยนสัญญาณแบบ ทั้งช่องสัญญาณว่างก่อนส่งออก(listen-before-talk) คือจะส่งข้อมูลเมื่อช่องสัญญาณว่าง ทำงานร่วมกับกับวิธีป้องกันการส่งพร้อมกันด้วยวิธีสุ่ม จากข้อเสนอเรียบง่ายและมีประสิทธิภาพทำให้พร็อกโคล รองรับการขยายตัวได้ และได้รับการใช้งานมาถึงปัจจุบัน

2.8 เครือข่ายแลนไร้สาย

เทคโนโลยีอิเล็กทรอนิกส์และเทคโนโลยีแลนไร้สายมีความแตกต่างกัน ถึงแม้ใช้เทคนิค time sharing เช่นเดียวกัน ความแตกต่างเกิดจากด้านประสิทธิภาพ จากที่เคยกล่าวมา และไร้สายใช้ช่องสัญญาณแบบแบ่งเหล็กไฟฟ้า ทำให้วิศวกรต้องออกแบบบวชิพิเศษสำหรับการจัดการคลื่นที่แพร่รอบทิศทาง นอกจากนี้การลดTHONสัญญาณโดยแบ่งเหล็กไฟฟ้านั้นลดลงอย่างมากถึงแม้ระยะห่างจากเครื่องส่งไม่เกินเมตร และยังมีปัญหาอย่างอื่นที่มากกว่า เป็นการการสื่อสารผ่านสายนำสัญญาณ ซึ่งจะกล่าวถึงในลำดับต่อไป

ข้อดีของการสื่อสารผ่านแม่เหล็กไฟฟ้าคือไม่ต้องการสายสัญญาณนั้นเป็นปะโยชน์อย่างมากในการสื่อสาร ทำให้นักวิทยาศาสตร์พยากรณ์แก้ปัญหาที่เกิดกับการส่งผ่านแม่เหล็กไฟฟ้า ซึ่งปัจจุบันมีการพัฒนา ก้าวหน้าอย่างมาก

2.8.1 ปัญหาทั่วไปของคลื่นแม่เหล็กไฟฟ้า

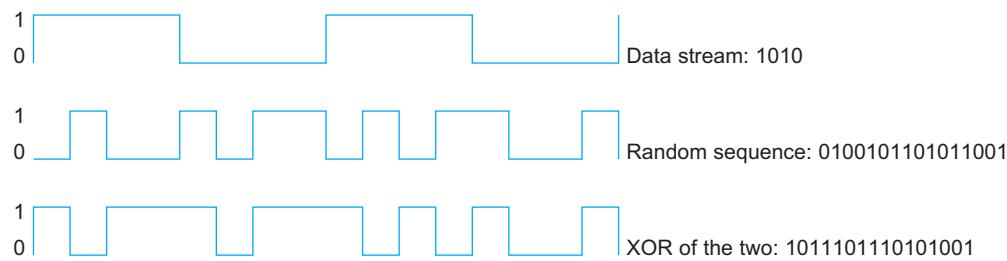
หากใช้แม่เหล็กไฟฟ้าความถี่เดียวกันจะทำให้เกิดการรบกวนระหว่างกัน ในแต่ละประเทศจึงกำหนดให้แม่เหล็กไฟฟ้าเป็นทรัพย์กรของประเทศที่ต้องมีหน่วยงานกำกับดูแล สำหรับประเทศไทยได้กำหนดให้หน่วยงานชื่อ กสทช. (คณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ) ทำหน้าที่กำกับควบคุมการใช้งานแม่เหล็กไฟฟ้า ทุกอุปกรณ์สื่อสารที่ใช้แม่เหล็กไฟฟ้าจะต้องผ่านการรับรองโดย กสทช. ซึ่งมีอุปกรณ์บางความถี่ได้รับการยกเว้น ซึ่งเป็นไปตามสากล ได้แก่อุปกรณ์ในย่านความถี่ 2.4GHz และ 5GHz ได้รับยกเว้นให้ใช้งานได้โดยกำลังส่งไม่เกินกำหนด ซึ่งส่งได้ประมาณ 30เมตร ใช้ได้โดยไม่ต้องขออนุญาต

เมื่อมีช่วงความถี่ที่สามารถใช้งานได้แล้ว ในการสื่อสารจริงที่มีอุปกรณ์สื่อสารจำนวนมากต้องการใช้ความถี่เดียวกัน เทคโนโลยีที่ทำให้อุปกรณ์สื่อสารจำนวนมากสามารถใช้งานในช่วงความถี่เดียวกันได้ ใช้เทคโนโลยี การแสเพกตัรัม(spread spectrum)

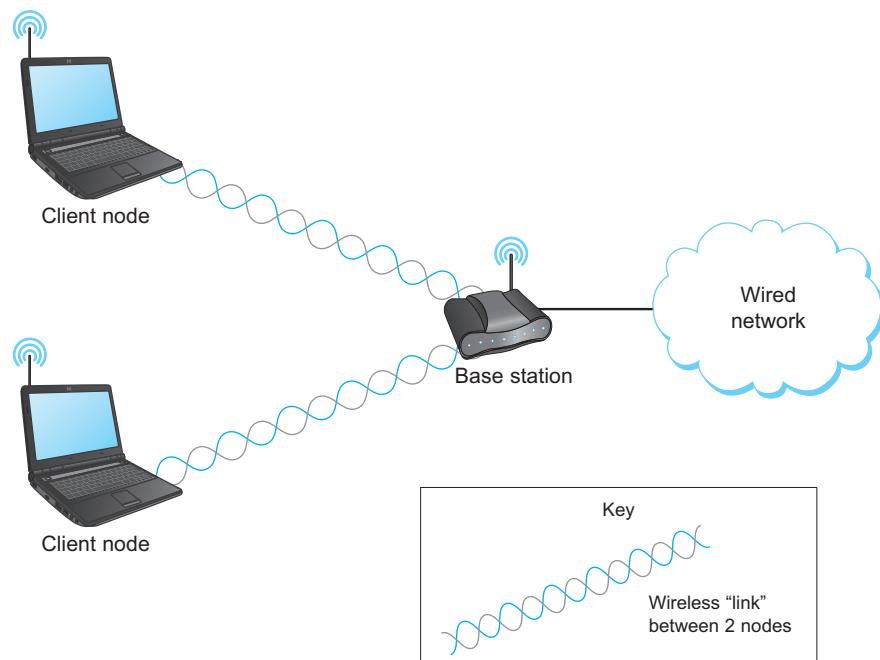
เป็นที่เข้าใจกันแล้วว่าการสื่อสารด้วยแม่เหล็กไฟฟ้า มีการส่งสัญญาณแบบแพร่กระจายรอบทิศทาง ทำให้ข้อมูลถูกส่งไปยังเครื่องอื่นที่ไม่ใช่คู่สื่อสารในการใช้เทคโนโลยีการแสเพกตัรัม เพื่อป้องกันอุปกรณ์สื่อสารที่ไม่ใช่คู่สื่อสารรวมกันระหว่างกันมีเทคนิคอยู่ 2 วิธีที่ถูกนำมาใช้ได้แก่ การกระจายความถี่(frequency spread) และ การเข้ารหัสสัญญาณโดยตรง (direct sequence) ทั้งสองวิธีนี้มีเพื่อเตรียมสัญญาณสำหรับจัดกลุ่มให้อุปกรณ์ทุกเครื่องที่มีการเชื่อมต่อ สถานีฐาน เดียวกันใช้มองเห็นข้อมูลเดียวกัน เทคโนโลยีที่เกี่ยวข้องได้แก่ FHSS(frequency hopping spread spectrum) และ DSSS

เทคโนโลยี FHSS ออกแบบให้ทุกคู่สื่อสารใช้งานความถี่เดียวกันทำให้อุปกรณ์อื่นที่ไม่ได้ใช้งานความถี่เดียวกันนี้ไม่สามารถสื่อสารหรือรบกวนสัญญาณได้

เทคโนโลยี DSSS ออกแบบให้มีการใช้โคดสำหรับบุคคลสื่อสารวิธีการนี้สามารถทำให้อุปกรณ์สื่อสารในเครือข่ายใช้ความถี่เดียวกันได้รูปแบบโดยเป็นไปตามรูปที่ 2.22 ลูกข่ายเครือข่ายไร้สายจะมีระดับสำหรับใช้สื่อสารเฉพาะเครื่อง ช่วยให้สถานีฐาน สามารถควบคุมทรัพยากรคลื่นแม่เหล็กไฟฟ้าสำหรับแต่ละลูกข่ายได้ ในรูปที่ 2.23 แสดงถึงการสื่อสารของลูกข่ายสองโดยมีสถานีฐานเครื่องเดียว สถานีฐานทำหน้าที่ส่งต่อสัญญาณ และควบคุมการเข้าถึงเครือข่ายของลูกข่ายทั้งสอง



รูปที่ 2.22; การเข้ารหัสด้วยเทคโนโลยี DSSS
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

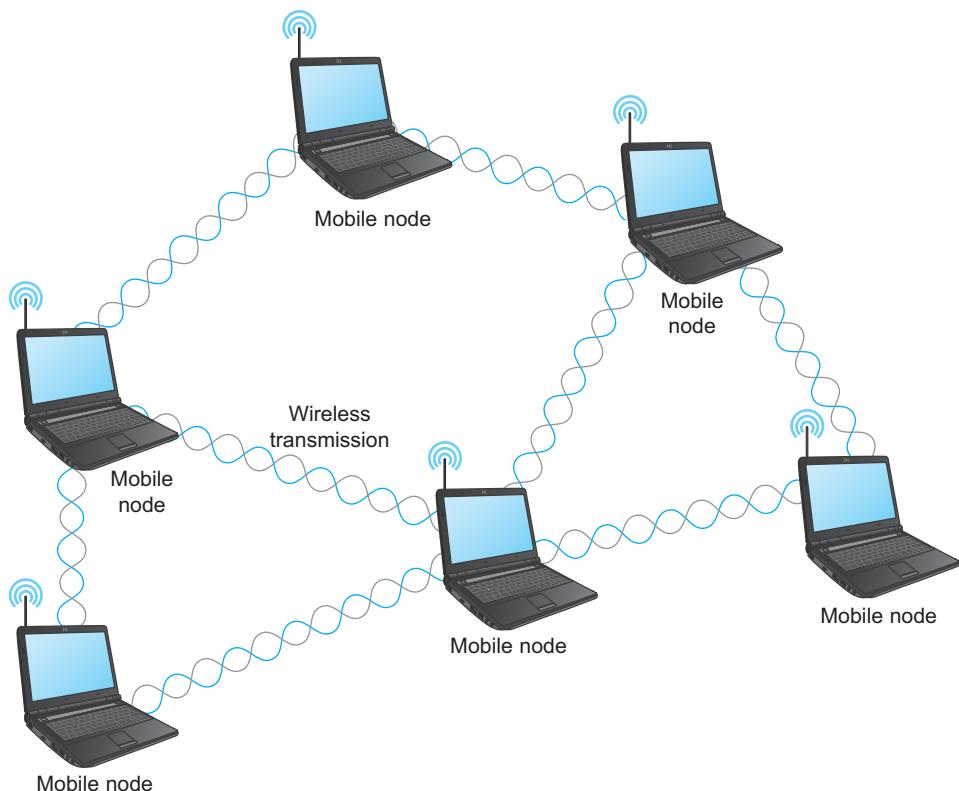


รูปที่ 2.23: ลูกข่ายแลนเร็วสายเชื่อมเครือข่ายผ่านสถานีฐาน
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

นอกจากนี้เครือข่ายไร้สาย ยังสามารถเชื่อมต่อ เครือข่ายไร้สายโดยไม่ต้องใช้สายสัญญาณเรียกเครือข่ายแบบนี้ว่าเครือข่าย เมช(mesh) และเครือข่าย เอพะกิก(ad-hoc) ดังรูปที่ 2.24 เครือข่ายแบบไม่ต้องการใช้สายมีการเชื่อมสัญญาณนี้จะเป็นประโยชน์ในพื้นที่ที่ติดตั้งสายสัญญาณและยากหรือการวางโครงข่ายพื้นฐานได้ลำบาก เช่นบริเวณห่างไกลได้อุปกรณ์จะทำหน้าที่สร้างโครงข่ายผ่านเครือข่ายไร้สายระหว่างกัน ตัวอย่างการใช้เครือข่ายเมชเช่นพื้นที่ป่า หรือบริเวณที่มีความอันตราย เช่นพื้นที่มีภัยมันตะพาบังสีเป็นต้น

2.8.2 มาตรฐาน IEEE 802.11/Wi-Fi

ทุกคนต้องได้เคยใช้งานเครือข่ายและในสายเป็นปกติอยู่แล้วอย่างเช่นในคณะวิศวกรรมศาสตร์ก็มีการให้บริการเครือข่ายและไร้สายในทุกพื้นที่โดยนักศึกษาสามารถเชื่อมต่อโครงข่ายได้โดยไม่มีค่าใช้จ่ายและอุปกรณ์โทรศัพท์คืนที่ทุกเครื่องรองรับการเชื่อมต่อผ่านและไร้สายถือว่าเป็นของหายที่มี ที่มีการใช้งานมากที่สุดโครงข่ายหนึ่ง

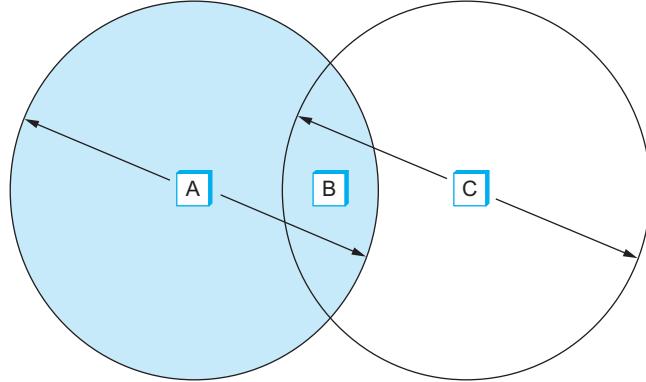


รูปที่ 2.24; รูปแบบการเชื่อมต่อแบบ Ad-hoc
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ลักษณะทางกายภาพของคลื่นแม่เหล็กไฟฟ้าแตกต่างกับสัญญาณไฟฟ้า สัญญาณไฟฟ้าต้องการตัวนำชิ้งเป็นสายนำสัญญาณขณะที่คลื่นแม่เหล็กไฟฟ้าไม่ต้องการตัวนำ แต่มีการแพร่กระจายสัญญาณทิศทางขึ้นไปไฟฟ้านั้นเกิดจากการกระแสไฟฟ้าสับเปลี่ยนตัวนำทำให้เกิดขั้นมากับฟ้าที่มีกำลังสัญญาณเพียงเล็กน้อย แต่คลื่นแม่เหล็กไฟฟ้าบางช่วงนั้นเป็นไม้ได้เกิดตามธรรมชาติทำให้มีถุกรบกวนจากคืนที่อยู่ตามธรรมชาติ จึงถูกใช้เป็นประโยชน์ในการสื่อสาร จากคุณสมบัติแม่เหล็กไฟฟ้ามีความถี่ในช่วงหนึ่งไม่ได้เกิดขึ้นเองโดยธรรมชาติทำให้เกิดสัญญาณใหม่ที่สามารถสื่อสารได้ระยะไกลโดยถุกรบกวนจากสัญญาณโดยธรรมชาติน้อย อย่างไรก็ตามคุณสมบัติคลื่นแม่เหล็กไฟฟ้ามีการลดTHONสัญญาณตามธรรมชาติรวดเร็วกว่าการลดTHONในสายนำสัญญาณ นอกจากเรื่องการลดTHONสัญญาณที่เป็นอุปสรรคต่อการสื่อสารในเครือข่ายไร้สายแล้วยังมีปัญหาลักษณะอื่นที่เกิดขึ้นกับเครือข่ายไร้สาย ได้แก่ โนนดซ่อน(hidden node) และ อีกชื่อโนน(exposed node)

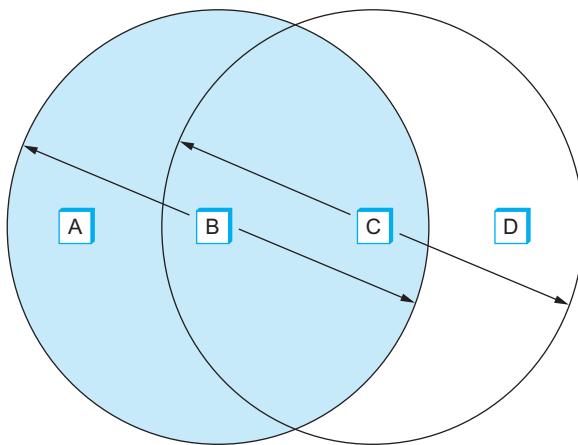
การสื่อสารเครือข่ายไร้สายแบบการเข้าถึงแบบหลายจุดมีวิธีป้องกันการส่งสัญญาณพร้อมกันด้วยการให้ถูกข่ายตรวจสอบการร่วงของช่องสัญญาณก่อนส่ง เมื่อพบว่าช่องสัญญาณว่างจึงส่งข้อมูลได้ ปัญหานอนดซ่อนอีกอย่างได้ในรูปที่ 2.25 ระหว่างการส่งสัญญาณของโนนด A ส่งให้โนนด B และ โนนด B อยู่ตำแหน่งตรงกางสามารถส่งสัญญาณถึงโนนด A และ โนนด C ได้ และ โนนด C ส่งถึงโนนด B เพียงโนนดเดียว ปัญหานอนดซ่อนเกิดขึ้นเมื่อโนนด A ต้องการส่งข้อมูลไปโนนด B เมื่อตรวจสอบช่องสัญญาณพบว่าในเครือข่ายมีโนนด B เครื่องเดียว และ โนนด B ไม่ได้กำลังส่งข้อมูลจึงคิดว่าช่องสัญญาณว่าง ขณะเดียวกันโนนด C ต้องการส่งข้อมูลไปโนนด B และตรวจสอบช่องสัญญาณพบว่าช่องสัญญาณว่างจึงส่งสัญญาณไปโนนด B ซึ่งในเวลาเดียวกัน A ก็

ส่งไป B เช่นเดียวกัน ทำให้เกิดเหตุการณ์ $A \rightarrow B$ และ $B \leftarrow C$ เกิดการรับกวนสัญญาณที่โหนด B โดยที่ A และ C ไม่รู้สาเหตุ เรียกปัญหานี้ว่า “โหนดซ่อน”



รูปที่ 2.25: โหนด A และ โหนด C เกิดปัญหา hidden node ระหว่างกัน
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

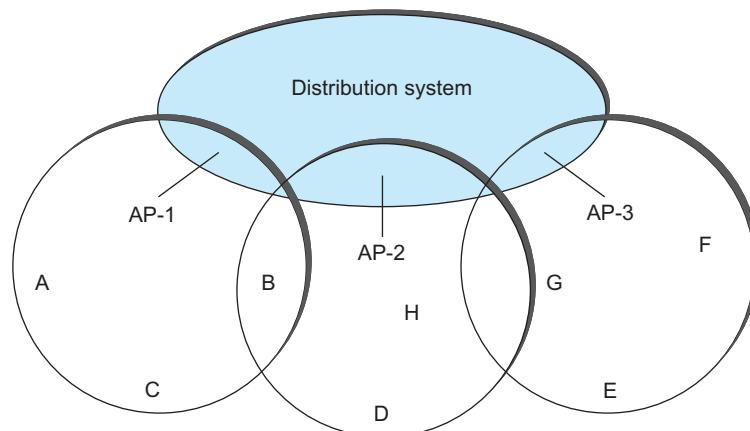
ปัญหามีอยู่อีกกรณีที่เครือข่ายอื่นที่ใช้ช่องสัญญาณเดียวกันได้ เช่นในรูปที่ 2.26 A และ B อยู่ในเครือข่ายเดียวกับใช้ช่องสัญญาณความถี่หนึ่ง โดยที่ B กับ C ไม่ได้อยู่เครือข่ายเดียวกันแต่ใช้ความถี่เดียวกัน C และ D อยู่เครือข่ายเดียวกัน เมื่อ A ต้องการสื่อสารถึง B และ D ต้องการสื่อสารถึง C จะทำให้ A และ C ไม่รู้เลยว่ามีอุปกรณ์อื่นใช้ความถี่เดียวกัน ทำให้เกิดการรบกวนจากอุปกรณ์ที่อยู่ห่างออกไป เรียกปัญหานี้ว่า “เอ็กซ์โพสโหนด”



รูปที่ 2.26: โหนด A และ D ไม่ทราบการมีอยู่ของกันและกัน เกิดปัญหา expose node
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

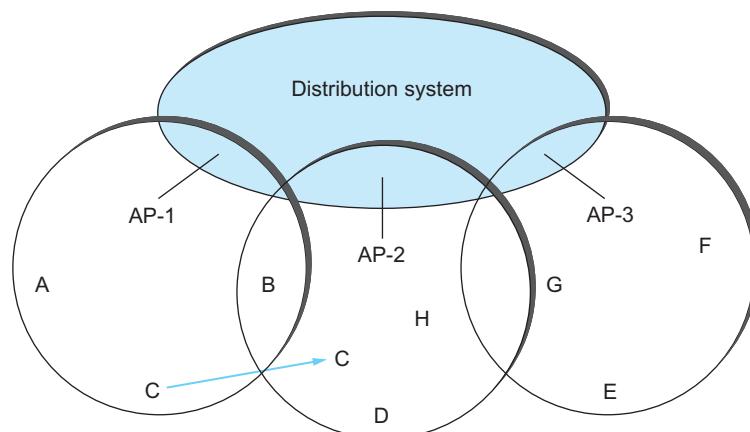
เครือข่ายและไร้สายสามารถเชื่อมต่อกับเครือข่ายแลนด์ได้ผ่าน DS แต่ยังเชื่อมต่อได้แลนไร้สายที่อยู่เครือข่ายอื่นได้ด้วย จากรูปที่ 2.27 มีเครือข่ายไร้สายทั้งหมดสามเครือข่ายได้แก่ AP-1 AP-2 และ AP-3 ทั้งสามเครือข่ายเชื่อมต่อระหว่างกันให้ผ่าน DS

เมื่อโหนดเคลื่อนที่ จะทำให้ไม่ต้องเริ่มกระบวนการเชื่อมต่อเครือข่ายใหม่ แต่เป็นเพียงการแลกเปลี่ยนข้อมูลระหว่างสถานีฐาน สังเกตในรูปที่ 2.28 เมื่อ C ต้องการเปลี่ยนไปเชื่อม AP-2 จะมีการแลกเปลี่ยนข้อมูล



รูปที่ 2.27: กรณีการเชื่อมต่อแลนไวร์ลีย์ตามจุดต่างๆทุกโหนดสามารถสื่อสารกันได้ผ่านทาง DS
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ระหว่าง AP-1 กับ DS ไปยัง AP-2 ซึ่งจะทำให้ C ไม่ได้รับผลกระทบจากการเปลี่ยนแปลงสถานะฐาน รูปที่ 2.29
อธิบายรูปแบบเฟรมมาตรฐานสำหรับการสื่อสารผ่านแลนไวร์ลีย์



รูปที่ 2.28: โหนดมีการเคลื่อนที่เปลี่ยนและขยายตัว
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

โครงสร้างเฟรม มาตรฐาน IEEE 802.11 แตกต่างจากเฟรมมาตรฐานอีเทอร์เน็ต อยู่หลายส่วน ส่วนที่แตกต่างกันชัดเจน เช่น Address เพิ่มขึ้นจากเดิม 2 แออดдрес เป็น 4 แออดdress สาเหตุมาตราฐานได้กำหนดให้แลนไวร์ลีย์ไม่สามารถติดต่อกับคู่ปลายทางโดยตรงแต่ต้องเชื่อมผ่าน base station ซึ่งในที่นี้คือ แออกเซสพอยต์ ซึ่งกรณีนี้ต้องการแออดdress จำนวน 3 แออดdress และอีกกรณีได้แก่การใช้แลนไวร์ลีย์เป็นการเชื่อมต่อแบบ บริดจ์ ซึ่งต้องการคู่สื่อสารจำนวน 2 คู่ ดังนั้นจึงมีจำนวน แออดdress 4 แออดdress



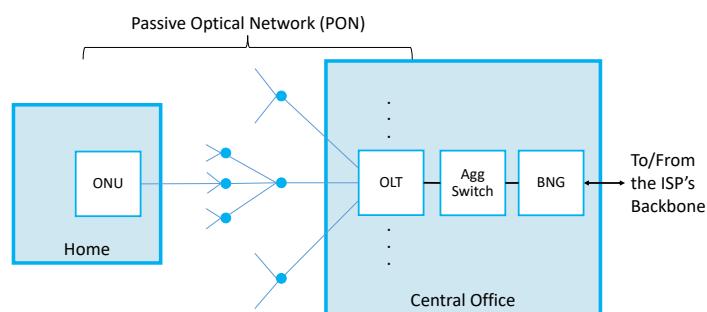
รูปที่ 2.29: เฟรมมาตรฐาน IEEE 802.11
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

2.9 Access Networks

มีการใช้งานเครือข่ายอินเทอร์เน็ตจะเป็นการเชื่อมต่อที่เรียกว่า Access Networks ผู้ใช้งานจะส่งข้อมูลไปถึงผู้ให้บริการเครือข่ายซึ่งอุปกรณ์ที่ให้บริการในส่วน Access Networks ที่คุณเคยได้แก่การเชื่อมต่อผ่านสายไฟบุ้น แก้วนำแสงหรือเรียกว่าPON และการเชื่อมต่อผ่านโครงข่ายเซลลูลาร์(cellular) ซึ่งทั้งสองกรณีอาจจะถูกเข้าใจว่าต่อจากเครือข่ายแลนหรือแลนไวร์ลีย์เพื่อออกสู่อินเทอร์เน็ต

2.9.1 Passive Optical Network

เทคโนโลยี PON เป็นเทคโนโลยีที่ทันสมัยที่สุดในปัจจุบัน เป็นการสื่อสารผ่านสายไฟเบอร์แก้วนำแสงส่งตรงจาก ไอ เอสพี ถึงบ้าน เทคโนโลยี PON ไม่ผ่านอุปกรณ์สวิตช์ใดๆ ไม่มีกระบวนการ store-and-forward ดังนั้นดีเลย์จึง น้อยมาก ดีเลย์น้อยเป็นเรื่องดีสำหรับการ ส่งเสียง ส่งภาพ หรือการประชุมออนไลน์ รูปที่ [2.30](#) ประกอบด้วย ONU(Optical Network Unit) เป็นจุดปลายทาง ผู้ใช้งาน เชื่อมต่อผ่านสายใยแก้วนำแสงไป OLT ซึ่งเชื่อมกับ เครื่อข่ายภายในสำนักงาน สำนักงานกลาง ซึ่ง ONU จะมีไฟส่องสี 1024 จุด เมื่อข้อมูลถูกส่งถึง OLT จะส่งต่อ ให้ Agg Switch ซึ่งทำหน้าที่รวบรวมข้อมูลจาก ONU ทั้งหมดแล้วส่งต่อไป BNG เพื่อเชื่อมเข้าสู่เครือข่ายที่ให้ บริการโดยไอเอสพี



รูปที่ 2.30: ตัวอย่างการเชื่อม PON กับ OLT ภายในสำนักงานก่อนส่งออกทาง BNG
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

2.9.2 Cellular Network

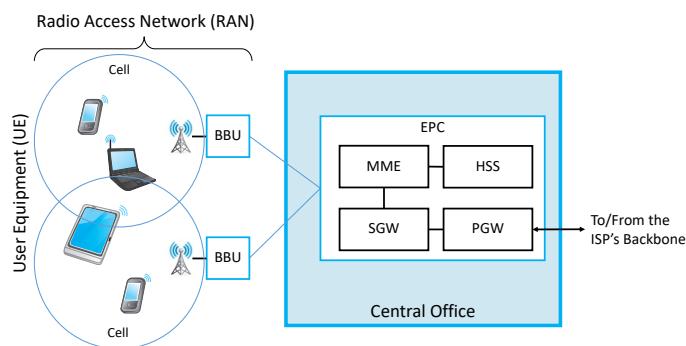
ขณะที่เครือข่ายโทรศัพท์เคลื่อนที่เติบโตอย่างรวดเร็ว ปัจจุบันจำนวนผู้ใช้โทรศัพท์เคลื่อนที่เพิ่มจำนวนมากกว่าคอมพิวเตอร์ไปแล้วนั้น โทรศัพท์เคลื่อนที่สามารถติดต่อเครือข่ายอินเทอร์เน็ตได้ผ่านผู้ให้บริการโทรศัพท์เคลื่อนที่ได้ในทุกที่ที่ต้องการ ซึ่งในปัจจุบันการสื่อสารได้ใช้ชั้นลึกแม่เหล็กไฟฟ้าหลาຍช่วงความถี่ เพื่อให้มีแบบดิจิตร์เพียงพอต่อความต้องการของผู้ใช้งาน สำหรับความถี่ที่ใช้ในโทรศัพท์เคลื่อนที่ที่เหมาะสมนั้นเริ่มตั้งแต่ 700-MHz ไปถึง 2400MHz และปัจจุบันเริ่มขยายช่วงการให้บริการไปใช้ความถี่ 6-GHz และ mmWave

(millimeter wave) ที่ความถี่ 24-GHz รูปที่ 2.31 อธิบายโครงข่ายแบบเซลลูลาร์ซึ่งเป็นเทคโนโลยีเริ่มต้นของโครงข่ายโทรศัพท์เคลื่อนที่

UE(User Equipment) ใช้แทนโทรศัพท์เคลื่อนที่ โดยที่จะเชื่อมผ่านเครือข่ายไร้สายโดยใช้พร็อโทคอลที่กำหนดในมาตรฐาน RAN สัญญาณที่ส่งผ่าน RAN นี้จะเป็นไปตามเทคโนโลยีที่ สถานีฐาน เป็นตัวกำหนด ข้อกำหนดให้ RAN จะกล่าวถึงการใช้งานแม่เหล็กไฟฟ้าเป็นหลัก เมื่อคลื่นแม่เหล็กไฟฟ้าเดินทางถึง สถานีฐาน จะส่งต่อให้ BBU เพื่อส่งข้อมูลผ่านสายใยแก้วนำแสงส่งต่อไปปัจจุบันกากลาง และส่งต่อไป ไอเอสพี

เครือข่ายเซลลูลาร์มีการทำงานคล้ายกับเทคโนโลยีแลนไร้สาย ระบบสื่อสารต้องการสถานีฐานสำหรับกระจายสัญญาณ สถานีฐานของเทคโนโลยีเซลลูลาร์เรียกว่า BBU โดยอุปกรณ์สำหรับผู้ใช้โครงข่ายเรียกว่า UE ระบบบริหารภายในสำนักงานกลางเรียกว่า EPC(Evolved Packet Core) โดยมีเทคโนโลยีสำหรับควบคุมการสื่อสารภาคแม่เหล็กไฟฟ้าเรียกว่า RAN

สำหรับ BBU อาจมีเรียกหลายชื่อ เช่น Evolved NodeB หรือ eNodeB หรือเขียนย่อ eNB ซึ่ง NodeB ทำหน้าที่ควบคุมสัญญาณในส่วนคลื่นแม่เหล็กไฟฟ้า



รูปที่ 2.31: โทรศัพท์เคลื่อนที่ส่งคลื่นวิทยุผ่าน RAN ไป BBU และส่งเข้าสำนักงานกลาง ก่อนส่งออกอินเทอร์เน็ต
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

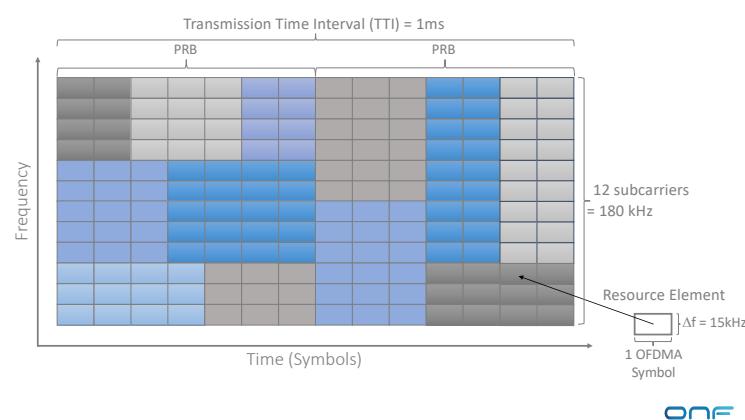
รูปที่ 2.31 อธิบายการสื่อสารแบบ End-to-End เริ่มต้นจาก UE เปรียบได้กับโทรศัพท์มือถือส่งข้อมูลไปยัง BBU และส่งข้อมูลต่อไปยังสำนักงานกลาง กระบวนการที่อยู่ภายใต้สำนักงานกลางจะมีระบบบริหารจัดการสัญญาณข้อมูล เช่น MME(Mobility Management Entity) ทำหน้าที่ควบคุมการย้ายสถานีฐาน SGW (Serving Gateway) ทำหน้าที่เป็นเกตเวย์(gateway) PGW(Packet Data Network Gateway) ควบคุมการทำงานแบบแพ็กเก็ตสิริทซิง และ HSS(Home Subscriber Server) สำหรับควบคุมบัญชีผู้ใช้งาน

เครือข่าย LTE(Long-Term Evolution) พัฒนาต่อยอดจากเครือข่าย UMTS(Universal Mobile Telecommunications System) ซึ่งได้ปรับปรุงด้านความปลอดภัย SAE(System Architecture Evolution) และด้านความเร็วในการสื่อสาร สำหรับการสื่อสาร LTE มีข้อจำกัด เช่น เดียวกับการสื่อสารไร้สายทั่วไปได้แก่ ความถี่เดียวกันไม่สามารถส่งพร้อมกันได้ เมื่อส่งความถี่เดียวกันในเวลาตรงกันทำให้เกิดการรบกวนสัญญาณ หากต้องการส่งพร้อมกันทำได้โดยส่งความถี่อื่น การส่งสัญญาณไร้สายไม่ได้ส่งแบบความถี่เดียวเป็นการแปลงข้อมูลดิจิตอลความเร็วสูงเป็นการส่งแบบขานๆ ที่ความถี่ต่ำหลายความถี่ จึงส่งได้พร้อมกันในหนึ่งเวลา และ

ที่ปลายทางแปลงจากขนาดเป็นอนุกรม ลักษณะนี้คือรูปแบบการแปลงสัญญาณขึ้นว่า OFDM(Orthogonal Frequency-Division Modulation)

การจัดสรรทรัพยากรในภาพกว้างโดยเสนอให้มองทรัพยากรเครือข่ายไร้สายเป็นกล่องสี่เหลี่ยม อธิบายในเฟรม LTE type 1 ชนิด FDD(Frequency Division Duplex) ความถี่ในการส่งข้อมูลเป็น Uplink และ Downlink ใช้ความถี่แยกกันในหนังสือบางเล่มเรียกว่า Unpaired และ TDD(Time Division Duplex) เรียกว่า Paired

การจัดสรรทรัพยากรด้านคลื่นวิทยุของLTEใช้วิธีเข้าถึงช่องสัญญาณแบบ OFDMAโดยเป็นทรัพยากรเป็นพื้นที่เรียกว่า PRB(Physical Resource Block) มีแกน x ใช้แทนช่วงเวลาแกน y ใช้แทนความถี่ sub-carrier ตามรูปที่ 2.32



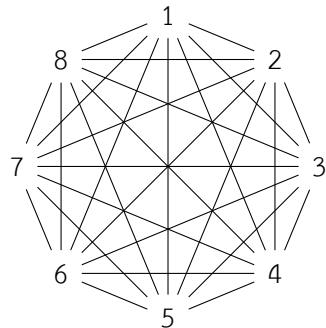
รูปที่ 2.32: วิธีจัดสรรทรัพยากรแบบ OFDMA
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

บทที่ 3

โพร์โทคอลชั้นเครือข่าย

ปัญหา : การเชื่อมต่อโดยตรงเกิดปัญหามีเครือข่ายมีขนาดใหญ่

ในการส่งข้อมูลจากต้นทางถึงปลายทาง วิธีเชื่อมต่อโดยตรงนั้นมีข้อจำกัดหลายประการ เช่น จำนวนช่องทาง สื่อสารที่ต้องการเท่ากับจำนวนอุปกรณ์ที่ต้องการติดต่อ หากใช้ช่องสัญญาณร่วมกันจะทำให้เกิดการรบกวน ระหว่างกัน ตัวอย่างเช่น ไม่สามารถส่งสัญญาณไฟฟ้า ผ่านสายนำสัญญาณด้วยความถี่เดียวกันได้ ทำให้เกิด อุปสรรคต่อการใช้งานสำหรับเครือข่ายที่อยู่ห่างไกล นอกจากนี้มีข้อจำกัดอีกหลายประการเมื่อเป็นการสื่อสารแบบเชื่อมโดยตรง เช่น ข้อจำกัดเส้นเชื่อมของสายสัญญาณอิบิายในรูปที่ 4.12

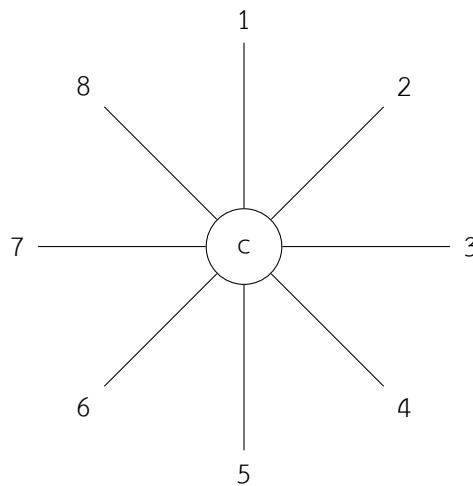


รูปที่ 3.1: กราฟปริบูรณ์มีโนนดจำนวน 8 โนนดต้องการ 28 เส้นเชื่อม
ลิขสิทธิภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

จากรูปที่ 4.12 เป็นกราฟที่มี 8 โนนด เมื่อต้องการให้ทุกโนนดเชื่อมต่อกันได้โดยตรง แต่ละโนนดต้องการเส้นเชื่อมจำนวน $N-1$ เส้น เมื่อมีโนนดทั้งหมด N โนนดทำให้ต้องการเส้นเชื่อม $N(N-1)$ แต่เส้นเชื่อม 1 เส้นเชื่อมให้ได้ 2 โนนด ในที่นี้ต้องการจำนวนเส้นเชื่อมจึงเลือก 1 เส้นจากทั้งหมด 2 เส้น คำนวณโดยนำ 2 หารจำนวนเส้นเชื่อมทั้งหมด ทำให้ได้ $\frac{N(N-1)}{2}$ หรือเขียนรูปพังชันของ N ได้ $f(N) = \frac{N^2-N}{2}$ จะเห็นได้ว่า มีการเติบโตของจำนวนเส้นเชื่อมเป็น $\mathcal{O}(N^2)$ ซึ่งทำให้มีอุปสรรคในการขยายจำนวนโนนด เมื่อมีลูกข่ายจำนวนมากทำให้จำนวนเส้นเชื่อมเพิ่มเป็นพังก์ชันเอกซ์โพเนนเชียล¹

ต่อมาปรับปรุงให้เส้นเชื่อมต่อตรงแบบโครงข่ายสถาาร์ตามภาพที่ 3.2 ทำให้ลดจำนวนเส้นเชื่อมได้ แลก กับการใช้อุปกรณ์กระจายข้อมูลร่วมกัน ถึงแม้ไม่สามารถส่งข้อมูลได้พร้อมกันในหนึ่งเวลา แต่การออกแบบให้มีอุปกรณ์หนึ่งทำหน้าที่กระจายสัญญาณแก่โไฮสต์ในเครือข่ายเสมือนว่าเป็นการเชื่อมต่อโดยตรงช่วยลดจำนวนพอร์ตในโไฮสต์ลงได้ เช่นจากเดิมที่เครือข่ายมี 8 โไฮสต์จะต้องการการ์ดเครือข่ายจำนวน 7 การ์ดเพื่อเชื่อมต่อโดยตรง แต่เมื่อมีอุปกรณ์ที่ทำหน้าที่กระจายสัญญาณจะทำให้โไฮสต์ใช้การ์ดเครือข่ายเพียงใบเดียว โดยใช้การสื่อสารแบบอีเทอร์เน็ตตามภาพที่ 3.3 อุปกรณ์อยู่ตรงกลางจะทำหน้าที่กระจายแพ็กเก็ตที่ได้รับไปยังพอร์ตอื่น

¹พังก์ชันเอกซ์โพเนนเชียล คือพังก์ชันเชิงในรูปเลขยกกำลังตัวอย่างเช่น $f(n) = n^2$



รูปที่ 3.2: กราฟบริบูรณ์มีโหนดจำนวน 8 โหนดต้องการ 8 เส้นเชื่อม
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

จากรูปที่ 3.4 เป็นอุปกรณ์ที่มีพอร์ตจำนวน 8 พอร์ต เมื่อมีหนึ่งพอร์ตได้รับข้อมูลจะกระจายไปที่พอร์ตที่เหลืออีก 7 พอร์ตอุปกรณ์ที่ทำหน้าที่นี้เรียกว่า หับ(hub)

หับทำหน้าที่กระจายข้อมูลเพียงอย่างเดียวทำให้มีการใช้งานเครือข่ายได้ไม่เต็มประสิทธิภาพ ยกตัวอย่างเช่นจากรูปที่ 3.4 เมื่อมีโถสต์ต้องการส่งข้อมูลไปอีกโถสต์ จะทำให้โถสต์อีก 6 เครื่องที่เหลือไม่อาจติดต่อเครือข่ายได้ ซึ่งอุปกรณ์สมัยใหม่สามารถแบ่งช่องสัญญาณจับคู่การสื่อสารได้ ทำให้คุ้สื่อสารสามารถสื่อสารได้พร้อมกัน อุปกรณ์ที่ทำหน้าที่ทดแทน หับมีชื่อว่า เนตเวิร์กสวิตช์(network switch)

3.1 พื้นฐานเนตเวิร์กสวิตช์

เนตเวิร์กสวิตช์ (network switch) หรือเรียกว่า สวิตช์ เป็นอุปกรณ์กระจายสัญญาณสำหรับการสื่อสารอีเทอร์เนต สวิตช์มีความสามารถเลือกช่องสัญญาณในการส่งแพ็กเก็ตไปปลายทางได้ ด้วยวิธีต่างๆ ซึ่งจะกล่าวในหัวข้อต่อไป

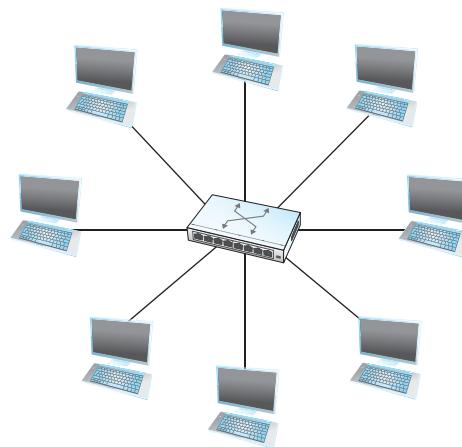
เนตเวิร์กสวิตช์ มีหน้าที่เชื่อมต่อเส้นเชื่อมต้นทางไปปลายทางให้เสร็จเป็นการเชื่อมโดยตรง

การส่งแพ็กเก็ตจากต้นทางไปปลายทางของเนตเวิร์กสวิตช์ไม่ได้กระจายสัญญาณไปทุกพอร์ตซึ่งเป็นวิธีที่ขาดข้างไม่ให้เครื่องอื่นใช้งานได้ วิธีส่งข้อมูลสามารถจัดกลุ่มได้ 3 วิธีดังนี้

1. Datagram (เดต้าแกรม)
2. Virtual circuit (วงจรเสมือน)
3. Source routing (การกำหนดเส้นทางจากต้นทาง)

3.1.1 เดتاแกรม

วิธีกำหนดเส้นทางแบบเดตาแกรม ใช้แนวคิดออกแบบให้เนตเวิร์กสวิตซ์บันทึก แสดงตรวจสอบโไฮสต์ที่เขื่อมอยู่ในแต่ละพอร์ตไว้ในหน่วยความจำในที่นี้เก็บในรูปแบบตารางเรียกว่า พอร์เวิร์ดดิงเทเบิล ยกตัวอย่างเครือข่ายในรูปที่ 3.4 มีสวิตซ์ทั้งหมดสามเครื่องแต่ละเครื่องเชื่อมโไฮสต์ A ถึง H โดยที่โไฮสต์ A C และ D เชื่อมกับ switch-1 สำหรับ โไฮสต์ E และ F เชื่อม switch-2 และโไฮสต์ B G และ H เชื่อม switch-3 สังเกตได้ว่าสวิตซ์ทั้งสามเครื่อง เชื่อมต่อระหว่างกัน ดังนี้ switch-1 พอร์ต 1 เชื่อมกับ switch-2 พอร์ต 3 และ switch-2 พอร์ต 0 เชื่อมกับ switch-3 พอร์ต 0



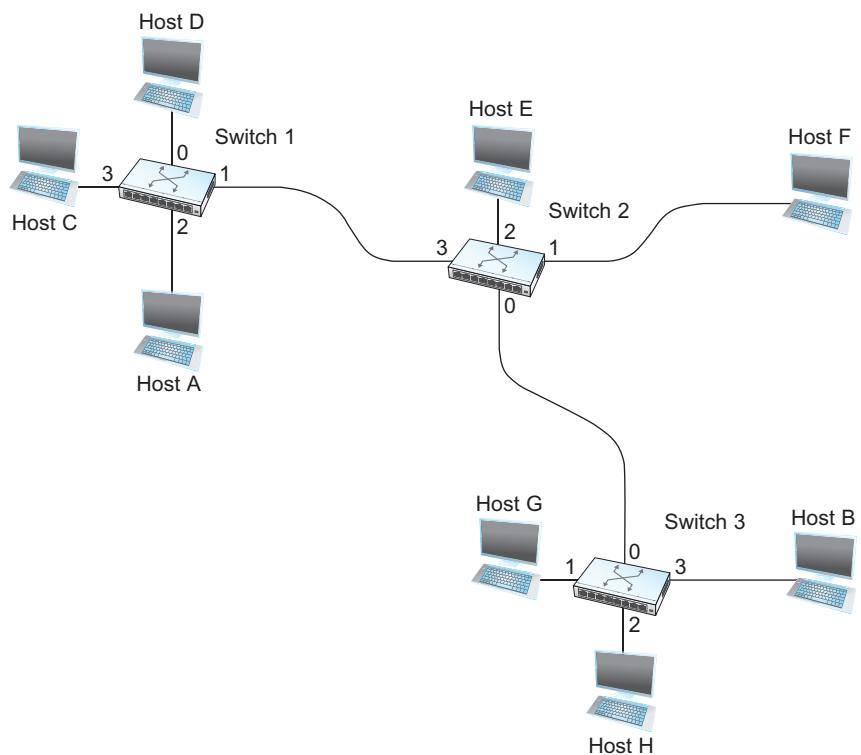
รูปที่ 3.3: การเชื่อมสวิตซ์แบบโครงข่ายสถาาร์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

สวิตซ์แต่ละเครื่องจะมีพอร์เวิร์ดดิงเทเบิลแตกต่างกันตามชื่อโไฮสต์และตำแหน่งพอร์ตที่โไฮสต์นั้นเชื่อม ต่อ ตัวอย่างพอร์เวิร์ดดิงเทเบิล ของ switch-1 และ switch-2 ตั้งตารางที่ 3.1 และ 3.2 ตามลำดับ

ตารางที่ 3.1: ตารางพอร์เวิร์ดของเนตเวิร์กสวิตซ์ 1

ปลายทาง	พอร์ต
A	2
B	1
C	3
D	0
E	1
F	1
G	1
H	1

ข้อมูลในตารางประกอบด้วยชื่อสำหรับระบุโไฮสต์ปลายทางและหมายเลขพอร์ตที่โไฮสต์ปลายทางกำลังเชื่อมต่อ ยกตัวอย่างเช่น เมื่อข้อมูลแบบเดตาแกรม จากโไฮสต์ A มีปลายทางเป็นโไฮสต์ B ข้อมูลเริ่มต้นที่ switch-1 ได้รับข้อมูลเดินทางเข้าพอร์ต 2 ข้อมูลที่เดินทางเข้าพอร์ตเรียกว่า เดินทางเข้า(incoming) เมื่อแพ็ก



รูปที่ 3.4: ตัวอย่างรูปแบบการเชื่อมต่อโดยใช้เนตเวิร์กสวิตช์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

เก็ตเดินทางถึงสวิตช์ จะเริ่มขั้นตอนคัดเลือกเส้นทางโดยอ่านข้อมูลจากส่วนเขตเดอร์ ซึ่งได้ค่าชื่อไฮสต์ปลายทาง นำค่าไฮสต์ปลายทางตรวจสอบกับฟอร์วีร์ดดิงเทเบิล จนได้ค่าพอร์ตที่จะส่งต่อ จากตารางที่ 3.1 เห็นได้ว่าหมายเลขพอร์ตที่จะส่งข้อมูลไปถึงไฮสต์ B คือหมายเลข 1 ข้อมูลจึงส่งออก(เดินทางออก(outgoing)) ไปที่พอร์ต 1 เมื่อข้อมูลเดินทางออก พอร์ต 1 จะกลายเป็น เดินทางเข้า พอร์ต 3 ของ switch-2 เมื่อ switch-2 ได้รับข้อมูลจะเริ่มอ่านเขตเดอร์ เพื่อดูชื่อไฮสต์ปลายทาง โดยที่อ่านได้ค่า B นำชื่อไฮสต์เทียบกับตารางที่ 3.2 พบว่าไฮสต์ B เชื่อมที่พอร์ต 0 ข้อมูลจึงเดินทางออก ไปที่พอร์ต 0 และเป็น เดินทางเข้า ของ switch-3 ที่พอร์ต 0 เมื่อ switch-3 ได้รับข้อมูลจึงทำขั้นตอนตรวจสอบ ฟอร์วีร์ดดิงเทเบิล แล้วจึงพบว่าส่งข้อมูลไปที่พอร์ต 3 ข้อมูลจึงเดินทางถึงไฮสต์ B เป็นอันสิ้นสุดการส่งข้อมูล

เรียกวิธีทำงานนี้ว่า “store and forward” คำว่า store หมายถึงทำหน้าที่เก็บข้อมูลลงหน่วยความจำแล้วประมวลผลเส้นทางหลังจากนั้นจึงส่งต่อ การส่งต่อเรียกว่า “forward” สำหรับคุณสมบัติการส่งข้อมูลแบบเด塔แกรม เป็นการส่งผ่านอุปกรณ์สวิตช์ที่มีคุณสมบัติเป็น store and forward มีดังต่อไปนี้

- ไฮสต์สามารถส่งข้อมูลได้จากทุกที่ทุกเวลา เมื่อสวิตช์ได้รับข้อมูลจะประมวลผลแล้วส่งต่ออย่างรวดเร็ว วิธีนี้ไม่ได้มีการกำหนดเส้นทางก่อนที่แพ็กเกจออกเดินทางจึงเรียกว่าเป็นวิธีการเชื่อมต่อแบบ ไม่กำหนดการเชื่อมต่อ(connectionless) ซึ่งแตกต่างจากวิธีการจัดเตรียมเส้นทางให้เรียบร้อยก่อนที่แพ็กเกจจะออกเดินทางเรียกว่า วิธีส่งแบบกำหนดการเชื่อมต่อ(connection-oriented)

ตารางที่ 3.2: ตารางฟอร์เวิร์ดของเนตเวิร์กสวิตช์ 2

ปลายทาง	พอร์ท
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0

- เมื่อแพ็กเก็ตออกเดินทางไปแล้วไม่มีทางทราบทะลุจจะเดินทางถึงปลายทางหรือไม่ หรือไม่ทราบแม้กระทั้งปลายทางเปิดอยู่หรือปิดไปแล้ว
- แต่ละแพ็กเก็ตที่ส่งนั้น ไม่ขึ้นต่อแพ็กเก็ตที่ถูกส่งไปก่อนหน้า ถึงแม้จะส่งไปปลายทางเดียวกัน ซึ่งยังคงต้องหาเส้นทางจากฟอร์เวิร์ดติงเทเบิลทุกรั้ง
- ในเหตุการณ์มีสวิตช์หรือลิงก์ขนาดไปจะไม่ส่งผลกระทบต่อเครือข่ายมากนักหากกระบวนการปรับปรุงฟอร์เวิร์ดติงเทเบิลยังค้นหาเส้นทางอื่นได้

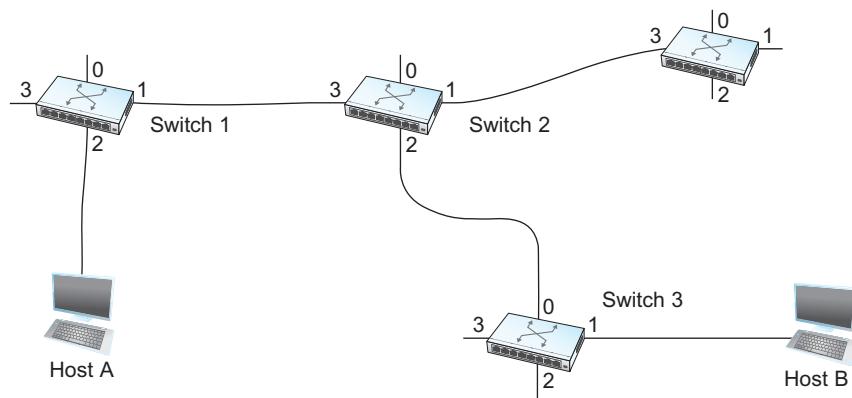
3.1.2 สวิตช์วงจรเสมือน

หัวข้อนี้กล่าวถึงเทคนิคการส่งแบบ Virtual circuit (วงจรเสมือน) ใช้อักษรย่อ VC เป็นเทคนิคที่สองต่อจากเดาแกรม วิธีส่งข้อมูลแบบนี้ต้องการการทำงาน 2 ขั้นตอน ขั้นตอนแรกกำหนดเส้นทาง ขั้นตอนที่สองจึงเริ่มส่งข้อมูล

ขั้นตอนกำหนดเส้นทาง เป็นการทำวงจรเสมือนไว้ล่วงหน้าก่อนจะส่งแพ็กเก็ต ซึ่งเป็นแนวคิดแบบกำหนดการเชื่อมต่อ ในการกำหนดเส้นทางนี้จะเป็นเส้นทางสำหรับหนึ่งคู่สื่อสาร หากมีหลายคู่สื่อสารจะมีจำนวนเส้นทางเท่ากับจำนวนการสื่อสารนั้น

ยกตัวอย่างเช่นการสื่อสารในรูปที่ 3.5 เมื่อต้องการส่งข้อมูลจากโไฮสต์ A ไปโไฮสต์ B จะมีเส้นทางจำนวน 1 เส้นทาง จาก A ไป B บันทึกในตารางเส้นทาง ตัวอย่าง switch-1 อธิบายในตารางที่ 3.1 กระบวนการในการเตรียมข้อมูลสำหรับป้อนลงตารางเรียกว่า “เตรียมการเชื่อมต่อ(connection setup)”

ขั้นตอนเตรียมการเชื่อมต่อ เป็นขั้นตอนสร้างเส้นทางจากต้นทางถึงปลายทาง มีกระบวนการสร้างสถานะของเส้นทางระหว่างสวิตช์แต่ละตัวเรียกว่า “สถานะเชื่อมต่อ(connection state)” เป็นการกำหนดเส้นทางจากต้นทางไปปลายทางโดยบันทึกไว้ในตารางวงจรเสมือน (VC table) สำหรับตารางวงจรเสมือน ประกอบด้วยข้อมูลต่อไปนี้



รูปที่ 3.5: วงจรเสมือนของการเชื่อม โไฮสต์ A กับ โไฮสต์ B
 ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

- ข้อมูล ตัวระบุวงจรเสมือน ใช้อักษรย่อ VCI สำหรับใช้กำหนดเส้นทางของแต่ละการเชื่อมต่อ เพื่อให้ระบุเส้นทางตลอดแนว นับจากต้นทางถึงปลายทาง
- เมื่อมีแพ็กเก็ตเดินทางเข้าอินเทอร์เฟซ(interface) จะกำหนดเป็น incoming VCI กำหนดแทนด้วยหมายเลขไม่ซ้ำกับ outgoing VCI
- เมื่อมีแพ็กเก็ตเดินทางออกอินเทอร์เฟซ กำหนดเป็น outgoing VCI กำหนดแทนด้วยหมายเลขไม่ซ้ำกับ incoming VCI

การกำหนดหมายเลข VCI เป็นไปได้ 2 แนวทางได้แก่ กำหนดโดยผู้ดูแลเครือข่าย ซึ่งค่า VCI นี้จะมีค่าไม่เปลี่ยนแปลง(permanent) จากการกำหนดค่าของผู้ดูแลระบบเรียกว่า วงจรเสมือนคงわり(permanent virtual circuit) และอีกแนวทางหนึ่งได้แก่การเปลี่ยนแปลงแบบใหมานิว(dynamic) การเปลี่ยนค่า VCI แบบนี้ทำให้มีการปรับปรุงค่าได้โดยไม่ต้องเป็นการเปลี่ยนแปลงโดยผู้ดูแลระบบ เรียกว่า วงจรเสมือนสวิตช์(switched virtual circuit)

ยกตัวอย่างการกำหนดค่า VCI โดยผู้ดูแลระบบ เมื่อต้องการสร้างการเชื่อมต่อจาก A ไป B อันดับแรกผู้ดูแลระบบต้องการกำหนดค่าตัวเลข VCI สำหรับใช้ระบุเส้นทางจากโไฮสต์ A ไป switch-1 ในที่นี้ผู้ดูแลระบบกำหนดให้เป็นหมายเลข 5 (incoming VCI = 5) ต่อมากู้ดูแลระบบกำหนดหมายเลขการเชื่อมต่อระหว่าง switch-1 กับ switch-2 โดยกำหนดให้มีค่าเท่ากับ 1 (outgoing VCI = 11) ข้อมูลในตาราง VC table จึงอธิบายได้ในตารางที่ 3.3

ตารางที่ 3.3: ตัวระบุวงจรเสมือน สวิตช์-1

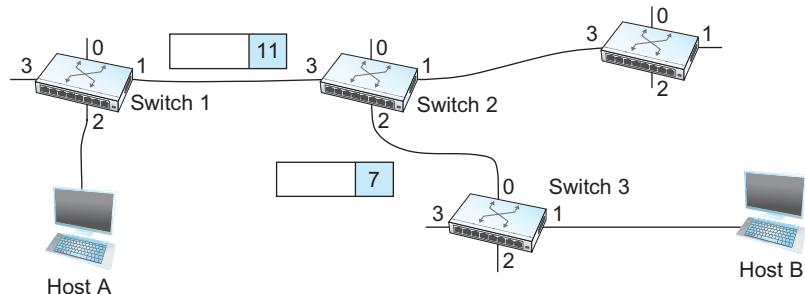
Incoming interface	incoming VCI	Interface outgoing	Outgoing VCI
2	5	1	11

หมายเลข outgoing VCI จะกล้ายเป็น incoming VCI ของสวิตช์ สำหรับ switch-2 เมื่อกำหนดให้ outgoing จาก switch-1 เท่ากับ 11 จึงทำให้ incoming VCI ของ switch-2 ที่อินเทอร์เฟซ 3 มีค่าเป็น 11

ต่อมาการเชื่อมจาก switch-2 ไป switch-3 กำหนดให้มีค่า VCI เท่ากับ 7 ดังนั้น outgoing VCI เท่ากับ 7 และการเชื่อมต่อระหว่าง switch-3 ไป B กำหนดให้ VCI มีค่า 4 ตามลำดับ เป็นตามตารางที่ 3.4 และ 3.5

เมื่อกำหนดค่า VCI จนเสร็จสิ้นแล้ว ขั้นตอนต่อไปเป็นการเริ่มต้นส่งข้อมูล อธิบายโดยใช้ภาพที่ 3.7 ในกรณี A ต้องการส่งข้อมูลถึง B อันดับแรก A จะส่งข้อมูลเข้าพอร์ต 2 ของ switch-1 และผู้ดูแลระบบได้กำหนดให้ไฮสต์ A บรรจุค่า VCI ลงในเอดเดอร์ มีค่าเท่ากับ 5 แล้วส่งไป switch-1 เมื่อ switch-1 ได้รับจากอินเทอร์เฟช 2 ซึ่งมี incoming interface เท่ากับ 2 จากตารางที่ 3.3 ได้กำหนดให้ incoming VCI ที่มี incoming interface = 2 จะส่งออก interface 1 ด้วยเลข VCI = 11 ดังนั้นข้อมูลจะส่งต่อไปยังอินเทอร์เฟช 1 และเขตเอดเดอร์ ให้มีค่า VCI = 11

ต่อมา switch-2 ได้รับข้อมูลที่อินเทอร์เฟช 3 และมีค่า VCI=11 จึงเทียบกับตารางที่ 3.4 และส่งแพ็กเก็ตไป switch-3 พร้อมกับกำหนดค่า VCI=7 อธิบายในรูปที่ 3.6 เมื่อ switch-3 ได้รับแพ็กเก็ตจาก switch-2 จะส่งข้อมูลไปไฮสต์ B โดยกำหนดค่า VCI=4 ทำให้แพ็กเก็ตเดินทางถึงไฮสต์ B สำเร็จ



รูปที่ 3.6: ตัวอย่างแพ็กเก็ตกำหนดเส้นทางด้วยวิธีจาระเมื่อ
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

เห็นได้ว่าเส้นทางข้อมูลจากไฮสต์ A สามารถเดินทางถึงไฮสต์ B มีการกำหนดค่า VCI เริ่มต้นจากไฮสต์ A และเส้นทางตลอดเครือข่ายโดยผู้ดูแลระบบ การกำหนดค่าโดยผู้ดูแลระบบจะเป็นประโยชน์เมื่อไม่มีการเปลี่ยนแปลงเส้นทางบ่อยนัก ซึ่งเครือข่ายระยะไกลและไม่ต้องการเปลี่ยนเส้นทางบ่อย เช่นเส้นทางเครือข่ายข้ามมหาสมุทรเป็นต้น หากปล่อยให้มีการเปลี่ยนแปลง VCI ตามหมายเลขแพ็กเก็ตจะทำให้ตัวเลข VCI จะมีจำนวนมากและเปลี่ยนแปลงจนอาจควบคุมไม่ได้ ด้วยสาเหตุนี้การใช้งานจริงจะพบการใช้งาน PVC เป็นส่วนใหญ่ และมีกำหนดเป็น SVC เฉพาะบางไฮสต์

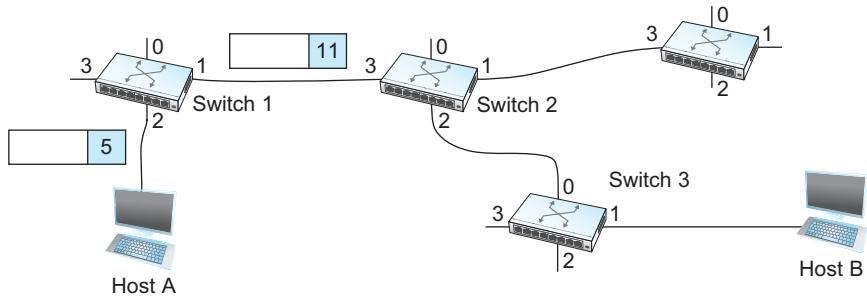
ตารางที่ 3.4: ตัวรับบุวงจรสเมื่อ สวิตช์-2

Incoming interface	incoming VCI	Interface outgoing	Outgoing VCI
3	11	2	7

ต่อมา ในการนี้การสร้าง connection state ด้วย วงจรเสมือนสวิตช์ เป็นการใช้สัญญาณ โดยสวิตช์เป็นอุปกรณ์กำหนดหมายเลข VCI ซึ่งจะต้องมีกลไกให้ไฮสต์ทราบค่าตัวเลขที่สวิตช์ได้เลือก กลไกที่สวิตช์ใช้บอกไฮสต์ได้แก่การส่งข้อมูลตอบกลับเป็นการส่งแพ็กเก็ตกลับไปยังไฮสต์ เรียกว่า แอ็คโนเผล็จเมนท์ เมื่อไฮสต์ได้รับแอ็คโนเผล็จเมนท์จะกำหนดเขตเดอร์มีค่า VCI ตรงตามที่ได้รับจาก แอ็คโนเผล็จเมนท์

ตารางที่ 3.5: ตัวระบุวงจรเสมือน สวิตช์-3

Incoming interface	incoming VCI	Interface outgoing	Outgoing VCI
0	7	1	4



รูปที่ 3.7: แพ็คเก็ตส่งผ่านวงจรเสมือน
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

อธิบายขั้นตอนการทำงานของ วงจรเสมือนสวิตช์ ใช้ตัวอย่างจากรูปที่ 3.6 ไฮสต์ A ต้องการส่งข้อมูลไป ไฮสต์ B ขั้นตอนแรกเป็นการทำขั้นตอนเตรียมการเชื่อมต่อ โดยส่งแพ็คเก็ตไปสอบถาม switch-1 เมื่อ switch-1 ได้รับแพ็คเก็ต ประเภท เตรียมการเชื่อมต่อ จะตรวจสอบว่ามีตัวเลข VCI ได้ว่างอยู่หลังจากพบทัวเลขว่างเป็นเลข 5 จึงกำหนด incoming VCI = 5 พร้อมกับส่งแพ็คเก็ตแอ็คโนเล็จเมนท์กลับไปยังไฮสต์ A ซึ่งทำให้ไฮสต์ A ทราบว่าต้องเช็คเดอร์ ด้วยค่า VCI ใด ลำดับต่อมา switch-1 พอร์ต 1 เชื่อมกับ switch-2 พอร์ต 3 โดย switch-2 จะเริ่มต้นหา VCI ที่ยังว่าง ในที่นี้เท่ากับ 11 จึงกำหนด outgoing VCI = 11 พร้อมกับตอบแอ็คโนเล็จเมนท์กลับไปที่ switch-1 ลำดับต่อมา switch-2 พอร์ต 2 เชื่อมกับ switch-3 พอร์ต 0 ทำขั้นตอนเตรียมการเชื่อมต่อ โดย switch-3 ค้นหา VCI ที่ยังว่าง แล้วแล้วเช็ค incoming VCI เท่ากับ 7 และตอบกลับด้วยแพ็คเก็ตแอ็คโนเล็จเมนท์ ไปยังไฮสต์ B เพื่อขอให้ไฮสต์ B บอกหมายเลข VCI สำหรับใช้เชื่อมต่อ เมื่อไฮสต์ B ได้รับจะตรวจสอบ VCI ที่ยังว่างแล้วตอบกลับด้วยแอ็คโนเล็จเมนท์กลับไปยัง switch-3 ในที่นี้ไฮสต์ B เลือก incoming VCI = 4 จะเห็นได้ว่า incoming VCI ได้กำหนดค่าเสร็จสิ้น สำหรับข้อมูล outgoing VCI ได้กำหนดค่าให้เป็นไปตาม incoming VCI ตัวอย่างเช่น switch-3 ได้รับ incoming VCI=4 จากไฮสต์ B จะทำให้เซตค่า outgoing VCI ที่เชื่อมกับไฮสต์ B เป็น 4 ตามค่าแอ็คโนเล็จเมนท์ที่ได้รับจากไฮสต์ B เป็นต้น ในกรณีที่ไฮสต์ A ไม่มีข้อมูลส่งไปยังไฮสต์ B จะส่งข้อมูลไปบอก switch-1 ให้ลบข้อมูล VCI ออก เมื่อ switch-1 ลบข้อมูล VCI ไปแล้วในการส่งข้อมูลที่มี VCI=5 ก็จะไม่สามารถส่งต่อข้อมูลได้

ในการส่งข้อมูลแบบวงจรเสมือนมีสิ่งที่ควรคำนึงดังนี้

- จากที่ไฮสต์ A จะต้องรอนานกว่าจะทำเตรียมการเชื่อมต่อเสร็จสิ้น ซึ่งเป็นระยะเวลาที่แพ็คเก็ตเดินทางไปและกลับ (round-trip time : RTT) ทำให้ระยะเวลาที่ไฮสต์ A ส่งข้อมูลจะใช้เวลาไม่น้อยไปกว่าระยะเวลา RTT

- แพ็คเก็ตสำหรับทำเตรียมการเชื่อมต่อต้องบรรจุข้อมูลของไฮสต์ปลายทาง ซึ่งอาจมีขนาดใหญ่กว่าข้อมูลที่ต้องการส่งจริงๆ
- เมื่อสวิตช์หรือลิงก์เกิดเสียหรือขาดทำให้ต้องมีการค้นหาเส้นทางใหม่ ข้อมูลในเส้นทางเก่าจะถูกลบเพื่อสำรองพื้นที่ให้เส้นทางใหม่
- ประดิษฐ์การเลือกลิงก์สำหรับส่งต่อให้ได้ผลลัพธ์ทางเครือข่ายที่เหมาะสม จะใช้กระบวนการการทำงาน ‘routing algorithm’ เพื่อตัดสินใจเลือกลิงก์

ข้อดีหนึ่งของวงจรเสมือนได้แก่ การเตรียมเส้นทางและทราบความพร้อมของปลายทางก่อนส่งข้อมูล ทำให้แต่ละจุดที่ข้อมูลจะเดินทางผ่านได้เตรียมทรัพยากรไว้รองรับได้ เช่น ขนาดหน่วยความจำที่เหมาะสมในการส่งข้อมูล ยกตัวอย่างโปรโตคอล X.25² ประกอบด้วยการทำงานสามส่วนดังนี้:

1. จัดหน่วยความจำสำหรับวงจรเสมือน
2. ใช้วิธีปรับขนาดการแพ็คเก็ตตามขนาดที่เหมาะสม เรียกว่าวิธี ลайдดิng วินโดว์(sliding window)
3. วงจรเสมือนปฏิเสธแพ็คเก็ตเมื่อมีสำรองบัฟเฟอร์สำหรับข้อมูลไม่เพียงพอ

การจัดเตรียมบัฟเฟอร์นี้เพื่อทำให้ยืนยันความพร้อมของไฮสต์ แนวคิดการปรับบัฟเฟอร์เรียกว่า ‘hop-by-hop flow control’

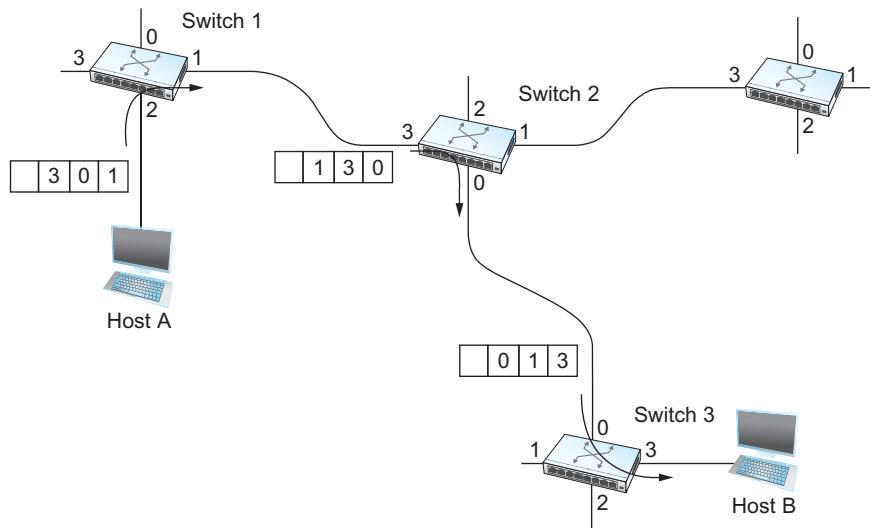
เปรียบเทียบระหว่างเด塔แกรมและวงจรเสมือน การสื่อสารแบบเด塔แกรมไม่ต้องการสร้างเส้นทาง ก่อนส่งข้อมูล ขณะที่วงจรเสมือนเสียเวลา กับการสร้างเส้นทางแต่ก็แลกกับความเชื่อถือได้ว่าข้อมูลจะส่งถึงปลายทาง

3.1.3 ต้นทางกำหนดเส้นทาง

การส่งข้อมูลแบบที่สาม วิธีต้นทางกำหนดเส้นทางเป็นอีกหนึ่งวิธีที่มีความแตกต่างจากทั้งสองวิธีแรก โดยให้ต้นทางเป็นผู้กำหนดเส้นทางข้อมูลได้ มีหลายแนวทางที่เป็นได้ในการเลือกเส้นทางโดยใช้ต้นทางกำหนดเส้นทาง วิธีหนึ่งคือกำหนดหมายเลขพอร์ตทางออกไว้ในเขตเดอร์ เมื่อสวิตช์อ่านเขตเดอร์พบหมายเลขพอร์ตที่เป็นทางออกส่งไปหมายเลขพอร์ตนั้น ดังตัวอย่างรูปที่ 3.8

จากตัวอย่าง แพ็คเก็ตจะเดินทางผ่านสวิตช์ทั้งหมดสามเครื่องเพื่อส่งข้อมูลจากไฮสต์ A ไปไฮสต์ B ที่ switch-1 มีหมายเลขพอร์ตส่งออกเป็นเลข 1 และสวิตช์ตัวต่อมาหมายเลขพอร์ตส่งออกเป็นเลข 0 และสวิตช์ตัวต่อมาหมายเลขพอร์ตส่งออกเป็นเลข 3 ดังนั้นข้อมูลหมายเลขพอร์ตที่จะเซตในเขตเดอร์จะมี พอร์ต(3,0,1) โดยสมมติให้สวิตช์อ่านหมายเลขพอร์ตจากข้อมูลความมือสุด ต่อมาก็จะเดินทางออกจาก switch-1 เข้า switch-2 ต้องการกำหนดหมายเลขพอร์ตของแพ็คเก็ต ใหม่โดยการหมุนข้อมูลไปทางขวา เป็น พอร์ต(1,3,0) ทำให้switch-2 ได้อ่านเขตเดอร์ พบรหัสหมายเลขพอร์ต 0 และจึงส่งต่อ ลำดับต่อมาเป็นการส่งแพ็คเก็ตไป switch-3 จึงเริ่มเปลี่ยนข้อมูลในส่วนเขตเดอร์ใหม่ด้วยวิธีหมุนอีกครั้ง จึงให้ข้อมูลเป็น พอร์ต(0,1,3) ทำให้ข้อมูลเดินทางถึงไฮสต์ B

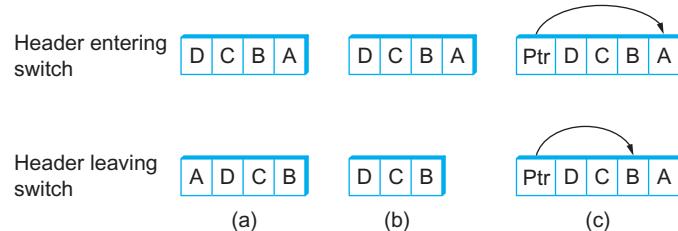
²วงจรเสมือนไม่มีการใช้งานแล้วปัจจุบัน



รูปที่ 3.8: การกำหนดเส้นทางแบบต้นทางกำหนดเส้นทาง(สวิตซ์อ่านเลขขวาสุดเสมอ)
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

มีประเด็นน่าสนใจสำคัญสำหรับวิธีให้ต้นทางเป็นผู้กำหนดเส้นทางข้อมูลหลายประการ ประการแรก วิธีนี้ สมมติให้ต้นทางเป็นเครื่องกำหนดเส้นทาง ซึ่งหมายถึง ต้นทางจะต้องทราบการเชื่อมเครือข่ายของอุปกรณ์ล่วงหน้า ทำให้ขัดแย้งกับวิธีค้นหาเส้นทาง

ประเด็นที่สอง เมื่อสวิตซ์เพิ่มจำนวนมากขึ้น จะทำให้ขนาดเฟดเดอร์โตขึ้นตามจำนวนสวิตซ์ในการนำไปใช้จริงอาจต้องจำกัดจำนวนสวิตซ์



รูปที่ 3.9: วิธีต้นทางกำหนดเส้นทางทั้งสามแบบ
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ประเด็นที่สาม การหมุนข้อมูล เหตุเดอร์สามารถปรับปรุงแนวทางได้ เช่น อธิบายในภาพที่ 3.9 รูปที่ 3.9(a) เป็นวิธีหมุนแบบปกติเมื่ออ่านข้อมูลเร็จแล้วหมุนข้อมูลต่อไปทางขวาสุด วิธีต่อมาใช้วิธีลบเมื่อส่งข้อมูลเร็จแล้วตามอธิบายในรูปที่ 3.9(b) และวิธีสุดท้ายรูปที่ 3.9(c) ใช้พอยน์เตอร์(pointer)ซึ่งข้อมูลแทนการหมุนข้อมูล

3.2 อีเทอร์เน็ตสวิตช์

จากหัวข้อที่แล้วได้กล่าวถึงรายละเอียดของภาพรวมเทคโนโลยีสวิตช์ หัวข้อนี้จะกล่าวถึงอุปกรณ์สวิตช์ที่เฉพาะเจาะจงมากขึ้น อุปกรณ์เครือข่ายทำหน้าที่กระจายสัญญาณในรูปแบบมาตรฐาน IEEE 802.3 หรือเรียกว่า อี-เทอร์เน็ตสวิตช์ นั้นเรียกว่า “สวิตช์”

อีเทอร์เน็ตสวิตช์ เป็นอุปกรณ์สวิตช์ที่มีความสามารถสำนักงาน ทำหน้าที่กระจายข้อมูลไปยังโไฮสต์ที่สื่อสารด้วยเทคโนโลยีเครือข่ายแลน อีเทอร์เน็ตสวิตช์ทำหน้าที่เชื่อมต่อระหว่างเครือข่ายคล้ายเป็นสะพานเชื่อมจึงถูกเรียกอีกชื่อว่า บริดจ์ ทำหน้าที่เชื่อมต่อระหว่างเครือข่ายแลนหลายเครือข่ายให้สามารถสื่อสารกันได้ ปัจจุบันการสื่อสารแบบอีเทอร์เน็ตเป็นการเชื่อมต่อแบบ point-to-point ซึ่งถือว่าเป็นเชื่อมต่อโดยตรงระหว่างอุปกรณ์ผ่านการทำางานของสวิตช์ เรียกอุปกรณ์สวิตช์ทำงานนี้ว่า “L2 switch”

3.2.1 Learning Bridges

อุปกรณ์ประเภทบริดจ์ทำหน้าที่เชื่อมต่อระหว่างสองเครือข่าย ที่มีประสิทธิภาพกว่าการกระจายข้อมูลทั้งหมด การทำงานของบริดจ์ไม่ได้กระจายทุกข้อมูลที่ได้รับออกสู่เครือข่าย อธิบายได้ในรูปที่ 3.10 เมื่อโไฮสต์ A ต้องการส่งข้อมูลไปโไฮสต์ B ข้อมูลจะกระจายผ่านเครือข่ายและถึงพอร์ต 1 อุปกรณ์บริดจ์จะรู้ว่าไม่ต้องส่งข้อมูลไปพอร์ต 2 เพราะโไฮสต์ B อยู่ในเครือข่ายเดียวกัน หากเป็นโไฮสต์ที่อยู่นอกเครือข่าย เช่น X บริดจ์จะทำหน้าที่เป็นสะพานเชื่อมข้อมูล ซึ่งการการส่งต่อของบริดจ์มีดังนี้

จำลองการทำงานของบริดจ์ได้ในตารางที่ 3.6 เมื่อบริดจ์ได้รับเฟรมที่พอร์ตจะอ่านแอดдресเครื่องปลายทางเทียบกับตาราง และส่งข้อมูลตามหมายเลขพอร์ต จากตัวอย่างเมื่อโไฮสต์ A ส่งข้อมูลถึงโไฮสต์ B ทำหน้าแอดдресเครื่องปลายทางเป็น B ทำให้บริดจ์กระจายข้อมูลในพอร์ต 1 และไม่ส่งแพ็กเก็ตไปพอร์ต 2

ตารางที่ 3.6: พอร์ตเวิร์ดดิ้งเทเบิล ภายใต้บริดจ์

โไฮสต์	พอร์ต
A	1
B	1
C	1
X	2
Y	2
Z	2

การเติมข้อมูลลงตารางพอร์ตเวิร์ดดิ้งเทเบิลอาจเติมโดย ผู้ดูแลระบบ(administrators) หรืออุปกรณ์ให้บริดจ์เรียนรู้จากแพ็กเก็ตที่ได้รับด้วยตนเอง แนวคิดการให้บริดจ์เรียนรู้จากแพ็กเก็ตทำได้โดย อ่านข้อมูลใน帧เดอร์ที่ประกอบด้วยหมายแอดdressโไฮสต์ต้นทาง และแอดdressโไฮสต์ปลายทาง ตัวอย่าง เช่น เมื่อโไฮสต์ A ส่งข้อมูลออกสู่เครือข่ายจะทำให้ บริดจ์อ่านเดอร์ได้ว่ามีแอดdressต้นทางโไฮสต์ A ปรากฏที่พอร์ต 1 เมื่อมีเครื่องอื่นใดที่ต้องการส่งถึงโไฮสต์ A จะทำให้บริดจ์ทราบว่าต้องส่งไปที่พอร์ต 1

3.2.2 Implementation

โค้ดต่อไปนี้ใช้อัลกอริทึมของบริดจ์ ทำความเข้าใจได้ง่าย โครงสร้างข้อมูลเรียกว่า BridgeEntry ใช้บันทึกตาราง bridge's forwarding เมื่อแพ็กเก็ต เดินทางถึงสวิตช์ จะตรวจสอบโดยใช้ MacAddr จากปลายทาง และอ่านค่า MAX_TTL ในการตรวจสอบว่า MacAddr นั้นมีอายุหรือยัง

```
#define BRIDGE_TAB_SIZE 1024 /* max size of bridging table */
#define MAX_TTL 120 /* time (in seconds) before an entry is
flushed */

typedef struct {
    MacAddr destination; /* MAC address of a node */
    int ifnumber; /* interface to reach it */
    u_short TTL; /* time to live */
    Binding binding; /* binding in the Map */
} BridgeEntry;

int numEntries = 0;
Map bridgeMap = mapCreate(BRIDGE_TAB_SIZE, sizeof(BridgeEntry));
```

ขั้นตอนการอัพเดต forwarding table กำหนดให้ updateTable โดยรับอาร์กิวเมนต์ 2 ตัวແປໄດ້ແກ່ MAC ต้นทาง (MacAddr src) และหมายเลขการ์ดเครือข่ายที่ข้อมูลนั้นเดินทางมา (int inif) โดย mapResolve() ทำหน้าที่เพิ่มข้อมูลลงตาราง forwarding table

```
void
updateTable (MacAddr src, int inif)
{
    BridgeEntry *b;

    if (mapResolve(bridgeMap, &src, (void **)&b) == FALSE )
    {
        /* this address is not in the table, so try to add it */
        if (numEntries < BRIDGE_TAB_SIZE)
        {
            b = NEW(BridgeEntry);
            b->binding = mapBind( bridgeMap, &src, b);
            /* use source address of packet as dest. address in table */
            b->destination = src;
            numEntries++;
        }
    }
}
```

```

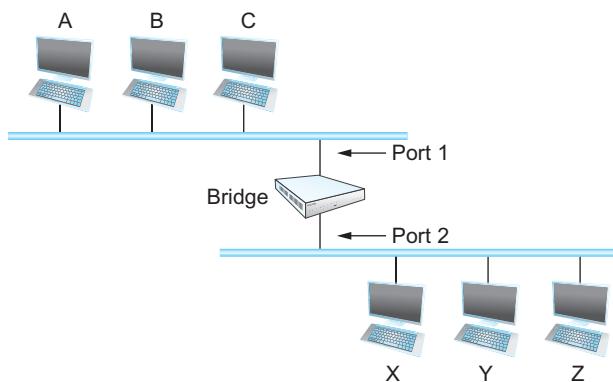
    }
    else
    {
        /* can't fit this address in the table now, so give up */
        return;
    }
}
/* reset TTL and use most recent input interface */
b->TTL = MAX_TTL;
b->ifnumber = inif;
}

```

การทำงานของโค้ดนี้นี้ใช้วิธีง่ายๆ แต่ในกรณีที่ bridge table เต็ม ทำให้ไม่สามารถเพิ่มข้อมูลใหม่ได้ สวิตช์อาจจะเลือกวิธีส่งต่อ หรือส่งต่อแพ็กเก็ตไปทุกพอร์ตก็ได้ ซึ่งสวิตช์ที่พบส่วนใหญ่มี bridge table เต็ม จะใช้วิธีส่งต่อแพ็กเก็ตทั้งหมดไปยังทุกพอร์ตเพื่อให้การสื่อสารเครือข่ายไม่สะคุต แต่ประสิทธิภาพการทำงานลดลงอย่างมาก และยังมีข้อเสียในด้านความมั่นคงปลอดภัยของข้อมูลซึ่งจะกล่าวถึงในบทที่ 6

3.2.3 อัลกอริทึมต้นไม้แบบทอดข้าม

กระบวนการต้นไม้แบบทอดข้ามเป็นการเชื่อมโยงเครือข่ายที่ไม่เกิดลูปในเครือข่าย การเกิดลูปในเครือข่ายคือ ข้อมูลสามารถเดินทางกลับมาที่จุดเริ่มต้นได้ ตัวอย่างรูปที่ 3.10 ในรูปมีลูปทั้งหมด 3 ลูปดังนี้ (S1-S2-S3-S4), (S1-S4-S6) และ (S1-S5-S7)



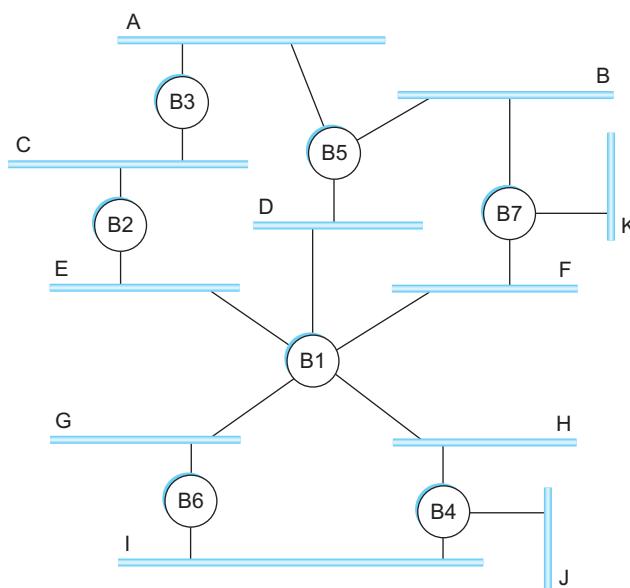
รูปที่ 3.10: บริดจ์เชื่อมการสื่อสารระหว่างสองเครือข่าย
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ถึงแม้เครือข่ายมีการเชื่อมต่อแบบลูปแต่เทคโนโลยีต้นไม้แบบทอดข้ามจะทำให้ตัดเส้นที่ทำให้เกิดลูปออกได้และยังทำให้เครือข่ายไม่ลูปออกจาก การเชื่อมต่อ

การเกิดลูปนั้นนอกจากเกิดจากความไม่ตั้งใจของผู้ดูแลระบบแล้ว อาจจะเกิดจากความตั้งใจได้ เช่น กัน เพื่อทำให้เครือข่ายมีลูปจะทำให้มีเส้นทางในการเดินทางไปถึงปลายทางได้หลายทาง หากเส้นได้เส้นหนึ่งขาด

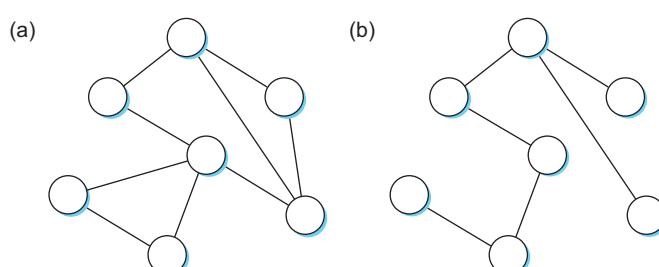
ระบบจะได้มีทางเลือกสำหรับเส้นทางใหม่ แต่การสื่อสารจะปล่อยให้เกิดลูปในเครือข่ายไม่ได้ ดังนั้นในหนึ่งเวลา จะมีเพียงต้นไม้แบบทอดข้ามเดียวที่เกิดในระบบ และเมื่อมลิงค์ขาดจึงเริ่มต้นสร้างต้นไม้แบบทอดข้ามใหม่

จากรูปที่ 3.11 อธิบายโดยใช้กราฟ ให้วอร์เท็กซ์(vertex)แทนสวิตซ์ และ เลี้ยวเชื่อม(edge)เส้นสาย เชื่อมระหว่างสวิตซ์ ได้เป็นรูปที่ 3.12



รูปที่ 3.11: อีเทอร์เน็ตสวิตซ์ที่มีลูป
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

รูปที่ 3.12(a) แทนกราฟที่มีลูป และรูปที่ 3.12(b) แทนกราฟต้นไม้แบบทอดข้าม



รูปที่ 3.12: เครือข่ายปกติและเครือข่ายแบบต้นไม้แบบทอดข้าม
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

แนวคิดต้นไม้แบบทอดข้ามทำความเข้าใจได้ง่าย เป้าหมายเพื่อให้เครือข่ายไม่เกิดลูปและยังคงทำให้ทุกอุปกรณ์ ในเครือข่ายยังคงเชื่อมกันได้ ส่วนงานที่ยากได้แก่ จะทำอย่างไรให้สวิตซ์ทุกเครื่องมองเห็นโครงข่ายทั้งหมด เหมือนกัน เพื่อให้ใช้ต้นไม้แบบทอดข้ามเดียวกัน ซึ่งสวิตซ์แต่ละตัวจะมองเห็นต้นไม้แบบทอดข้ามไม่เหมือนกัน คำตอบอยู่ที่ต้นไม้แบบทอดข้ามโพรโทคอล ซึ่งได้รับการเสนอโดย เรเดีย เพิร์ลแมน(รูปที่ 3.13)

ต้นไม้แบบทอดข้ามพัฒนาโดย เรเดีย เพิร์ล
แมน ได้คิดค้นวิธีขยายการเชื่อมต่อเครือข่ายอีเทอร์
เน็ตโดยใช้ต้นไม้แบบทอดข้ามโพรโทคอล ยังได้แต่กี



อธิบายการทำงานต้นไม้แบบทอดข้ามโพร์โทคอลໄว้
ในงานวิจัย ([Perlman, 1985](#)) ดังนี้

Algorhyme

I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.
A tree which must be sure to span
So packets can reach every LAN.
First the root must be selected.
By ID it is elected.
Least cost paths from root are traced.
In the tree these paths are placed.
A mesh is made by folks like me
Then bridges find a spanning tree.

เรเดีย เพิร์ลแมน

สิ่งที่เรเดีย เพิร์ลแมน คิดขึ้นมาได้เปลี่ยนการสื่อสารอีเทอร์เน็ตจากเดิมที่สื่อสารได้ไม่ก่อร้ายให้เป็น
เทคโนโลยีการสื่อสารที่ขยายขนาดได้ รองรับการจัดการเครือข่ายขนาดใหญ่ และเป็น wang รากฐานสำหรับเครือ
ข่ายอินเทอร์เน็ตในปัจจุบัน

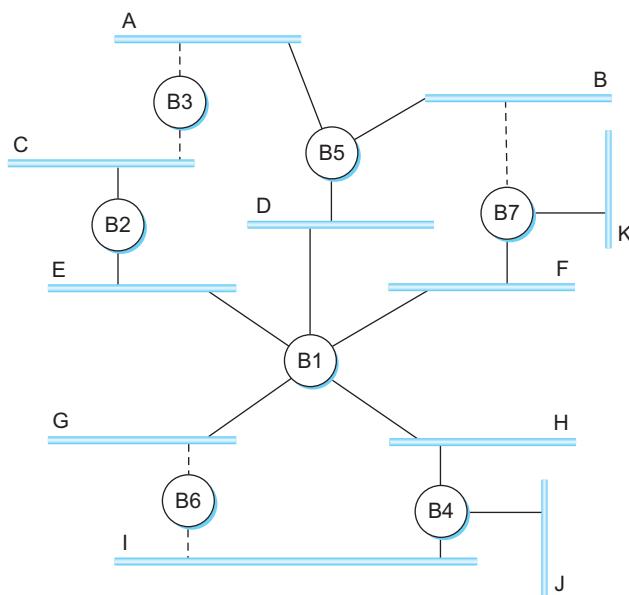
ลูปเป็นตัวปัญหาในอีเทอร์เน็ตเมื่อเกิดการเชื่อมแلن เป็นวงกลมทำให้เกิดการส่งข้อมูลไม่หยุดเรียกปัญหานี้ว่า
Broadcast storm

ໄอยเดียของต้นไม้แบบทอดข้ามโพร์โทคอลคือการกำหนดให้สวิตช์เลือกพอร์ตที่อนุญาตให้ส่งข้อมูลและ
ปิดพอร์ตที่ไม่อนุญาตให้ส่ง เมื่อจะควบคุมทุกสวิตช์ได้จะต้องมีสวิตช์หนึ่งที่ตรวจสอบเพียงตัวเดียวเพื่อไม่ให้
ทำงานขัดแย้งกัน เรียกว่าสวิตช์ราก(*root*) การเลือกรากทำได้โดยเลือกค่า ไอดี(*identifier*) ของสวิตช์ที่มีค่า
น้อยที่สุดให้หน้าที่เป็นราก สวิตช์ที่ทำหน้าที่เป็นรากจะมีสิทธิ์ส่งข้อมูลออกได้ทุกพอร์ต โดยมีข้อมูลที่ทำหน้าที่
ตรวจสอบเส้นทางที่สั้นที่สุดที่เดินทางไปถึงรากสวิตช์ ซึ่งสวิตช์ที่นอกเหนือจากรากจะจัดพอร์ตที่ส่งข้อมูลไปทาง
รากที่สั้นที่สุดไว้ และปิดพอร์ตอื่นที่ทำให้การเดินทางข้อมูลยาว

ขณะที่ในมุมมองคนงานคิดเป็นโครงข่ายตามรูปที่ [3.11](#) แต่เมื่อประมวลผลด้วยต้นไม้แบบทอดข้ามโพ
ร์โทคอลจะเป็นดังรูปที่ [3.14](#) เมื่อเส้นทึบใช้แทนการอนุญาตให้ส่งข้อมูลและเส้นประใช้แทนเส้นที่ไม่ถูกใช้งาน

การกำหนดข้อมูลให้เขตเดอร์มี 3 ส่วน

1. ไอดีของสวิตช์ที่ส่งออก



รูปที่ 3.14: เครือข่ายเชื่อมแบบบริดจ์ในมุมมองโลจิคัล
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

2. ไอเดียของสวิตซ์ที่คาดว่าเป็นรูห
3. d (ระยะทาง), ที่สวิตซ์ผู้ส่งแพ็คเก็ตประเมิน เป็นจำนวนระยะห่างทางเครือข่าย จากสวิตซ์ผู้ส่งเทียบกับรูห

หากพบรูหแพ็คเก็ตใหม่มีจำนวนระยะห่างทางเครือข่ายสั้นกว่าที่เคยบันทึก สวิตซ์จะยกเลิกข้อมูลเก่า และใช้ข้อมูลจากแพ็คเก็ตเดอร์รีใหม่แทน ถ้าแพ็คเก็ตนั้นส่งมาจาก ruth ค่าระยะห่างทางเครือข่ายจะมีค่าเป็น 0

เมื่อเริ่มต้นสวิตซ์ทุกเครื่องถูกกำหนดให้เป็นรูห และต่อมาเริ่มส่งแพ็คเก็ตแล้วจึงเริ่มประมวลผลระยะห่างระหว่างกันและจุดท้าย Ruth จะเหลือเพียงหนึดเดียวที่มี $d=0$

กำหนดให้ X แทนสวิตซ์ต้นทาง และ Y แทน Ruth สวิตซ์ และ d แทนค่าระยะห่าง โดยที่ข้อมูลจัดเรียงตามลำดับ (Y,d,X) อธิบายลำดับการทำงานของ S3 เป็นลำดับดังนี้

1. S3 ได้รับข้อมูล $(S2,0,S2)$
2. จากที่ $2 < 3$, S3 รับ S2 เป็นรูห
3. S3 เพิ่มระยะทาง $d+1$ โดยที่ $S2(0)$ และส่งข้อมูล $(S2, 1, S3)$ ไปยัง S5
4. ขณะเดียวกัน S2 รับ S1 เป็นรูห และส่ง $(S1,1,S5)$ ไปยัง S3
5. S5 รับ S1 เป็นรูห และส่ง $(S1,1,S5)$ ไปยัง S3
6. S3 รับ S1 เป็นรูห และบันทึกว่าทั้ง S2 และ S5 อยู่ใกล้ Ruth แต่ S2 มีไอเดียน้อยกว่า จึงเหลือเส้นทาง S3 เป็นต้นไม้แบบทอดข้ามต่ำสุด(minimum spanning tree)ไปยัง Ruth

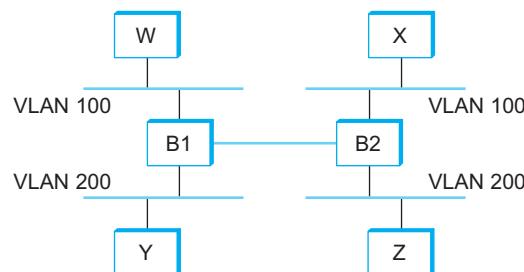
3.2.4 บродคาสต์ และ มัลติคาสต์(multicast)

นอกจากการส่งแพ็กเก็ตในส่วนข้อมูลที่ต้องการส่งจากต้นทางถึงปลายทางโดยตรงแล้ว ในการบริหารจัดการเครือข่ายยังต้องการให้สวิตช์ข้อมูลที่มีการกระจายสู่ปลายทางได้หลายเครื่อง การกระจายข้อมูลสู่ปลายทางหลายเครื่องแบ่งเป็น 2 กลุ่มได้แก่ บродคาสต์ และ มัลติคาสต์ บродคาสต์เป็นวิธีกำหนดปลายทางด้วยแอดเดรสเดียวแต่หมายถึงการกระจายไปทุกอุปกรณ์ในเครือข่าย มัลติคาสต์ เป็นวิธีกำหนดแอดเดรสให้สำหรับการเลือกเฉพาะกลุ่มที่ต้องการส่งข้อมูลไปทาง

3.2.5 Virtual LANs (วีแลน)

หนึ่งในข้อจำกัดของสวิตช์คือไม่สามารถเพิ่มจำนวนเครือข่ายในสวิตช์ตัวเดียวได้ ซึ่งข้อจำกัดเกิดจากอีเทอร์เน็ตสวิตช์ต้องการใช้บroadcast domain สำหรับส่งแพ็กเก็ตเลือกเส้นทาง ทำให้การแบ่งเครือข่ายในสวิตช์ทำไม่ได้ เพราะไม่สามารถแบ่งส่วนของบroadcast domain ออกจากสวิตช์ได้ สำหรับส่วนที่เป็นข้อมูลบroadcast domain เรียกว่า “บroadcast domain(broadcast domain)” ใน การแบ่งบroadcast domain ในหนึ่งสวิตช์ทำได้ด้วยการใช้เทคโนโลยีวีแลน (virtual LANs) เทคโนโลยีวีแลนใช้วิธีกำหนดไอดีให้แก่แต่ละบroadcast domain เรียกว่า “วีแลนไอดี (VLAN ID)”

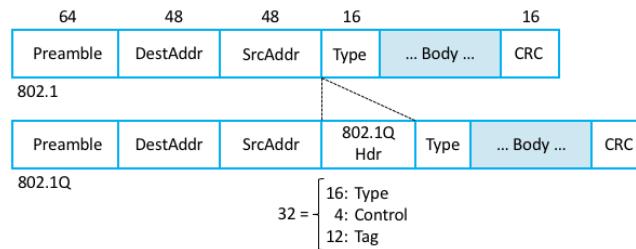
อธิบายการทำงานวีแลนจากตัวอย่างในรูปที่ 3.15 จากรูปมีไฮสตร์ทั้งหมดสี่ไฮสตร์ มีสวิตช์สองเครื่องสำหรับสวิตช์สองเครื่องมีการแบ่งบroadcast domain เป็นกลุ่มโดยกำหนดหมายเลขวีแลนไอดีเป็น 100 และ 200 ให้ไฮสตร์ W, X อยู่ในกลุ่มวีแลนไอดี 100 และ Y, Z อยู่ในกลุ่มวีแลนไอดี 200 ในการกำหนดหมายเลขวีแลนนั้นห้ามระบบจะเป็นผู้กำหนดค่าลงในสวิตช์



รูปที่ 3.15: เครือข่ายคู่ร่วมกันที่ใช้สวิตช์ร่วมกัน
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

เมื่อมีแพ็กเก็ตส่งออกจากไฮสตร์ X เดินทางถึง S2 สวิตช์อ่านข้อมูลจากพอร์ตและพบว่าพอร์ตได้กำหนดให้มีวีแลนไอดีเป็น 100 ก็จะเขียนข้อมูลลง帧เดอร์ให้ค่าวีแลนไอดี=100 ก่อนส่งไปยังสวิตช์อื่นต่อไป เมื่อ S2 ได้รับข้อมูลจาก S1 จะเริ่มอ่าน帧เดอร์และอ่านวีแลนไอดีเมื่อได้ข้อมูลเป็นวีแลนไอดี 100 จะตรวจสอบว่าพอร์ตใดมีวีแลนไอดีตรงกันจึงส่งไปพอร์ตตั้งพร้อมกับลงข้อมูล帧เดอร์ให้เป็นปกติ จากรูปนี้จะเห็นได้ว่าการกำหนดวีแลนไอดีเป็นการทำงานของผู้ดูแลระบบ โดยที่ไฮสตร์สามารถทำงานได้โดยไม่ต้องกำหนดค่าใดๆ

ข้อมูลในส่วนของ帧 header ที่ต้องการเพิ่มให้รองรับ VLAN เป็นการเขียนข้อมูลเพิ่มเติมจาก帧 header มาตรฐาน 802.1 โดยเพิ่มชุดข้อมูลขนาด 12-บิต เป็นวีแอลไอดี(VID) อยู่ระหว่าง ‘SrcAddr’ และ ‘Type’ ตามรูปที่ 3.16



รูปที่ 3.16: 802.1Q VLAN tag แทรกข้อมูลใน帧 header มาตรฐานอีเทอร์เน็ต(802.1)
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

3.3 อินเทอร์เน็ตโพรโทคอล (IP)

จากหัวข้อที่ผ่านมาได้กล่าวถึงการเชื่อมเครือข่ายด้วยบริดจ์และอีเทอร์เน็ตสวิตช์ ซึ่งมีข้อจำกัดจากการสื่อสารเป็นทอดๆ และการขยายเครือข่ายให้มีขนาดใหญ่ สำหรับหัวข้อนี้จะกล่าวถึงการขยายเครือข่ายให้ใหญ่ขึ้นด้วยการสร้างและตรวจสอบการทำงานเครือข่ายเครือข่าย VLAN ซึ่งเป็นที่มาของคำว่า ‘อินเทอร์เน็ตเวิร์ก’

3.3.1 แนะนำแนวคิด Addressing

ความหมายของอินเทอร์เน็ตเวิร์ก

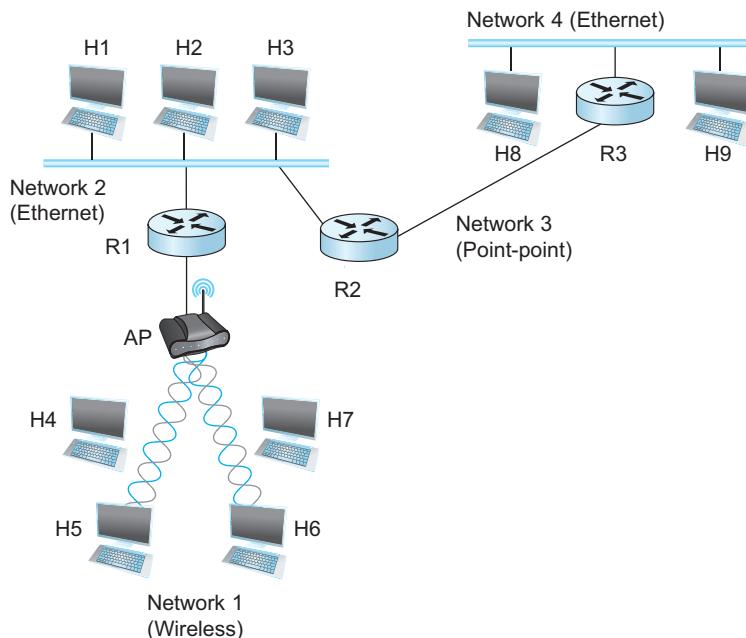
คำว่าอินเทอร์เน็ตเวิร์กหรือบางครั้งพิมพ์สั้นลงด้วยคำว่า ‘อินเทอร์เน็ต³’ เพื่ออ้างถึงคุณสมบัติในการเชื่อมโยงการทำงานระหว่างเครือข่าย ให้มองเห็นเป็นการส่งแพ็กเก็ตแบบ host-to-host โดยตรง

ส่วนประกอบของอินเทอร์เน็ตมีหลายส่วนที่อาจทำให้สับสนในเวลานี้ เช่น ความท่างระหว่าง เนตเวิร์ก, ชั้บเน็ตเวิร์ก(subnetwork) และ อินเทอร์เน็ตเวิร์ก ในที่นี้ขอยกเนื้อหา ชั้บเน็ตเวิร์ก ไปกล่าวถึงในหัวข้อต่อไป

ในที่นี้เนตเวิร์กคือการเชื่อมต่อโดยตรงไม่ว่าจะเป็นการเชื่อมผ่านสายเล่นเครือข่ายอีเทอร์เน็ต หรือ เป็นการเชื่อมต่อแบบไม่มีสายเล่นเครือข่ายแลนไร้สาย สำหรับอินเทอร์เน็ตเวิร์กเป็นการนำเนตเวิร์กที่กล่าวมาข้างบนเชื่อมต่อกัน

รูปที่ 3.17 เป็นตัวอย่างเครือข่ายอินเทอร์เน็ตเวิร์ก ที่เชื่อมเครือข่ายกับเครือข่าย “network of networks” จากรูปมีเครือข่ายขนาดเล็กจำนวน 4 เครือข่ายแล้วเชื่อมเข้าด้วยกันผ่านอุปกรณ์ที่เรียกว่า “เร้าเตอร์” จากรูปเป็นเครือข่ายเครือข่ายแลน (Ethernet) 2 เครือข่าย เครือข่ายแลนไร้สาย (Wireless) 1 และเป็นเครือข่ายแบบ point-to-point ที่เชื่อมระหว่างเร้าเตอร์กับเร้าเตอร์ 1 เครือข่าย

³internet อักษร ‘i’ ตัวพิมพ์เล็กในที่นี้แตกต่างจาก Internet ที่มี I ตัวพิมพ์ใหญ่



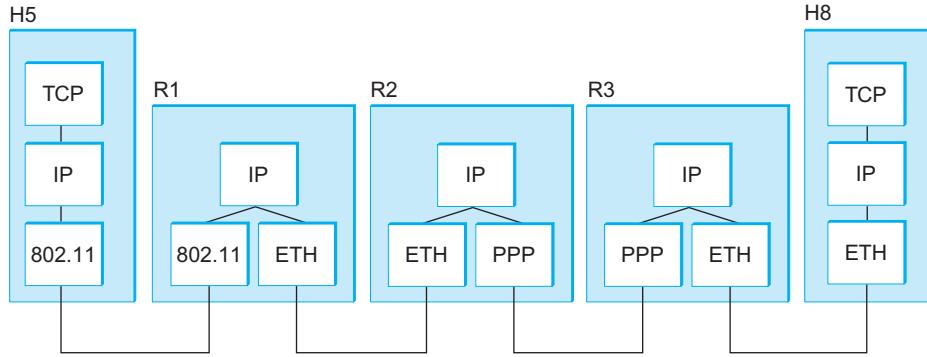
รูปที่ 3.17: อินเทอร์เน็ตเวิร์กที่พับโดยตามปกติ H แทนโฮสต์และ R แทนเร้าเตอร์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

อินเทอร์เน็ตโพรโทคอลเป็นกุญแจสำคัญที่ทำให้เครือข่ายคอมพิวเตอร์ขยายขอบเขตได้ขนาดใหญ่ เช่นในปัจจุบัน มีการเชื่อมโยงกับเครือข่ายขนาดเล็กที่แตกต่างกันเข้าด้วยกันได้ เช่นสามารถติดต่อเครือข่าย อีเทอร์เน็ตกับเครือข่ายโทรศัพท์เคลื่อนที่ได้ ต้นกำเนิดอินเทอร์เน็ตโพรโทคอล พัฒนาโดย ดร.โรเบิร์ต คาห์น (Dr. Robert Kahn) และ ดร.วินตัน เซร์ฟ(Dr. Vinton Cerf) ทั้งสองท่านได้รับรางวัล TURING AWARD ปี ค.ศ. 2004 ([Hyman, 2012](#)) สำหรับการคิด อินเทอร์เน็ตโพรโทคอล หรือรู้จักในชื่อ Kahn-Cerf โพรโทคอล ([Leiner และคณะ, 2009](#)) มาจากชื่อผู้คิดค้นทั้งสองท่าน

สามารถทำความเข้าใจไอพีได้จากแนวคิดต้องการให้อุปกรณ์ทุกเครื่องในเครือข่ายมีหมายเลขประจำตัวแบบไม่ซ้ำกัน หมายเลขอันดับกำหนดให้ชี้ว่าหมายเลขไอพี ทุกอุปกรณ์ในเครือข่ายอินเทอร์เน็ตจะยอมรับข้อตกลงนี้ โดยหมายเลขไอพีจะไม่ซ้ำกันเลยในเครือข่ายอินเทอร์เน็ต หน้าที่ของไอพีคือกำหนดที่ระบุชื่อที่ไม่ซ้ำของอุปกรณ์นั้นเอง เมื่ออุปกรณ์ทุกตัวมีชื่อไม่ซ้ำกันแล้วทำให้ข้อมูลเดินทางถึงเป้าหมายได้

การเสนอให้มีหมายเลขไอพีทำให้อุปกรณ์ที่มีเทคโนโลยีต่างกันสามารถสื่อสารกันได้โดยใช้ชื่อไอพีในรูปแบบเดียว ดังตัวอย่างรูปที่ 3.18 ที่แสดง H5 และ H8 มีต้นทางเป็นเทคโนโลยีอีเทอร์เน็ตมีการระบุแอดเดรสที่เป็นไปตามเทคโนโลยีอีเทอร์เน็ต ซึ่งเรียกว่าพิสิคัลแอดเดรส(physical address) และ H8 เป็นเทคโนโลยีแลนไวร์ลีย์ มีการทำหนดแอดเดรสเป็นไปตามเทคโนโลยีแลนไวร์ลีย์ ซึ่งเป็น พิสิคัลแอดเดรสในรูปแบบของแลนไวร์ลีย์ แต่ทั้งสองเทคโนโลยีสามารถเชื่อมต่อกันได้ผ่านอินเทอร์เน็ตโพรโทคอล

ในทวีปนี้ได้กล่าวถึงภาพรวมของไอพี และทวีปต่อไปกล่าวถึงไอพีในรายละเอียดที่ลึกซึ้ง อินเทอร์เน็ตโพรโทคอลมีความน่าสนใจตรงที่เป็นตัวกำหนดเครือข่ายอินเทอร์เน็ตขยายได้และเป็นที่ยอมรับใช้งานมากทุกวันนี้ และอินเทอร์เน็ตโพรโทคอลก็ยังเป็นตัวที่จำกัดไม่ให้อินเทอร์เน็ตโพรโทคอลขยายไปมากกว่านี้ได้จนนำไปสู่การออกแบบไอพีรุ่น ๖



รูปที่ 3.18: อินเทอร์เน็ตเวิร์กอย่างง่าย มีการเชื่อมต่อกันระหว่าง H5 และ H8 ผ่านโพรโทคอล แลนไวร์ลีย์และ อีเทอร์เน็ต

ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

Service Model

เป็นจุดเริ่มต้นที่ดีที่จะพูดถึงหน้าที่การทำงานของอินเทอร์เน็ตเวิร์กโดยมาอธิบายจากหน้าที่ตัวแบบบริการ (service model) เนื้อหาที่จะกล่าวถึงต่อไปนี้คือการส่งข้อมูลแบบ host-to-host สิ่งที่ต้องคำนึงถึงได้แก่ จะทำอย่างไรให้โฮสต์สามารถสื่อสารได้ถึงปลายทางทั้งที่ยังขึ้นอยู่กับเทคโนโลยีของการสื่อสารจากความต่างทาง เทคโนโลยีที่มีการติดต่อแตกต่างกันในทางกายภาพ ยกตัวอย่าง เช่น คงไม่ได้แน่ถ้าหากเครือข่ายต้องการส่งข้อมูล โดยรับรองว่าข้อมูลจะเดินทางถึงปลายทางได้ภายใน 1 มิลลิวินาที หรือน้อยกว่านั้น ทั้งที่ต้องเกี่ยวข้องกับการ สื่อสารระหว่างโพรโทคอลหลายขั้นตอน ดังนั้นในส่วน IP จะออกแบบให้สื่อสารให้เร็วที่สุดเท่าที่เป็นได้หรือ เรียกว่า “best effort” อย่างไรก็ตามวิธีนี้ไม่สามารถ garant ได้ว่าข้อมูลจะเดินทางเร็วหรือเดินทางช้า ในส่วน IP จะไม่ดำเนินการแก้ไขใดๆ เพราะพยายามดีที่สุด(best-effort)แล้ว

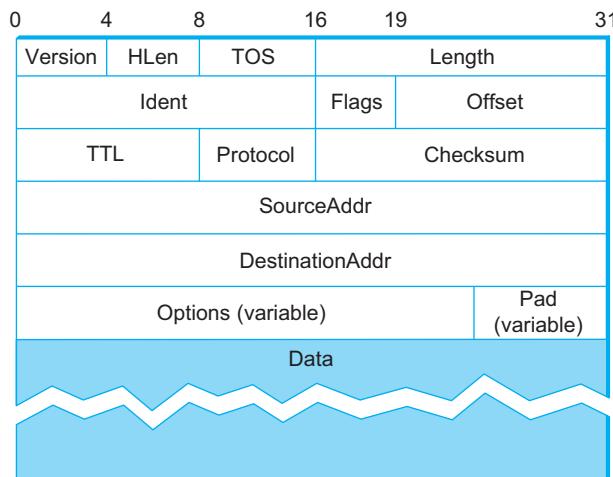
การขนส่งเดต้าแกรม

อินเทอร์เน็ตโพรโทคอลเดต้าแกรมเป็นพื้นฐานของอินเทอร์เน็ตโพรโทคอล ทบทวนในหัวข้อที่แล้วเดต้าแกรม เป็นแพ็กเก็ตประเภทหนึ่งที่มีคุณสมบัติเป็นไม่กำหนดการ เชื่อมต่อ ทุกแพ็กเก็ต ที่เดินทางเข้าสู่เครือข่ายจะถูก ส่งไปยังปลายทางได้ถูกต้อง โดยไม่ต้องกำหนดค่าใดๆ เพิ่มเติม ในที่นี้พยายามดีที่สุด จึงหมายถึงการดำเนินการ ได้โดยกตามที่จะทำให้ส่งผิดส่งไม่สำเร็จมีข้อมูลผิดพลาดบางส่วน

พยายามดีที่สุด เป็นบริการแบบไม่กำหนดการ เชื่อมต่อที่มีรูปแบบการสื่อสารอย่างง่าย ผู้ส่งสามารถ ขอใช้บริการส่งแพ็กเก็ตแบบไม่กำหนดการ เชื่อมต่อจากบริการกำหนดการ เชื่อมต่อได้ และเครือข่ายจะให้บริ การแบบไม่กำหนดการ เชื่อมต่อ จะเห็นได้ว่าวิธีนี้เป็นวิธีที่สื่อสารได้ง่าย ไม่ต้องใช้การออกแบบซับซ้อน สำหรับ การส่งข้อมูลที่ต้องการการันตีว่าข้อมูลเดินทางถึงปลายทางได้แน่นอนนั้นจะปล่อยให้เป็นหน้าที่ของเลเยอร์ขั้น ต่อไป

รูปแบบแพ็คเก็ต

สำหรับแพ็คเก็ตประเภทไอโอพินั่นเหมือนกับแพ็คเก็ตที่ว่าไปที่ประกอบด้วยヘดเดอร์ ภายในเขตเดอร์บรรจุข้อมูลที่จำเป็นสำหรับอินเทอร์เน็ตโพรโทคอล ข้อมูลส่วนเหลือของไอโอพีชิบะยในรูปที่ 3.19



รูปที่ 3.19: เฮดเดอร์ของไอโอพีรุ่น ๔
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ดูรายละเอียดในแต่ละฟิลด์ของแพ็คเก็ตนั้นทำความเข้าใจได้ง่ายมาก เพราะออกแบบให้มีแนวคิดพยาามดีที่สุด ดังนั้นในแต่ละฟิลด์จึงมีไม่มาก ส่วนแรกได้แก่ฟิลด์เวอร์ชัน(version) สังเกตได้ว่าฟิลด์เวอร์ชันจะถูกกำหนดให้อยู่ส่วนแรกของヘดเดอร์เพื่อให้โปรแกรมสามารถอ่านข้อมูลได้ก่อนประมวลผลฟิลด์ต่อไป เพื่อใช้เป็นข้อมูลสำหรับการกำหนดรูปแบบヘดเดอร์ ในฟิลด์Version ส่วนนี้ใช้ระบุถึงหมายเลขรุ่นของไอโอพีเมื่อมีค่าตัวเลข 4 หมายถึงไอโอพีรุ่น ๔ ลำดับต่อมาได้แก่เหตุผลมีขนาดヘดเดอร์เท่ากับ 32 บิตค่าหนึ่งบิตของข้อมูล HLen หมายถึง 4 บิต ลำดับต่อมาก็ ชนิดการบริการ(type of service) TOS มีการใช้งานแตกต่างจากที่ออกแบบไว้แต่แรก และมีการเปลี่ยนแปลงมาโดยตลอด ซึ่งก็เป็นไปตามวัตถุประสงค์ของฟิลด์ ที่ออกแบบให้สามารถเปลี่ยนแปลงได้ ซึ่งปัจจุบัน TOS ใช้สำหรับกำหนดลำดับความสำคัญของแพ็คเก็ตเพื่อบรรบค่าดีเลย์ของแพ็คเก็ตให้เหมาะสมกับการใช้งาน

อีก 16 บิตลำดับต่อมาก็ใช้กำหนดค่า Length ของแทกต่างๆจากเดาแกรมรวมヘดเดอร์ด้วย ค่านี้มีความแตกต่างจาก HLen สำหรับ HLen นับจำนวน word ขณะที่ Length นับจำนวนไบต์ ค่าความยาวข้อมูลของ IP มีสูงสุดได้ไม่เกินค่าสูงสุดของ 32บิต หรือเท่ากับ $2^{32} = 65,535$ อย่างไรก็ตามเมื่อแพ็คเก็ตมีขนาดใหญ่กว่าที่ชั้นกายภาพจะรับได้ จะมีการแบ่งข้อมูลหรือเรียกว่า Fragmentation และ Reassembly ชุดข้อมูลลำดับต่อมาก็ได้แก่ TTL(time to live) มีขนาดหนึ่งไบต์ ข้อมูลส่วนนี้ใช้ระบุให้เราเตอร์ได้อ่านเพื่อใช้กำหนดอายุให้แพ็คเก็จที่ส่งออกสู่เครือข่ายนั้นจะมีจำนวนระยะห่างทางเครือข่ายได้ไม่เกินค่าที่ระบุใน TTL เมื่อแพ็คเก็ตเดินทางผ่านเราเตอร์จะทำให้ค่า TTL เอาลดลงหนึ่งและลดลงไปเรื่อยๆจนกระทั่ง TTL=0 และข้อมูลยังเดินทางไม่ถึงปลายทางแพ็คเก็ตนั้นจะถูกลบออกจากระบบ

ลำดับต่อมาก็เป็นโพรโทคอล เป็นข้อมูลสำหรับการใช้คอมมาร์ทสำหรับเลเยอร์ที่สูงขึ้นตัวอย่างเช่นข้อมูลถูกส่งด้วย IP และส่งต่อขึ้นไปยังโพรโทคอล TCP หรือ UDP ข้อมูลในส่วนเหลือจะใช้ระบุว่าเลเยอร์ที่จะส่งไป

นั้นเป็น TCP = 6 หรือเป็น UDP = 17 ลำดับต่อมา Checksum ค่าระบุใน Checksum ได้จากการคำนวณข้อมูลในส่วน IP เข้าเดอร์ที่มีความยาว 16 บิตและนำมาใช้กระบวนการ CRC สองฟิล์ดสุดท้ายในเข้าเดอร์ได้แก่ SourceAddr และ DestinationAddr ใช้กำหนดหมายเลขอพีตันทางและหมายเลขอพีปลายทาง

Protocol version ใช้กำหนดรูปแบบเข้าเดอร์ กรณีproto Kol/IoP รุ่นที่ ๔ อธิบายด้วยรูปตามรูปที่ 3.20 จากรูปเรียงบิตจากซ้ายไปขวาและ 32บิต เริ่มต้นด้วย version ใช้ขนาด 4 บิต **version** ใช้ระบุข้อมูลในเพรมนั้นเป็น IoP รุ่น ๔ เมื่อหัสดามรูปที่ 3.20 หรือ รุ่น ๖ มีรหัสในรูปที่ 3.21 โดยเรียงจากบิตน้อยสุดอยู่ซ้ายมือเมื่อการ์ดเครื่องข่ายอ่านข้อมูลถึงส่วน version จะตรวจสอบประเภทของ IoP เมื่อทราบแล้วจึงเตรียมระบบให้อ่านได้ตรงกับรุ่น

0	1	2	3
0	1	0	0

รูปที่ 3.20: รหัสไบนาเรียกกำหนด proto Kol/IoP รุ่น ๔
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

รูปที่ 3.20 เริ่มต้นเรียงจากบิตที่ 0 ถึงบิตที่ 3 โดยการอ่านข้อมูลกำหนดให้บิตน้ำหนักมากสุดอยู่ด้านซ้ายมือ(MSB(most significant bit)) เมื่อแปลงเลขฐานสองให้เป็นเลขฐานสิบจะได้ค่าเท่ากับ 4 ซึ่งตรงกับหมายเลขอพีเวอร์ชัน ๔ สำหรับรูปที่ 3.21 เมื่อแปลงเลขฐานสองของ $(0110)_2$ เป็นเลขฐานสิบจะได้ ๖ ซึ่งตรงกับ IoP รุ่น ๖

0	1	2	3
0	1	1	0

รูปที่ 3.21: รหัสไบนาเรียกกำหนด proto Kol/IoP รุ่น ๖
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

HLen ใช้ระบุขนาดความยาวไบต์ของเข้าเดอร์ มีขนาด 4 บิตแต่ใช้อ้างถึงข้อมูลขนาด 32บิต หรือเท่ากับ 4 ไบต์ โดยกำหนดให้หนึ่งบิตแทนด้วย 4 ไบต์ เช่นอ่านค่าได้ $(0101)_2 = 5$ หมายถึงความยาวข้อมูลเข้าเดอร์มีขนาด $4 \times 5 = 20$ ไบต์ ตัวอย่างในรูปที่ 3.22

4	5	6	7
0	1	0	1

รูปที่ 3.22: ตัวอย่างรหัสไบนาเรียรระบุขนาดเข้าเดอร์มีขนาด 20 ไบต์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ECN(Explicit Congestion Notifications) ได้กำหนดใช้นับจากปี ค.ศ.2001 ระบุใน RFC-3168 ([RFC3168 \(2001\)](#)) ใช้รายงานความคับคั่งข้อมูลในขณะนั้น มีขนาด 2 บิต ความหมายอธิบายได้ในตารางที่ 3.22 เมื่อการสื่อสารระหว่างต้นทางกับปลายทางรองรับ การทำงานแบบ ECN จะอ่านค่าจากช่องอาจจะได้รับ ECT(0) หรือ ECT(1)

ตารางที่ 3.7: รหัสระบุประเภทความคับคั่ง

b_{14}	b_{15}	ความหมาย
0	0	Non ECN-Capable Transport
0	1	ECN Capable Transport, ECT(1)
1	0	ECN Capable Transport, ECT(0)
1	1	Congestion Encountered, CE

Fragmentation และ Reassembly

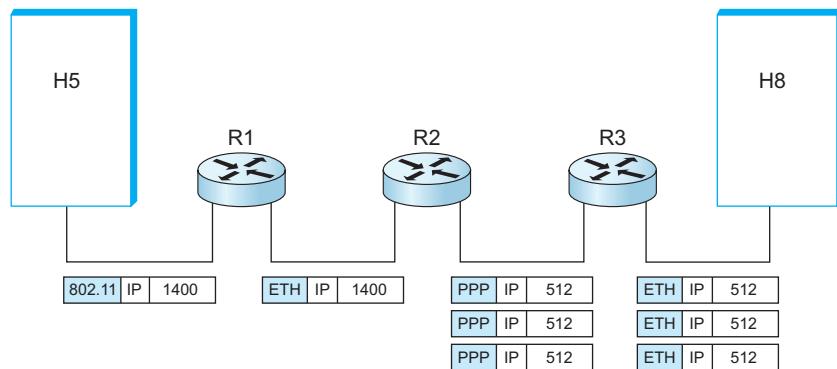
การหักแพ็กเก็ต และ การประกอบแพ็กเก็ต(reassembly) ตามที่ได้กล่าวมาข้างต้น อินเทอร์เน็ตเวิร์ก เกิดจากการรวมตัวกันของอุปกรณ์เน็ตเวิร์กที่มีความแตกต่างกัน เรียกว่า ความหลากหลาย(heterogeneous) ยกตัวอย่างเช่นเครือข่ายอินเทอร์เน็ตในอดีตส่งข้อมูลได้ครั้งละไม่เกิน 1500 ไบต์ และเทคโนโลยีเทอร์เน็ตปัจจุบันสามารถส่งแพ็กเก็ตได้ขนาดใหญ่ขึ้นเรียกว่าชัมโบ(jumbo)แพ็กเก็ต ขนาดข้อมูลสูงสุด 9000 ไบต์ เมื่อทั้งสองเทคโนโลยี ติดต่อกันผ่าน IP จะทำให้พบปัญหาเมื่อมีการส่งจ้มไปแพ็กเก็ตไปยังเครือข่ายอินเทอร์เน็ตรุ่นเก่าเมื่อส่งข้อมูลขนาด 9000 ไบต์ไปยังการ์ดเครือข่ายที่รองรับได้เพียง 1500 ไบต์จะทำให้ข้อมูลไม่สามารถส่งได้สำเร็จ

ปัญหานี้เกิดขึ้นได้กับทุกเครือข่ายเป็นข้อจำกัดทางกายภาพของเทคโนโลยีนั้นๆ ซึ่งมีในทุกเครือข่าย ข้อจำกัดหนึ่งที่พบได้คือขนาดแพ็กเก็ตที่สามารถส่งได้ในแต่ละครั้ง ซึ่งเรียกว่า เอ็มทีบี ขนาดเอ็มทีบีนี้เป็นขนาดของแพล็อกไม้บาร์วเมดเดอร์

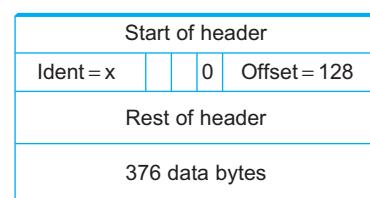
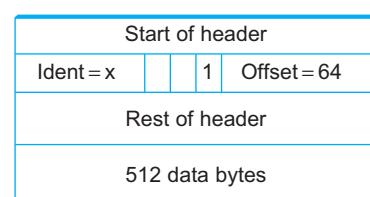
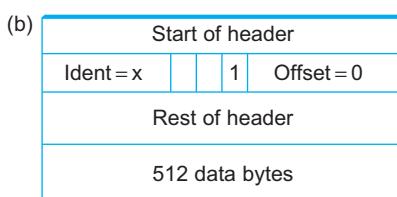
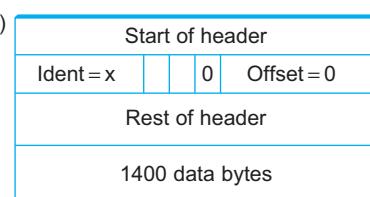
เมื่อโฮสต์จะส่งໄอีพีเดต้าแกรมสิ่งแรกที่ต้องทำการคือกำหนดขนาดของแพ็กเก็ตให้มีขนาดไม่เกิดเอ็มทีบีของเครือข่าย ซึ่งแพ็กเก็ตอาจเดินทางผ่านเครือข่ายด้วยเทคโนโลยีชั้นลิ๊งค์แตกต่างกัน แต่การตรวจสอบเอ็มทีบีของเครือข่ายนั้นทำได้ยาก ดังนั้นในชั้นໄอีพีจะตรวจสอบเฉพาะข้อจำกัดทางกายภาพที่อุปกรณ์เชื่อมต่อในปัจจุบันเท่านั้น เมื่อข้อมูลถูกส่งต่อไปยังเราเตอร์อิกด้านหนึ่งของเราเตอร์ที่มีการเชื่อมต่อกับเทคโนโลยีเครือข่ายอื่นจะทำการตรวจสอบเอ็มทีบีที่เหมาะสมและเมื่อพบว่าแพ็กเก็ตมีขนาดใหญ่กว่าเอ็มทีบีจึงเริ่มทำการหักแพ็กเก็ตข้อมูลให้มีขนาดไม่เกินเอ็มทีบีแล้วจึงส่งข้อมูล

ยกตัวอย่างเครือข่ายในรูปที่ 3.23 มีความต้องการส่งข้อมูลจาก H5 ไป H8 เมื่อ H5 เชื่อมต่อด้วยเทคโนโลยีแลนรีสาย และ H8 เชื่อมต่อด้วยเทคโนโลยีเทอร์เน็ต เส้นทางข้อมูลที่เดินทางจาก H5 ให้ H8 จะต้องส่งผ่านเครือข่ายอีเทอร์เน็ตและ point-to-point ซึ่งในที่นี้เทคโนโลยีอินเทอร์เน็ตสามารถส่งได้ขณะข้อมูลไม่เกิน 1400 ไบต์ขณะที่เทคโนโลยี point-to-point ส่งได้ไม่เกิน 512 ไบต์ดังนั้นเมื่อข้อมูลเดินทางผ่านเราเตอร์ R1 แพ็กเก็ตจะไม่ถูกการหักแพ็กเก็ตและส่งไปยังเราเตอร์ R2 แต่เมื่อแพ็กเก็ตจาก R2 ต้องการส่งไปยัง R3 พบว่าแพ็กเก็ตมีขนาดใหญ่กว่าเอ็มทีบีของ point-to-point ในขั้นตอนนี้จึงเกิดการหักแพ็กเก็ต แบ่งข้อมูลออกเป็นสามส่วนละ 512 ไบต์แล้วจึงส่งข้อมูลไปยัง R3 และ R3 ยังคงส่งข้อมูลขนาด 512 ไบต์จำนวนสามชุดไป H8 เมื่อ H8 ได้รับข้อมูลจะเริ่มการประกอบแพ็กเก็ตแพ็กเก็ตให้เป็นแพ็กเก็ตเดียว

เมื่อแพ็กเก็ต เกิดการหักแพ็กเก็ตจากหนึ่งแพ็กเก็ตเป็นสามแพ็กเก็ตจะต้องมี ตัวระบุว่าทั้งสามแพ็กเก็ตนั้นเกิดจากแพ็กเก็ตเดียวกันในไฟล์ໄอีพี ได้เพิ่มส่วนระบุความเป็นแพ็กเก็ตเดียวกันซึ่งชื่อว่า Ident (Identification) มีขนาด 2 ไบต์ ซึ่งรูปที่ 3.24 เกิดจากการแบ่งข้อมูลขนาด 1400 ไบต์เป็นข้อมูลเป็น 3 ส่วนมีขนาด 512 ไบต์ จำนวนสองแพ็กเก็ต และ 376 ไบต์



รูปที่ 3.23: การเดินทางของอินเทอร์เน็ตโปรโตคอลเดต้าแกรมผ่านเทคโนโลยีเครือข่ายหลายเทคโนโลยีโดยที่ R2 เชื่อมกับ R3 แบบ PPP มีอั้มที่ยุ่งตัวที่สุด
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>



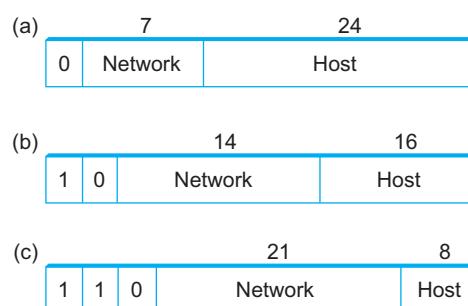
รูปที่ 3.24: แพ็กเก็ตเมดเดอร์ของเดต้าแกรมที่เกิดการหักแพ็กเก็ต
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

จากรูปที่ 3.24(a) แสดงแพ็กเก็ตตันฉบับมีขนาดข้อมูล 1400 ไบต์ และรูปที่ 3.24(b) แสดงถึงข้อมูลที่มีการหักแพ็กเก็ตแล้วสามแพ็กเก็ต เมื่อทั้งสามแพ็กเก็ตเดินทางถึง H8 จะเริ่มต้นการประกอบแพ็กเก็ต ด้วยอ่านข้อมูลในเขตเดอร์ จากรูป 3.24(b) ที่มีเลข Ident ตรงกันโดยที่แพ็กเก็ตที่มีเกิดการหักแพ็กเก็ตจะมีค่า

Flag=1 ยกเว้นแพ็กเก็ตสุดท้าย เมื่อเรียงลำดับตามค่า Offset จะครบทุกแพ็กเก็ต แม้ H8 จะได้รับแพ็กเก็ตเหมือนต้นทางทุกประการ

3.3.2 Global Addresses

จากที่กล่าวมาข้างต้นได้กล่าวถึงการให้บริการของ IP ซึ่งมีข้อมูลส่วนหนึ่งที่สำคัญสำหรับการให้บริการได้แก่ การระบุตัวอุปกรณ์ที่ไม่ซ้ำกันเรียกว่า ‘addressing’ ซึ่งเทคโนโลยีเทอร์เน็ตก็มีการระบุแอดเดรสแตกต่างกับการระบุแอดเดรสในระดับ IP ในการสื่อสารในระดับ IP ใช้วิธีกำหนดค่าตัวเลขจำนวน 32 บิต โดยที่ 32 บิตนี้แบ่งออกเป็นสามกลุ่มเรียกว่าคลาสได้แก่คลาส A คลาส B และคลาส C ตามอธิบายในรูปที่ 3.25



รูปที่ 3.25: อินเทอร์เน็ตโพรโทคอลแอดเดรส: (a) คลาส A; (b) คลาส B; (c) คลาส C
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ในการเป็นคลาสนั้น ใช้วิธีกำหนดค่าใน MSB ถ้าบิตที่อยู่ช้ายมีสุดมีค่าเท่ากับ 0 ถือว่าไอพีนั้นอยู่ในคลาส A ถ้าบิตแรกเป็น 1 และบิตต่อมาเป็น 0 ถือว่าไอพีนั้นอยู่คลาส B และ ถ้าสองบิตแรกเป็น 1 และบิตที่สามเป็น 0 ถือว่าไอพีนั้นอยู่คลาส C

การแบ่งคลาสนี้มีเพื่อกำหนดจำนวนโಯสต์ในเครือข่าย ซึ่งคลาส A มีจำนวนโโยสต์สูงที่สุด รองลงมาเป็นคลาส B และคลาส C มีจำนวนโโยสต์น้อยที่สุด

จากรูปที่ 3.25 คลาส A มีจำนวนเนตเวิร์กทั้งหมด 7 บิต และมีจำนวนโโยสต์ที่เป็นได้ทั้งหมด 24 บิตดังนั้นคลาส A จะมีจำนวนโโยสต์ได้สูงสุด 126 เครือข่าย (ค่า 0 และ 127 เป็นเลขสำรองไว้ใช้ในงานอื่น) และมีจำนวนโโยสต์ได้สูงสุด $2^{24} - 2$ (เช่นเดียวกับส่วนเนตเวิร์กสำรองค่าที่มีทุกบิตเป็น 0 ทั้งหมด และเป็น 1 ทั้งหมด ไม่สำหรับงานอื่น) ทำให้มีโโยสต์ได้สูงสุดประมาณ 16 ล้านโโยสต์ คลาส B มีจำนวนเนตเวิร์ก 14 บิตและมีจำนวนโโยสต์ 16 บิต ทำให้คลาสบี B หนึ่งเนตเวิร์กมีโโยสต์ได้สูงสุด $2^{16} - 2 = 65,534$ โโยสต์ คลาส C มีแอดเดรสเพียง 8 บิตทำให้หนึ่งเนตเวิร์กมีโโยสต์ได้สูงสุด $2^8 - 2 = 254$ โโยสต์ ($ip = 0$ ใช้ระบุชื่อเครือข่าย และ $ip = 255$ ใช้ระบุครอบคลาส) แต่สิ่งที่คลาส C ได้มาเพิ่มได้แก่มีจำนวนเนตเวิร์กจำนวนทั้งสิ้น 2^{21} เนตเวิร์ก

ก่อนที่จะกล่าวถึงวิธีการใช้งานไอพีแอดเดรส มีสิ่งสำคัญที่ควรกล่าวถึงได้แก่การเขียนไอพีในงานจริง ไอพีแอดเดรสในการใช้งานจริงนั้นเขียนในรูปแบบเลขฐานสิบ โดยแบ่งฐานสองครั้งละหนึ่งไปตัวและคั่นด้วยเครื่องหมาย “.” ตัวอย่างเช่น 110.78.83.4 แบ่งจากเลขฐานสอง

$(01101110010011100101001100000100)_2$

อย่างไรก็ตามการใช้งานปกติไม่ได้ใช้อิปเพดเดรสโดยตรง แต่ระบบอินเทอร์เน็ตใช้ชื่อที่ประกอบจากตัวอักษรภาษาอังกฤษ(รหัสรหัสและสกุล) ตัวอย่างเช่น www.npu.ac.th แทนการระบุหมายเลขไอพีแต่ระบบอินเทอร์เน็ตจะแปลงชื่อเป็นไอพีโดยใช้โพรโทคอลชื่อว่า DNS ทำให้ได้ผลลัพธ์เป็นหมายเลขไอพี 110.78.83.4 สำหรับอุปกรณ์ใช้สื่อสารระหว่างกัน และอีกเรื่องที่สำคัญได้แก่ไอพียังถูกใช้ในการระบุหมายเลขไอพีเร้าเตอร์ด้วย

3.3.3 การฟอร์เวิร์ด(forward)เดต้าแกรมโดยใช้อินเทอร์เน็ตโพรโทคอล

หัวข้อต่อไปนี้กล่าวถึงวิธีที่เร้าเตอร์ฟอร์เวิร์ดเดต้าแกรม ภายในอินเทอร์เน็ตเวิร์ก ทบทวนแนวคิดในขั้นตอนการฟอร์เวิร์ดแพ็กเก็ต เริ่มต้นจากการรับแพ็กเก็ตเข้าสู่เนตเวิร์กและส่งต่อผ่านอุปกรณ์ที่เกี่ยวข้องไปจนกระทั่งถึงปลายทาง ในการส่งต่อข้อมูลนี้มีกระบวนการหนึ่งเรียกว่าการทำหนดเส้นทาง เพื่อคัดเลือกเส้นทางที่เหมาะสมให้แพ็กเก็ตเดินทางถึงปลายทางได้ กระบวนการกำหนดเส้นทางจะกล่าวถึงในหัวข้อต่อไป

ประเด็นสำคัญที่จะกล่าวถึงขั้นตอนการฟอร์เวิร์ดอินเทอร์เน็ตโพรโทคอลเดต้าแกรมมีดังต่อไปนี้

- ทุกอินเทอร์เน็ตโพรโทคอลจะมีอิปเพดเดรสของโหนดปลายทางบรรจุในไอพีเยดเดอร์
- ในส่วน IP ที่ใช้ส่วนเนตเวิร์กจะเป็นไอพีที่ไม่ซ้ำกันกับทุก IP เนตเวิร์กที่อยู่ในระบบเครือข่ายอินเทอร์เน็ต
- ทุกโไฮสต์และเร้าเตอร์ใช้อิปที่มีค่าเนตเวิร์กไอพีเดียวกันเพื่อเป็นการสื่อสารกันได้โดยตรง
- ทุกการเชื่อมต่อทางกายภาพซึ่งเป็นส่วนหนึ่งของอินเทอร์เน็ตจะมีเร้าเตอร์อย่างน้อยหนึ่งตัวเป็นอุปกรณ์สำหรับเชื่อมต่อ เร้าเตอร์นี้ทำหน้าที่สลับข้อมูลระหว่างเครือข่ายที่ใช้เทคโนโลยีแตกต่างกัน

เมื่อแพ็กเก็ตออกจากเครื่องต้นทาง มีการกำหนดให้เป็นแพ็กเก็ตชนิดไอพีเดต้าแกรม แพ็กเก็ตนี้จะส่งต่อผ่านเร้าเตอร์หลายตัวโดยที่เร้าเตอร์แต่ละตัวมีการเชื่อมต่อ กับโไฮสต์ที่อยู่ในเครือข่ายเดียวกันดังนั้นก่อนที่แพ็กเก็ตจะถูกส่งไปเร้าเตอร์ โไฮสต์ต้นทางตรวจสอบปลายทางของแพ็กเก็ตนั้นอยู่ในเครือข่ายเดียวกัน หากพบว่าปลายทางมีอิปอยู่ในเครือข่ายเดียวกันจะส่งแพ็กเก็ตตรงไปทางเครือข่ายปลายทาง แต่เมื่อพบว่าปลายทางอยู่ภายนอกเครือข่าย ทราบโดยตรวจสอบจากหมายเลขไอพีเนตเวิร์กปลายทางไม่ตรงกับไอพีเนตเวิร์กต้นทาง จึงส่งแพ็กเก็ตไปเร้าเตอร์ และ เมื่อเร้าเตอร์ได้รับข้อมูลจะเริ่มขั้นตอน เช่นเดียวกับโไฮสต์ โดยการตรวจสอบหมายเลขไอพีปลายทางหากอยู่ในเครือข่ายเดียวกันจะส่งข้อมูลไปยังเครื่องนั้นโดยตรง แต่เมื่อพบว่าไม่อยู่ในเครือข่ายเดียวกันจะส่งแพ็กเก็ตไปเร้าเตอร์ และเร้าเตอร์เครื่องต่อไปจะทำขั้นตอนเช่นนี้ต่อไปจนกว่าแพ็กเก็ตเดินทางถึงปลายทาง หรือจนกว่า TTL=0

ยกตัวอย่างจากรูปที่ 3.17 เมื่อ H1 ต้องการส่งข้อมูลไป H2 ถือว่าสามารถส่งได้โดยตรง เพราะอยู่ในเนตเวิร์กเดียวกัน H1 สามารถติดต่อกับ H2 โดยใช้อิเนตเวอร์เน็ตแล็ปเดรส ตามเทคโนโลยีอิเนตเวอร์เน็ต อีกตัวอย่าง เช่น H4 ต้องการติดต่อ H5 จะสื่อสารผ่านเทคโนโลยีแลนแล้วสายสามารถติดต่อกันได้โดยใช้แลนแล้วสายแลดเดรส ตามที่เทคโนโลยีแลนแล้วสายได้กำหนดไว้ แต่เมื่อไรที่ H1 ต้องการติดต่อ H4 จะพบว่าทั้งสองโไฮสต์ไม่ได้อยู่ในเนตเวิร์กเดียวกันทำให้ต้องสื่อสารผ่านเร้าเตอร์

ในการเลือกเส้นทางของเร้าเตอร์ยังใช้แนวคิดการบันทึกข้อมูลในตาราง ตัวอย่างตารางกำหนดเส้นทางของ R2 แสดงในตารางที่ 3.8 เมื่อต้องการส่งแพ็กเก็ตจาก H5 ไป H8 เริ่มต้นจาก H5 เห็นได้ว่า H5 ไม่

สามารถส่งข้อมูลตรงถึง H8 เพราะไม่อยู่ในเครือข่ายเดียวกัน จึงส่งข้อมูลไปเร้าเตอร์ที่ถูกเลือกให้เป็นเกตเวย์ ในที่นี้ได้แก่ R1 เมื่อ R1 ได้รับข้อมูลแล้วตรวจสอบไอพีปลายทางพบว่าไม่อยู่ในเครือข่ายเดียวกันจึงต้องส่งต่อ R1 จะส่งต่อไปเกตเวย์ที่ผู้ดูแลระบบได้เซ็ตค่าเอาไว้ ในที่นี้คือ R2 สมมุติตารางฟอร์เวิร์ด R2 เป็นไปตามตารางที่ 3.8 R2 จะค้นหาเส้นทางส่งไป H8 (อยู่ใน network-4) จากตารางดังกล่าว เห็นได้ว่าการติดต่อ network-4 จะต้องส่งผ่าน R3 จากที่ R3 อยู่ในเน็ตเวิร์กวงเดียวกันจึงเพิ่มข้อมูลลงตารางฟอร์เวิร์ดเป็นไปตามตารางที่ 3.9

ตารางที่ 3.8: ตารางฟอร์เวิร์ด R2

NetworkNum	NextHop
1	R1
4	R3

ตารางที่ 3.9: ตารางฟอร์เวิร์ดสมบูรณ์ของ R2

NetworkNum	NextHop
1	R1
2	Interface 1
3	Interface 0
4	R3

เห็นได้ว่าสำหรับทุกเน็ตเวิร์ก ที่ส่งไป R2 จะรู้ว่าต้องส่งข้อมูลไปทางใดผ่านทางตารางฟอร์เวิร์ด และ R2 ทราบว่าเครื่องใดสามารถติดต่อได้โดยตรง จากตารางที่ 3.9 พบว่าติดต่อ network-2 และ network-3 ได้โดยตรง

เมื่อ H5 ต้องการส่งข้อมูลไป H8 จากที่ทั้งสองไฮสตรีมไม่ได้เชื่อมต่อกันโดยตรง จึงต้องส่งข้อมูลผ่านเร้าเตอร์ ที่กำหนดค่าเร้าเตอร์เริ่มต้นไว้(ในเครื่อง H5) H5 ส่งข้อมูลไปยัง R1 ซึ่งเป็นค่าเริ่มต้นที่ H5 กำหนดไว้แต่แรก และมีอยู่เพียงตัวเลือกเดียว เรียกว่า “default router” เมื่อ H1 ส่งข้อมูลไปยัง R1 ในขั้นตอนที่ไปพาหนะนั่ง จะให้พิจารณาเพื่อตรวจสอบว่ามี IP อยู่ในเครือข่ายวงเดียวกันหรือไม่ หากการตรวจสอบพบว่า IP ปลายทาง เป็น H8 จึงไม่อยู่ในเน็ตเวิร์กวงเดียวกัน ดังนั้น R2 จึงส่งข้อมูลไปเร้าเตอร์ที่กำหนดไว้ใน ฟอร์เวิร์ดดิงเทเบิล ในที่นี้ได้แก่ R2 เมื่อ R2 ได้รับข้อมูลจะอ่านเขตเดอร์เพื่อดูค่าและเดรสนปัลยาทาง และตรวจสอบว่าอยู่ในเครือข่ายเดียวกันหรือไม่ หากว่าปลายทางไม่ได้อยู่ในเครือข่ายเดียวกันจึงส่งข้อมูลต่อไปยัง R3 เมื่อ R3 ได้รับข้อมูลตรวจสอบและเดรสนปัลยาทางพบว่าเป็น H8 ซึ่งอยู่ในเครือข่ายเดียวกันทำให้ R3 สามารถส่งต่อข้อมูลไปยัง H8 ได้โดยตรง

3.3.4 การแบ่งไอพีเป็นเน็ตเวิร์กย่อย(subnetting) และ ไอพีแอดเดรสแบบไม่ระบุคลาส

จากที่กล่าวมาข้างต้นมีการแบ่งไอพีออกเป็นสามกลุ่มได้แก่คลาส A คลาส B และ คลาส C การแบ่งเป็นกลุ่มนี้มีขนาดใหญ่สามกลุ่มนี้อาจสร้างปัญหาได้ ยกตัวอย่างเช่นในเครือข่ายที่ต้องการใช้งานไอพีจำนวน 255 แอดเดรส จะไม่สามารถใช้เน็ตเวิร์กคลาส C ได้เพราะคลาส C มี IP ได้สูงสุด 254 แอดเดรส แต่มีอย่างคลาส A ไปเป็นเครือ

ข่ายคลาส B จะทำให้มี การใช้งานเครือข่ายขนาดใหญ่เกินความจำเป็น เพราะคลาส B มีแอดเดรสได้สูงสุด 2¹⁴ (16,000 แอดเดรส)

ปัจจุบันจึงยอมรับให้มีการแบ่งไอพีเนตเวิร์กขนาดย่อยเพิ่มเติมจากคลาสหลักๆทั้งสามคลาส เรียกว่า “การแบ่งไอพีเป็นเนตเวิร์กย่อย” หรือเรียกสั้นๆว่า การแบ่ง“ซับเน็ต” การแบ่งซับเน็ต คือการใช้เนตเวิร์กที่มีอยู่ แบ่งออกเป็นเนตเวิร์กย่อยลงไป

ถึงแม้ว่าจะเนตเวิร์กถูกแบ่งเป็นซับเน็ต แต่การค้นหาเส้นทางของเร้าเตอร์ยังคงอ่านข้อมูลจากส่วน เขตเดอร์ที่เป็นเนตเวิร์กใหญ่ แล้วส่งข้อมูลไปยังเร้าเตอร์ที่ดูแลเนตเวิร์กนั้นแล้วหลังจากนั้นเนตเวิร์กที่ถูกแบ่ง เป็นซับเน็ตจะส่งต่อไปยังซับเน็ตย่อยต่อไป

การแบ่งไอพีเป็นเนตเวิร์กย่อยเป็นวิธีแบ่งเนตเวิร์กคลาสปกติให้มีขนาดเล็กลง ในกรอบแบบเพื่อ แบ่งซับเน็ตจะมีการเลือกบิตที่ทำหน้าที่เป็นเนตเวิร์ก และบิตที่ทำหน้าที่เป็นแอดเดรส โดยแบ่งจากทั้งหมด 32 บิต หากแบ่งจำนวนบิตให้เนตเวิร์กมาก จะทำให้เหลือจำนวนบิตสำหรับแอดเดรสสนับสนุน

สำหรับการเลือกบิตที่จะเป็นส่วนระบุเนตเวิร์ก กำหนดให้แทนด้วยบิต = 1 และ บิต = 0 ใช้แทนบิต แอดเดรสตัวอย่างเช่น มีไอพี

$$(01101110.01001110.01010011.00000100)_2$$

เมื่อทำการแบ่งไอพีเป็นเนตเวิร์กย่อยจะมีการระบุช่วงของบิตที่ทำหน้าที่เป็นเนตเวิร์ก ตัวอย่างเช่น ต้องการให้เนตเวิร์กมีหมายเลขไอพีไม่เกิน 254 แอดเดรส ดังนั้นจึงกำหนดให้มีเนตเวิร์กบิตเท่ากับ 24 บิต หรือ เขียนได้เป็น

$$(11111111.11111111.11111111.00000000)_2$$

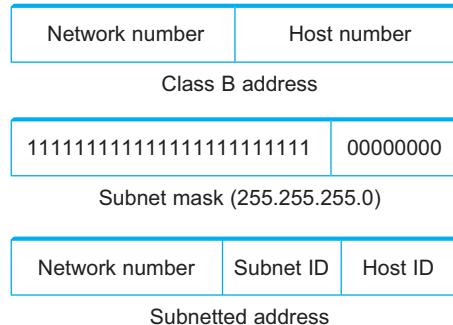
เรียกหมายเลขในนารีที่ใช้กำหนดเนตเวิร์กนี้ว่า ซับเน็ตมาสก์ จากเลขในนารีทั้งสองชุด ในการใช้งานระบบไอพี แบบมีซับเน็ต จะมีหมายเลขไอพีและซับเน็ตมาสก์ใช้งานร่วมกันเสมอ ในที่ที่นี้

$$\text{IP} = (01101110.01001110.01010011.00000100)_2 = 110.78.83.4$$

$$\text{subnet mask} = (11111111.11111111.11111111.00000000)_2 = 255.255.255.0$$

ในการส่งข้อมูลแบบการแบ่งไอพีเป็นเนตเวิร์กย่อย ใช้ข้อมูลไอพี ส่งส่วนส่วนแรกได้แก่ไอพีแอดเดรส ส่วนที่สองได้แก่ไอพีซับเน็ตมาสก์ตามที่อธิบายในรูปที่ 3.26 ใน การแบ่งไอพีแอดเดรส กำหนดให้บิต 1 ที่ปรากฏ อยู่ในไอพีซับเน็ตมาสก์เป็นตัวระบุไอพีของเนตเวิร์กและส่วนที่เป็นบิต 0 จะใช้กำหนดไอพีแอดเดรสโดยส่วนต์ จาก รูปที่ 3.26 ค่า IP เริ่มต้นเป็นค่า IP ที่อยู่ในคลาส B เมื่อต้องการแบ่งซับเน็ต ทำได้โดยเปลี่ยนบิตค่า 1 จำนวน เนตเวิร์กบิต จากเดิมที่มีบิต 1 อยู่ จำนวน 16 บิต เป็น 24 บิต เมื่อแปลงค่าซับเน็ตมาสก์เป็นเลขฐานสิบจะ ได้ 255.255.255.0 และทำให้เกิดสำเร็จได้เพิ่มขึ้นจากเดิมอีก 8 บิต เมื่อทำการแบ่งซักเม็ดแล้วจะทำให้หนึ่ง

เน็ตเวิร์คของไอทีที่เกิดจากการแบ่งชั้บเดэмีจำนวนเล็กลงจากเดิมที่มีขนาด 2^{14} 16,000 โไฮสต์ทำให้เหลือเพียง 254 โไฮสต์



รูปที่ 3.26: ชั้บเน็ตมาสก์ไอพีคลาส B
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ในการตรวจสอบเน็ตเวิร์คไอพีของเร้าเตอร์ มีขั้นตอนเพียงแค่ไอพีแอดเดรสมา AND กับไอพีชั้บเน็ต มาสก์ ทำให้ได้เน็ตเวิร์คแอดเดรส ตัวอย่างจากรูปที่ 3.27 โไฮสต์ H1 มีไอพีแอดเดรส 128.96.34.15 และมีชั้บเน็ตมาสก์ 255.255.255.128 (ทุกโไฮสต์ที่มีเน็ตเวิร์คไอพีเดียวกันหมายถึงเป็นโไฮสต์อยู่ในวงเดียวกัน สามารถสื่อสารกันได้โดยตรง) ดังนั้นในกรณีนี้ $128.96.34.15 \text{ AND } 255.255.255.128 = 128.96.34.0$ ถือว่าอยู่ใน เน็ตเวิร์ค เดียวกับ $128.96.34.1 \text{ AND } 255.255.255.128 = 128.96.34.0$ และเมื่อนำไอพีปลายทางหมายเลข 128.96.34.130 AND 255.255.255.128

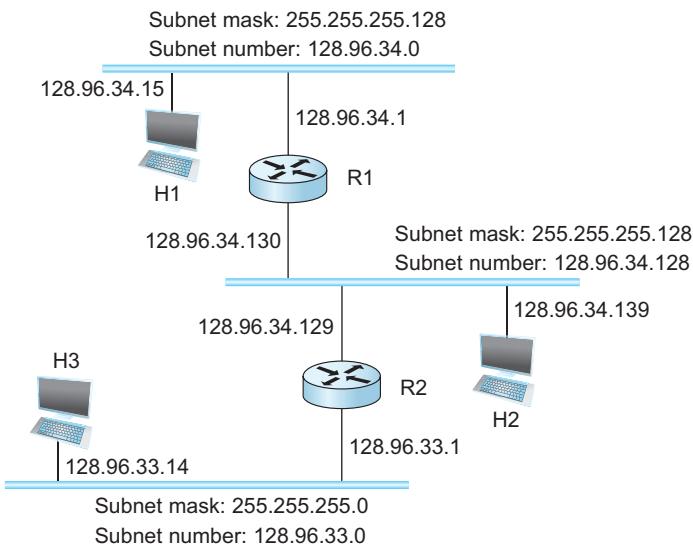
$$\begin{aligned} & 10000000.01100000.00100010.10000010 \text{ AND} \\ & 11111111.11111111.11111111.10000000 = \\ & 10000000.01100000.00100010.10000000 = \\ & 128.96.34.128 \end{aligned}$$

มาทดสอบจะพบว่ามีค่าเน็ตเวิร์คไอพีแอดเดรสไม่ตรงกันจึงถือว่าเป็น IP ภายนอกเครือข่ายที่ต้องส่ง ข้อมูลผ่านเร้าเตอร์

ฟอร์เวิร์ดดิงเทเบิล ของเร้าเตอร์จะมีการเปลี่ยนแปลงเมื่อมีการเปลี่ยนชั้บเน็ตจากที่ผ่านมา ฟอร์เวิร์ด ดิงเทเบิลประกอบด้วยข้อมูล (networkNum, NextHop) เพื่อให้ระบบรองรับกับชั้บเน็ตมาสก์ จึงเพิ่มข้อมูล เป็น (SubnetNumber, SubnetMask, NextHop) ตามตารางที่ 3.10

ตารางที่ 3.10: ฟอร์เวิร์ดดิงเทเบิลที่แบ่งชั้บเน็ต

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2



รูปที่ 3.27: เครือข่ายมีเร้าเตอร์จำนวนสองเครื่องมีสามเครือข่าย
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ต่อมาเมื่อมีข้อมูลส่งออกจาก H1 ไป H2 เมื่อ R1 ได้รับข้อมูลจะตรวจสอบโดยใช้หมายเลขไอพีปลายทางที่อ่านได้จากเซดเดอร์ ทำการ AND กับ SubnetMask และตรวจสอบว่าผลลัพธ์ตรงกับตารางแล้วได้จึงส่งไปตาม NextHop ที่ปรากฏในตารางนั้น ขั้นตอนวิธีการทำงานในโค้ดต่อไปนี้

```
D = destination IP address
for each forwarding table entry (SubnetNumber, SubnetMask, NextHop)
    D1 = SubnetMask & D
    if D1 = SubnetNumber
        if NextHop is an interface
            deliver datagram directly to destination
        else
            deliver datagram to NextHop (a router)
```

จากที่กล่าวมาเป็นตัวอย่างที่เร้าเตอร์ เลือกเส้นทางส่งข้อมูลด้วยการทำ AND กับเบื้องปลายทาง สำหรับองค์กรที่มีไฟจำนวนมากอาจจะมีเร้าเตอร์เพียงตัวเดียวที่ทำหน้าที่เป็นเกตเวย์ขององค์กร หากเร้าเตอร์นี้ต้องทำการประมวลผล AND สำหรับทุกแพ็คเก็ตเป็นการสื่อสารที่ขาดประสิทธิภาพ ตัวอย่างเช่นรูปที่ 3.27 แทนที่ฟอร์เวิร์ดติงเทเบิลของเร้าเตอร์เกตเวย์จะมีรายการซับเน็ตครบทุกซับเน็ต สามารถใช้เนตเวิร์ก 128.96 แทนเครือข่ายทั้งหมดได้ ช่วยให้การค้นหาข้อมูลซับเน็ตจากฟอร์เวิร์ดติงเทเบิลมีประสิทธิภาพขึ้น

Classless Addressing

การแบ่งซับเน็ตต้องการข้อมูลคู่กับหมายเลขไอพี บางครั้งเรียกว่า “supernetting” แต่ส่วนใหญ่จะเรียกว่า “CIDR(classless interdomain routing)” ออกเสียงว่า “ไซเดอร์”

CIDR พัฒนาเพื่อปี คศ.1990 เพื่อใช้เป็นรูปแบบมาตรฐานในการเลือกเส้นทางในระบบอินเทอร์เน็ต ก่อนปรับใช้CIDR ระบบอินเทอร์เน็ตใช้วิธีจัดกลุ่มไอพีเป็น 3 คลาสได้แก่คลาส A B และคลาส C ทำให้เกิดการจัดสรรกลุ่มไอพีได้ไม่เป็นประสิทธิภาพเช่นเครือข่ายที่มีไอพีคลาส A ซึ่งเป็นคลาสที่มีไอพีแอดเดรสขนาดใหญ่ จะต้องให้เราเตอร์เพียงตัวเดียวในการรับข้อมูลทั้งเครือข่าย ขณะเดียวกันลักษณะแบบไอพีเป็นชั้บเน็ตก็จะทำให้บริหารแบบร่วมศูนย์ เช่นองค์กรมีหน่วยงานภายในหลายหน่วยงานจะทำได้ไม่สะดวก เพราะเมื่อทำชั้บเน็ตไปแล้วจะต้องมีเราเตอร์ทำงานที่ดูแลชั้บเน็ตนั้น

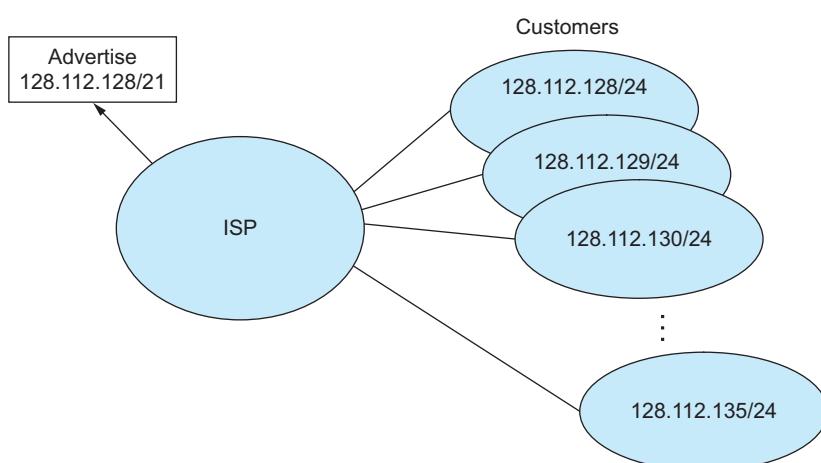
แนวคิดCIDRออกแบบให้มี subnetting เพื่อใช้ระบุเนตเวิร์กไอพีในระดับย่อยลงมาได้ ทำให้มีเนตเวิร์กไอพีจำนวนมากกว่าการแบ่งคลาสในรูปแบบเดิม รูปแบบในการระบุไอพีแบบCIDR มีดังนี้

192.168.12.0/23

จากตัวอย่างเป็นไอพีที่มีการกำหนดให้จำนวนบิตสำหรับระบุเนตเวิร์กไอพีนับจากซ้ายมีจำนวน 23 บิต หากต้องการแบ่งเนตเวิร์กเป็นชั้บเน็ตย่อยอีก 2 เนตเวิร์กจะแบ่งได้ดังนี้

192.168.12.0/23 = 192.168.12.0/24 + 192.168.13.0/24

ยกตัวอย่างการแบ่งชั้บเน็ตด้วยCIDR จากรูปที่ 3.28 สมมติผู้ให้บริการอินเทอร์เน็ต มีไอพีกลุ่มใหญ่ เป็น 128.112.128/21 ต้องการแจกจ่ายไอพีให้ลูกค้าเป็นกลุ่ม กลุ่มละ 254 ไอพี จึงทำให้มีจำนวนบิตในฝั่ง แอดเดรส เป็น 8 บิต ($2^8 = 256$ ยกเว้น 2 ไอพีสำหรับ ใช้ระบุเครือข่าย และ บรรดาสถาน) เมื่อผู้ให้บริการมีชั้บเน็ต 21 บิต และต้องการไอพีสำหรับแต่ละกลุ่มเป็น 8 บิต ดังนั้นเนตเวิร์กใหม่ที่แบ่งได้ โดยคำนวนจำนวนบิตที่ผู้ให้บริการสามารถแบ่งได้เท่ากับ $32-21 = 11$ บิต และต้องแบ่งให้เป็นบิตสำหรับแอดเดรส 8 บิต ทำให้เหลือบิตสำหรับเนตเวิร์กแอดเดรส เท่ากับ $11-8 = 3$ บิต หรือจะมีจำนวนเนตเวิร์กที่แบ่งชั้บเน็ตแล้วได้ทั้งหมด $2^3 = 8$ เนตเวิร์ก ซึ่งอธิบายตามรูปที่ 3.28



รูปที่ 3.28: การแบ่งชั้บเน็ต /21 เป็น /24
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

Address Translation(ARP)

จากหัวข้อที่ผ่านมาได้กล่าวถึงการค้นหาเส้นทางและส่งต่อข้อมูลผ่านเราเตอร์ ซึ่งขั้นตอนสุดท้ายมีการส่งข้อมูลโดยตรงระหว่างไฮสต์กับไฮสต์ ซึ่งอยู่ในเนตเวิร์กเดียวกัน จากที่ผ่านมาจึงไม่กล่าวถึงวิธีการที่เราเตอร์หรือไฮสต์ใช้ค้นหาและตรวจสอบภายใน IP แต่เมื่อต้องการส่งข้อมูลไปทางเครื่องปลายทางโดยใช้เทคโนโลยีชั้นลิ้งก์ จึงต้องมีกระบวนการในการค้นหาและตรวจสอบในชั้นลิ้งก์ กระบวนการในการสอบถามเพื่อให้ได้แอดдресในชั้นลิ้งก์เรียกว่า ARP

ARP เป็นโปรโตคอลที่ออกแบบสำหรับการแปลงไอพี ซึ่งเป็น MAC แอดเดรสของ/Card เครือข่ายในระบบอินเทอร์เน็ต ค่า MAC แอดเดรสจะใช้ระบุอุปกรณ์ที่อยู่ในเนตเวิร์ก เดียวกันแล้วส่งข้อมูลผ่านเทคโนโลยี-เครือข่ายนั้นมา ไฮสต์ได้โดยตรง อธิบายวิธีการทำงานของแม่กับเด็กได้ทั้งนี้ เมื่อไฟสต์ได้รับ IP ปลายทางจะทำการตรวจสอบว่าในเครือข่ายนั้นมีไฮสต์ใดมีไอพีแอดเดรสตรงกับไอพีปลายทาง โดยข้อมูลมีการสร้างรูปแบบแพ็กเก็ตเดียวที่เป็นไปตามรูปแบบ协议 ARP ซึ่งภายในไฟล์ มีไอพีแอดเดรสที่ต้องการทราบ MAC แอดเดรส และส่งไปให้ทุกเครื่องในเครือข่ายได้โดยกำหนดให้ ARP แพ็กเก็ตมีค่า MAC แอดเดรสปลายทางเป็น FF:FF:FF:FF:FF:FF รูปแบบ protocol ARP เป็นตามรูปที่ 3.29

0	8	16	31		
Hardware type = 1		ProtocolType = 0x0800			
HLen = 48	PLen = 32	Operation			
SourceHardwareAddr (bytes 0–3)					
SourceHardwareAddr (bytes 4–5)		SourceProtocolAddr (bytes 0–1)			
SourceProtocolAddr (bytes 2–3)		TargetHardwareAddr (bytes 0–1)			
TargetHardwareAddr (bytes 2–5)					
TargetProtocolAddr (bytes 0–3)					

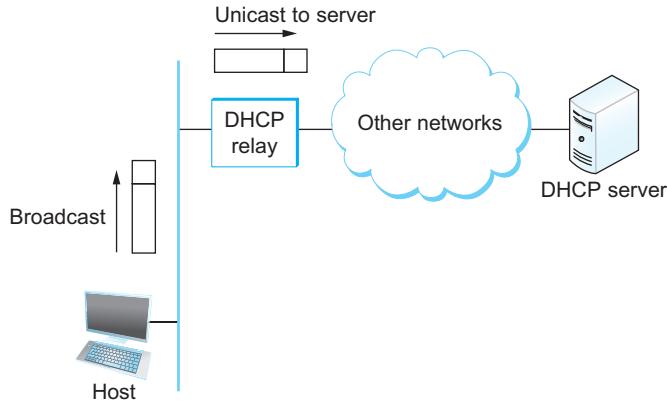
รูปที่ 3.29: โครงสร้าง protocol ARP
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

จากรูปเป็นรูปแบบ protocol การสื่อสารแบบอินเทอร์เน็ต โดยกำหนดให้มีจีบีที่กำหนดให้เป็น 224 บิต หรือ 28 ไบต์ ส่องในแรร์กใช้กำหนดชนิดของฮาร์ดแวร์ Hardware type สำหรับ

3.3.5 Host Configuration (DHCP)

ในการสื่อสารด้วยระบบอินเทอร์เน็ตต้องการสื่อสารผ่านไอพี ผู้ดูแลระบบจึงมีหน้าที่กำหนดไอพีแอดเดรสให้กับไฮสต์ ที่ต้องการเข้าสู่การสื่อสาร หน้าที่นี้สร้างภาระให้ผู้ดูแลระบบได้ เมื่อเครือข่ายมีไฮสต์จำนวนมาก ระบบอินเทอร์เน็ต ในปัจจุบันได้กำหนดให้มี protocol ที่ทำหน้าที่แจกจ่ายไอพีแอดเดรสให้กับไฮสต์แบบอัตโนมัติ มีชื่อว่า “DHCP”

DHCP เป็น protocol ทำหน้าที่เป็นโคลอนต์-เซิร์ฟเวอร์ เมื่อไฮสต์ที่ยังไม่มีไอพีแอดเดรสจะทำการสร้างแพ็กเก็ตในรูปแบบ DHCP protocol ตามการอธิบายในรูปที่ 3.31 ลำดับการสื่อสารระหว่าง โคลอนต์-เซิร์ฟเวอร์ อธิบายโดยใช้รูปที่ 3.30 เมื่อไฮสต์ยังไม่มีไอพีต้องการสื่อสารผ่านระบบอินเทอร์เน็ต จึงส่งไปยังเครื่อง



รูปที่ 3.30: โฉมสั่งบรรดาศัตรุเครื่อข่ายเพื่อขอหมายเลขไอพี
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

DHCP เชิญชวนเพื่อขอรับไอพีแอดเดรส เมื่อเชิญชวนได้รับแพ็กเก็ต จะทำการตรวจสอบในฐานข้อมูลว่ามี IP ใดที่ยังไม่แจกจ่ายบ้าง เมื่อพบไอพีแอดเดรส ว่างอยู่แล้วจะตอบกลับ พร้อมกับกำหนดค่าไอพีแอดเดรสลงในฐานข้อมูล เมื่อโฉมได้รับไอพีแล้วจะจึงกำหนดค่าไอพีแอดเดรสในเครื่องโฉม แล้วจึงเริ่มต้นเข้าสู่การสื่อสารผ่านเครือข่าย อินเทอร์เน็ต



รูปที่ 3.31: โครงสร้างโปรโตคอล DHCP
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

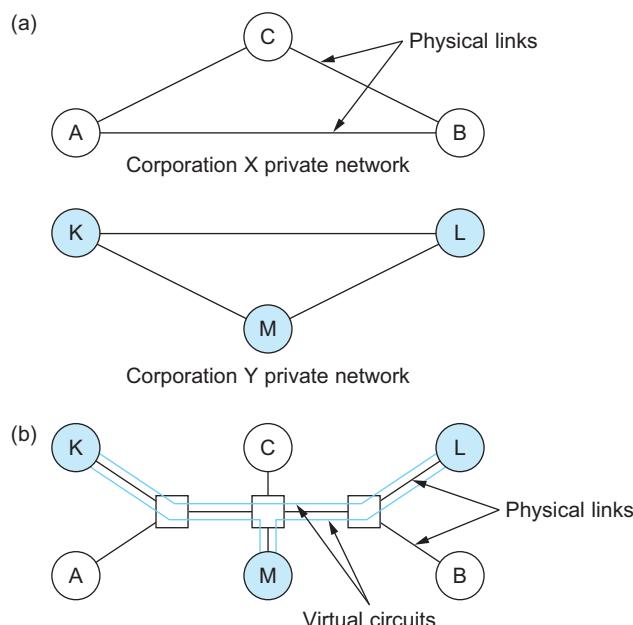
3.3.6 Error Reporting (ICMP)

ประเด็นต่อมาของการสื่อสารผ่านเครือข่ายอินเทอร์เน็ตที่มีการเลือกเส้นทางผ่านไอพีแอดเดรสนั้น มีแพ็กเก็ตที่ต้องเดินทางผ่านเร้าเตอร์จำนวนหนึ่ง ซึ่งอาจเกิดปัญหาระหว่างการเดินทางไปสู่ปลายทาง ดังนั้นจึงมีprotoคอลทำหน้าที่รายงานสถานะในการเดินทางของแพ็กเก็ต เปรียบเสมือนตำรวจจราจร เมื่อพบว่าแพ็กเก็ตไม่สามารถเดินทางถึงปลายทางได้จะส่งแพ็กเก็ต ตามprotoคอล ICMP RFC792 (1981) กลับไปบอกต้นทาง

3.3.7 Virtual Networks และ Tunnels

จากที่ผ่านมาการเชื่อมผ่านเครือข่ายอินเทอร์เน็ตที่ส่งผ่านเร้าเตอร์ต้องมีเนตเวิร์กไอพี อยู่คุณจะเห็นว่าเครือข่ายอินเทอร์เน็ตใช้วิธีตรวจสอบด้วยการตรวจสอบเนตเวิร์กไอพี หากมีบางโปรแกรมที่ต้องใช้งานผ่านการติดต่อแบบอินเทอร์เน็ตเท่านั้น จะไม่สามารถใช้การสื่อสารในรูปแบบอินเทอร์เน็ตปกติได้ ซึ่งการแก้ปัญหานี้ทำได้โดยการใช้เทคโนโลยี VPN(เครือข่ายส่วนตัวเสมือน)

เทคโนโลยี เครือข่ายส่วนตัวเสมือน ได้กำหนดรูปแบบการเชื่อมต่อผ่านเครือข่ายอินเทอร์เน็ตโดยการสร้างไอพีเสมือนเพื่อให้โอดัตตันทางกับโอดัตต์ปลายทางสามารถเชื่อมต่อได้โดยใช้เนตเวิร์กไอพีวงเดียวกัน คุณสมบัติของเครือข่ายส่วนตัวเสมือนทำให้เชื่อมเครือข่ายอินเทอร์เน็ตได้จากระยะไกล และยังทำให้อุปกรณ์เครือข่ายนั้นมองเห็นเสมือนอยู่ในเครือข่ายเดียวกัน ดังตัวอย่างรูปที่ 3.32(a) เป็นเครือข่ายมีการเชื่อมโยงหน่วยงาน A B C ผ่านเครือข่ายปกติ และมีหน่วยงาน K L M มีเครือข่ายส่วนตัวที่เชื่อมต่อกัน เมื่อต้องการให้ทั้งสองเครือข่ายแชร์มีเดียร่วมกันเพื่อประหยัดค่าใช้จ่ายในการลากสายสัญญาณใหม่ อาจใช้เทคโนโลยีจำลองเครือข่ายเพื่อให้ทั้งสองเครือข่ายยังคงทำงานเสมือนแยกจากกัน ตามอธิบายในภาพที่ 3.32(b) การแบ่งแยกทั้งสองเครือข่ายออกจากกันทั้งที่ใช้สื่อนำสัญญาณเดียวกันทำได้โดยแนวทาง เช่น กำหนดไฟล์สำหรับกำหนดโดดของเครือข่ายที่แตกต่างกัน และถอดได้ออติที่ปลายทาง เทคโนโลยีนี้มีชื่อว่า MPLS(multi-protocol label switching) หรือการสร้างไอพีทันเนล ภายใต้ทันเนลจะมีโปรโตคอลชั้นลิ้งค์อยู่ภายในอีกชั้นหนึ่ง



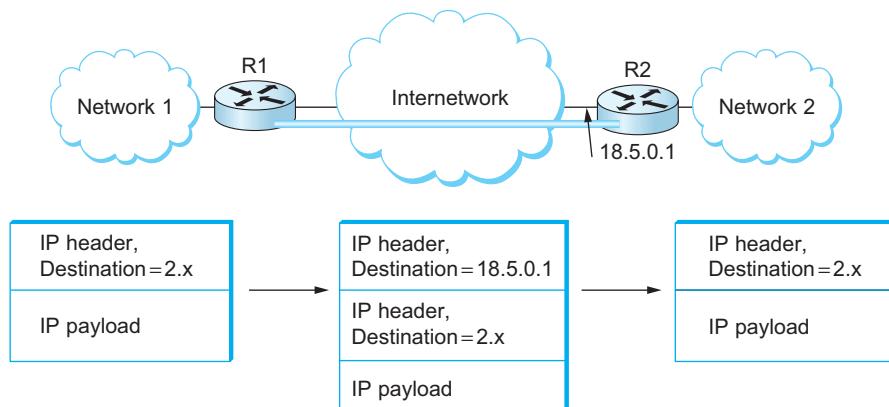
รูปที่ 3.32: ตัวอย่างเครือข่ายเครือข่ายส่วนตัวเสมือน: (a) สองเครือข่ายแยกกันทางกายภาพ (b) เครือข่ายใช้สายสัญญาณร่วมกันแต่แยกกันโดยใช้ช่วงจูรเสมือน
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

เมื่อมีการสร้างทันเนลชั้นแล้วอุปกรณ์จะส่งข้อมูลเข้าทันเนลทำได้โดยง่ายเพียงกำหนดค่าในฟอร์มรีด ดึงเทเบิลให้เข้าไปที่อินเทอร์เน็ตทันเนล ตามตัวอย่างในตารางที่ 3.11

ตารางที่ 3.11: ฟอร์มเดิมของร้าเตอร์ R1

NetworkNum	NextHop
1	Interface 0
2	Virtual Interface 0
Default	Interface 1

ในรูปที่ 3.33 เป็นตัวอย่างการเชื่อมเครือข่ายส่วนตัวเสมือนโดยใช้การสร้างทันเนลจาก R1 ไปยัง R2 โดยที่ปกติแล้ว R1 และ R2 ติดต่อ กันผ่านเครือข่ายอินเทอร์เน็ต แต่ในภาวะปกติการเชื่อมต่อจาก Network-1 ไปยัง Network-2 ต้องใช้หมายเลขไอพีในระบบอินเทอร์เน็ต(public IP) ซึ่งในกรณีที่ต้องการให้ Network-1 และ Network-2 เชื่อมต่อ กันได้เสมือนอยู่ในเครือข่ายเดียวกัน จะทำได้โดยการสร้างทันเนลสำหรับทำหน้าที่เป็นท่อนนำข้อมูลไปถึงปลายทางโดยตรง จากรูปจะเห็นได้ว่ามีการสร้างทันเนลเมืองหมายเลขไอพีในกลุ่ม 2.x ซึ่งเมื่อส่งข้อมูลออกจาก R1 และจะยังใช้ public IP ไปจนถึง R2 เมื่อ R2 ถอดเดอร์เพบเลเยอร์อีพีจึงสามารถอ่านหมายเลขไอพีกลุ่ม 2.x ได้ เสมือนว่าอยู่ในตัวเวิร์กงงานเดียวกัน



รูปที่ 3.33: การเชื่อมต่อทันเนลผ่านเครือข่าย 18.5.0.1 ทำให้ R1 และ R2 เชื่อมต่อ กันได้โดยตรงผ่านเครือข่าย 2.x
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

3.4 ขั้นตอนวิธีเลือกเส้นทาง

กำหนดเส้นทางเป็นกระบวนการที่ใช้ในการค้นหาเส้นทางที่ทำหน้าที่เดินทางจากต้นทางไปถึงปลายทาง มีประสิทธิภาพที่สุด ซึ่งค่าประสิทธิภาพที่สนใจในการค้นหาเส้นทาง เช่น ระยะทางสั้นที่สุด สื่อสารได้เร็วที่สุด มีดีเลย์น้อยที่สุด เป็นต้น ปัญหาการค้นหาเส้นทางนิยมอธิบายด้วยกราฟ ดังจะกล่าวถึงในหัวข้อต่อไป

ตารางที่ 3.12: ตัวอย่างตารางเรทติ้ง

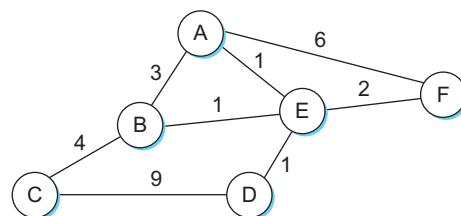
Prefix/Length	Next Hop
18/8	171.69.245.10

ตารางที่ 3.13: ตัวอย่างฟอร์เวิร์ดดิงเทเบิล

Prefix/Length	Interface	MAC Address
18/8	if0	8:0:2b:e4:b:1:2

3.4.1 อธิบายเนตเวิร์กด้วยกราฟ

การค้นหาเส้นทางเป็นปัญหาหากประเภทหนึ่งของเครือข่ายคอมพิวเตอร์ เพื่อจำกัดกรอบเฉพาะปัจจัยสำคัญจึงนิยมใช้วิธีอธิบายการเชื่อมต่อผ่านทางกราฟดังรูปที่ 3.34 ประกอบด้วยโหนดจำนวนหกแห่งได้แก่ A B C D E และ F แต่ละโหนด เชื่อมตอกันผ่านเส้นเชื่อมโดยแต่ละเส้นเชื่อมจะมีหมายเลขกำกับซึ่งหมายเลขกำกับของเส้น เชื่อมนี้ถือเป็นค่าใช้จ่ายที่ใช้ในการเดินทางไปปลายทาง

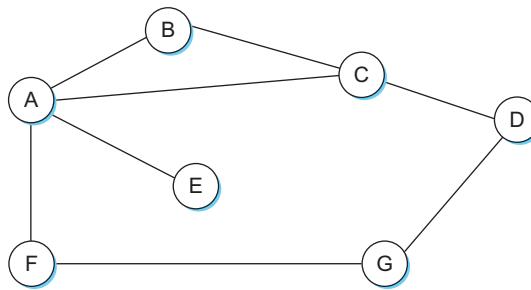


รูปที่ 3.34: เขียนการเชื่อมเครือข่ายด้วยกราฟ
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

จากราฟในรูปที่ 3.34 ไม่มีการระบุประเภทของเส้นเชื่อมหรือเทคโนโลยีสำหรับการสื่อสาร ซึ่งการใช้กราฟนี้จะสนใจวิธีการในการเลือกเส้นทางที่มีผลรวมของค่าใช้จ่ายน้อยที่สุดเพื่อเดินทางไปให้ถึงปลายทางวิธีการในการค้นหาเส้นทางสามารถจัดกลุ่มได้สองแนวได้แก่ distance vector และ link state

3.4.2 Distance-Vector (RIP)

แนวคิดการค้นหาเส้นทางแบบ DV นี้มีจุดเริ่มจากข้อเสนอของ เบลแมน-ฟอร์ด-มัว Bellman-Ford-Moore (Bang-Jensen และ Gutin, 2008, Section 2.3.4: The Bellman-Ford-Moore algorithm)(ทั้งสามท่านคิดขั้ลกอริทึมโดยไม่เข้าต่อ กัน และเสนอในเวลาไล่เลี่ยกัน จึงใช้ชื่อผู้คิดค้นทั้งสามท่านร่วมกัน เรียงตามตัวอักษร) กำหนดให้โหนด มีการเดียวกันเป็นหนึ่งมิติระยะห่างระหว่างโหนดกำหนดให้เป็น "distance" (ค่าใช้จ่าย(cost)) โหนดต้นทางสามารถส่งข้อมูลไปยังปลายทางในระยะใกล้ได้โดยฝากส่งโหนดที่ประชิดกับตนเอง ซึ่งอาจมีเส้นที่เป็นไปได้หลายเส้นทาง เป้าหมายของการค้นหาเส้นทางที่สั้นที่สุดคือเส้นทางไปถึงปลายทางที่มีค่าใช้จ่ายน้อยที่สุด ข้อตกลงตั้งต้นของ DV สมมติให้แต่ละโหนดรู้ค่าใช้จ่ายของลิงค์ที่ประชิดกับตนเองเท่านั้น สำหรับลิงค์ที่ไม่ได้ต่อโดยตรงจะกำหนดให้มีค่าเป็นอนันต์ (∞) ตัวอย่างในรูปที่ 3.35



รูปที่ 3.35: Distance-vector กำหนดเส้นทาง: ตัวอย่างเครือข่าย
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

จากรูปที่ 3.35 สามารถเขียนตารางสำหรับกำหนดค่าใช้จ่ายที่แต่ละโหนดของเห็นตามตารางที่ 3.14 โดยตารางนี้สรุปรวมข้อมูลทั้งหมดของเครือข่าย กำหนดให้โหนดที่ใช้อีมต่อ กันโดยตรงมีค่าใช้จ่ายเป็น 1 โหนดที่เชื่อมกับตนเองมีค่าเป็น 0 และถ้าสามารถเชื่อมได้โดยตรงให้มีค่าใช้จ่ายเป็น ∞

ตารางที่ 3.14: ข้อมูลเริ่มต้นในแต่ละโหนดสำหรับคำนวณ DV

	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	8	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

อธิบายตารางที่ 3.14 ได้ดังนี้ ที่โหนด A เส้นเชื่อมไป A มีค่าใช้จ่ายเป็น 0 ค่าใช้จ่ายจาก A ไป B มีค่าใช้จ่ายเป็น 1 จาก A ไป C มีค่าใช้จ่ายเป็น 1 ค่าใช้จ่ายจาก A ไป D มีค่าใช้จ่ายเป็น ∞ ค่าใช้จ่ายจาก A ไป E มีค่าใช้จ่ายเป็น 1 ค่าใช้จ่ายจาก A ไป F มีค่าใช้จ่ายเป็น 1 และค่าใช้จ่ายจาก A ไป G มีค่าใช้จ่ายเป็น 0 ทำจนครบทุกโหนดจะได้ตารางที่ 3.14 จากรูปที่ 3.35 เป็นไดกราฟ(directed graph)ทำให้ค่าใช้จ่ายเดินทางไปและเดินทางกลับมีค่าใช้จ่ายเท่ากัน

ตารางที่ 3.14 เป็นภาพใหญ่ของเครือข่าย หากมองเฉพาะจุดที่โหนด A จะมีข้อมูลตามตารางที่ 3.14

เมื่อได้ข้อมูลแล้ว ลำดับต่อมากำหนดให้แต่ละโหนดส่งแพ็กเก็ตไปบอกโหนดเพื่อนบ้าน (neighbors node) ถึงโหนดอื่นที่ตนเองเชื่อมได้โดยตรง ตัวอย่างเช่น โหนด F ส่งแพ็กเก็ตไปบอกโหนด A ว่าสามารถติดต่อกับโหนด G ด้วยค่าใช้จ่ายเป็น 1 ซึ่งหมายถึงว่า ถ้าโหนด A ต้องการติดต่อไปโหนด G จะต้องใช้ค่าใช้จ่ายเป็น 2 เพราะต้องส่งผ่าน F และ F ส่งต่อไป G ดังนั้นข้อตารางเราที่ตั้งไว้เป็นไปตามตารางที่ 3.16

เมื่อทำเช่นนี้ครบทุกโหนดจะทำให้ได้ตารางภาพรวมของเครือข่ายเป็นไปตามตารางที่ 3.17

ทำความเข้าใจการปรับค่าตาราง เมื่อมีเส้นเชื่อมขาดได้ดังนี้ ยกตัวอย่าง เมื่อโหนด F ตรวจพบลิงก์ระหว่าง F ไป G ขาด อันดับแรก F จะกำหนดค่าใช้จ่ายไป F เป็น ∞ และส่งข้อมูลบอก A จากที่ A มีข้อมูลใน

ตารางที่ 3.15: ข้อมูลตารางเรขาติ้งที่หนนด A

Destination	Cost	NextHop
B	1	B
C	1	C
D	∞	-
E	1	E
F	1	F
G	∞	-

ตารางที่ 3.16: ข้อมูลตารางเรஹติ้งที่หนนด A

Destination	Cost	NextHop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

ตารางถ้าจะส่งข้อมูลไป G มีค่าใช้จ่าย 2 หน่วย จะทำให้ A เปลี่ยนค่าใช้จ่ายระหว่าง A ไป G เป็น ∞ และในการปรับค่าตารางรอบต่อไป C ได้ส่งข้อมูลบอกว่าสามารถเชื่อมกับ G โดยมีค่าใช้จ่ายเป็น 2 ทำให้ A สามารถส่งไป G ผ่าน C โดยมีค่าใช้จ่ายเป็น 3 ซึ่งมีค่าน้อยกว่า ∞ ทำให้ A เปลี่ยนเส้นทางจาก F ไปทาง C เมื่อต้องการส่งไป G ใน การปรับค่าตารางนี้ F จะเดินทางไปทาง G ได้จะมีค่าใช้จ่ายเป็น 4

วิธี DV ใช้การกำหนดการปรับค่าอยู่ 2 ทางเลือก เมื่อไม่มีการเปลี่ยนแปลงใดๆ จะปรับค่าตารางเป็นค่าบเวลา (periodic) แต่ในกรณีมีเหตุการณ์ที่ทำให้ลิงค์เปลี่ยนแปลง เช่นเส้นเชื่อมขาดจะทำการปรับทันที โดยโหนดจะส่งแพ็กเก็ตไปยังโหนดข้างเคียง วิธีการปรับค่านี้อาจทำให้เกิดปัญหาการนับไม่รู้จบ (count to infinity problem) ตัวอย่าง เช่นโหนด A ขาดจาก E จะเกิดการอัพเพทให้ค่าใช้จ่ายจาก A ไป E เป็น ∞ แต่โหนด B และ C บอกว่ามีค่าใช้จ่ายไป E เป็น 2 และในช่วงขณะนั้น B มีข้อมูลว่าจะไป C ผ่าน E มีค่าใช้จ่ายเป็น 2 จึงส่งไปบอก A ว่า A สามารถส่งผ่าน B เพื่อไป E ได้โดยใช้ค่าใช้จ่าย 4 และถ้าส่งผ่าน C จะมีค่าใช้จ่ายเป็น 5 ขณะที่เส้นเชื่อม A-E ยังขาด ทำให้ยังมีการปรับค่าตารางไม่สิ้นสุดจึงกล้ายเป็นปัญหาการนับไม่รู้จบ

วิธีแก้ปัญหานี้มีหลายแนวทางที่เป็นไป ด้วยอย่างเช่น กำหนดจำนวนการปรับให้มีค่าใช้จ่ายไม่เกินค่าสูงสุด เช่น ค่าใช้จ่ายไม่เกิน 15 ทำให้ค่าใช้จ่ายในระบบสูงสุดเปลี่ยนจาก ∞ เป็น 16 ถ้าหากนับได้ 16 ถือว่าเป็น ∞ จะทำให้การปรับค่าตารางมีได้ไม่เกินขอบเขตและไม่เกิดปัญหาการนับไม่รู้จบ

อีกหนึ่งวิธีเรียกว่า split horizon ใช้แนวคิดต่อยอดจากการกำหนดค่าใช้จ่ายสูงสุด (ในที่นี้เป็น 16) โดยทำสวนทางจากปกติ แทนที่จะให้โหนดข้าง ๆ เป็นตัวปรับค่าตาราง ก็ให้โหนดที่เห็นว่าเส้นเชื่อมขาดนั้น

ตารางที่ 3.17: ข้อมูลเมื่อเลือกเส้นทางเสร็จสิ้น

	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	8	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

เป็นผู้ส่งแพ็กเก็ตโดยกำหนดให้เส้นเชื่อมเป็นหนทางมีค่าเป็น 16 แทน ทำให้บางครั้งเรียกว่าเป็นวิธี “poison reverse”

ในการเขียนโค้ดนั้นทำได้อย่างตรงไปตรงมา จากตัวอย่างโค้ดต่อไปนี้ อธิบายเฉพาะส่วนที่เป็นพื้นฐานของขั้นตอนการค้นหาเส้นทาง โดยกำหนดโครงสร้างข้อมูลที่จำเป็นในการใช้ค้นหาเส้นทางและกำหนดค่า MAX_TTL ไว้เป็นค่าสูงสุดของจำนวนเส้นเชื่อมก่อนลบแพ็กเก็ตทิ้ง และกำหนดให้มีค่าคงที่ MAX_ROUTES ใช้แทนค่าใช้จ่ายสูงสุด

```
#define MAX_ROUTES      128      /* maximum size of routing table */
#define MAX_TTL          120      /* time (in seconds) until route expires */

typedef struct {
    NodeAddr Destination;    /* address of destination */
    NodeAddr NextHop;        /* address of next hop */
    int      Cost;           /* distance metric */
    u_short TTL;            /* time to live */
} Route;

int      numRoutes = 0;
Route   routingTable[MAX_ROUTES];
```

สำหรับการปรับค่าตารางเป็นไปตามโค้ด mergeRoute โดยทำงานเป็นคากโดยตรวจสอบรายการในตารางพร้อมกับตรวจสอบค่า TTL กำหนดให้ค่าเริ่มต้น TTL มีค่าเท่ากับ 120 และจะลดค่าลงทุกครั้งที่ผ่านเราเทอร์ เมื่อ TTL ลดเหลือ 0 จะลบแพ็กเก็ตออกจากระบบ

```
void
mergeRoute (Route *new)
{
    int i;
```

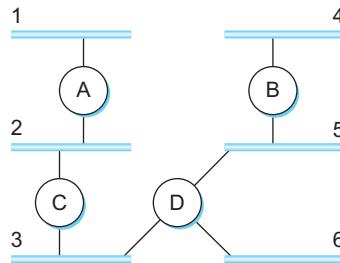
```

for (i = 0; i < numRoutes; ++i)
{
    if (new->Destination == routingTable[i].Destination)
    {
        if (new->Cost + 1 < routingTable[i].Cost)
        {
            /* found a better route: */
            break;
        } else if (new->NextHop == routingTable[i].NextHop) {
            /* metric for current next-hop may have changed: */
            break;
        } else {
            /* route is uninteresting---just ignore it */
            return;
        }
    }
    if (i == numRoutes)
    {
        /* this is a completely new route; is there room for it? */
        if (numRoutes < MAXROUTES)
        {
            ++numRoutes;
        } else {
            /* can't fit this route in table so give up */
            return;
        }
    }
    routingTable[i] = *new;
    /* reset TTL */
    routingTable[i].TTL = MAX_TTL;
    /* account for hop to get to next node */
    ++routingTable[i].Cost;
}

```

3.4.3 Routing Information Protocol (RIP)

หนึ่งในโพรโทคอล ที่นิยมใช้ในสำนักงานเพื่อค้นหาเส้นทางเครือข่ายได้แก่ Routing Information Protocol (RIP) ซึ่งเป็นโพรโทคอลทำงานโดยใช้แนวคิด DV ได้รับการติดตั้งมาพร้อมกับระบบปฏิบัติการยูนิกซ์ รุ่น Verkeley Software Distribution of Unix การค้นหาเส้นทางแบบ RIP(routing information protocol)นี้



รูปที่ 3.36: เครือข่ายที่มีสี่เร้าเตอร์บนเน็ตเวิร์ก
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

กำหนดให้จำนวนเส้นเชื่อมเป็นเงื่อนไขในการค้นหาเส้นทาง การสื่อสารที่ผ่านแต่ละเส้นเชื่อม หรือเรียกว่าระยะห่างทางเครือข่าย นั้นจะเพิ่มขึ้นเมื่อผ่านเร้าเตอร์หนึ่งเครื่องและเส้นทางที่สั้นที่สุดคือเส้นทางที่มีจำนวนเส้นเชื่อมน้อยที่สุด ยกตัวอย่างตามรูปที่ 3.36 เส้นทางจากเน็ตเวิร์ก 1 ไปหา เน็ตเวิร์ก 6 มีจำนวนลิงค์(ระยะห่างทางเครือข่าย)ทั้งหมด 4 ระยะห่างทางเครือข่าย

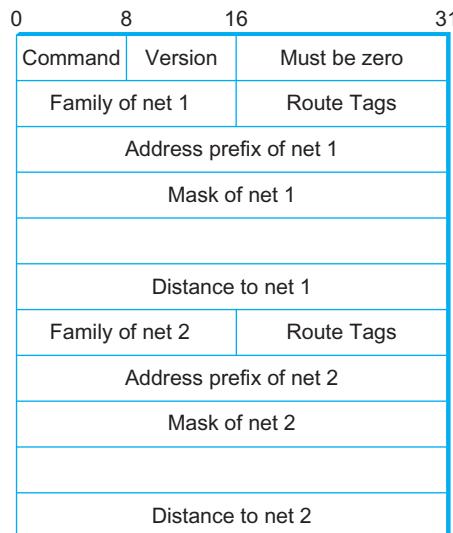
การใช้งานจริงของเร้าเตอร์นั้นมีความแตกต่างจากการวิเคราะห์ด้วยทฤษฎีกราฟ ในแง่ของการใช้งานจริงไม่จำเป็นต้องคำนึงถึงจำนวนเส้นทางที่มีอยู่ในแต่ละเส้นทาง แต่จะคำนึงถึงระยะห่างทางเครือข่ายที่ต้องผ่านเร้าเตอร์ใดบ้าง ตัวอย่างเช่น ทางจาก เน็ตเวิร์ก 1 ไป เน็ตเวิร์ก 6 จะมีเส้นทางที่ระยะห่าง 4 ระยะห่างทางเครือข่าย คือ 1->A->B->D->6 และทางจาก เน็ตเวิร์ก 1 ไป เน็ตเวิร์ก 5 จะมีเส้นทางที่ระยะห่าง 3 ระยะห่างทางเครือข่าย คือ 1->A->C->5 และทางจาก เน็ตเวิร์ก 1 ไป เน็ตเวิร์ก 4 จะมีเส้นทางที่ระยะห่าง 2 ระยะห่างทางเครือข่าย คือ 1->A->2

RIP กำหนดให้มีการปรับค่าตารางเราระยะต่อๆ กันเป็น 30 วินาที โดยจะส่งข้อมูลบอกเร้าเตอร์ที่อยู่ติดกันเป็นระยะ ปัจจุบัน RIP พัฒนาถึงเวอร์ชันที่สอง RIP version 2 (RIPv2) แพ็กเก็ต RIP มีโครงสร้างตามรูปที่ 3.37

3.4.4 Open Shortest Path First Protocol (OSPF)

การค้นหาเส้นทางแบบ การจัดเลี้นทางแบบพลวัต(link state) เสนอแนวคิดแตกต่างจาก DV จากที่ RIP มีตารางเราระยะต่อๆ กัน สำหรับกำหนดเส้นทางข้อมูลไว้ล่วงหน้าและมีการเปลี่ยนแปลงตามเวลา ขณะที่ LS ออกแบบให้เร้าเตอร์ค้นหาเส้นทางทุกครั้งที่ต้องการส่งข้อมูล เมื่อ LS ได้เส้นทางแล้วจะเรียกสถานะนั้นว่า link-state การค้นหาเส้นทางแบบ LS เหมาะสำหรับเครือข่ายที่มีการเปลี่ยนแปลงลิงก์บ่อย

อธิบายการทำงาน LS ได้ดังนี้ เมื่อเร้าเตอร์ต้องการค้นหาเส้นทางจะส่งข้อมูล广播出去ให้ทุกเร้าเตอร์ที่อยู่ติดกัน สำหรับเร้าเตอร์ที่อยู่ติดกันจะ广播ค่าสต็อปไปจนถึงปลายทาง เรียกวิธีนี้ว่า “flooding”(ผลักดัน) ข้อมูลที่บรรจุอยู่ภายใน广播ค่าสต็อปแพ็กเก็ต เรียกว่า LSP ประกอบด้วยข้อมูลต่อไปนี้



รูปที่ 3.37: โครงสร้างเฟรม RIP
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

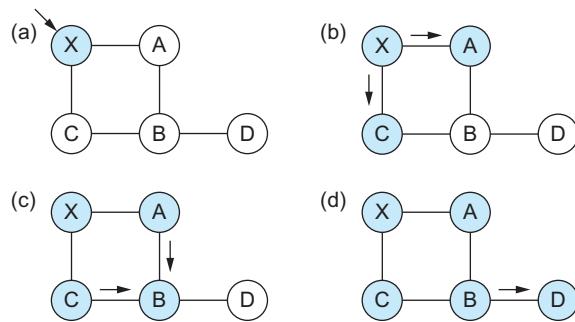
- ID ของโหนดที่สร้าง LSP
- รายการเร้าเตอร์ที่อยู่ติดกันและค่าใช้จ่ายที่เชื่อมกับเร้าเตอร์
- sequence number
- TTL ของแพ็กเก็ต

ข้อมูลสองส่วนแรกของ LSP ใช้สำหรับค้นหาเส้นทางและสองส่วนหลังใช้สำหรับரอดคาสต์ อธิบายการทำงานกระบวนการฟลัตบอร์ดคาสต์แพ็กเก็ต โดยใช้รูปที่ 3.38 พิจารณาโหนด X ได้รับข้อมูล LSP จากโหนด Y และ X ตรวจสอบว่ามี LSP ในระบบหรือยัง ถ้ายังไม่มีจะบันทึกข้อมูลไว้ในตัวเอง การตรวจสอบว่าเคยมีข้อมูลหรือไม่นั้นตรวจสอบจาก sequence number แต่เมื่อตรวจสอบแล้วเป็นข้อมูลใหม่ที่ไม่เคยบันทึกมากกว่า(ข้อมูลใหม่มีค่า sequence number สูงกว่าข้อมูลเดิม) จะนำค่าที่ได้ใหม่ทดแทนค่าเดิม เสร็จแล้ว X

เริ่มจากรูปที่ 3.38(a) อธิบายขั้นตอนการฟลัต เริ่มจาก X ได้รับ LSP หลังจากตรวจสอบ sequence number แล้วจะกระจายข้อมูลไปทุกเร้าเตอร์ที่อยู่ข้างๆ ยกเว้นเร้าเตอร์ที่ส่งให้ X ในที่นี้ได้แก่ A และ C เป็นไปตามรูปที่ 3.38(b) เมื่อ A และ C ได้รับ LSP จะทำวิธีเดียวกับ X ได้แก่ตรวจสอบ sequence number แล้วส่ง LSP ไป B ตามรูปที่ 3.38(c) เห็นได้ว่า B ได้รับ LSP จำนวนสองแพ็กเก็ตแต่มี sequence number ซ้ำกันจึงลบ LSP ที่ซ้ำกันและส่งต่อไป B เพียงแพ็กเก็ตเดียว ตามรูปที่ 3.38(d)

การคำนวนเส้นทาง

การคำนวนเส้นทางเป็นไปได้สองแนวทาง ได้แก่ หาเส้นทางໄວ่ล่วงหน้า และ หาเส้นทางให้ทุกครั้งเมื่อต้องการส่ง ซึ่งได้แบ่งเป็น 2 วิธี วิธี DV เป็นการคำนวนเวลาตามเวลา ขณะที่ LS คำนวนเส้นทางเมื่อต้องการส่งข้อมูล



รูปที่ 3.38: การฟลั๊ดใน LS: (a) LSP เข้าโนนด X; (b) X ฟลั๊ด LSP ไป A และ C; (c) A และ C ฟลั๊ดไป B แต่ไม่ส่งไป X; (d) ฟลั๊ดครบทุกโนนด

กระบวนการหาเส้นทางจะเป็นไปตามอัลกอริทึมพื้นฐาน หรือเรียกว่า State-of-the-art⁴ ซึ่งแบบนิยมมี 2 รูปแบบได้แก่ (Bang-Jensen และ Gutin, 2008, Section 2.3.4: The Bellman-Ford-Moore algorithm) และ ขั้นตอนวิธีของไดก์สตรา(dijkstra algorithm)(Cormen และคณะ, 2001, Section 24.3: Dijkstra's algorithm) เรียงตามลำดับ

อธิบายขั้นตอนการหาเส้นทางโดยใช้ข้อมูลตัวอย่างในรูปที่ 3.39 สมมติให้มีระบบส่งแพ็กเก็ต LSP ไปทุกโนนด กำหนดให้ $l(i,j)$ ใช้อังถึงข้อมูลที่ที่โนนด i ส่งไปโนนด j ด้วยค่าค่าใช้จ่าย $l(i,j)$ กำหนดให้ $l(i,j)=\infty$ หมายถึงไม่มีเส้นเชื่อมเข้ามายอดจากโนนด i ไปโนนด j

ต่อไปกล่าวถึงขั้นตอนการค้นหาเส้นทางแบบขั้นตอนวิธีของไดก์สตรา เป็นอัลกอริทึมที่เป็นที่รู้จักทั่วไปในการค้นหาเส้นทางสั้นที่สุด ในการค้นหาเส้นทางจะเป็นเส้นทางที่โนนดจำนวน N โนนด เดินทางไปโนนด $N-1$ โนนด

อธิบายขั้นตอนวิธีขั้นตอนวิธีของไดก์สตราจากโค้ดด้านล่าง ได้ดังนี้ เริ่มต้นแทนค่า M ด้วยโนนดเริ่มต้น ในที่นี้คือ s วนลูปทุกโนนด ทุกโนนดทราบข้อมูลค่าใช้จ่ายของแต่ละโนนด โดยกำหนดค่าใช้จ่าย ที่โนนด s เดินทางไปโนนด g ใดๆ โดยใช้เส้นทาง w ($C(w)$) และเลือกเส้นทาง w ที่มีค่าใช้จ่ายรวมน้อยที่สุด และปรับปรุงค่าในตารางเรขาที่ g วนทำเช่นนี้จนกระทั่งครบทุกโนนด

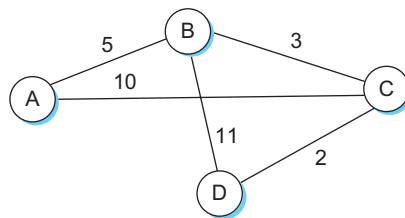
```

M = {s}
for each n in N - {s}
    C(n) = l(s,n)
while (N != M)
    M = M + {w} such that C(w) is the minimum for all w in (N-M)
    for each n in (N-M)
        C(n) = MIN(C(n), C(w)+l(w,n))

```

จากรูปที่ 3.39 กำหนดตัวอย่างกราฟที่มีสี่โนนดมี 5 เส้นเชื่อม มีเส้นทางไปแต่ละโนนดหลายเส้นทาง เมื่อถูกที่โนนด D มีลำดับการทำงานตามตารางที่ 3.18

⁴รูปแบบ(ที่ดีที่สุด)ใช้เป็นตัวเปรียบเทียบวิธีอื่น



รูปที่ 3.39: เครื่อข่ายตัวอย่างการทำงานแบบ LS
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

การค้นหาเล่นแบบRLมีคุณสมบัติน่าสนใจ ทำให้เห็นได้ว่าทำให้ระบบมีความเสถียรและสื่อสารข้อมูลบนเครือข่ายไม่มาก

การค้นหาเส้นทางด้วย OSPF ใช้แนวทางแบบ TOT ซึ่งไม่ได้เตรียมเส้นทางไว้ล่วงหน้า ซึ่งหมายความว่ากับเครือข่ายขนาดใหญ่ขึ้น เพื่อลดปริมาณแพ็กเก็ตที่ใช้สำหรับค้นหาลง โพร์โทคอลแบบ OSPF มีโครงสร้างแพ็กเก็ตเดียวกันเป็นมาตรฐานที่ [3.40](#) สำหรับ Version มีค่าเป็น 2 และ Type มีค่าอยู่ระหว่าง 1 ถึง 5 สำหรับ SourceAddr ใช้กำหนดโดยสัตต้นทาง และ Areald มีขนาด 32-บิต ใช้กำหนดพื้นที่ที่รับผิดชอบ Checksum ใช้เก็บข้อมูล checksum สำหรับตรวจสอบด้านความสมบูรณ์ของเมตadata และ Authentication ใช้ในด้านความปลอดภัย

แพ็คเกจมีทั้งหมด 5 ประเภท เรียกว่า type เพื่อใช้เป็นข้อความ “hello” โดยที่เราเตอร์ จะส่งให้เราเตอร์ข้างเคียงเพื่อบอกว่าต้นเองยังพร้อมทำงาน ส่วน type ที่เหลือใช้เพื่อ request send และ acknowledge

0	8	16	31
Version	Type	Message length	
SourceAddr			
AreaId			
Checksum		Authentication type	
Authentication			

รูปที่ 3.40: รูปแบบเซดเดอร์ OSPF
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

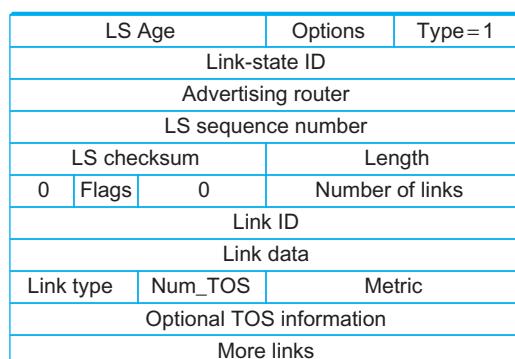
จากรูปที่ 3.41 เป็นรูปแบบแพ็กเก็ต Type-1 ทำหน้าที่ advertisement สำหรับ Type-1 LSA(link-state advertisement) ทำหน้าที่กระจายข้อมูลค่าใช้จ่ายไปเร้าเตอร์ข้างเคียง สำหรับ Type 2 สำหรับบอกเครือข่ายว่ามีเร้าเตอร์ตัวใดที่ต้องมารับฟัง

3.5 ໂພຣໂທຄອລໄອຟີຣຸນທີ່ ۶

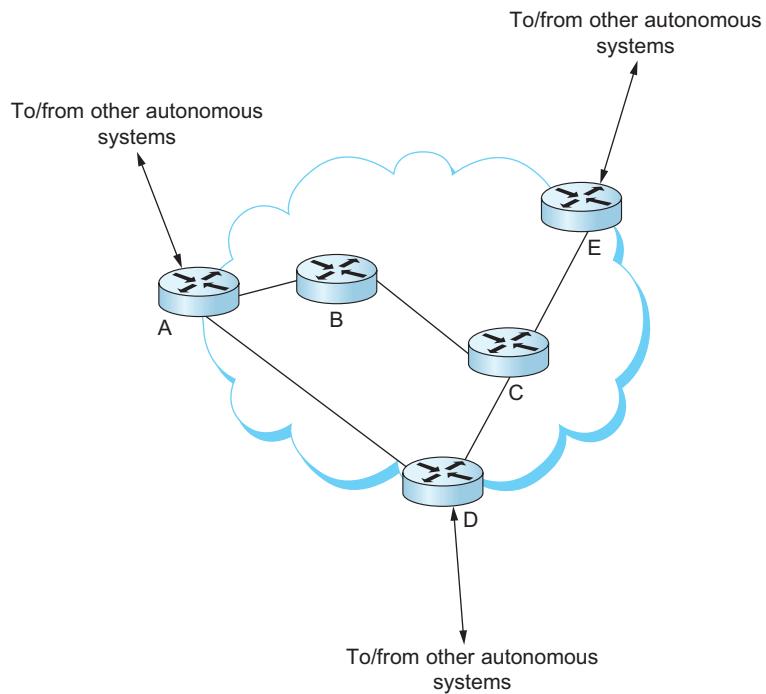
ไอพีรุ่น ๖ พัฒนาขึ้นหลังจากมีการใช้งานเครือข่ายอินเทอร์เน็ตโดยใช้อิ皮รุ่น ๕ เพิ่มจำนวนขึ้นอย่างรวดเร็วและผู้ให้บริการมองเห็นถึงความเป็นไปได้ที่ไอพีจะไม่เพียงพอต่อความต้องการ เพราะหมายเลขอิพินั่นไม่ได้ถูกใช้กับไฮสตรีทเพียงอย่างเดียว แต่ถูกใช้กับอุปกรณ์เครือข่ายที่สร้างโครงข่ายด้านในรูปที่ 3.42

ตารางที่ 3.18: ลำดับการสร้างตารางเราท์ติ้งสำหรับโหนด D

Step	Confirmed	Tentative	Comments
1	(D,0,-)		Since D is the only new member of the confirmed list, look at its LSP.
2	(D,0,-)	(B,11,B) (C,2,C)	D's LSP says we can reach B through B at cost 11, which is better than anything else on either list, so put it on "Tentative" list; same for C.
3	(D,0,-) (C,2,C)	(B,11,B)	Put lowest-cost member of "Tentative" (C) onto "Confirmed" list. Next, examine LSP of newly confirmed member (C).
4	(D,0,-) (C,2,C)	(B,5,C) (A,12,C)	Cost to reach B through C is 5, so replace (B,11,B). C's LSP tells us that we can reach A at cost 12.
5	(D,0,-) (C,2,C) (B,5,C)	(A,12,C)	Move lowest-cost member of "Tentative" (B) to "Confirmed", then look at its LSP.
6	(D,0,-) (C,2,C) (B,5,C)	(A,10,C)	Since we can reach A at cost 5 through B, replace the "Tentative" entry.
7	(D,0,-) (C,2,C) (B,5,C) (A,10,C)		Move lowest-cost member of "Tentative" (A) to "Confirmed", and we are all done.

รูปที่ 3.41: รูปแบบ advertisement ภายในแพ็กเก็ต OSPF
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ตั้งนั้นเพื่อให้สามารถขยายจำนวนอุปกรณ์จะขยายได้เพียงพอจึงมีการออกแบบ IP เวอร์ชันหลักให้เพิ่มหมายเลขแอดเดรสจากเดิม 32-บิตเป็น 128-บิต จากการขยายจำนวนแอดเดรสนี้ทำให้สามารถยืนยันได้แน่นอนว่าทุกอุปกรณ์เครือข่ายจะสามารถเชื่อมต่อสู่อินเทอร์เน็ตได้โดยไม่มีทางที่แอดเดรสจะไม่เพียงพอ



รูปที่ 3.42: การเขื่อมต่อระหว่างเครือข่ายใช้หมายเลขไอพีสำหรับตารางเรขาตั้ง
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ความเป็นมาของรุ่น ๖

องค์กร IETF(Internet Engineering Task Force) ปัญหาของการขยายตัวของผู้ใช้งานเครือข่ายอินเทอร์เน็ต เริ่มมีหมายเลขอี้รอน้อยลงนับตั้งแต่ปี 1991 ได้เสนอแก้ไขปัญหานี้อยู่หลายรูปแบบ ทั้งหมดไม่พ้นการเพิ่ม พื้นที่สำหรับหมายเลขอี้รี ซึ่งไอพีแสดงเดรสบรรจุอยู่ในเขตเดอร์ทุกๆ ไอพีแพ็กเก็ต เพียงเพิ่มขนาดพื้นที่สำหรับ เก็บแอดเดรสก์จะขยายจำนวนแอดเดรสได้ แต่สิ่งที่ตามมานั้นคือการปรับเปลี่ยนหมายเลขอเวอร์ชันของอินเทอร์ เน็ตโพรโทคอล ซึ่งจะต้องเปลี่ยนแปลงซอฟต์แวร์ในแต่ละโญาสต์และตารางเรขาตั้งให้รู้จักโพรโทคอลไอพีรุ่นใหม่ นี้ ซึ่งต้องมีการเปลี่ยนแปลงอย่างระมัดระวัง เพราะจะส่งผลกระทบต่ออุปกรณ์ที่อยู่ในเครือข่ายอินเทอร์เน็ต

การออกแบบพูดกับไอพีรุ่นใหม่จะแก้ปัญหาที่เกิดกับไอพีรุ่นเก่าได้ด้วยคุณสมบัติสำคัญของไอพีรุ่น ๖ นี้ ได้แก่

- รองรับการให้บริการแบบ real-time
- มีระบบความปลอดภัยสูงขึ้น
- มีระบบกำหนดหมายเลขอี้รี
- ปรับปรุงประสิทธิภาพการค้นหาเส้นทาง

ไอพีรุ่น ๖ ได้รับการกล่าวถึงมาตลอดระยะเวลา 30 ปีที่ผ่านมา เพื่อทดแทนการทำงานไอพีรุ่น ๔ แต่ยังขาด จำนวนได้ไม่นานนัก มีอุปกรณ์เครือข่ายที่ใช้ไอพีรุ่น ๔ นั้นมีจำนวนมากและไม่สามารถปรับปรุงซอฟต์แวร์ได้ และบางส่วนถูกตัดตั้งไปในระบบที่มีความสามารถสูงและการเปลี่ยนถ่ายเทคโนโลยีจึงมีปัญหาอย่างอ่อนไหว กว่าเดิม

Addresses และ Routing

จากการแบ่งกลุ่มไอพี แบบ CIDR เป็นวิธีมีประสิทธิภาพในไอพีรุ่น ๔ ถูกนำมาใช้กับไอพีรุ่น ๖ เป็นไอพีแบบ classless โดยยังคงใช้พีร์พิคบิต เป็นตัวกำหนดกลุ่มเครือข่าย ตามตารางที่ 3.19

ตารางที่ 3.19: หมายเลขพีร์พิคของไอพีรุ่น ๖

พีร์พิค	การใช้งาน
00...0(128-บิต)	ยังไม่ใช้งาน
00...1 (128-บิต)	Loopback
1111 1111	Multicast addresses
1111 1110 10	Link-local unicast
ทุกไอเพินอกเหนือจากด้านบน	Global Unicast

จากตารางที่ 3.19 เป็นการจัดสรรหมายเลขไอพีที่ไม่ซับซ้อน แคลรอกใช้กำหนดและตรวจสอบที่ยังไม่ได้ใช้งาน 例外ต่อมาเป็นหมายเลขไอพีที่ใช้สำหรับการทดสอบเนตเวิร์กจะเดปเตอร์ อันดับที่สามเป็นแอดเดรสสำหรับ Multicast อันดับที่สี่เป็นแอดเดรสสำหรับเครือข่ายภายใน และอันดับสุดท้ายเป็นหมายเลขไอพีที่ใช้ในการเชื่อมต่อเครือข่ายอินเทอร์เน็ต

Address Notation

ของ IP เวอร์ชั่นใหม่นั้นจะเขียนแทนด้วยเลขฐาน 16 จำนวน 32-ดิจิท(digit) รวมเป็นเลขใบารีขนาด 128-บิต โดยการเขียนเลขไอพีจะคั่นระหว่างแต่ละ 4-ดิจิทของเลขฐาน 16 ด้วยเครื่องหมายโคลอน(:) ตั้งต่อไปนี้

47CD:1234:4422:AC02:0022:1234:A456:01234

รูปแบบการเขียนไอพีรุ่น ๖แบบเต็มรูปแบบ ซึ่งมีข้อกำหนดมาตรฐานให้สามารถลดรูปแบบการเขียนลงได้ในกรณีที่มีเลข 0 ติดต่อกัน ตัวอย่างเช่น

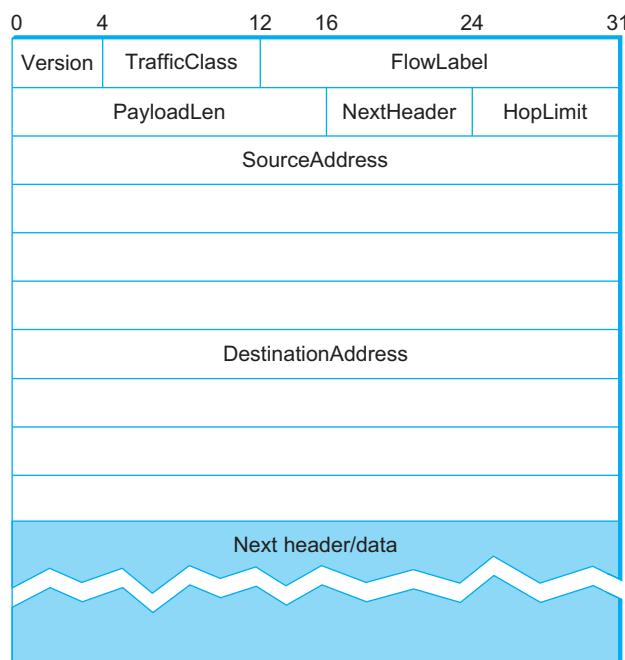
47CD:0000:0000:0000:0000:0000:A456:0124

สามารถลดรูปในกรณีมี 0 ติดต่อกัน ได้เป็น

47CD::A456:0124

โครงสร้างแพ็กเก็ต

สร้างข้อมูลของ IP เวอร์ชั่นใหม่นั้นเป็นไปตามรูปที่ 3.43 เวอร์ชั่นหลักได้ตัดส่วนสิ่งที่ทำให้ขนาดของแพ็กเก็ตไม่คงที่ออกไป ทำให้ขนาดแพ็กเก็ตของ ไอพีรุ่น ๖ มีค่าคงที่เท่ากับ 40-ไบต์



รูปที่ 3.43; โครงสร้างヘッเดอร์ของไอพีรุ่น ๖
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

บทที่ 4

ໂພຣໂທຄອລ ແບ End-to-End

จากบทที่ 3 ได้กล่าวถึงวิธีการกำหนดแอดเดรสให้กับโพสต์ในอุปกรณ์เครือข่ายที่ต้องการเชื่อมต่อแต่ยังไม่ได้กล่าวถึงวิธีการนำพาข้อมูลจากโค้ดต้นทางให้ส่งไปถึงโพสปลายทางในการส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ตนั้นกำหนดให้มีรูปแบบการส่งข้อมูลออกเป็นสองประเภทประเภทแรกได้แก่การส่งข้อมูลโดยไม่สามารถการันตีว่าเดินทางถึงปลายทางได้ เรียกว่า “connectionless” ประเภทที่สองได้แก่การส่งข้อมูลที่สามารถการันตีได้ว่าข้อมูลจะเดินทางถึงปลายทางอย่างสมบูรณ์ เรียกว่า “connection oriented”

ปัญหา : การประมวลผลข้อมูลเครือข่าย

ปัญหาที่เกิดกับการส่งข้อมูลผ่านเครือข่ายคอมพิวเตอร์ที่ผู้ออกแบบแบบໂ proletokolcurve คำนึงถึงมีดังนี้

- ឃើញឈើណាតែវាទៅខ្លួនមូលចិត្តភាពយោទាហេ
 - សេងខ្លួនរើសរាយការបង្កើតអំពីរបាយការណ៍
 - ដោយសេងខ្លួនជូនឈើណា
 - រងរបារការសេងខ្លួនបានណាតុល្លឹម
 - សេងខ្លួនដើរឲ្យអត្ថបន្ទើរួចរាល់ពីរបាយការណ៍របៀបដែល
 - ឧប្បាសាទែងក្រោមគោរពអត្ថបន្ទើរួចរាល់ការសេងខ្លួនពីរបាយការណ៍ដែល
 - រងរបារការសេងខ្លួនពីរបាយការណ៍ដែលបានរាយការណ៍របៀបដែល

จากรายการด้านบนปัญหาเกิดขึ้นจากเส้นทางในการส่งข้อมูลนั้นจะพบอุปสรรคในการนำพาข้อมูลซึ่งในชั้นทรายสปอร์ตนี้จะแก้ไขโดยอุปสรรคที่เกิดขึ้น ปัญหาที่เกิดในการส่งข้อมูลมีดังนี้

- มีข้อมูลสูญหายระหว่างทาง
 - ข้อมูลเดินทางถึงปลายทางไม่เป็นลำดับ
 - เกิดการส่งข้อมูลช้า
 - ขนาดข้อมูลถูกจำกัดในบาง ลิงก์
 - บางข้อมูลอาจใช้เวลาเดินทางนาน

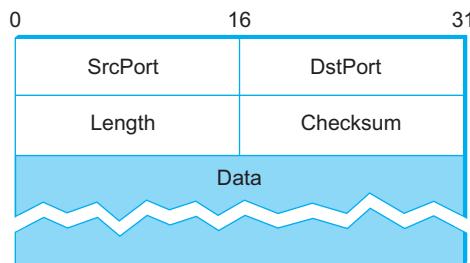
ปัญหาตามที่กล่าวมานี้อาจไม่ได้เกิดขึ้นตลอดแต่เมื่อเกิดขึ้นแล้วกระบวนการขนส่งครัวสามารถที่จะแก้ไขได้ จะมีการออกแบบรองรับวิธีการแก้ปัญหานี้เบ่งเป็นสองแบบแบบแรกคือไม่คำนึงถึงปัญหาเหล่านี้เลยโดยมีความเชื่อว่าโอกาสที่จะเกิดปัญหาเหล่านี้มีโอกาสสนอยวิธีการเหล่านี้จะใช้การส่งข้อมูลแบบ connectionless ขณะที่เข้าที่คำนึงถึงปัญหาและพยายามแก้ไขจะเลือกใช้วิธีการส่งข้อมูลแบบ connection oriented

4.1 โพรโทคอล UDP

โพรโทคอล UDP ([RFC768, 1980](#)) เป็นโพรโทคอลส่งข้อมูลอย่างง่าย มีหน้าที่รับและนำส่งถึงปลายทางโดยไม่คำนึงถึงองค์ประกอบเกี่ยวข้องอื่นใด การไม่คำนึงถึงองค์ประกอบอื่นใดทำให้โพรโทคอลทำงานเรียบง่าย และปล่อยให้โพรโทคอล TCP([RFC793, 1981](#)) ที่ทำหน้าที่คำนึงถึงองค์ประกอบอื่นซึ่งมีการทำงานซับซ้อนขึ้น

หน้าที่ของ UDP หรือ TCP ทำหน้าที่นำส่งข้อมูลจากต้นทางให้ถึงปลายทาง เช่นเดียวกัน แต่มีการควบคุมคุณภาพการส่งข้อมูลแตกต่างกัน การสื่อสารในขั้นนี้เรียกว่าเป็นการสื่อสารชั้น transport (Layer-4) โพรโทคอล UDP เป็น connectionless ไม่คำนึงถึงปัญหาที่จะเกิดในเครือข่าย โดยออกแบบโพรโทคอลให้รองรับการส่งข้อมูลจากหลายแอปพลิเคชันที่ต้องการส่งไปหลายเครื่อง โดยการกำหนดหมายเลขพอร์ตเป็นตัวแทนของแอปพลิเคชัน

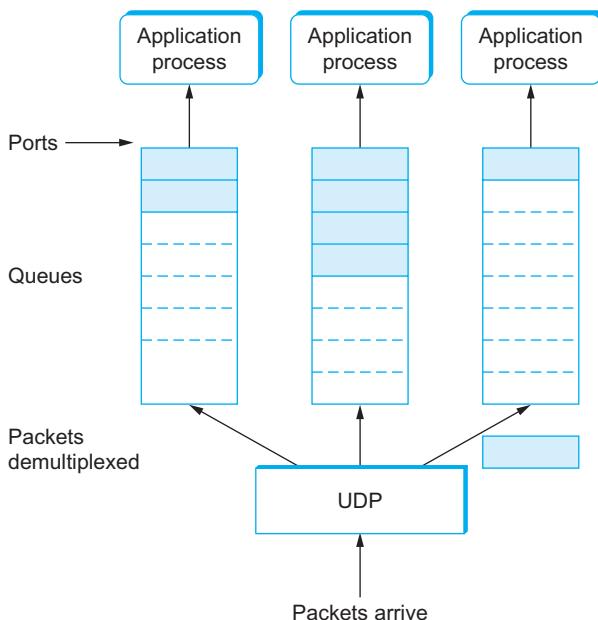
หมายเลขพอร์ตที่ใช้ในการกำหนดแอปพลิเคชัน แบ่งได้ 2 ส่วนได้แก่ well-known และ port ทั่วไป สำหรับ well-known จะเป็นหมายเลขพอร์ต มีหมายเลขอยู่ในช่วง 0-1023 โครงสร้างเขตเดอร์ โพรโทคอล UDP เป็นตามรูปที่ [4.1](#)



รูปที่ 4.1: โครงสร้างเขตเดอร์ UDP
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

UDP เขตเดอร์มีโครงสร้างเขตเดอร์ สีฟลัด์ ได้แก่ SrcPort DstPort Length และ Checksum สำหรับ SrcPort และ DstPort ใช้กำหนดหมายเลขพอร์ตต้นทางและปลายทาง Length ใช้บอกความยาวข้อมูล และ Checksum ใช้ตรวจสอบความสมบูรณ์ของเขตเดอร์

เมื่อไอดีมีหมายเลขแอปพลิเคชันต้องการส่งข้อมูลไปปลายทาง ก็มีการกำหนดหมายเลขพอร์ตที่แตกต่างกัน เป็นไปตามรูปที่ [4.2](#) จากรูปมีแอปพลิเคชันทั้งหมดสามไปเคชั่นที่ต้องการส่งข้อมูลไปที่ปลายทางในแต่ละแอปพลิเคชันจะมีการกำหนดหมายเลขพอร์ตที่แตกต่างกันเมื่อข้อมูลส่งถึงในแต่ละพอร์ตจะถูกนำไปเข้าสู่ระบบคิวซึ่งอยู่ภายใต้ โอเอส(Operating System)ของเครื่องที่ต้องการส่งนั้นหลังจากนั้นตัวโอเอสจะทำการคัดเลือกข้อมูลจากคิวเพื่อส่งข้อมูลออกไปถึงปลายทางลำดับต่อไป



รูปที่ 4.2: เมื่อหlays และ application ต้องการใช้ UDP ส่งข้อมูลจะใช้หมายเลขพอร์ตแต่ละต่างกัน
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

4.2 โพรโทคอล TCP

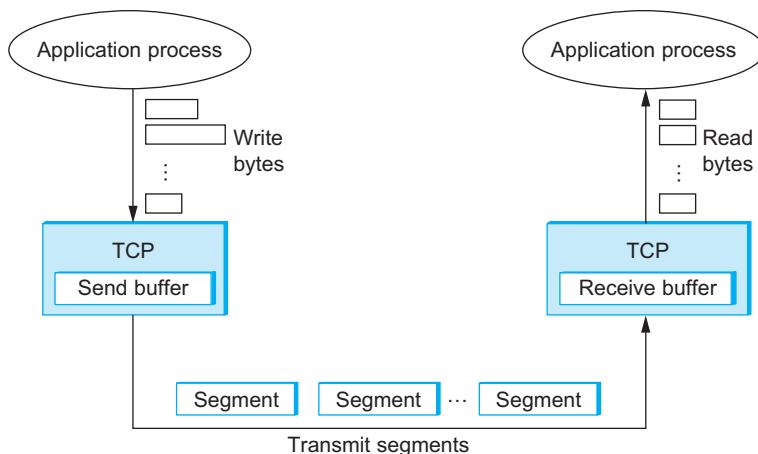
โพรโทคอล TCP เป็นการส่งข้อมูลแบบ connection oriented หรือเรียกว่า “Reliable Byte Stream” การออกแบบ TCP คำนึงถึงปัญหาในการส่งข้อมูลตามที่ได้กล่าวในข้างต้น(4) จึงออกแบบให้ TCP รองรับความต้องการด้านคุณภาพมากกว่า UDP ซึ่งขั้นตอนการทำงานจะคล้ายๆ กันในลำดับต่อไป

4.2.1 End-to-End Issues

หัวใจสำคัญของ TCP คือ sliding window มีหลักการคล้ายกับ sliding window ในชั้นลิงก์ แต่ชั้นลิงก์ เป็นการปรับ sliding window ระหว่างสองเครื่อง แต่สำหรับ TCP ต้องปรับ sliding window กับการเชื่อมต่อตลอดเส้นทาง จึงทำให้การทำงานซับซ้อนกว่าที่พบรูปในชั้นลิงก์

4.2.2 Segment Format

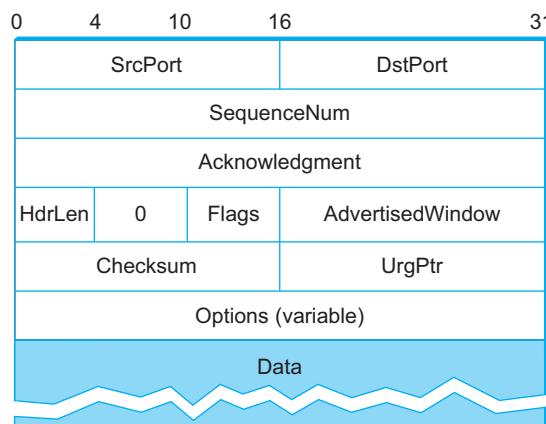
โพรโทคอล TCP มีคุณสมบัติเป็นส่งข้อมูลโดยคำนึงถึงการเรียงลำดับไบต์ที่ถูกต้อง (byte-oriented protocol) ซึ่งหมายถึง เครื่องส่งและเครื่องรับจะมีลำดับเหมือนกัน ถึงแม้ข้อมูลที่ผ่านการเดินทางไปถึงปลายทางไม่เป็นลำดับก็ตาม เมื่อเครื่องรับได้รับข้อมูลแล้วจะเรียงลำดับให้เหมือนเครื่องส่งก่อนจะส่งให้โพรโทคอลเลเยอร์ที่สูงขึ้น ในรูปที่ 4.3 อธิบายการส่งข้อมูลผ่านระบบ TCP โดยแบ่งข้อมูลเป็นส่วนๆ แต่ละส่วนเรียกว่า เซกเมนต์ เครื่องส่งจะส่งข้อมูลลง Send buffer และระบบจะอ่านข้อมูลจาก Send buffer นั้นส่งไปปลายทาง เมื่อปลายทางได้รับข้อมูลจะบันทึกข้อมูลลง Receive buffer และเรียงลำดับให้ตรงกับลำดับเครื่องส่งก่อนส่งข้อมูลขึ้นไปเลเยอร์ต่อไป



รูปที่ 4.3: การบุริหารจัดการข้อมูลของโปรโตคอล TCP
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

โครงสร้างเชกเม้นต์เป็นตามรูปที่ 4.4 มี SrcPort และ DstPort เมื่อเป็น UDP สำหรับข้อมูลพอร์ตนี้จะใช้คู่กับหมายเลขไอพี ประกอบเป็น 4 ชุดข้อมูล(4-tuple) เพื่อกำหนดเป็นตัวอ้างถึงข้อมูลต้นทางและปลายทาง
 $(SrcPort, SrcIPAddr, DstPort, DstIPAddr)$

เรียกข้อมูล $(SrcPort, SrcIPAddr, DstPort, DstIPAddr)$ ว่าเป็น ชี้อ กเก็ต(socket) การมีชี้อ กเก็ตทำให้การส่งข้อมูลจากแอปพลิเคชันต้นทางได้หลายแอปพลิเคชัน และเครื่องปลายทางมีแอปพลิเคชันหลายตัวที่ SrcIPAddr และ DstIPAddr คงเดิมแต่เปลี่ยนหมายเลขพอร์ต

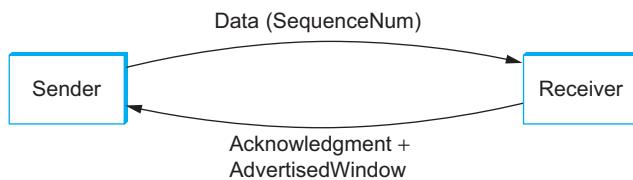


รูปที่ 4.4: เขตเดอร์โปรโตคอล TCP
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ต่อมากำหนด SequenceNum, Acknowledgement และ AdvertiseWindow เป็นฟีลด์สำหรับการทำงานของ sliding window รวมถึงการตรวจสอบการเรียงลำดับของข้อมูล ซึ่งไม่ปรากฏใน UDP

รูปที่ 4.5

ต่อมากำหนด Flags ขนาด 6-บิต ใช้ควบคุมการส่งต่อข้อมูลของแต่ละเชกเม้นต์ แฟล็ก(flags)ที่เป็นไปได้ทั้งหมดประกอบด้วย SYN FIN RESET PUSH URG และ ACK สำหรับ SYN และ FIN ใช้สำหรับเริ่มต้นเชื่อมต่อ และสิ้น



รูปที่ 4.5: TCP มีสิ่ง SequenceNum และเครื่องรับได้ตอบกลับ
ลิขสิทธิภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

สุดการเชื่อมต่อ สำหรับ ACK จะถูกเซตทุกครั้งที่มี Acknowledgement หมายถึงเครื่องที่ได้รับจะต้องให้ความสนใจ Acknowledgement นี้ ต่อมาแฟล็ก URG สำหรับบอกถึงเซกเมนต์มีความต้องการเร่งด่วนบางประการ เมื่อแฟล็ก URG เซตแล้วจะประกอบข้อมูล UrgPrt เพื่อระบุข้อมูลประกอบการเร่งด่วน ต่อมา PUSH ใช้สำหรับกำหนดให้ข้อมูลจากการไม่ต้องรอถูกส่งและรอรับการจัดการโดยโปรแกรมผู้รับแฟล็ก RESET ใช้ปิดการเชื่อมต่อแบบ

สุดท้าย Checksum เป็นการตรวจสอบความถูกต้องของเซดเดอร์ เมื่อใน UDP

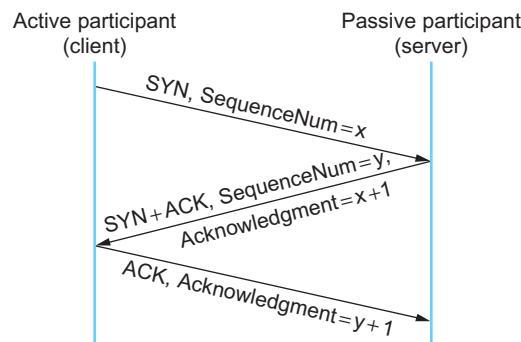
4.2.3 Connection Establishment และ Termination

ก่อนเริ่มต้นการส่งข้อมูลผ่านระบบ TCP นั้นกำหนดให้จะต้องจัดเตรียมเส้นทางการเชื่อมต่อให้พร้อมตั้งแต่ต้นทางไปจนถึงปลายทางเรียกวิธีการจัดเตรียมการเชื่อมต่อว่า “Connection Establishment” และเมื่อสื่อสารเสร็จแล้วจะต้องปิดการเชื่อมต่ออย่างสมบูรณ์เรียกว่า “Termination”

Three-Way Handshake

การตรวจสอบความพร้อมในการส่งและรับข้อมูล ใช้วิธีสอบถามไปสอบถามกลับ เป็นการสนทนารอบ 3 รอบ เรียกว่า “Three-Way Handshake” จากรูปที่ 4.6 เครื่องไคลเอนต์ หรือเรียกว่า คอลล์เลอร์(caller) ส่งเซกเมนต์แรกไป เชิร์ฟเวอร์ หรือเรียกว่า คอลล์ลีร์(callee) เมื่อเชิร์ฟเวอร์ได้รับแล้วจะตอบกลับเป็นเซกเมนต์ที่สอง เพื่อยืนยันว่าได้รับข้อมูล แต่เครื่องเชิร์ฟเวอร์จะยังไม่ทราบว่าเครื่องไคลเอนต์ได้รับข้อมูลการตอบกลับหรือไม่ ดังนั้นในเซกเมนต์สุดท้ายจะเป็นการยืนยันว่าเครื่องไคลเอนต์ได้รับข้อมูลเสร็จสมบูรณ์ ซึ่งมีขั้นตอนการสนทนาทั้งหมดสามขั้นตอน

แนวคิดเบื้องหลังการแลกเปลี่ยนข้อมูลระหว่างทั้งสองเครื่องนี้เป็นการออกแบบระบบที่ต้องการให้เกิดการตกลงในชุดข้อมูลตรงกัน โดยที่ทั้งสองเครื่องมีชุดข้อมูลเป็นของตัวเอง อธิบายการทำงานจากรูปที่ 4.6 โดย คอลล์เลอร์ ต้องการเชื่อมต่อ คอลล์ลีร์ ส่งข้อมูลโดยกำหนดแฟล็กเป็น (Flags=SYN,SequenceNum=x) เมื่อ x แทนเลขใดๆ เมื่อ คอลล์ลีร์ ได้รับข้อมูลแล้วจะต้องตอบกลับโดยการตอบกลับเพื่อยืนยันการได้รับนั้นจะเป็นข้อมูลที่สัมพันธ์กับแฟล็กที่ คอลล์เลอร์ ส่งมา ซึ่งข้อมูลตอบกลับเป็นการนำค่า x+1 ดังนี้ (Flags=ACK, Ack = x+1) และต่อมา คอลล์ลีร์ จะส่งชุดข้อมูลของตัวเองบ้าง โดยการส่ง (Flags = SYN, SequenceNum = y) ซึ่ง y ใช้แทนเลขใดๆ ที่ คอลล์ลีร์ กำหนดขึ้นเอง และส่งไปยัง คอลล์เลอร์ เมื่อ คอลล์เลอร์ ได้รับข้อมูลจะตอบกลับด้วย (Flags = ACK, Ack = y+1)



รูปที่ 4.6: ขั้นตอน Three-way Handshake ในระบบ TCP
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

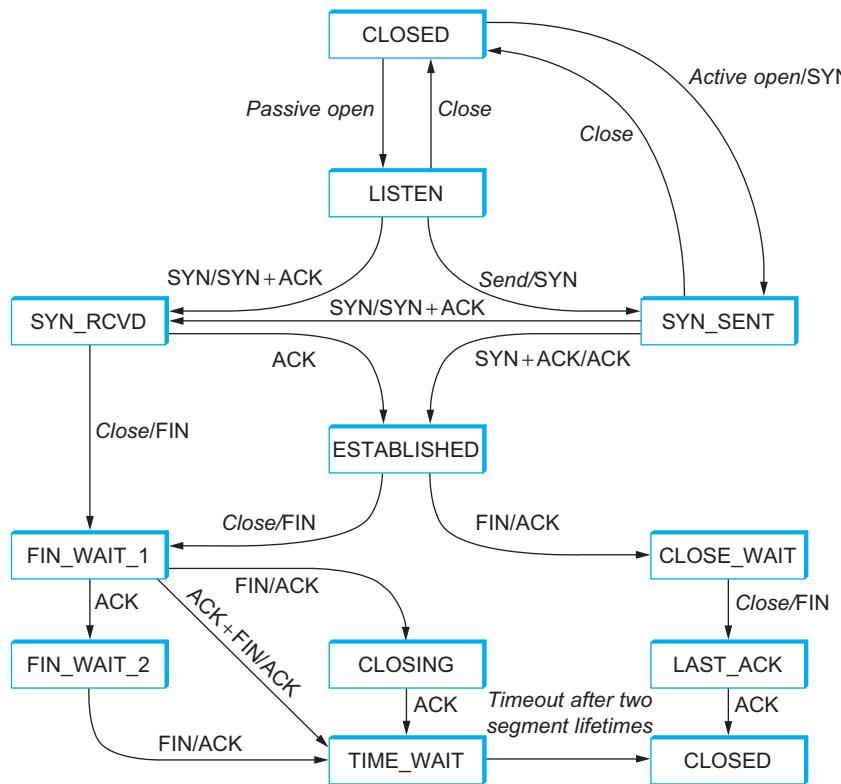
State-transition diagram

การทำงาน TCP มีหลายขั้นตอน ซึ่งสามารถทำให้เข้าใจง่ายด้วยการอธิบายโดยใช้ state-transition diagram ตามรูปที่ 4.7 จากรูปสถานะของแต่ละเซกเมนต์ มองเป็น 2 กลุ่มใหญ่ กลุ่มสถานะเปิดการเชื่อมต่ออยู่ด้านบน ESTABLISHED และสถานะปิดการเชื่อมต่ออยู่ด้านล่าง ESTABLISHED โดยแยกเป็นสถานะย่อยทั้งหมด 12¹ สถานะดังนี้

1. CLOSED สถานะปิดการเชื่อมต่อ
2. LISTEN รอให้ใหม่ขอเชื่อมต่อ
3. SYN_RECV ส่งคำตอบกลับคำขอเชื่อมต่อไปยังโคลเอนต์
4. SYN_SENT ส่งคำขอเชื่อมต่อไปยังเซิร์ฟเวอร์
5. ESTABLISHED สถานะเชื่อมต่อสำเร็จ และกำลังเชื่อมต่อ
6. FIN_WAIT_1 เมื่อได้รับแฟล็ก Close/FIN
7. FIN_WAIT_2 เมื่อได้รับแฟล็ก ACK
8. CLOSING เมื่อได้รับ FIN/ACK
9. CLOSE_WAIT รอสัญญาณ Close/FIN
10. LAST_ACK สถานะส่ง แอ็คโนเล็จเม้นท์ ครั้งสุดท้าย
11. TIME_WAIT เมื่อได้รับแฟล็ก FIN/ACK

สถานะเริ่มต้นยังไม่มีการเชื่อมต่อทาง TCP ทุกเครื่องโคลเอนต์อยู่ในสถานะ CLOSED และปลายทางอยู่สถานะ LISTEN เมื่อโคลเอนต์ต้องการติดต่อสื่อสารจะเริ่มต้นสร้างการเชื่อมต่อ(connection) โดยการส่งเซกเมนต์ SYN ไปเครื่องที่ต้องการติดต่อ และเปลี่ยนสถานะตัวเองไปอยู่ SYN_SENT เป็นขั้นตอน handshake ขั้นแรก เมื่อเซิร์ฟเวอร์ได้รับ SYN จะตอบกลับโคลเอนต์ด้วย SYN+ACK และเปลี่ยนสถานะตัวเองไปเป็น SYN_RECV ลำดับต่อมา เป็นลำดับ handshake ขั้นสอง และสุดท้ายโคลเอนต์ตอบกลับด้วย SYN+ACK/ACK เป็นขั้นตอนที่สาม จึงครบขั้นตอนการทำ three-way handshake

¹ จากรูปที่ 4.7 มีสถานะ CLOSSED ซ้ำกัน เพื่อครุ่นเจาะขึ้น



รูปที่ 4.7: TCP state-transition diagram
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

สำหรับขั้นตอนปิดการเชื่อมต่อ จะเริ่มจากไคลเอนต์ หรือ เซิร์ฟเวอร์ก็ได้ ในการปิดการเชื่อมต่อ ทำได้ 3 แนวทางดังนี้

- ส่งคำขอปิดการเชื่อมต่อ : **ESTABLISHED** → **FIN_WAIT_1** → **FIN_WAIT_2** → **TIME_WAIT** → **CLOSED**.
- ได้รับคำสั่งปิดการเชื่อมต่อ : **ESTABLISHED** → **CLOSE_WAIT** → **LAST_ACK** → **CLOSED**
- ทั้งสองฝ่ายตกลงปิดพร้อมกัน : **ESTABLISHED** → **FIN_WAIT_1** → **CLOSING** → **TIME_WAIT** → **CLOSED**.

4.2.4 Sliding Window สำหรับ TCP

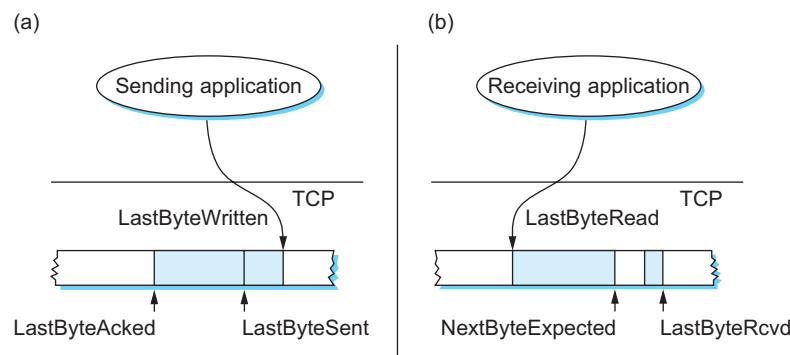
วัตถุประสงค์การมี sliding window ในระบบ TCP มีดังนี้

- เพื่อทำให้ข้อมูลมีเสถียรภาพในการส่งข้อมูลเพิ่มขึ้น
- เพื่อให้ข้อมูลสามารถเดินทางถึงเป้าหมายและมีการส่งให้เป็นตามลำดับ(ordered delivery)
- ใช้ควบคุมอัตราเร็วการส่งข้อมูลระหว่างเครื่องรับและเครื่องส่งได้เหมาะสม

ในส่วนที่ sliding window ใน TCP แตกต่างจาก sliding window ในการทำงานชั้นลิ้งค์ เกี่ยวข้องกับการทำงานในข้อ 1 และ 2 แทนที่การกำหนด sliding window จะคงที่ของลิ้งค์ การทำงาน sliding window ใน TCP จะมีข้อมูลตอบกลับจากเครื่องรับเรียกว่า advertises ซึ่งข้อมูลดังกล่าวจะกำหนดค่า AdvertisedWindow ในTCP เฮดเดอร์ฟล็อต ค่านี้จะเปลี่ยนแปลงโดยขึ้นอยู่กับหน่วยความจำคงเหลือของเครื่องรับ²

การมีเสถียรภาพในการส่งข้อมูลและการส่งให้เป็นตามลำดับ

หัวข้อนี้อธิบายระบบ sliding window ใน TCP ส่งผลต่อการมีเสถียรภาพในการส่งข้อมูลและการส่งให้เป็นตามลำดับอย่างไร



รูปที่ 4.8: ความสัมพันธ์ระหว่างบัฟเฟอร์ เครื่องส่ง (a) และ บัฟเฟอร์เครื่องรับ (b)
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

อธิบายโดยใช้รูปที่ 4.8 สมมติให้เครื่องส่ง (a) มีข้อมูลต้องการส่งไปเครื่องรับ (b) จากรูปที่ 4.8(a) แบบสิขาข่ายมีแต่ข้อมูลที่ส่งออกไปแล้วและเครื่องรับได้ตอบกลับแอ็คโนเล็จเม้นท์ ถือว่าส่งได้สำเร็จ ซึ่งແບสีขาวจะซึ้งที่ตำแหน่งข้อมูลที่ยืนยันว่าส่งสำเร็จกำหนดให้แทนด้วยตัวแปร **LastByteAcked** ต่อมากับสีฟ้าถัดมาใช้แทนเซกเมนต์ที่ส่งออกไปแล้วยังไม่ได้รับ แอ็คโนเล็จเม้นท์ ไปจนถึงตำแหน่งล่าสุดที่ข้อมูลได้ส่งออกแล้วแต่ยังไม่ได้รับแอ็คโนเล็จเม้นท์ กำหนดให้แทนด้วยตัวแปร **LastByteSent** สีฟ้าถัดจากนั้นเป็นส่วนหน่วยความจำที่เหลืออยู่ที่สามารถรองรับข้อมูลที่ต้องการส่งเพิ่มได้ เท่านั้น ถ้าสีฟ้าคือขนาดหน่วยความจำที่มีอย่างจำกัดของเครื่องส่ง และยังไม่ได้รับแอ็คโนเล็จเม้นท์ กำหนดให้แทนด้วยตัวแปร **LastByteWritten** ที่ต้องการส่งในคราวนี้จะถูกส่งไปเครื่องรับก็มีหน่วยความจำอย่างจำกัดดังรูปที่ 4.8(b) มีแบบสีฟ้าไว้ใช้กำหนดพื้นที่หน่วยความจำทั้งหมดที่จะส่งได้ซึ่งสังเกตช่วงว่าสีขาวหมายถึงข้อมูลนั้นได้รับข้อมูลสำเร็จแล้วในตำแหน่งต่างกัน

จากรูปที่ 4.8(a) เมื่อเซกเมนต์แรกส่งออก จะเกี่ยวข้องกับตัวแปรสามตัวดังนี้ **LastByteAcked**, **LastByteSent**, และ **LastByteWritten** ทำหน้าที่จัดสรรพื้นที่บัฟเฟอร์ ของเครื่องส่ง ในที่นี้จะเรียกว่า “คอลล์เลอร์”

จำนวนตัวเลขที่กำหนดใน **LastByteAcked** และ **LastByteSent** ใช้แทนขนาดข้อมูลหน่วยเป็นไบต์ โดยที่ขนาดข้อมูลที่เครื่องรับได้รับเท่ากับ **LastByteAcked** นั้นมีขนาดน้อยกว่าหรือเท่ากับข้อมูลที่ส่ง

²เครื่องรับไม่ได้รับข้อมูลจากเครื่องส่งเพียงเครื่องเดียว ในเวลาเดียวกันอาจมีหลายเครื่องพยายามส่งไปเครื่องรับเดียว ทำให้หน่วยความจำลัดลง

LastByteSent ถ้าข้อมูล LastByteAcked มีขนาดน้อยกว่า LastByteSent หมายถึงข้อมูลมีการสูญหายระหว่างทาง เนื่องเป็นความสัมพันธ์ดังนี้

LastByteAcked <= LastByteSent

ที่เครื่องรับคอลล์ลีร์ เมื่อได้รับข้อมูลจะรับข้อมูลเข้าหน่วยความจำตามความสามารถที่รับได้ และตอบกลับแอ็กโนเหล็จเมนท์ ตามจำนวนเบต์ที่ได้รับ ขณะที่ส่งข้อมูลส่งออกลำดับ LastByteSent ที่จะมีค่าน้อยกว่า หรือเท่ากับข้อมูลที่ส่งออกจากเครื่องส่ง LastByteWritten เป็นความสัมพันธ์ดังนี้

LastByteSent <= LastByteWritten

เมื่อข้อมูลส่งเสร็จสิ้นจะลบข้อมูลส่งสำเร็จออกจากบัฟเฟอร์ ดังนั้นจะไม่มีข้อมูลพอยน์เตอร์ที่อยู่ทางซ้ายมือ LastByteAcked และข้อมูลที่ยังส่งไม่ยืนยันโดยแอ็กโนเหล็จเมนท์ จะยังบันทึกในบัฟเฟอร์ เป็นข้อมูล LastByteWritten

ต่ำมาดูที่เครื่องรับคอลล์ลีร์ ใช้ตัวแปรพอยน์เตอร์สามตัวเช่นเดียวกับเครื่องส่ง ได้แก่ LastByteRead, NextByteExpected และ LastByteRcvd

ค่าตัวแปร LastByteRead อยู่ทางซ้ายมือใช้ชี้ตำแหน่งข้อมูลในหน่วยความจำที่ได้ส่งแอ็กโนเหล็จเมนท์ตอบกลับไปแล้ว และ NextByteExpected ใช้ชี้หน่วยความจำที่ได้สำรองพื้นที่ไว้สำหรับรับข้อมูล ค่า LastByteRead มีจำนวนน้อยกว่า NextByteExpected เสมอ ดังความสัมพันธ์ดังนี้

LastByteRead < NextByteExpected

จาก $\text{LastByteRead} < \text{NextByteExpected}$ เป็นจริงได้ เพราะข้อมูลไม่อาจส่งได้ถึงขั้นแอปพลิเคชัน หากเครื่องรับไม่ได้รับข้อมูล

เมื่อคอลล์ลีรับข้อมูลประมวลผลเสร็จแล้วจะปรับค่า LastByteRcvd เพิ่มอีก 1 ไบต์ เพื่อลบข้อมูลออกจากหน่วยความจำ และอนุญาตให้คอลล์ลีร์ส่งข้อมูลมาเพิ่มได้อีก 1 ไบต์

NextByteExpected <= LastByteRcvd + 1

เมื่อข้อมูลเดินทางเป็นลำดับ ตัวแปร NextByteExpected จะซึ้งไปยังจุดที่อยู่หลังจาก LastByteRcvd ที่ซึ่งหากข้อมูลเดินทางมาถึงไม่เป็นลำดับจะมีการเว้นช่องว่างไว้ดังอธิบายในรูปที่ 4.8(b)

Flow Control

จากที่กล่าวมาข้างต้น การปรับขนาดบัฟเฟอร์มีวิธีทำงานตามพื้นฐาน sliding window มีส่วนที่แตกต่างอย่างเดียวได้แก่ขั้นตอนการเติมช่องว่างลงบัฟเฟอร์เมื่อข้อมูลเดินทางไม่เป็นลำดับ

เพื่อทำความเข้าใจหลักการทำงาน มาดูการทำงานอัลกอริทึมสองที่แตกต่างจาก sliding window ที่ว่าไป ก่อนกำหนดขอบเขตของบัฟเฟอร์ ใช้ตัวแปร MaxSendBuffer และ MaxRcvBuffer (ตอนนี้ยังไม่ต้อง

กังวลส่วนรายละเอียดของทั้งสองตัวแปร จะกล่าวถึงในส่วนถัดไป) ในเวลานี้ให้ทราบว่า มีตัวแปรสองตัวที่เครื่องส่ง(MaxSendBuffer)และเครื่องรับ(MaxRecvBuffer)ใช้เป็นตัวบอกขีดจำกัดของบัฟเฟอร์

กลับมาที่โปรโทคอล sliding window กำหนดให้มี window ใช้แทนการรับข้อมูลโดยที่ไม่ต้องรอแอ็กโนเลจเม้นท์ ดังนั้นเครื่องรับจะบีบอัตราส่งของเครื่องส่งโดยการบอกรหานาด window ผ่านทาง แอ็คโนเลจเม้นท์ ซึ่งคือการรักษาขนาด window ไม่ให้เกิด MaxRcvBuffer

LastByteRcvd - LastByteRead <= MaxRcvBuffer

การป้องกันไม่ให้ใช้ บัฟเฟอร์ มากกว่า MaxRcvBuffer เป็นการป้องกันไม่ให้เกิดปัญหา บัฟเฟอร์โอเวอร์โฟล์ว(buffer overflow) โดยขนาดพื้นที่บัฟเฟอร์ที่เหลืออยู่ส่งผ่าน AdvertiseWindow

AdvertisedWindow = MaxRcvBuffer - ((NextByteExpected - 1) - LastByteRead)

เมื่อข้อมูลที่เดินทางถึงเครื่องรับทีละแพ็คเก็ต แต่ละแพ็คเก็ตมีขนาดไม่เท่ากัน เมื่อเครื่องรับได้รับข้อมูลจะตอบกลับด้วยแอ็กโนเลจเม้นท์ พร้อมกันจะงพื้นที่บัฟเฟอร์ของเครื่องรับ และรายงานหน่วยความจำคงเหลือผ่าน AdvertisedWindow เครื่องรับได้รับข้อมูลจะปรับ LastByteRcvd ตามจำนวนไบต์ที่ได้รับ และเลื่อนไปทางขวาของรับข้อมูลต่อไป ข้อมูลจะถูกนำออกจากบัฟเฟอร์เมื่อมีการอ่านข้อมูลจากชั้นแอปพลิเคชัน การอ่านข้อมูลนี้จะทำการปรับเพิ่มตัวแปร LastByteRead ซึ่งหมายถึงข้อมูลถูกอ่านเสร็จแล้ว ซึ่งส่งผลให้บัฟเฟอร์เหลือมากขึ้น และจะส่งกลับไปบอกเครื่องส่งว่ามีบัฟเฟอร์เหลือมากกว่าขึ้นผ่านทางการปรับขนาด AdvertisedWindow

LastByteSent - LastByteAcked <= AdvertisedWindow

หรือกล่าวอีกทางหนึ่งได้ว่า เครื่องส่งจะสามารถส่งได้ด้วยขนาด window เท่ากับ EffectiveWindow

EffectiveWindow = AdvertisedWindow - (LastByteSent - LastByteAcked)

EffectiveWindow จะมีขนาดมากกว่า 0 ก่อนที่เครื่องส่งจะส่งข้อมูล ดังนั้นเมื่อได้รับ แอ็คโนเลจเม้นท์ จำนวน x ไบต์ จะทำให้เครื่องส่งสามารถส่งข้อมูลเพิ่มได้มากกว่าเดิมจำนวน x ไบต์ ดังนั้น

LastByteWritten - LastByteAcked <= MaxSendBuffer

ถ้าเครื่องส่ง ได้ส่งข้อมูลจำนวน y ไบต์ ผ่าน TCP แต่

(LastByteWritten - LastByteAcked) + y > MaxSendBuffer

แล้ว TCP จะบอกว่าไม่สามารถส่งข้อมูลได้มากกว่า MaxSendBuffer ซึ่งเกิดจาก AdvertisedWindow = 0

Protecting Against Wraparound

หัวข้อนี้จะกล่าวถึงขนาดของ SequenceNum และ AdvertisedWindow ที่กำหนดในโปรโทคอล TCP ที่ส่งผลต่อประสิทธิภาพการส่งข้อมูล สำหรับ SequenceNum มีขนาด 32-บิต และ AdvertisedWindow มีขนาด

16-บิต หมายถึง TCP จะปรับ sliding window โดยมี SequenceNum มีขนาดใหญ่กว่า AdvertisedWindow $2^{32} >> 2 \times 2^{16}$ ถือว่ามากกว่า อย่างไรก็ตามความท่างของทั้งสองฟีลด์นี้ไม่สำคัญนัก พิจารณาได้ดังนี้

ขนาด SequenceNum ขนาด 32-บิต กำหนดให้มีขนาดใหญ่พอจะที่ไม่เกิดการใช้ SequenceNum ซ้ำในแต่ละรอบการเชื่อมต่อ การไม่เกิด SequenceNum ทำให้ TCP เรียงลำดับข้อมูลได้ถูก ลำดับข้อมูลใช้อ้างถึงเซกเมนต์จำนวนหนึ่งชุด ถ้าส่งด้วยอัตราเร็วสูงขึ้นจะทำให้การเพิ่มจำนวน SequenceNum สูงขึ้นตามไปด้วย ตารางที่ 4.1 อธิบายข้อมูลที่จะใช้ SequenceNum เที่ยวนครบรรบุ โดยเทียบกับอัตราเร็วในการส่งข้อมูล

ตารางที่ 4.1: หน่วยเวลาขนาด 32-บิต

แบบตัวดิจิต	หน่วยเวลาจนกว่าจะตัดการเชื่อมต่อ
T1 (1.5 Mbps)	6.4 hours
T3 (45 Mbps)	13 minutes
Fast Ethernet (100 Mbps)	6 minutes
OC-3 (155 Mbps)	4 minutes
OC-48 (2.5 Gbps)	14 seconds
OC-192 (10 Gbps)	3 seconds
10GigE (10 Gbps)	3 seconds

จากตาราง 4.1 เห็นได้ว่าขนาด 32-บิต SequenceNum จะใช้หน่วยวัดเร็วขึ้นเมื่อมีการส่งด้วยอัตราเร็วสูง สังเกต OC-192 เป็นอัตราเร็วอินเทอร์เน็ต(เครือข่าย backbone)ในปัจจุบัน ซึ่งเครื่องแม่ข่ายที่อยู่ใน data center สื่อสารได้ด้วยความเร็ว 10Gbps ทำให้ขนาด 32-บิต SequenceNum สำหรับเครื่องแม่ข่ายเหล่านี้ ถือว่าขนาดเล็กกว่าที่ควรเป็นอยู่มาก โชคดีที่ IETF ได้ปรับปรุงขนาด SequenceNum ไปแล้วในปัจจุบันซึ่งจะกล่าวถึงในลำดับต่อไป

Keeping the Pipe Full

ต่อมาสำหรับ AdvertisedWindow ขนาด 16-บิต เป็นขนาดใหญ่ที่สุดที่ใช้บอกเครื่องส่งจะส่งข้อมูลได้เต็มที่โดยไม่ต้องรอแอ็คโนเวล็จเม้นท์ และเครื่องรับจะไม่รับข้อมูลที่มีขนาดมากกว่า AdvertisedWindow

ในกรณีนี้ไม่ได้เป็นเพียงการบอกขนาดแบบตัวดิจิตของเครือข่าย แต่เป็นการบอกถึง $\text{delay} \times \text{bandwidth}$ product ขนาดของ window ต้องมีขนาดใหญ่กว่า $\text{delay} \times \text{bandwidth}$ เพื่อให้สามารถส่งได้เต็มความสามารถ ในตารางที่ 4.2 กำหนดขนาด $\text{delay} \times \text{bandwidth}$ ของแต่ละเทคโนโลยี

จากตารางเห็นได้ว่า AdvertisedWindow ส่งปัญหามากกว่า SequenceNum ซึ่งเริ่มมีขนาดไม่เพียงพอกับเทคโนโลยี T3 ซึ่งขนาด 16-บิต เปิดให้มี AdvertisedWindow เพียง 64 KB อย่างไรก็ตามปัญหานี้ได้รับการแก้ไขแล้วในปัจจุบันซึ่งจะกล่าวถึงในลำดับต่อไป

ตารางที่ 4.2: ต้องการ Window Size สำหรับ 100-ms รwanต์ทริปไทม์

แบบวิดีธ	Delay×Bandwidth Product
T1 (1.5 Mbps)	18 KB
T3 (45 Mbps)	549 KB
Fast Ethernet (100 Mbps)	1.2 MB
OC-3 (155 Mbps)	1.8 MB
OC-48 (2.5 Gbps)	29.6 MB
OC-192 (10 Gbps)	118.4 MB
10GigE (10 Gbps)	118.4 MB

4.2.5 Triggering Transmission

หัวข้อนี้จะพิจารณาปัญหาเชิงลึกมากขึ้น มาดูว่า TCP พิจารณาส่งเซกเมนต์อย่างไร เมื่อได้รับข้อมูลจากชั้นแอปพลิเคชันมาต่อเนื่อง เมื่อถึงชั้นTCP จะเริ่มตัดข้อมูลออกเป็น เซกเมนต์ จะมีขั้นตอนตัดสินใจอย่างไร มีปัจจัยใดบ้างที่ควบคุมการตัดสินใจนี้

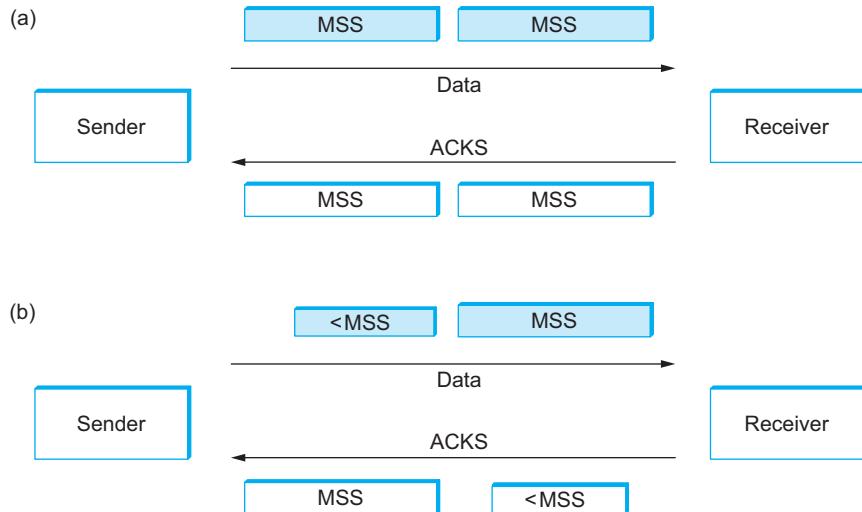
หากไม่สนใจผลกระทบจากการทำงาน flow control กรณีที่การเชื่อมต่อครั้งแรก TCP จะทำงานทั้งหมดสามขั้นตอน ขั้นตอนแรกตรวจสอบขนาด MSS(maximum segment size) และจะส่งเซกเมนต์ทันทีที่มี รวบรวมไปต่อกันถึง MSS จากกระบวนการส่ง MSS เป็นตัวกำหนดขนาดเซกเมนต์สูงสุดของ TCP ที่สามารถส่งได้โดยไม่ก่อให้เกิดการทำงานเพิ่มในชั้นอินเทอร์เน็ตโพรโทคอล หากส่งขนาดใหญ่เกินไปจะทำให้ชั้น IP ต้องแยกส่วน(fragmentation) ค่า MSS จะต้องตรวจสอบตลอดเส้นทางที่ส่งออกจากต้นทางไปถึงปลายทาง ซึ่งแต่ละเส้นทางนั้นมี เอ็มทียู แตกต่างกัน การเลือกขนาดใหญ่ที่สุดของเส้นทางคือการเลือกขนาดเล็กที่สุดของ เอ็มทียู ตลอดเส้นทาง

Silly Window Syndrome

จากการทำงานในหัวข้อที่ผ่านมาได้เลี่ยงปัญหาที่เกิดจากไม่มีระบบ flow control ซึ่งไม่อาจมองข้ามการทำงาน flow control หัวข้อนี้จะกล่าวถึงการควบคุม flow control เครื่องส่งจะเพิ่มอัตราเร็วขึ้นเรื่อยๆ จนกว่า window จะปิดรับและเมื่อได้รับ แอ็คโนเล็จเมนท์ จะทำให้ส่งข้อมูลได้อีก สมมติว่าเครื่องส่งทำการส่งได้สักพักโดยมีการควบคุมอัตราส่งด้วย แอ็คโนเล็จเมนท์ จนเหลือขนาดข้อมูลส่งอยู่ที่ $MSS/2$ คำตามอยู่ที่เครื่องส่งจะส่งด้วยขนาด $MSS/2$ หรือจะรอจนกว่าได้รับ แอ็คโนเล็จเมนท์ ให้สามารถส่งได้ขนาดเต็ม MSS มาตรฐานดังเดิมไม่ได้ระบุควรเลือกวิธีใดอย่างชัดเจน ปัจจุบัน TCPเลือกวิธีส่ง $MSS/2$

ในสถานการณ์ที่เครือข่ายมีการสื่อสารหนาแน่น ทำให้เกิดเหตุการณ์ที่เรียกว่า Silly Window Syndrome อธิบายตามรูปที่ 4.9 ปัญหา Silly window syndrome : (a) ระบบที่ส่งข้อมูลไม่เกิน MSS และเครื่องรับมีแอ็คโนเล็จเมนท์ทุก MSS ระบบทำงานได้ปกติ (b) เมื่อเครื่องส่งทำการส่งข้อมูลขนาดน้อยกว่า MSS หรือเครื่องรับตอบแอ็คโนเล็จเมนท์น้อยกว่าขนาด MSS ข้อมูลขนาดเล็กไม่เกินกว่าระบบส่งได้ ทำให้ระบบยังมีเสถียรภาพ

เมื่อใช้ TCP ส่งข้อมูลมีขนาดเท่ากับ MSS และเครื่องส่งลดขนาดข้อมูลลงให้น้อยกว่า MSS สามารถแก้ปัญหาได้ด้วยการลดขนาด MSS ลง เช่น MSS/2



รูปที่ 4.9: ปัญหา Silly window syndrome
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

Nagle's Algorithm

ในการนี้ผู้ส่ง TCP ส่งข้อมูลมีขนาด MSS แต่มี window เล็กกว่า MSS จากข้อสงสัยที่ผ่านระบบจะส่งด้วยขนาดเท่า window หรือรอให้ window มีขนาดเท่า MSS ถ้ารอให้มีขนาด window เท่า MSS อาจส่งผลต่อซอฟต์แวร์ที่ต้องการทำงานต่อเนื่องเช่น telnet แต่ถ้ารอแล้วได้ส่งข้อมูลได้ขนาดใหญ่ที่สุดจะทำให้ส่งข้อมูลได้เร็วและลดโอเวอร์ヘด(overhead) ที่เกิดจากแพ็กเกจเดอร์

ในการประยุกต์ใช้งานจึงสามารถใช้ระบบ timer สำหรับนับเวลาที่เหมาะสม ตัวอย่างเช่น รอจนกว่าถึง 100 ms วินี้ได้รับการเสนอโดย Nagle (RFC896, 1984) ซึ่งเป็นส่วนหนึ่งของการควบคุมความคับคั่ง หลักการทำงานเป็นไปตามอัลกอริทึมต่อไปนี้

```

When the application produces data to send
if both the available data and the window >= MSS
    send a full segment
else
    if there is unACKed data in flight
        buffer the new data until an ACK arrives
    else
        send all the new data now

```

เมื่อแอปพลิเคชันมีข้อมูลต้องการส่งออก ส่งมายังชั้น TCP จะเริ่มตรวจสอบขนาด window มีขนาดใหญ่กว่า MSS หรือไม่ หากมีขนาดใหญ่กว่าจะส่งด้วย เชกเม้นต์ ขนาดเต็ม MSS แต่ถ้าไม่เข่นนั้นจะส่งไปและ

จับเวลาจนกว่าได้รับ แอ็คโนเผล็จเมนท์ หรือไม่ก็ส่งข้อมูลใหม่ ค่าเริ่มต้นของ TCP มีการใช้ Nagle's algorithm สามารถเลิกการทำงานโดยกำหนดค่า TCP_NODELAY

Adaptive Retransmission

TCP ออกแบบมาเพื่อเป็นระบบรับผิดชอบส่งข้อมูลจากต้นทางไปจนถึงปลายทาง ดังนั้นมีข้อมูลบางส่วนเดินทางไปไม่ถึงปลายทาง ระบบ TCP จะต้องสามารถแก้ไขปัญหานั้นได้โดยไม่ต้องรายงานถึงแอปพลิเคชัน ส่วนงานย่อยของ TCP ที่ทำหน้าที่ดูแลการส่งข้อมูลช้าเรียกว่า “Adaptive Retransmission”

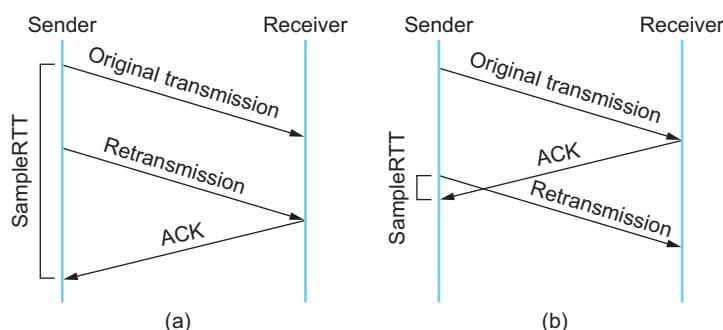
Original Algorithm การทำความเข้าใจเริ่มจากดูการทำงานพื้นฐานของระบบ timeout ระหว่างทั้งสองเครื่อง ซึ่งวิธีนี้ได้กำหนดให้มาตรฐานการสื่อสารของ TCP ในช่วงเริ่มแรก โดยการตรวจสอบค่าเฉลี่ย RTT (average RTT) โดยค่า timeout จะเป็นพังก์ชันของ RTT เมื่อมี แอ็คโนเผล็จเมนท์ จากแต่ละเซกเมนต์ ระบบ TCP โดยนำความแตกต่างของเวลาสองชุดมาเปรียบเทียบกันได้ค่าค่า SampleRTT เพื่อใช้คำนวณ EstimateRTT ตามสมการต่อไปนี้

$$\text{EstimatedRTT} = \alpha \times \text{EstimatedRTT} + (1 - \alpha) \times \text{SampleRTT}$$

ค่า α มีเพื่อปรับ EstimateRTT ให้เรียบขึ้น การปรับค่า α เพียงเล็กจะทำให้ EstimatedRTT เปลี่ยนแปลงไปมาก ค่าพื้นฐานของ α อยู่ในช่วง 0.8 ถึง 0.9 เมื่อได้ EstimatedRTT นำไปคำนวณค่า TimeOut ได้ดังนี้

$$\text{TimeOut} = 2 \times \text{EstimatedRTT}$$

Karn/Partridge Algorithm เมื่อใช้งาน อินเทอร์เน็ต ผ่านมาหลายปี เริ่มมองเห็นปัญหา Original Algorithm จากการคำนวณ SimpleRTT ดังรูปที่ 4.10(a) อย่างกรณีปกติเมื่อ แอ็คโนเผล็จเมนท์ เดินทางกลับมาถึงภายในระยะเวลา TimeOut และในรูปที่ 4.10(b) แอ็คโนเผล็จเมนท์ เดินทางกลับมาช้ากว่า TimeOut ทำให้เกิดการส่งข้อมูลอีกครั้ง



รูปที่ 4.10: การทำงานกับ แอ็คโนเผล็จเมนท์: (a) การทำงานปกติ (b) การส่ง ส่งข้อมูลอีกครั้ง
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

Jacobson/Karels Algorithm

วิธีของ Karn/Partridge algorithm ได้ยกระดับการทำงานของ TCP ขึ้นจากเดิมเมื่อเครือข่ายมีการใช้งานหนาแน่น ทำให้การคำนวณ RTT แม่นยำขึ้นถึงแม้เครือข่ายพบปัญหาจากความคับคั่ง แต่ที่ผ่านมาปัญหาจากความคับคั่งไม่ได้หายไป ในปี ค.ศ. 1988 นักวิจัยสองคน Jacobson (1988) และ Karels และคณะ (1988) ได้เสนอวิธีรับมือกับปัญหาการเพิ่มขึ้นของความคับคั่ง ซึ่งจะกล่าวถึงโดยละเอียดในหัวข้อต่อไป ในหัวข้อนี้จะสนใจแนวคิดที่เกี่ยวข้องกับการคำนวณเวลาเพื่อใช้ในการตัดสินการ ส่งข้อมูลอีกครั้ง

ในการทำความเข้าใจ timer มีความเกี่ยวข้องกับการควบคุมความคับคั่ง อย่างไรนั้น เริ่มต้นจาก เมื่อไก่ล้วน timeout จะไม่สนใจการ ส่งข้อมูลอีกครั้ง เชกเมนต์ เพราะเป็นการเพิ่มปริมาณข้อมูลในเครือข่าย เมื่อทำให้ปริมาณโหลดเครือข่ายลดลงจะทำให้การคำนวณ RTT แม่นยำขึ้น การควบคุม timeout นี้จึงส่งผลทาง อ้อมเป็นการควบคุมความคับคั่งไปในตัว

ปัญหาหลักของการคำนวณ RTT มาจากการคำนวณไม่ได้ใช้ค่า ความแปรปรวน(variance) ของ sample RTT หากค่าความแปรปรวนของ RTT มีน้อยจะทำให้ EstimateRTT มีความเที่ยงตรงขึ้น แนวทาง การคำนวณจึงเป็นดังต่อไปนี้

$$\begin{aligned} \text{Difference} &= \text{SampleRTT} - \text{EstimatedRTT} \\ \text{EstimatedRTT} &= \text{EstimatedRTT} + (\text{delta} \times \text{Difference}) \\ \text{Deviation} &= \text{Deviation} + \text{delta} (|\text{Difference}| - \text{Deviation}) \end{aligned}$$

เมื่อ delta เป็นค่าอัตราส่วนมีค่าอยู่ระหว่าง 0 ถึง 1 คำนวณจากค่าเฉลี่ยของ RTT

ซึ่ง TCP คำนวณค่า timeout ได้จากการ EstimateRTT และ Deviation ตามสมการต่อไปนี้

$$\text{TimeOut} = \mu \times \text{EstimatedRTT} + \phi \times \text{Deviation}$$

จากการทดลอง mu กำหนดให้มีค่าเป็น 1 และ phi มีค่าเป็น 4 เมื่อ ความแปรปรวนน้อย TimeOut จะใกล้เคียงกับ EstimateRTT แต่เมื่อ ความแปรปรวน มีค่ามากจะทำให้ค่า Deviation ชดเชยค่าให้เที่ยงตรงขึ้น

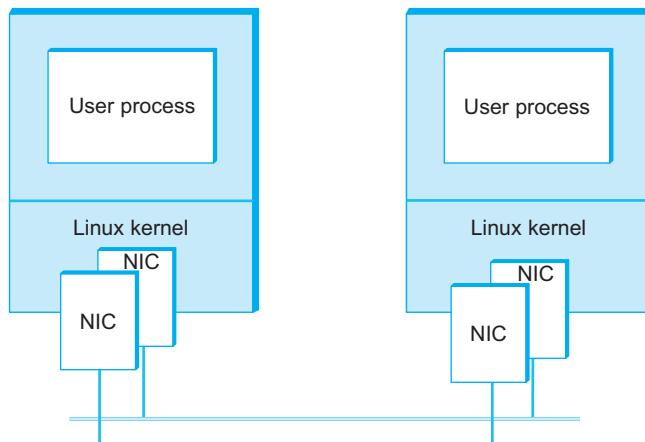
4.2.6 ค่าประสิทธิภาพ

จากบทที่ 1 ได้กล่าวถึงค่าประสิทธิภาพจำนวนสองค่าได้แก่ latency และ throughput ซึ่งมีปัจจัยเกี่ยวข้อง กับปัจจัยหลายประการหลายส่วนอาทิเช่น propagation delay และ link bandwidth รวมถึงการทำงานของ ซอฟต์แวร์ สำหรับค่าประสิทธิภาพที่จะกล่าวถึงในหัวข้อนี้จะเกี่ยวข้องกับการทำงานของซอฟต์แวร์ที่ส่งผลต่อ ประสิทธิภาพ

ต่อไปนี้จะเติมเต็มความสมบูรณ์ของการใช้ซอฟต์แวร์ในการวัดทดสอบประสิทธิภาพและ อธิบายถึง ความหมายของการทดสอบประสิทธิภาพ ความสำคัญในการวัดที่มองเห็นในระดับชั้นแอปพลิเคชัน

เนื่องจากการวัดและทดสอบประสิทธิภาพโดยการออกแบบการทดลอง แต่กรณีนี้มีการทดสอบใน เครื่องคอมพิวเตอร์จำนวนสองเครื่องมีชีพิชญ์ Xeon Dual core 2.4-GHz รันด้วยระบบปฏิบัติการลีนุกซ์(linux) มีความเร็วในการสื่อสารมากกว่า 1 Gbps เชื่อมต่อผ่านทางการ์ดเครือข่ายอีเทอร์เน็ต เทียบชื่อว่า NIC อธิบาย

ในรูปที่ 4.11 ในรูปได้เขียนต่อระหว่างเครื่องสองเครื่องทำให้ล่าเดย์ค่า propagation delay ได้ในการทดลองนี้จะวัดค่าประสิทธิภาพของ processor/software overheads ได้ โดยการรันโปรแกรมให้เปิดช่องการการสื่อสารผ่านซีอกเก็ต



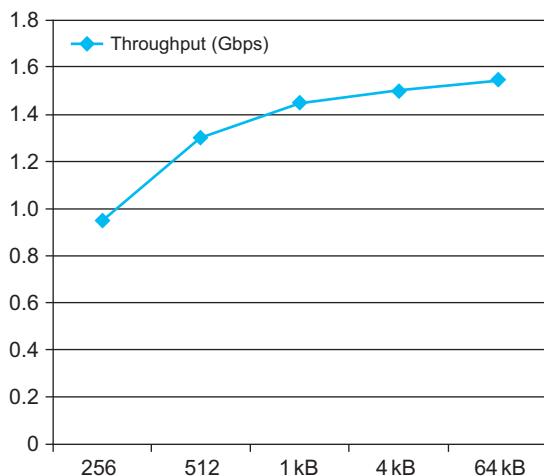
รูปที่ 4.11: การทดสอบระบบ: ลินก์สองเครื่องเชื่อมต่อกันโดยใช้การต่อสายเครื่อข่ายเครื่องละสองใบ
ลิขสิทธิภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ผู้อ่านอาจจะสงสัยว่าการทดลองนี้ไม่ได้ระบุขอบเขตของฮาร์ดแวร์หรืออัตราเร็วของลิงก์ ในหัวข้อนี้ไม่ได้ต้องการศึกษาความเร็วของโปรโตคอลแต่ต้องการแสดงให้เห็นถึงรูปแบบวิธีการวัดการทดลองและการบันทึกการทดสอบประสิทธิภาพของโพรโทคอลเป็นหลัก

การทดสอบทรูพุต สำหรับแพ็กเก็ตที่มีขนาดแตกต่างกันใช้เครื่องมือทดสอบชื่อว่า TTCP ผลการทดสอบเป็นตามรูปที่ 4.12 จุดสังเกตสำคัญของกราฟอยู่ที่การเพิ่มขึ้นของทรูพุต ช่วงแรกเมื่อแพ็กเก็ตมีขนาดเพิ่มขึ้นทำให้ทรูพุตเพิ่มขึ้นตามสัดส่วนและเข้าใกล้ค่าสูงสุดทรูพุต 2 Gbps เป็นความสามารถของลิงก์

เครื่อข่ายที่ทดลองนี้เป็นเครื่อข่ายในพื้นที่ควบคุม หรือเรียกว่า มีการทำงานสมบูรณ์แบบ “perfect” เป็นเครื่อข่ายไม่ได้รับผลกระทบจาก delay หรือ loss ที่เกิดจากปัจจัยภายนอก ดังนั้นปัจจัยเพียงอย่างเดียวที่มีผลต่อประสิทธิภาพเป็นการทำงานของ TCP และฮาร์ดแวร์และซอฟต์แวร์ของเครื่อง โดยผลกระทบจะเริ่มเห็นผลชัดเมื่อต้องการสื่อสารกับเครื่อข่ายที่อยู่ไกล จะเห็นว่าค่าประสิทธิภาพได้รับผลกระทบจากข้อจำกัดของแบบดิจิทัลและเกิดข้อมูลสูญหายง่ายในเครื่อข่ายไร้สาย ซึ่งทั้งหมดนี้ส่งผลกระทบต่อประสิทธิภาพโดยรวมของเครื่อข่าย

ดังนั้นการทดสอบประสิทธิภาพพื้นฐานของการขนส่งข้อมูลจากต้นทางไปปลายทาง ซึ่งความประสิทธิภาพยังมีค่าอื่นที่น่าสนใจ เช่น ค่าเดลaiy ค่าการใช้หน่วยประมวลผล ค่าปริมาณหน่วยความจำ ค่าข้อมูลสูญหาย และอื่นๆ ซึ่งเกิดขึ้นกับเครื่อข่ายที่ใช้งานจริง



รูปที่ 4.12: การทดลองวัดค่า throughput โดยมีแพ็กเก็ตหลายขนาด
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

บทที่ 5

การควบคุมความคับคั่ง

ปัญหา : การจัดสรรทรัพยากร

ปัญหาการจัดสรรทรัพยากรเป็นปัญหาที่พบโดยทั่วไป เปรียบได้กับปากาหนึ่งแห่งแต่ที่มีผู้คนต้องการใช้งานจำนวนมาก หากต้องการจัดสรรให้ใช้ได้โดยเกิดปัญหาจะต้องมีกระบวนการจัดสรรทรัพยากรดังกล่าว ยกตัวอย่างเช่น ปัญหารถติด เกิดกับถนนที่มีขนาดไม่พอสำหรับรถยนต์ทุกคันทำให้เกิดจากการแย่งเข้าไปอยู่ในพื้นที่ถนน บรรทุกทรัพยากรที่นี่คือถนน และผู้ต้องการใช้งานทรัพยากรคือรถยนต์ จึงมีข้อตกลงที่ระบุให้รถยนต์ทุกคันตกลงร่วมกันกลยุทธ์เป็นมีกฎหมายด้านการจราจร เป็นต้น สังเกตได้ว่าจากตัวอย่างปัญหานี้คือถนนมีขนาดจำกัดไม่เพียงพอต่อจำนวนรถยนต์ ซึ่งเป็นสิ่งไม่อาจหลีกเลี่ยงได้ มีข้อจำกัดหลายประการที่ไม่สามารถทำขนาดถนนให้เพียงพอสำหรับทุกคน และในความเป็นจริงไม่จำเป็นต้องสร้างถนนให้มีขนาดเพียงพอต่อรถทุกคัน เช่นเดียวกับ ปัญหาที่พบกับเครื่อข่ายคอมพิวเตอร์ มีความต้องการและข้อจำกัดไม่แตกต่างกันนัก มีคอมพิวเตอร์จำนวนมากต้องการใช้เครือข่าย แต่มีข้อจำกัดไม่สามารถจัดสรรให้มีเครือข่ายเพียงพอสำหรับคอมพิวเตอร์ทุกเครื่อง อย่างไรก็ตามไม่ใช่คอมพิวเตอร์ทุกเครื่องต้องการติดต่ออินเทอร์เน็ตตลอดเวลาหรือต้องการสื่อสารด้วยความเร็วสูงสุดตลอดเวลา ซึ่งนอกจากนี้ยังมีประเด็นอื่นที่น่าสนใจในการใช้แก้ปัญหาที่จะกล่าวถึงในลำดับต่อไป

จากที่ผ่านมาได้กล่าวถึงนั้นการแบ่ง เลเยอร์ เพื่อใช้ศึกษาพรโทคอลได้ง่าย ลำดับต่อไปจะกล่าวถึง วิธีการแบ่งปันทรัพยากรในมุมมองของความเท่าเทียมกันหรือการแบ่งอย่างมีประสิทธิภาพนั้นต้องการการควบคุมจำนวนผู้ใช้ ซึ่งผู้ใช้จำนวนมากก็ต้องการให้ตนเองได้สิทธิส่งข้อมูลด้วยความเร็วสูงเป็นระยะเวลานานเท่าที่เป็นได้ แต่เมื่อมีหนึ่งคนได้สิทธิส่งข้อมูลความเร็วเต็มที่ของเครือข่ายแล้วจะทำให้ผู้อื่นที่อยู่ในเครือข่ายเดียวกันนั้นไม่สามารถใช้เครือข่ายนั้นได้ ในการแบ่งใช้ทรัพยากรนี้ต่างฝ่ายต่างต้องการได้ทรัพยากรสูงที่สุด ดังนั้นการจัดสรรทรัพยากรให้แก่ผู้ใช้จำนวนมากนั้นยังถือว่าเป็นปัญหาที่มีความซับซ้อนอยู่ภายใน

จากบทที่ผ่านมาได้เรียนรู้การทำงานเป็นลำดับชั้นในแต่ละโพรโทคอลที่แยกอย่างເลື່ອພօສໍາຫຼັບເຮັດວຽກ รູ້ທີ່ລະສ່ວນໂດຍง່າຍ ในการແລກປ່ຽນຂໍ້ມູນແຕ່ລະເລຍືອຣີໃໝ່ມີມະບວນການອ່ານເຊີພາສ່ວນເຂດເດວກ່ອງຈິງເຖິງປີໄດ້ກັບກາຮັບສ່າງ ເອີເປີ(Application Programming Interface)ໃນດ້ານການເຂີຍໂປຣແກຣມ

วิธีจัดสรรทรัพยากรให้มีประสิทธิภาพ และ ทุกแพັກເກີດໄດ້ສິຫຼືທີ່ເທົ່າເຖິງກັນ ທັງໃນການນຶື່ມໂໂສຕ໌ จำนวนมาก หรือເພີ່ມໂໂສຕ໌ເດີຍກົດຕາມ เป็นปัญหาของการแบ่งทรัพยากรที่มีอย่างจำกัด ซึ่งทรัพยากรนີ້หมายถึง แบบດົວດົງຂອງລິງກົດແລະ ບັຟເພື່ອບັນເຮົາເຕືອນ ອ້າວີ ສວິທີ ແພັກເກີດເຕີນທາງມາຈາກຫລາຍເສັ້ນທາງມີປລາຍທາງແຕກຕ່າງກັນອອກໄປ ແຕ່ໃຫ້ทรัพยากรໃນການຂັ້ນສ່ວນກັນ ເປົ້າໝາຍຂອ້ພັກເກີດຄືອເດີນທາງໃຫ້ປລາຍທາງໂດຍຮົວ ແຕ່ຈະມີເພີ່ມຫົ່ງແພັກເກີດໃນຫົ່ງເວລາທ່ານັ້ນທີ່ໄດ້ສ່ວນຂໍ້ມູນ (ຈາກການທຳມະນຸດແບບເໜັງຫຼືສັ້ນຫຼັບສ່ວນ) ແພັກເກີດທີ່ແຍ່ງຂ່ອງສັ້ນຫຼັບສ່ວນໄມ້ໄດ້ຈັດໃຫ້ຮອຍໃນຄົວ ການແຍ່ງກັນຂອງແພັກເກີດເພື່ອເຂົ້າບັຟເພື່ອແອົກຈາກບັຟເພື່ອທີ່ເຮົາເຕືອນ ເປັນເພື່ອໃຫ້ຄືອຮອງລິງກົດ ເນື່ອແພັກເກີດແຍ່ງກັນນັກນາມເກີນໄປ ແຕ່ມີເພີ່ມລິງກົດເດີຍ ຈະໃຫ້ຄົວເຕັມຈິງມີຜລເສີຍຫາຍາມມາເຊັ່ນ : ແພັກເກີດຈະດຽວປະຈຸບັນແລ້ວເກີດກາສົງໄໝມີອີກຄັ້ງເປັນກາເພີ່ມປຣິມານເຄືອຂ່າຍເຂົ້າສູ່ຮະບບໜ້າອີກ

และในกรณีที่ lever ร้ายที่สุด จากแพ็กเก็ตที่ส่งซ้ำจากการส่งไม่สำเร็จทำให้เกิด คิวเกิดปัญหา โอลูออร์ฟลั่ว และทุกแพ็กเก็ตจะถูกดรอป เพื่อจัดการกับสถานการณ์ตั้งกล่าวเครือข่ายส่วนใหญ่มีระบบควบคุมความคับคั่ง

การควบคุมความคับคั่งและการจัดสรรทรัพยากรเบรียบเสมือนหรือยุส滂ด้าน ในแต่หนึ่งหากออกแบบเครือข่ายให้ป้องกันไม่ให้เกิดความคับคั่งแบบไม่ให้เกิดความคับคั่งเลย เช่น จัดตารางเวลาแบบวงจรสวิตชิ่ง อาจทำให้แพ็กเก็ตได้เวลาในการใช้ลิงก์เท่ากันและหลีกเลี่ยงความคับคั่งได้ แต่วิธีนี้จะส่งผลให้มีการใช้ลิงก์โดยไม่เต็มประสิทธิภาพ ยกตัวอย่างเช่น มีโอสต์ที่ไม่ต้องการส่งข้อมูล แต่ทรัพยากรยังคงจัดสรรช่วงเวลาให้กับโอสต์นั้น แต่หากเป็นการออกแบบระบบที่ดี จะทำให้นำช่วงเวลาที่โอสต์ไม่ได้ส่งไปให้โอสต์อื่นได้ส่งจะทำให้โอสต์อื่นไม่เสียเวลาออนไลน์ช่วงดังกล่าวได้ แต่การจัดสรรทรัพยากรเครือข่ายด้วยความแม่นยำเป็นเรื่องยาก อย่างไรก็ตาม ทรัพยากรมีการกระจายไปทั่วเครือข่ายบริเวณหนึ่งของเครือข่ายอาจใช้ทรัพยากรเต็ม แต่บริเวณอื่นอาจไม่ถูกใช้ หากเลือกวิธีอนุญาตให้ส่งต้นทางของแพ็กเก็ตส่งข้อมูลได้มากเท่าที่ต้องการ แต่จะควบคุมเมื่อเกิดความคับคั่งขึ้น วิธีนี้เป็นวิธีที่ง่าย แต่อาจทำให้เกิดปัญหาได้ เนื่องจากแพ็กเก็ตอาจถูกดรอปจากเครือข่ายในบริเวณที่เครือข่ายมีความคับคั่ง เพราะมีแพ็กเก็ตเดินทางมาจนล้นออกจากคิวได้

การควบคุมความคับคั่งและการจัดสรรทรัพยากรเกี่ยวข้องกับทั้งโอสต์และองค์ประกอบรวมเครือข่าย เช่น เร้าเตอร์ ในอุปกรณ์เครือข่าย ซึ่งสัญญาณ เพื่อให้สามารถจัดระบบคิว เพื่อควบคุมการส่งแพ็กเก็ตเป็นไปอย่างมีลำดับ และตรวจสอบแพ็กเก็ตได้ถูกดรอป การจัดระบบคิวทำให้สามารถป้องกันไม่ให้แพ็กเก็ตของโอสต์ใดโอสต์หนึ่งลือครองทรัพยากรเพียงโอสต์เดียว ซึ่งวิธีการควบคุมความคับคั่ง จะเกี่ยวข้องกับการทำหนดความร่วงของข้อมูลต้นทางที่อนุญาตให้ส่งแพ็กเก็ต การดำเนินการนี้ทำขึ้นเพื่อพยายามป้องกันไม่ให้เกิดความคับคั่งตั้งแต่แรก

บทนี้อธิบายภาพรวมของการควบคุมความคับคั่ง และการจัดสรรทรัพยากร จากนั้นจะกล่าวถึงระบบคิวแบบต่างๆ ที่สามารถนำไปใช้กับเร้าเตอร์ ตามด้วยการอธิบายการทำงานของอัลกอริทึม ในกระบวนการควบคุมความคับคั่ง ของ TCP บนโอสต์ และกล่าวถึงเทคนิคิวต่างๆ ที่เกี่ยวข้องภายในเร้าเตอร์และโอสต์ ที่มีวัตถุประสงค์เพื่อเลี่ยงการเกิดความคับคั่ง

5.1 ประเด็นการจัดสรรทรัพยากรเครือข่าย

การจัดสรรทรัพยากรและการควบคุมความคับคั่งเป็นปัญหาที่ซับซ้อนซึ่งเป็นปัญหาอันดับแรกๆ ของการศึกษาด้านการออกแบบเครือข่าย ยังคงมีปัญหาวิจัยให้ศึกษาอยู่ตลอดเวลา ปัจจัยหนึ่งที่ทำให้ปัญหาการจัดสรรทรัพยากรมีความซับซ้อนคือข้อมูลไม่ถูกตัดแยกตั้งแต่ต้นทาง การจัดสรรทรัพยากรมีการทำงานเป็นส่วนๆ มีเร้าเตอร์ สวิตช์ และกระบวนการแขร์ลิงก์ภายในเครือข่าย และการทำงานในส่วนซึ่งกันส่ง ความเกี่ยวข้องกับองค์ประกอบจำนวนมากนี้มีโอกาสทำให้เกิดการทำงานขัดแย้งกันได้ ซึ่งมีแนวทางอีกแนวทางเรียกว่า “cross-layer design” ซึ่งเป็นวิธีที่เข้าไปควบคุมในทุกเลเยอร์ (ทำให้เกิดความซับซ้อนมากกว่าเดิม) อย่างไรก็ตามการแบ่งการทำงานเป็นเลเยอร์ช่วยลดความซับซ้อนในการทำความเข้าใจพร้อมกันได้ ทำให้วิธีนี้ยังคงได้รับความนิยมกว่า cross-layer design แต่มีปัญหาด้านการทำงานที่อาจขัดแย้งกัน

ก่อนลงในรายละเอียดมากล่าวถึงคำศัพท์ที่เกี่ยวข้องในบทนี้ แนวคิดการจัดสรรทรัพยากร หมายถึงกระบวนการในการจัดการเครือข่ายเพื่อให้ตอบสนองความต้องการที่มีผู้ใช้งานจำนวนมากและทรัพยากรกัน ซึ่ง

ผู้ที่เข้ามาແຍ່ງອາຈຈະເປັນແອປພລິເຄີບທີ່ຢູ່ໃນເຄື່ອງເດີວກນ ອົງຈາກອຸປະກິດອື່ນໃນເຄື່ອງຂ່າຍກີ້ໄດ້ ສໍາຮັບທັກພາກເຄື່ອງຂ່າຍໃນທີ່ນີ້ຈະເຂື່ອມໂຍງກັບແບນດົວດົກ ແລະ ຂາດບັບເພື່ອໃນຮ້າເຕອຣ໌ ອົງສວິຕີ່ຫຼື ອົງໃນການດົກເຄື່ອງຂ່າຍ ຈຶ່ງກາວະປົກທີ້ນີ້ມີທັກພາກນ້ອຍກວ່າຄວາມຕ້ອງການ ໝາຍຄວາມວ່າມີຜູ້ໃຊ້ຫຼື ແອປພລິເຄີບບາງຕ້ອງຈຳໄດ້ຮັບທັກພາກເຄື່ອງຂ່າຍນ້ອຍກວ່າທີ່ຕ້ອງການ ຈຶ່ງກາວະຄວບຄຸມໃຫ້ຜູ້ໃຊ້ບາງຄົນໄດ້ໃຫ້ບາງຄົນໄໝໃຫ້ທັກພາກເຄື່ອງຂ່າຍເປັນສ່ວນໜຶ່ງຂອງປັນຫາກາຈັດສຽງທັກພາກຄືກາຕົກສິນໃຈວ່າຈະປົງເສີມເມື່ອໄດ້ແລະປົງເສີມໃຈ

ໃນກາຮືກຂາສ່ວນແຮກຈະກ່າວຄືງຕັ້ງແບບເຄື່ອງຂ່າຍ ໂດຍຈຳກັດກອບກາຮືກຂາເພື່ອໃຫ້ສາມາດທຳຄວາມເຂົ້າໃຈຮາຍລະເອີຍດີທີ່ສໍາຄັນຂອງປັນຫາຄວາມຄັບຄັ້ງ ແລະ ຕັດອົງປະກອບທີ່ໄໝເກີ່ວຂ່ອງອອກໄປກ່ອນ

ກາຮືກຄຸມຄວາມຄັບຄັ້ງໃຫ້ອົບຍາກການທຳການຂອງໂທນດ ອົງໂອສຕໍ່ ດ້ວຍກາຮືກຄຸມໃຫ້ໂທນດໄໝສ່ວນໜຸ່ມ ຈົນທຳໃຫ້ເຄື່ອງຂ່າຍເກີດຄວາມຄັບຄັ້ງ ຈະເປັນສາເຫດໃຫ້ເກີດໄວ້ວົງໄຫລດ(overload) ເຄື່ອງຂ່າຍທີ່ໄໝມີກາຮືກຄຸມປົກຕິແລ້ວຈະມີຄວາມຄັບຄັ້ງ ທາກນີ້ຄື່ງວິທີກາຮືກຄຸມໄໝໃຫ້ເກີດຄວາມຄັບຄັ້ງອ່າງຈຳກັດ ສິ່ງແຮກທີ່ຈະນີ້ຄື່ງຄື່ສັ່ງໃຫ້ລູກຂ່າຍ 2-3ໂອສຕໍ່ຫຼຸດສ່ວນໜຸ່ມ ວິທີນີ້ຈະຫ່ວຍໃຫ້ຄົນອື່ນທີ່ອ່າຍ່ວ່າມີເຄື່ອງຂ່າຍໄດ້ຮັບສ່ວນທີ່ມີມືການເສີຍສະລະ ອ່າງໄຮກ້ຕາມ ກະບວນກາຮືກຄຸມຄວາມຄັບຄັ້ງ ຕາມປົກຕິມັກໃຫ້ຄວາມເປັນຮຽມກັບລູກຂ່າຍທຸກຄົນເສມອກາຄັກນ(ໃນບາງແຈ່ນຸ່ມ) ກ່າວຄື່ອງກະບວນກາປົກຕິຈະພາຍາມແປ່ງໃຫ້ທຸກຄົນໄດ້ຄວາມເດືອນຮັນເທົ່າກັນທັງໝົດ ແທນທີ່ຈະໃໝ່ມືການໄດ້ທັກພາກມາກ ແລະ ມືການທີ່ຕ້ອງເສີຍລະເພື່ອໃຫ້ຜູ້ໃໝ່ເກີ່ວຂ່ອງມີເຄື່ອງຂ່າຍໄດ້ທັກພາກມາກກວ່າຄົນອື່ນ ດັ່ງນັ້ນກະບວນກາຮືກຄຸມຄວາມຄັບຄັ້ງ ຈະມີຂັ້ນຕອນຈັດສຽງທັກພາກທີ່ມີຮາຍລະເອີຍດັບຫຼືອ່ານອຸ່ງກາຍໃນ

ສິ່ງທີ່ການທຳຄວາມເຂົ້າໃຈຄື່ອງການແຕກຕ່າງຮະຫວ່າງ ກາຮືກຄຸມໂຟລົວ ແລະ ກາຮືກຄຸມຄວາມຄັບຄັ້ງ ກາຮືກຄຸມໂຟລົວເກີ່ວຂ່ອງກັບກາຮືກຄຸມໃຫ້ຜູ້ສ່ວນຍັງຄົງໄດ້ສ່ວນເຮົວທີ່ສຸດເທົ່າທີ່ເຄື່ອງຮັບທຳໄດ້ ໃນທາງທຽບກັນຂໍ້າມ ກາຮືກຄຸມຄວາມຄັບຄັ້ງມີຈຸດມຸ່ງໝາຍເພື່ອປັບກັນໄມ້ໃຫ້ຜູ້ສ່ວນຈຳນວນມາກສ່ວນໜຸ່ມເຂົ້າເຄື່ອງຂ່າຍເຄື່ອງຂ່າຍ ມາກເກີນໄປເນື່ອຈາກທັກພາກເຄື່ອງຂ່າຍໃນບາງເສັ້ນທາງໄໝເພີຍພວ

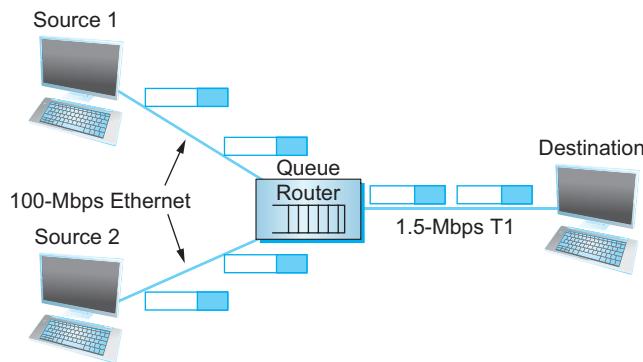
5.1.1 ຕັ້ງແບບເຄື່ອງຂ່າຍ

ຕັ້ງແບບເຄື່ອງຂ່າຍຄື່ອງກາຮືກຄຸມໂຟລົວໃນແບນນາມຮຽມ ໂດຍຕັດສ່ວນທີ່ໄໝສໍາຄັນຕ່ອງກາຮືກຄຸມທີ່ອກໄປໃຫ້ເຫຼືອເພາະສ່ວນທີ່ສັນໃຈ ສໍາຮັບຕັ້ງແບບເຄື່ອງຂ່າຍໃນທີ່ນີ້ຈະກ່າວຄື່ງເຄື່ອງຂ່າຍທີ່ໄໝຄວາມສັນໃຈກາຈັດສຽງທັກພາກເຄື່ອງຂ່າຍ ຕັ້ງແບບປະກອບດ້ວຍ ໂທນດ ແບນດົວດົກ ກາຮືກຄຸມ ເຂື່ອມຕ່ອ ແລະ ບັບເພື່ອໃຫ້ຮ້າເຕອຣ໌ ໃນແຕ່ລະໂທນດ ຕາມອົບຍາຍໃນຮູ້ທີ່ 5.1

ເຄື່ອງຂ່າຍແບບແພັກເກີດສວິຕີ່

ກາຈັດສຽງທັກພາກໃນການສື່ອສາງເຖິງໂລຢີແພັກເກີດສວິຕີ່(ຮະບບອິນເທୋຣັນເນື້ອແປັນແພັກເກີດສວິຕີ່) ແລະ ມີສ່ວນປະກອບອືກມາກ ເຫັນ ລົງກົງແລະສວິຕີ່ຫລາຍດ້ວຍ ອົງເຮົາເຕອຣ໌ ເພຣະການທຳການສ່ວນໃຫຍ່ເປັນການທຳການສ່ວນຕິດຕ່ອນເທୋຣັນເນື້ອ ດັ່ງນັ້ນອຸປະກິດເຄື່ອງຂ່າຍໃນທີ່ນີ້ຈະກ່າວຄື່ງເພາະເຮົາເຕອຣ໌ ໃນທີ່ນີ້ຈະໃໝ່ເຮົາເຕອຣ໌ແທນອຸປະກິດທີ່ໄໝເກີດເຄື່ອງຂ່າຍແລະລະເວັນການທຳການຂອງສວິຕີ່ໄວ້ໃນຈານທີ່ເຫຼົ້າໃຈວ່າມີກາຍໃນກາຈັດສຽງຂັ້ນລົງກົງ

ໃນສານກາຮືກຄຸມເຄື່ອງຂ່າຍແບບນີ້ ມີໂອສຕໍ່ຈຳນວນມາກຍູ່ໃນເຄື່ອງຂ່າຍແລະ ຕ້ອງການສ່ວນໜຸ່ມຕົວດ້ວຍ ຈຶ່ງຄວາມຈຸຂອງເຄື່ອງຂ່າຍຈະມີເພີຍ ມີປັນຫາທີ່ນໍາສັນໃຈກາຮືກຄຸມໂຟລົວໃນຮູ້ທີ່ 5.1 ເຄື່ອງຕັນທາງມີຈຳນວນມາກ ຄື່ງແມ້



รูปที่ 5.1: การเกิดปัญหาของขดกับเร้าเตอร์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ลิงก์จากต้นทางจะมีขนาดใหญ่เพียงพอในการรองรับเครือข่ายจากแต่ละเครื่องแต่ที่ปลายทางมีเครือข่ายขนาดเล็กทำให้เกิดปัญหาของขดกับเร้าเตอร์ หากลองขยายปัญหานี้อีกด้านหนึ่งคือในเครือข่ายอาจมีลิงก์ที่ไม่ได้ใช้งานเลยก็ได้และหากออกแบบระบบไม่เดียวจะทำให้เกิดการกระจายตัวของข้อมูลในบางลิงก์ และไม่ได้ใช้งานลิงก์ที่ยังว่างอยู่ (ปัญหานี้ยังคงเกิดกับระบบ TCP ในปัจจุบัน)

โฟล์วแบบไม่กำหนดการเชื่อมต่อ

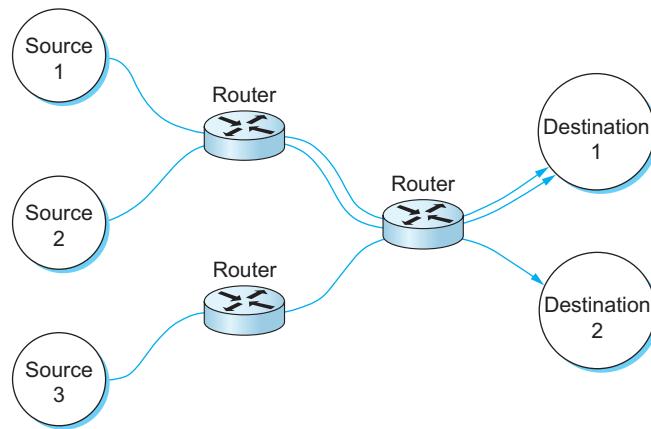
ไม่กำหนดการเชื่อมต่อเป็นการส่งข้อมูลแบบไม่ตรวจสอบความสำเร็จ ซึ่งเป็นวิธีที่ใช้ส่งข้อมูลนับจากชั้นกายภาพมาจนถึงชั้นส่งข้อมูล ยกเว้นการส่งข้อมูลแบบ TCP ที่ออกแบบให้มีการตรวจสอบการส่งสำเร็จ

เรียกแพ็กเก็ตที่มีคุณสมบัติเป็นไม่กำหนดการเชื่อมต่อ ว่า เดتاเกรม เดตาเกรม ไม่มีกระบวนการตรวจสอบข้อมูลมากนักทำให้ทำงานได้เร็ว แต่อาจเดินทางลึกลายไม่เป็นลำดับ มีการส่งข้อมูลด้วยอัตราเร็วเกินกว่ารับทัน และปัญหาอื่นๆซึ่งจะกล่าวถึงในหัวข้อต่อไป โฟล์วของเดตาเกรมคือข้อมูลที่เดินทางจากต้นทางถึงปลายโดยประกอบด้วยข้อมูลส่วนเยดเดอร์ดังนี้

(SrcPort, SrcIPAddr, DstPort, DstIPAddr)

โฟล์วสามารถเป็นการสื่อสารไฮสต์ต่อไฮสต์โดยตรง เช่น มีที่อยู่ไฮสต์ต้นทาง/ปลายทางเหมือนกัน หรือเป็นการส่งแบบมีต้นทางเดียวหน้าปลายทาง หรือในกรณีเป็นการสื่อสารของ ไฮสต์/พอร์ตต้นทาง/ปลายทางเหมือนกัน เร้าเตอร์สามารถมองเห็นโฟล์วโดยการอ่านข้อมูลจากแพ็กเก็ตเยดเดอร์ ข้อมูลภายในเยดเดอร์จะแสดงถึงข้อมูลการเชื่อมต่อแบบ end-to-end ในรูปที่ 5.2 แสดงโฟล์วต่างๆ ที่เหล่านเร้าเตอร์

เนื่องจากเดตาเกรมเกี่ยวข้องกับโฟล์วที่เหล่านเร้าเตอร์แต่ละตัว โฟล์วที่ผ่านเร้าเตอร์ จะเก็บสถานะบางอย่างไว้สำหรับใช้ตัดสินใจในการจัดสรรทรัพยากรเครือข่าย สถานะนี้เรียกว่า “สถานะผ่อนปรน(soft state)” ขณะที่ “สถานะเข้มงวด(hard state)” เป็นการตรวจสอบสถานะการเชื่อมต่อแบบเข้มงวด (พบในระบบ TCP) ความแตกต่างคือ สถานะผ่อนปรน ไม่จำเป็นต้องสร้างและลบสถานะโดยการส่งสัญญาณควบคุม ซึ่งสถานะผ่อนปรน แสดงถึงจุดที่เครือข่ายที่ไม่มีการเชื่อมต่อได้โดยไม่จำเป็นต้องระบุสถานะในเร้าเตอร์ สำหรับเครือข่ายที่เน้นความซัคเจนในการเชื่อมต่อยังคงใช้สถานะเข้มงวด



รูปที่ 5.2: ไฟล์ผ่านเร้าเตอร์จำนวนหลายไฟล์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

โดยทั่วไป การทำงานของเครือข่ายไม่ได้ขึ้นอยู่กับสถานะผ่อนปรน ที่มีอยู่ โดยแต่ละแพ็กเก็ตกำหนดเส้นทางได้โดยไม่ต้องคำนึงถึง สถานะผ่อนปรน) แต่หากมีการใช้ สถานะผ่อนปรน ในเร้าเตอร์ เพื่อรักษาสถานะแพ็กเก็ต จะสามารถจัดการกับแพ็กเก็ตได้ดีกว่า

เมื่อมีไฟล์ในระบบ จะทำให้เร้าเตอร์สามารถกำหนดทรัพยากรให้กับแต่ละไฟล์ได้ง่าย เช่นควบคุมอันหนึ่งมาก่อนมาหลัง

ตัวแบบบริการ

จากที่ผ่านมา ได้กล่าวถึงกระบวนการที่ต้องการให้อินเทอร์เน็ตบริการได้ดีที่สุด เป็นการทำงานที่ให้แพ็กเก็ตได้รับโอกาสเท่าเทียมกัน โดยโไฮสต์ปลายทางไม่มีสิทธิขอสิทธิในการส่งข้อมูลด้วยบริการพิเศษ เช่นต้องการได้รับบริการที่สามารถกำหนดความเร็วขั้นต่ำได การขอรับบริการพิเศษหรือเรียกว่า “การประกันคุณภาพของบริการ(Quality of Service)” เป็นความต้องการบริการที่ขอให้มีให้บริการกำหนดขอบเขตขั้นต่ำที่ตนเองจะได้รับบริการ เช่น การรับประกันแบบเดวิดอทที่จำเป็นสำหรับการสตรีมวิดีโอ รูปแบบการบริการที่กล่าวถึงนี้เกี่ยวข้องกับแนวทางที่ออกแบบให้แต่ละไฟล์ได้รับการรับประกันเชิงคุณภาพ

5.1.2 Taxonomy

วิธีการจัดสรรทรัพยากรมีการเสนอในรูปแบบที่แตกต่างกันหลายรูปแบบสำหรับวิธีการทำความเข้าใจ จะจัดกลุ่มเป็นทีละสองกลุ่ม การจัดประเภทวิธีการจัดสรรทรัพยากรเป็นสามแนวทางดังนี้

Router-Centric เทียบกับ Host-Centric

Router-Centric คือให้เร้าเตอร์เป็นศูนย์กลางในการจัดสรรทรัพยากร และ Host-Centric คือให้ไฮสต์เป็นศูนย์กลางในการตัดสินในการจัดสรรทรัพยากร Router-Centric ให้สิทธิเร้าเตอร์ เป็นผู้ตัดสินใจว่าจะให้ข้อมูลได้สิทธิส่งข้อมูล และข้อมูลใดจะต้องรอไปก่อน ซึ่งการทำงานด้วยวิธี Router-Centric ยังต้องการความร่วม

มือจากไฮสต์ เช่น เร้าเตอร์พบว่าเครือข่ายกำลังเกิดความคับคั่งจะส่งคำสั่งให้บานไฮสต์หยุดส่งข้อมูลชั่วคราว หากไฮสต์ไม่ตอบสนองก็จะทำให้เครือข่ายยังคงมีความความคับคั่ง

Reservation-Based เทียบกับ Feedback-Based

Reservation-Based เป็นการจองทรัพยากรไว้ล่วงหน้า และ Feedback-Based จะเกิดการจัดสรรทรัพยากรเมื่อได้รับการร้องขอ

Window Based เทียบกับ Rate Based

Window Based ใช้วิธีอ่านค่าพื้นที่หน่วยความจำของบัฟเฟอร์ ระบบจะควบคุมให้การส่งข้อมูลอยู่ภายในกรอบ window สำหรับ Rate Based ใช้วิธีตรวจสอบอัตราเร็วของข้อมูลและควบคุมอัตราเร็วอยู่ในระดับไม่เกิดขีดจำกัด

สรุปวิธีจัดกลุ่มแนวทางบริหารทรัพยากร

การจัดประเภทวิธีการจัดสรรทรัพยากร จำกที่กล่าวมาได้แบ่งรูปแบบจาก 3 กลุ่ม โดยแบ่งกลุ่มละสองด้านที่แตกต่างกัน จากการสังเกตอาจเห็นได้ว่ามีแนวทางทั้งหมดที่แนวนอนทั้งหมดที่แตกต่างกัน แต่ในทางปฏิบัติ การแบ่งเป็นสองด้านทำความเข้าใจง่ายกว่า การนำทุกวิธีมาใช้ในการอธิบาย ใน การแบ่งวิธีจัดสรรทรัพยากร สามารถสรุปได้ดังนี้

ด้านหนึ่ง ใช้รูปแบบการบริการแบบ Best-effort โดยปล่อยให้ระบบทำงานโดยควบคุมน้อยที่สุด นี้คือวิธีทั่วไปที่ใช้ในอินเทอร์เน็ต

อีกด้านหนึ่ง ใช้รูปแบบบริการที่ใช้ การประกันคุณภาพของบริการ หมายถึงวิธีการส่วนทรัพยากรไว้สำหรับบางบริการ หรือการเปิดให้ระบบยอมรับคำขอจากลูกค้าย เช่น การขออัดตัวซึ่งจะมีความแตกต่างกันไปตามระดับของทรัพยากรที่ส่วนไว้ ซึ่งจะกล่าวถึงหัวข้อนี้ในหัวข้อต่อไป

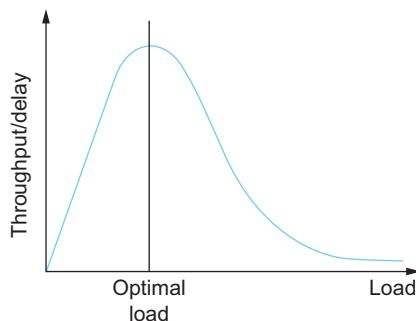
5.1.3 ปัจจัยในการประเมินประสิทธิภาพ

ปัจจัยอย่างง่ายที่ใช้ในการประเมินประสิทธิภาพของการจัดสรรทรัพยากรเป็นการ พิจารณาจากสองประการ ได้แก่ ทรัพุตและดีเลย์ โดยทรัพุตมีเป้าหมายให้ได้ค่ามากที่สุด ขณะที่ดีเลย์มีเป้าหมายให้ได้ค่าน้อยที่สุด ปัจจัยทั้งสองประการนี้มีความต้องการขัดแย้งกันเอง วิธีหนึ่งที่ใช้เป็นอัลกอริทึมการจัดสรรทรัพยากรเพื่อเพิ่มปริมาณทรัพุตคืออนุญาตให้แพ็กเก็ตส่งได้มากที่สุดเท่าที่เป็นไปได้ เพื่อที่จะทำให้มีการใช้ทรัพยากรลิงก์เต็ม 100% เพื่อไม่ให้มีลิงก์ไม่ได้ใช้งาน เนื่องจากลิงก์ที่ไม่ได้ใช้งานมักจะส่งผลต่อทรัพุต แนวทางนี้เกิดปัญหาตามมาคือการเพิ่มจำนวนแพ็กเก็ตในเครือข่ายยังเป็นการเพิ่มความยาวของคิวในเรอเตอร์แต่ละตัวด้วย ซึ่งในทางกลับกัน คิวที่ยาวขึ้น หมายความว่าแพ็กเก็ตจะมีดีเลย์เพิ่มขึ้นในเครือข่าย

เพื่ออธิบายความสัมพันธ์ของทั้งสองค่า ได้เสนอให้มีอัตราส่วนของทรูพุตต่อดีเลย์เป็นค่าในการชี้วัดประสิทธิภาพของการจัดสรรทรัพยากร อัตราส่วนนี้เรียกว่า “Power of the network (กำลังงานทางเครือข่าย)” เป็นตามสมการต่อไปนี้:

$$\text{Power} = \text{Throughput} / \text{Delay}$$

ค่ากำลังงานเครือข่ายนี้ไม่ได้แปรผลตรงไปตรงมา เป็นหลังเป็นทฤษฎีของระบบคิวแบบ M/M/1¹ โดยที่อักษร M มาจาก márko維n(Markovian) คือทั้งการมาและการให้บริการเป็นเอ็กซ์โพเนนเชียล



รูปที่ 5.3: จุดเดียวอยู่ตรงจุดที่ทำให้ทรูพุตกับดีเลย์มีโหลดสูงสุด
ลิขสิทธิภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

เป้าหมายเพื่อให้ได้ค่า Power สูงสุด ซึ่งมีทรูพุตสูงสุด ดีเลย์น้อยสุด ซึ่งเกิดจากปริมาณโหลดปราศในเครือข่าย ในทางกลับกัน โหลดสามารถกำหนดโดยกระบวนการจัดสรรทรัพยากรูปที่ 5.3 กำหนดให้มีส่วน Power เป็นตัวแuren ของค่าสูงสุดของระบบ วิธีการจัดสรรทรัพยากรจะทำงานที่ควบคุมให้ระบบเข้าสู่จุดสูงสุดของส่วนโคงนี้ ทางด้านซ้ายมีอัตราจุดยอด แทนปริมาณโหลดกำลังเพิ่มขึ้น ด้านขวา มีอัตราจุดยอดแทนปริมาณโหลดกำลังลดลง

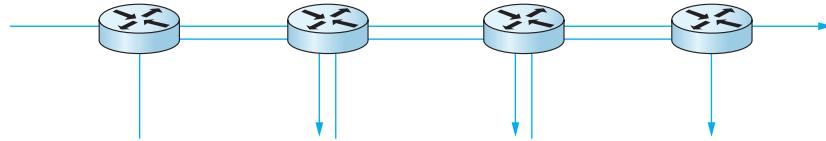
การจัดสรรทรัพยากรอย่างยุติธรรม

การใช้ทรัพยากรเครือข่ายอย่างมีประสิทธิภาพไม่ได้เป็นเพียงเกณฑ์เดียวของการจัดสรรทรัพยากร ในเครือข่ายคอมพิวเตอร์มีเรื่องความเป็นธรรมอยู่ด้วย อย่างไรก็ตาม หัวข้อนี้ยัง hacความชัดเจนของการจัดสรรความยุติธรรมได้ยาก หากต้องการระบุว่าอะไรคือการจัดสรรทรัพยากรอย่างยุติธรรม ยังคงดูเป็นการทำหนดคำจำกัดความได้ยาก ตัวอย่างเช่น รูปแบบการจัดสรรทรัพยากรตามการจอง โครงการก่อนได้ใช้ทรัพยากรคือว่า yutিธรรม สำหรับคนที่มาจองก่อน แต่ถ้าในระบบมีคน(อุกรณ์)ที่ทำงานเร็วกว่าคนอื่นจะต้องลดเวลาคนอื่นไม่ได้รับบริการเลยยังถือว่า yutิธรรมอยู่หรือไม่

ในกรณีที่ไม่มีข้อมูลชัดเจนว่าจะสังเกตความ yutิธรรมด้านใดนั้น ในกรณีมีหลายไฟล์ลิงก์ร่วมกันระบบต้องการให้แต่ละไฟล์ได้รับส่วนแบ่งแบบดีวิดร์เท่ากัน คำจำกัดความของความ yutิธรรมในที่นี้คือ ส่วนแบ่งแบบดีวิดที่ที่ yutิธรรมหมายถึงส่วนแบ่งแบบดีวิดร์เท่ากัน แต่ถึงแม้จะมีการจองพื้นที่ไม่เท่ากัน อาจก่อ

¹อัตราห่างของการเดินทางแท็กเก็ตเข้ามาในระบบมีความน่าจะเป็นของการกระจายค่าแบบ บัวช(poisson) ใช้ระยะเวลาการให้บริการมีค่าความน่าจะเป็นในการใช้เวลาให้บริการแบบ เอ็กซ์โพเนนเชียล(exponential) มีผู้ให้บริการเดียว มีความยาวค่าไม่จำกัด และการจัดลำดับคิวเป็นแบบ FIFO(first in, first out)

ให้เกิดความไม่ยุติธรรมขึ้นได้ จึงควรพิจารณาความยาวของเส้นทางที่ไฟล์ต้องการใช้เพื่อส่งถึงปลายทางมาเปรียบเทียบด้วยหรือไม่? อธิบายตามตัวอย่างดังนี้รูปที่ 5.4 การแบ่งปันแบบด์วิดธ์อย่างไรให้ยุติธรรมระหว่างหนึ่งไฟล์ต้องการทรัพยากรในการส่งสื่อระยะห่างทางเครือข่าย กับสามไฟล์ต้องการทรัพยากรไฟล์ละหนึ่งระยะห่างทางเครือข่าย



รูปที่ 5.4: ใช้เปรียบเทียบความเท่าเทียม หนึ่งไฟล์ต้องการสื่อระยะห่างทางเครือข่าย กับ สามไฟล์ต้องการไฟล์ละหนึ่งระยะห่างทางเครือข่าย
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

สมมติความหมายของความยุติธรรมคือมีเท่ากันและทุกเส้นทางต้องใช้ระยะเวลาส่งเท่ากัน มีการศึกษาของ Jain และคณะ (1999) นักวิจัยด้านเครือข่ายได้เสนอตัวชี้วัดที่จะใช้วัดความเป็นธรรมเป็นปริมาณได้ซึ่งสามารถนำไปใช้เพื่อควบคุมความคับคั่งได้ เรียกตัวชี้วัดนี้ว่า “ตัวชี้ของเจน(Jain's index)” ซึ่งกำหนดได้ดังนี้

$$(x_1, x_2, \dots, x_n) \quad (5.1)$$

ค่า x_n บันทึกวัดในหน่วย บิต/วินาที พังก์ชันต่อไปนี้กำหนดตัวชี้ความเป็นธรรมให้กับแต่ละไฟล์:

$$f(x_1, x_2, \dots, x_n) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2} \quad (5.2)$$

ตัวชี้ของเจนมีผลลัพธ์อยู่ระหว่าง 0 ถึง 1 เมื่อ โดยที่ 1 แสดงถึงความเป็นธรรมมากที่สุด เพื่อให้เข้าใจพื้นฐานเบื้องหลังการวัดนี้ ให้พิจารณาระนีมี n ไฟล์ได้รับปริมาณงาน 1 หน่วยของข้อมูลต่อวินาที จะเห็นว่าตัวชี้ของเจน ในกรณีนี้คือ

$$\frac{x^2}{n \times n} = 1 \quad (5.3)$$

สมมติว่าหนึ่งไฟล์ได้รับปริมาณงาน $1 + \Delta$ ค่าตัวชี้ของเจน เป็น

$$\frac{((n - 1) + 1 + \Delta)^2}{n(n - 1 + (1 + \Delta)^2)} = \frac{n^2 + 2n\Delta + \Delta^2}{n^2 + 2n\Delta + n\Delta^2} \quad (5.4)$$

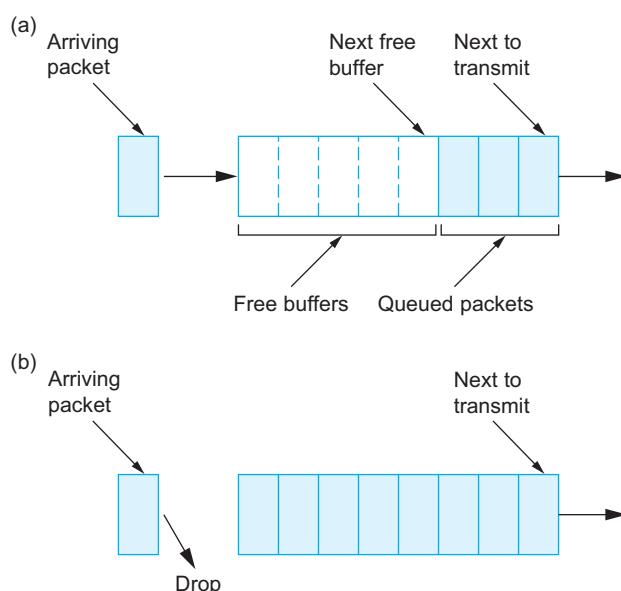
สังเกตว่าส่วนที่เกินเข้ามาคือ $(n - 1)\Delta^2$ ดังนั้น ไม่ว่าไฟล์ จะมากกว่าหรือน้อยกว่า Δ ยังคงมีผลให้ตัวชี้ของเจนลดค่าลงน้อยกว่า 1 ซึ่งหมายถึงความเป็นธรรมลดลง อีกตัวอย่างหนึ่ง เช่น ระบบสามารถทำให้ k ไฟล์จากทั้งหมด n ไฟล์ ได้รับทรัพุตเท่ากัน จึงทำให้ส่วนที่เหลือ $n - k$ ไฟล์ จะได้รับทรัพุตเป็น 0 ซึ่งกรณีนี้ความยุติธรรมทำได้เพียง k/n

5.2 รูปแบบคิว

คิวคือการรอ ระบบคิวคือวิธีการจัดสรรคิว เกี่ยวข้องกับรูปแบบการรับเข้าคิว รูปแบบการออกจากคิว พื้นที่สำหรับการได้รับบริการ เป็นต้น ซึ่งการรับเข้าคิวและการออกจากคิว เรียกว่า “รูปแบบคิว (Queueing Disciplines)” ใช้อธิบายถึงแพ็กเก็ตเมื่อเดินทางถึงคิวจะเก็บข้อมูลแพ็กเก็ตลงบัฟเฟอร์อย่างไร และอ่านบัฟเฟอร์อย่างไร รูปแบบการเข้าคิวที่จะกล่าวถึงนี้มี 2 แบบดังนี้ FIFO และ แฟร์คิว(fair queueing)

5.2.1 FIFO

แนวคิดของการจัดคิวแบบ FIFO หรือที่เรียกว่าการจัดคิวแบบFCFS(first come, first served) ทำความเข้าใจได้ด้วย: เมื่อแพ็กเก็ตเดินทางมาถึงเร้าเตอร์ จะเป็นแพ็กเก็ตแรกที่จะส่งออก แสดงไว้ในรูปที่ 5.5(a) ซึ่งแสดง FIFO มีพื้นที่บัฟเฟอร์ แบ่งเป็น 8 ช่อง (หรือเรียกว่า “สล็อต(slot)”) กล่าวได้ว่ามีคิวยาว 8 ช่อง เมื่อแพ็กเก็ตเดินทางถึงเร้าเตอร์ ตัวเร้าเตอร์จะตรวจสอบบัฟเฟอร์ที่ว่างอยู่โดยอ่านตัวชี้ Next free buffer เมื่อพบบัฟเฟอร์ที่ว่างจะนำเข้าคิว ในกรณีอ่านข้อมูลในคิวแบบFIFOจะเลือกอ่านแพ็กเก็ตที่มาก่อน จึงอ่านข้อมูลที่พอยน์เตอร์ Next to transmit ชี้อยู่ เมื่ออ่านเสร็จแล้วจะคืนพื้นที่ให้บัฟเฟอร์ ปกติเราเตอร์มีบัฟเฟอร์จำกัด เมื่อแพ็กเก็ตเดินทางถึงคิว (พื้นที่บัฟเฟอร์) ที่เต็มแล้ว เราเตอร์จะดรอปแพ็กเก็ตทิ้ง ดังแสดงในรูปที่ 5.5(b) การดรอปแพ็กเก็ตนี้ไม่คำนึงถึงลำดับความสำคัญของโอล์วหรือสิทธิพิเศษใดๆ การดรอปนี้บางครั้งเรียกว่าดรอปท้ายแคล เป็นผลจากการจัดคิวแบบ FIFO ที่เดินทางมาถึงในช่วงคิวเต็ม(ท้ายคิว)จะถูกดรอปทิ้ง



รูปที่ 5.5: คิวชนิด FIFO (a) การตัดคิวแบบดรอปท้ายแคล ของ FIFO(b)
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

สิ่งที่จะต้องคำนึงระหว่าง ดรอปท้ายแคล และ FIFO เป็นแนวคิดที่แยกกันไม่ออก FIFO ใช้กำหนดลำดับในการส่งแพ็กเก็ตเข้าลำดับในบัฟเฟอร์ สำหรับดรอปท้ายแคล เป็นวิธีการดรอป ซึ่งเป็นตัวกำหนดว่าแพ็กเก็ตใดจะถูกดรอป เนื่องจาก FIFO และ ดรอปท้ายแคล เป็นตัวอย่างที่ง่ายที่สุดของ วิธีการจัดระบบคิวและวิธีการ

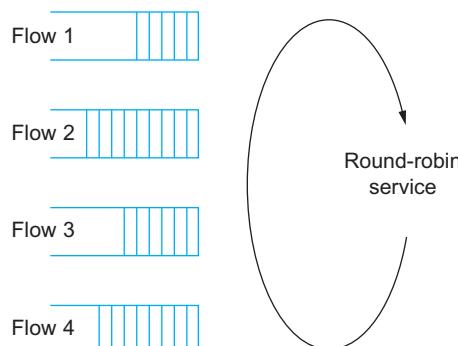
ดรอร์ป บางครั้งจึงถูกมองรวมกัน แล้วเรียกว่า คิวแบบ FIFO ซึ่งจริงๆแล้วควรเรียกว่า FIFO ที่มีดรอร์ปท้ายๆ แคล

ในลำดับต่อไปจะกล่าวถึงแนวทางการดรอร์ปแบบอื่น ซึ่งใช้อัลกอริทึมที่ซับซ้อนกว่า ตรวจสอบเพียงว่า “มีบีฟเฟอร์ว่างหรือไม่” เพื่อเป็นตัวตัดสินใจว่าจะปล่อยแพ็กเก็ตออกจากคิวเมื่อใด กระบวนการการดรอร์ปดังกล่าวอาจใช้กับ FIFO หรือวิธีการอื่นซับซ้อนกว่านี้ได้

5.2.2 Fair Queueing

ปัญหานี้ของ FIFO คือไม่สามารถคัดแยกความแตกต่างของต้นทางได้ เมื่อข้อมูลเดินทางเข้าคิวทั้งหมดจะทำงานตามเวลาที่เข้าสู่คิว แพ็กเก็ตใดเข้าคิวก่อนจะได้รับสิทธิก่อน โดยไม่อาจแยกลำดับความสำคัญได้ เช่น เครื่องต้นทางส่งข้อมูลจำนวนมากตลอดเวลาทำให้ข้อมูลเข้าสู่คิวมากที่สุดเป็นการปิดกั้นไม่ให้เครื่องอื่นได้ส่งข้อมูล

รูปแบบการเข้าคิวแบบแฟร์คิว ออกแบบให้มีการตรวจสอบเครื่องต้นทางก่อนอนุญาตให้เข้าคิว มีการทำงานเป็นตามรูปที่ 5.6 ในรูปที่ไฟล์ทั้งหมดสีไฟล์จากแหล่งที่มาสีเครื่องแต่ละเครื่องต้องการใช้ทรัพยากรเครือข่าย ในการเข้าคิวตามรูปจะใช้วิธีวนรอบ สมมติรอบแรกให้ Flow 1 ได้ส่งข้อมูลเสร็จแล้วจะวนรอบให้ Flow 4 ต่อมาเป็น Flow 3 และ Flow 2 ก่อนจะวนกลับมาที่ Flow 1 การวนรอบนี้จะกำหนดเวลาให้แต่ละไฟล์ได้เท่ากัน



รูปที่ 5.6: บริการ Round-robin จากทั้งหมดสีไฟล์เข้าเร้าเตอร์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ผลพลอยได้จากการทำงานของระบบคิวแบบแฟร์คิว เป็นการควบคุมความคับคั่งซึ่งสามารถควบคุมต้นทางถึงปลายทางได้ สำหรับการทำความเข้าใจคิว ทำได้โดยแยกการทำงานคิวออกจากภาระ ซึ่งการขึ้นกับตัวอุปกรณ์ เพราะ แฟร์คิวไม่ได้อยู่ในการทำงานของเร้าเตอร์ แต่ให้มองเป็นภาระของระบบโดยใช้มุมมองแบบนามธรรม จะระบบนี้จะตรวจสอบต้นทางได้ ในระบบมีต้นทางจำนวนมาก จะมีคำถามต่อมาได้แก่ ตรวจสอบอย่างไร? จะกล่าวถึงในลำดับต่อไป

การแบ่งทรัพยากรของ แฟร์คิว อาจสร้างความสับสนได้ เพราะในสถานการณ์ปกติ ข้อมูลที่สื่อสารในเครือข่ายแต่ละแพ็กเก็ตมีความยาวไม่เท่ากัน ซึ่งการแบ่งปันทรัพยากรเพื่อให้เกิดความเป็นธรรมนั้นจะคำนึงถึงความของของแพ็กเก็ตด้วยหรือไม่ ตัวอย่างเช่น ในระบบสื่อสารมีลิงก์แบบวิดีโอ 800-bps (100-Byte-per-

second) มีแพ็กเก็ตเดินทางเข้าระบบ แพ็กเก็ตแรกมีขนาด 1000-ไบต์ และแพ็กเก็ตที่สองมีความยาว 500-ไบต์ หากพิจารณาเที่ยงจำนวนแพ็กเก็ต ให้หั้งสองแพ็กเก็ตได้สิทธิเท่ากัน ครั้งแรกแพ็กเก็ตแรกได้เข้าระบบก่อน จะทำให้แพ็กเก็ตแรกใช้เวลาในการส่ง $1000/100 = 10$ วินาที ต่อมาแพ็กเก็ตสองได้ส่งข้อมูล ซึ่งใช้เวลาส่ง $500/100 = 5$ วินาที สังเกตได้ว่าแพ็กเก็ตแรกได้ใช้เวลาถือครองทรัพยากรเครือข่ายนานกว่าแพ็กเก็ตสองถึงสองเท่าตัว หากคิดในระบบการแข่งทรัพยากรเครือข่ายจากระยะเวลาระหว่างการถือครองทรัพยากรถือว่าไม่ยุติธรรมต่อแพ็กเก็ตที่สอง

ระบบคิวแบบ ร่วนด์โรบิน ใช้การมองความเป็นธรรมจากการระยะเวลาเท่ากันโดยแบ่งการส่งข้อมูลเป็นหน่วยเล็กที่สุด ในที่นี้คือหน่วย บิต จากตัวอย่าง มีแพ็กเก็ต สองชุด ขนาด 1000-ไบต์ และ 500-ไบต์ หรือขนาด 8000-บิต และ 4000-บิต ตามลำดับ ระบบร่วนด์โรบิน จะทำการสลับทีละบิต โดยให้แพ็กเก็ตแรกได้ส่งบิตที่หนึ่ง แล้วสลับไปแพ็กเก็ตที่สองส่งบิตที่หนึ่ง แล้วจึงสลับกลับไปแพ็กเก็ตแรก จนกว่าจะส่งครบ การทำงานเช่นนี้ทำให้ เกิดความเท่าเทียมในด้านเวลา สังเกตจากเมื่อแพ็กเก็ตที่แรกส่งถึงไบต์ที่ห้าร้อย จะเห็นได้ว่าแพ็กเก็ตที่สองจะส่งถึงไบต์ที่ห้าร้อยเช่นกัน และแพ็กเก็ตที่สองได้ส่งครบแล้ว โดยหั้งสองแพ็กเก็ตได้ใช้ทรัพยากรในการส่งข้อมูลจำนวน 500-ไบต์เท่ากัน ซึ่งแพ็กเก็ตแรกจะยังส่งต่อจนกว่าจะครบ

ทำความสะอาดเข้าใจการทำงาน RR แบบ บิต-ต่อ-บิต โดยกำหนดตัวแบบคิวให้มี 2 ขั้นตอนได้แก่ คิว และมีหน่วยบริการ(service) เมื่อแพ็กเก็ตเดินทางมาถึงจะถูกจัดเข้าคิว หลังจากถึงเวลาจะโอนข้อมูลจากคิวเข้าหน่วยบริการ เมื่อออกจากหน่วยบริการจะถือว่าส่งข้อมูลสำเร็จ

อธิบายการทำงานดังนี้ ให้เราเตอร์มีหนึ่งแพ็กเก็ตอยู่ในคิว (บันทึกอยู่ในบัฟเฟอร์ของเร้าเตอร์ หรือเรียกว่า “active flow”) มีความยาวเท่ากับ P ให้ C แทนการนับสัญญาณนาฬิกา และ C_P แทนจำนวนสัญญาณนาฬิกาที่เราเตอร์ใช้อ่านแพ็กเก็ตขนาด P ออกจากคิว ตัวอย่างเช่น มีแพ็กเก็ต 1-ไบต์ ($P=8$ -บิต) หากระบบสามารถอ่าน 1 บิตโดยใช้ 1 สัญญาณนาฬิกาจะทำให้ $C_{8-bit} = 8$ ส่วนต่อมาเป็นหน่วยบริการ กำหนดให้ C_S แทนจำนวนสัญญาณนาฬิกาที่เราเตอร์ใช้ดำเนินการภายในหน่วยบริการ และให้ C_F แทนจำนวนสัญญาณนาฬิกาทั้งหมดทั้งหมด P อยู่ในคิวจนแล้วส่งเข้าหน่วยบริการและสุดท้ายออกจากหน่วยบริการ คือจำนวนสัญญาณนาฬิกาในการอ่านแพ็กเก็ตขนาด P ออกจากคิว บวกกับจำนวนสัญญาณนาฬิกาที่ใช้ในการหน่วยบริการ เป็นตามสมการที่ (5.5)

$$C_F = C_S + C_P \quad (5.5)$$

กำหนดให้อักษร i แทนการอ้างถึงแพ็กเก็ตใด ๆ ตั้งนั้น P_i คือแพ็กเก็ตที่ i มีความยาว P และ C_{P_i} แทนจำนวนสัญญาณนาฬิกาที่ประมวลในคิวของแพ็กเก็ต i ซึ่งมีความยาว P และ T_{S_i} คือเวลาที่เราเตอร์เริ่มส่งแพ็กเก็ตที่ i กล่าวได้ว่า จำนวนสัญญาณนาฬิกาที่ใช้ในการส่งแพ็กเก็ต i ใดๆ อธิบายได้ในสมการที่ (5.6)

$$C_{F_i} = C_{S_i} + C_{P_i} \quad (5.6)$$

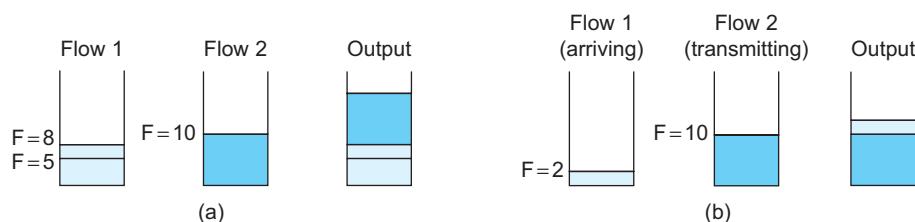
ในกรณีมีแพ็กเก็ตจำนวนมาก แต่ละแพ็กเก็ตแทนด้วย i แพ็กเก็ตที่ i จะได้ส่งข้อมูลนั้นขึ้นอยู่กับสถานะของคิวในเร้าเตอร์ ซึ่ง 2 สองสถานะได้แก่ ไม่มีแพ็กเก็ตอยู่ในคิว กับ มีแพ็กเก็ตอยู่ในคิว กรณีไม่มีแพ็ก

เก็ตอยู่ในคิวจะทำให้แพ็กเก็ตได้รับการประมวลผลทันที คือได้รับการส่งต่อเข้าหน่วยบริการโดยไม่ต้อง ขณะที่ กรณีมีแพ็กเก็ตอยู่ในคิวจะทำให้ แพ็กเก็ตที่ i ต้องรอให้แพ็กเก็ตก่อนหน้าออกจากคิวหมดก่อนแล้วจึงจะได้รับ บริการ โดยที่แพ็กเก็ตก่อนหน้า (แพ็กเก็ตที่ $i-1$) จะออกจากคิวได้แล้วว่าหน่วยบริการได้ดำเนินการแพ็กเก็ตที่ $i-1$ เสร็จแล้ว หรือเขียนได้ว่า F_{i-1}

ซึ่งกล่าวได้ว่าระบบจะได้เริ่มบริการจาก คิวว่างแล้วได้บริการแทนด้วย A_i และคิวไม่ว่างต้องรอให้ $i-1$ บริการเสร็จก่อนแทนด้วย F_i เลือกตัวที่ใช้เวลามากที่สุดดังนั้น $S_i = \max(A_i, F_{i-1})$ เป็นไปตาม สมการ (5.7)

$$F_i = \max(A_i, F_{i-1}) + P_i \quad (5.7)$$

ยกตัวอย่างรูปที่ 5.7(a) มีโฟล์จำนวนสองโฟล์แทนด้วย Flow 1 และ Flow 2 จากรูป Flow 1 มี ส่องแพ็กเก็ต โดยแพ็กเก็ตแรกสัญญาณพิกา 5-clock และอีกแพ็กเก็ตใช้ 3-clock รวมเป็น 8-clock ขณะ ที่ Flow 2 ใช้สัญญาณพิกา 10-clock ระบบจะเลือกส่ง Flow 1 ก่อน เพราะเร็วกว่า Flow 2 ในรูป ที่ 5.7(b) เร้าเตอร์กำลังส่งแพ็กเก็ตจาก Flow 2 ระหว่างนี้มี Flow 1 เดินทางเข้าระบบทำให้ Flow 1 ชิงส่งเร็ว มากกว่าได้ช่วงเวลาในการส่งก่อน



รูปที่ 5.7: ตัวอย่าง fair queueing : (a)แพ็กเก็ตที่ส่งเร็วกว่าจะได้รับสิทธิส่งก่อน (b) Flow 2 กำลังส่ง และมี Flow 1 เข้ามาแทรก
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

มีจุดสังเกตอยู่สองเรื่องเกี่ยวกับแฟร์คิว ประการแรกได้แก่ ลิงก์ในระบบแฟร์คิวจะไม่ว่าง ทราบได้ที่ มีอย่างน้อยหนึ่งแพ็กเก็ตในคิว การทำงานรูปแบบนี้เรียกว่า “การอนุรักษ์งาน(work conserving)” ผลการ ทำงานของการอนุรักษ์งานคือหากมีทรัพยากริบิ้งเหลืออยู่และไม่มีเครื่องต้องการใช้ แต่ไม่ได้สิทธิส่ง จะทำให้มี สิทธิส่ง ได้เต็มแบบดั่งเดิม และเมื่อมีโฟล์ใหม่เข้าระบบ โฟล์ที่เคยได้ส่วนแบ่งแบบดั่งเดิมมากจะหารด้วยโฟล์ ใหม่ ทำให้ได้แบบดั่งเดิมลดลง ประการที่สอง หากลิงก์ถูกใช้งานเต็มเช่นมี โฟล์จำนวน n โฟล์ จะไม่สามารถได้ รับการจัดสรรแบบดั่งเดิมได้มากกว่า $1/n^{th}$ ของแบบดั่งเดิมที่มี

พื้นฐานของ แฟร์คิว คือแบ่งแบบดั่งเดิมจำนวนเท่ากัน แต่เป็นไปได้ที่จะแบ่งด้วยอัตราส่วนไม่เท่ากัน เรียกว่า แฟร์คิวตามน้ำหนัก(Weighted fair queuing) ซึ่งเปิดทางให้กำหนดความสำคัญให้แต่ละโฟล์ไม่เท่า กัน

5.3 TCP Congestion Control

หัวข้อนี้กล่าวถึงตัวอย่างการทำงานของ Congestion Control แบบ end-to-end ซึ่งใช้ในระบบ TCP หากกล่าวถึงการทำงานของระบบส่งข้อมูลผ่านอินเทอร์เน็ตส่วนสำคัญยังอย่างหนึ่งที่ทำให้เครือข่ายอินเทอร์เน็ตทำงานจนแพร่หลายในปัจจุบันคือการทำให้ระบบมีเสถียรภาพในการส่งข้อมูลโดยเบื้องหลังคือออกแบบ Congestion Control ใน TCP

จาคีอบ ฟาน อูเทรคต์ ([Jacobson, 1988](#)) เสนอวิธีการควบคุมความคับคั่ง ใน TCP โดยตีพิมพ์ใน ACM SIGCOMM Computer Communication Review ปี 1988 และ 8 ปีโดยประมาณถูกนำมาใช้ในระบบอินเทอร์เน็ต กำหนดมาตรฐานแบบการทำงานใน [RFC2001 \(1997\)](#) หลักการพื้นฐานมาจากไฮสต์จะสามารถส่งข้อมูลได้สูงสุดไม่เกินกว่า advertised window อนุญาตในการเกิดความคับคั่งจะพบโดยเร้าเตอร์จะพบว่ามีการตอบแพ็กเก็ตและส่งผลให้เกิดการส่งซ้ำซึ่งเป็นการส่งผลให้เกิดความคับคั่งหนักขึ้น



รูปที่ 5.8: จาคีอบ ฟาน อูเทรคต์ ผู้เสนอใช้ Congestion Control สำหรับ TCP ลิขสิทธิ์ ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

กล่าวได้ว่า แนวคิดการควบคุมความคับคั่งที่พับใน TCP มีขึ้นเพื่อให้ไฮสต์ต้นทางแต่ละไฮสต์ได้เตรียมความพร้อมสำหรับควบคุมอัตราเร็วที่เหมาะสมก่อนส่งเข้าเครือข่าย ไฮสต์จะทราบจำนวนแพ็กเก็ตที่สามารถส่งได้โดยไม่เกิดปัญหา เมื่อไฮสต์ต้นทางมีแพ็กเก็ตจำนวนมากที่ต้องการส่ง จะใช้การตอบกลับของ ACK เป็นตัวบอกว่าแพ็กเก็ตได้รับ การส่งออกในเครือข่าย ดังนั้นจึงทำให้ยืนยันได้ว่าอัตราเร็วนั้นส่งแพ็กเก็ตสำเร็จ ในลำดับต่อมาเมื่อมีแพ็กเก็ตใหม่เข้าสู่เครือข่ายจะไม่เป็นการเพิ่มความคับคั่งขึ้นในเครือข่าย โดยวิธีการใช้ ACK นี้สามารถกำหนดความเร็วของการส่งแพ็กเก็ต TCP ได้ หรือเรียกว่า “self-clocking”

5.3.1 Additive Increase/Multiplicative Decrease

การส่งแบบ TCP มองแต่ละแพ็กเก็ตมีหลายสถานะ(state) ตั้งแต่เริ่มจนกระทั่งส่งสำเร็จ ตัวแปรที่ใช้เก็บสถานะความคับคั่งของเครือข่ายมีชื่อว่า CongestionWindow โดยไฮสต์ต้นทางจะอ่านค่า CongestionWindow เพื่อใช้ควบคุมอัตราเร็วในการปล่อยแพ็กเก็ต ค่า CongestionWindow จะเปลี่ยนแปลงตามค่า ACK เมื่อไฮสต์ต้นทางไม่ได้รับ ACK (หรือเรียกว่า unacknowledge) จะลดขนาด CongestionWindow ลง เป็นตามสมการต่อไปนี้

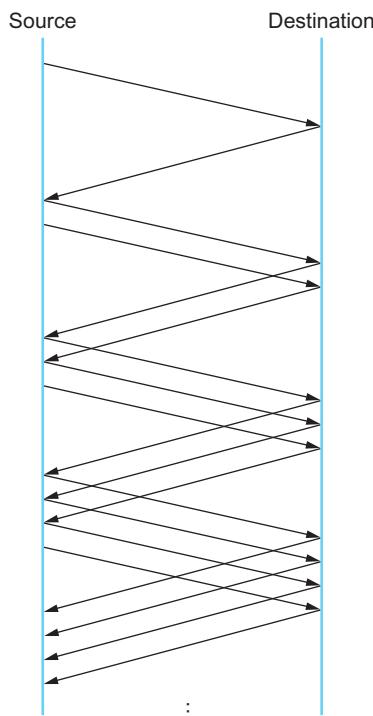
$$\text{MaxWindow} = \text{MIN}(\text{CongestionWindow}, \text{AdvertisedWindow})$$

$$\text{EffectiveWindow} = \text{MaxWindow} - (\text{LastByteSent} - \text{LastByteAcked})$$

เห็นได้ว่าใช้ MaxWindow แทน AdvertisedWindow ในการคำนวณ EffectiveWindow ทำให้โฮสต์ต้นทางส่งได้เร็วไม่เกินความสามารถต่อสุดในเส้นทาง

ปัญหานี้คือโฮสต์จะหาค่าที่เหมาะสมสำหรับกำหนด CongestionWindow ได้อย่างไร ซึ่งไม่เหมือนกับ AdvertisedWindow ซึ่งตอบกลับจากผู้รับ แต่ TCP ไม่มีเครื่องดึงส่งค่า CongestionWindow ไปยังฝั่งส่งของ TCP คำตอบคือโฮสต์ต้นทางกำหนดค่า CongestionWindow ขึ้นตามระดับความคับคั่งที่พึ่งในเครือข่าย โดยฟัง unacknowledge หากบว่ามี unacknowledge (แพ็คเก็ตไม่มี ACK หลัง timeout) จะทำการลด CongestionWindow ลงเมื่อระดับความแออัดเพิ่มขึ้น และจะเริ่ม CongestionWindow กระบวนการปรับขั้น-ลงนี้ เรียกว่า AIMD(additive increase/multiplicative decrease) จะกล่าวในรายละเอียดในลำดับถัดไป

ตัวแปร CongestionWindow มีหน่วยเป็นไบต์ แต่การปรับให้ขึ้นเร็วหรือลงเร็วจะทำให้การเพิ่ม-ลดอัตราเร็วเครือข่ายทำได้เร็วขึ้น เช่นกัน ตัวอย่างเช่น สมมติว่า CongestionWindow ปัจจุบันตั้งค่าเป็น 16 แพ็คเก็ต หากตรวจพบแพ็คเก็ตเกิดสูญหาย(lost) สามารถปรับ CongestionWindow เป็น 8 (โดยปกติตรวจพบสูญหายเมื่อเกิด timeout ซึ่ง TCP มีกระบวนการตรวจสอบแพ็คเก็ตสูญหาย) หากพบสูญหายเพิ่มขึ้นทำให้ CongestionWindow ลดลงเหลือ 4 แล้ว 2 และในที่สุดก็ถึง 1 แพ็คเก็ต โดย CongestionWindow มีค่าน้อยที่สุดเท่ากับ 1 แพ็คเก็ต



รูปที่ 5.9: แพ็คเก็ตขนาดเดินทางมีการเพิ่มไฟล์ขึ้นเรื่อยๆ
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

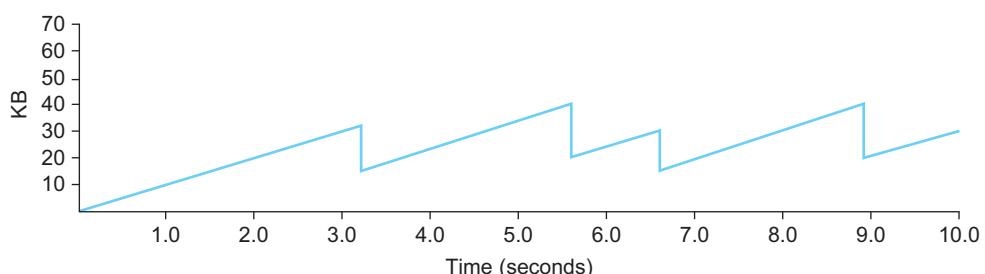
การควบคุมความคับคั่ง ด้วยการลดขนาด Window ลงถือว่าเป็นวิธีดั้งเดิม สามารถปรับปรุงประสิทธิภาพโดยกำหนดขอบเขตของความคับคั่งได้ เพื่อสำรองให้กับโฮสต์ใหม่ที่อาจจะเข้ามาเพิ่ม เรียกว่า “additive increase” เป็นการทำงานส่วนหนึ่งของ AIMD มีการทำงานดังนี้ ทุกครั้งที่โฮสต์ต้นทางส่ง CongestionWindow สำเร็จแล้ว แต่ละแพ็คเก็ตที่ส่งออกไปกลับมีไดร์รับ ACK จะเพิ่ม CongestionWindows ทีละ 1 แพ็คเก็ต การ

เพิ่มขึ้นเชิงเส้นนี้แสดงไว้ในรูปที่ 5.9 ซึ่งในทางปฏิบัติ TCP ไม่ได้รอให้ได้รับ ACK จนครบ Window แต่จะค่อยๆ เพิ่ม CongestionWindow ขึ้นทีละน้อยเมื่อ ACK ที่มาถึง :

$$\text{Increment} = \text{MSS} \times (\text{MSS}/\text{CongestionWindow})$$

$$\text{CongestionWindow} += \text{Increment}$$

นั่นคือแทนที่จะเพิ่ม CongestionWindow ทีละไปต์ โดยรอให้ได้รับ CongestionWindow จนครบ MSS ระบบจะรอในแต่ละ RTT ซึ่งเป็นข้อมูลบางส่วนของ MSS ซึ่งทุกครั้งที่รับ ACK จะนำมาปรับ CongestionWindow ทันที



รูปที่ 5.10: การทำงาน TCP มีทรรศน์เป็นพื้นเลื่อย
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

การเพิ่มและลดความคับคั่งนี้จะทำตลอดระยะเวลาการสื่อสาร หากนำค่า CongestionWindow มาพล็อต(plot)ให้แกน x แทนเวลา ได้รูปแบบกราฟเป็นพื้นเลื่อย ดังแสดงในรูปที่ 5.10

5.3.2 Slow Start

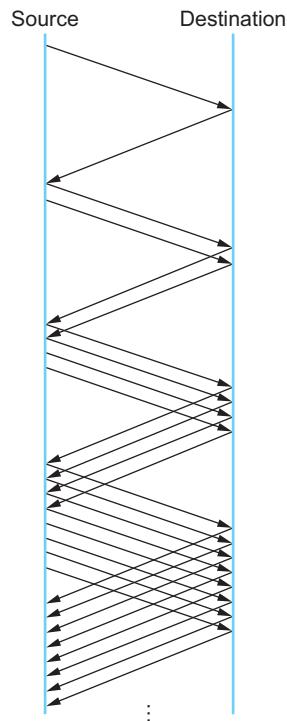
กระบวนการปรับ Contention Window ในหัวข้อนี้จะเริ่มต้นจาก ContentionWindow = 1 และรอรับ ACK หากได้รับจะเพิ่มขึ้นเป็น 2 เท่าแล้วรอรับ ACK เพิ่มแบบนี้ขึ้นเรื่อยๆ จนกว่าจะไม่ได้รับ ACK มีการทำงานตามรูปที่ 5.11

เหตุที่ไม่เริ่มต้นเร็วเลย เพราะจะทำให้เกิดสูญหายตั้งแต่ครั้งแรก เหตุที่เริ่มต้นช้า(ContentionWindow=1) เพราะการเข้มต่อผ่านอินเทอร์เน็ตเป็นการส่งผ่านลิงก์จำนวนมาก แต่ละลิงก์มีความเร็วเป็นไปได้ตั้งแต่ 1-Mbps ไปถึง 40-Gbps ซึ่งผู้ส่งข้อมูลไม่มีทางทราบได้เลยว่าในเส้นทางมีลิงค์ใดแบบวิดีโอน้อยที่สุด ดังนั้นการปรับ ContentionWindow จึงถูกนำมาใช้

การทำงาน Congestion Control ใน TCP อธิบายด้วยโค้ดภาษา C ดังนี้

```
{
    u_int     cw = state->CongestionWindow;
    u_int     incr = state->maxseg;

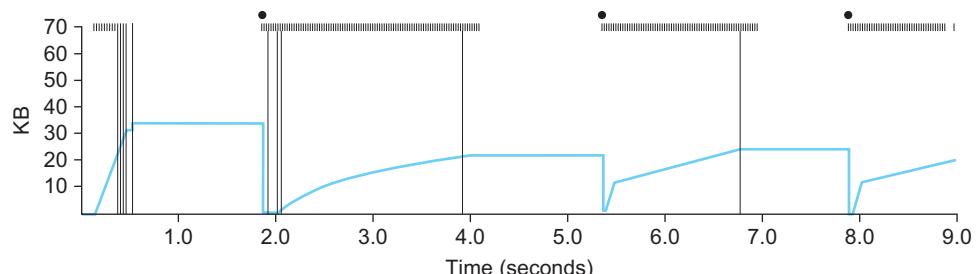
    if (cw > state->CongestionThreshold)
        incr = incr * incr / cw;
```



รูปที่ 5.11: แพ็กเก็ตทำงานแบบ slow start
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

```
state->CongestionWindow = MIN(cw + incr, TCP_MAXWIN);
}
```

เมื่อ state ใช้แทนสถานะของแพ็กเก็ต และกำหนดค่าต่ำสุดและสูงสุดของ ContentionWindow เป็นการทำงานร่วมกันระหว่าง Slow start และ AIMD



รูปที่ 5.12: การทำงานของระบบ ควบคุมความคับคั่ง สำหรับ TCP
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

รูปที่ 5.12 อธิบาย CongestionWindow เพิ่มขึ้นและลดลงอย่างไรตามเวลา และแสดงให้เห็นถึงการทำงานร่วมกันของ Slow Start และ AIMD

บทที่ 6

ความปลอดภัยทางเครือข่าย

ปัญหา: การโจมตีทางเครือข่าย

เครือข่ายคอมพิวเตอร์เป็นระบบที่พัฒนาเพื่อตอบสนองให้แอปพลิเคชันจำนวนมากแข็งแกร่งที่รับประทานกัน เครือข่ายคอมพิวเตอร์ที่มีขนาดใหญ่ เช่นระบบอินเทอร์เน็ตมีการแข่งขันอย่างก้าวกระโดด การแข่งขันนี้ได้มาจากการที่มีความจำเป็นทางธุรกิจ การแลกเปลี่ยนข้อมูลของภาคธุรกิจไม่เว้นแม้กระทั่งการซื้อขายทางการที่ทำผิดระบบความมั่นคงปลอดภัยทางคอมพิวเตอร์ซึ่งถูกพัฒนาขึ้นเพื่อปกป้องไม่ให้มีการนำเครือข่ายไปใช้ในการกระทำผิด

ยกตัวอย่างเช่น ภัยคุกคามจากการใช้เว็บไซต์ สมมติบื้อบเป็นลูกค้าเว็บขายสินค้าออนไลน์ ใช้บัตรเครดิตชำระเงินค่าผ่านระบบออนไลน์ ภัยคุกคามที่เกิดขึ้นจากผู้บุกรุกทำได้โดยการดักฟังข้อมูลที่บื้อบส่งไปยังเว็บไซต์เพื่อชำระเงินตัวอย่างเช่น ชื่อหมายเลขบัตรเครดิตวันหมดอายุและรหัสหลังบัตรซึ่งข้อมูลเหล่านี้ เป็นข้อมูลจำเป็นที่ใช้ในการตัดบัตรเครดิต เมื่อผู้บุกรุกได้รับข้อมูลบัตรเครดิตของบื้อบแล้วสามารถนำข้อมูลนั้นไปใช้จ่ายที่เว็บไซต์อื่นได้ แล้วผู้บุกรุกดักฟังข้อมูลผ่านเครือข่ายได้อย่างไร่นั้น การสื่อสารผ่านเครือข่ายไร้สาย หรือเครือข่ายแลนสามารถตัดข้อมูลตามเส้นทางของสัญญาณ หากเป็นข้อมูลส่งผ่านเครือข่ายไร้สายสามารถ เขตเครื่องรับให้ดักฟังได้ทุกบริเวณรอบสัญญาณ หากเป็นการสื่อสารผ่านเครือข่ายแลนสามารถดูเส้นทางข้อมูล ได้ตามสายแลนที่พาดผ่าน นอกเหนือนี้ผู้ข่ายยังสามารถติดตั้งตรวจสอบตัวเว็บที่ทำหน้าที่เป็นสปายคอลดักฟัง ข้อมูลในเครือข่ายคอมพิวเตอร์หรือในตารางเรขาตั้ง การดักฟังข้อมูลในอุปกรณ์แม่ข่ายเกิดขึ้นได้หากข้อมูลนั้น มีความสำคัญ เช่นเป็นภัยคุกคามที่เกิดในระดับชาติ หาผู้อ่านทราบถึงกระบวนการการสื่อสารผ่านเครือข่าย คอมพิวเตอร์จะทำให้เข้าใจถึงปัญหาในด้านความปลอดภัยและนำไปสู่การป้องกันที่เหมาะสม

การสื่อสารผ่านเครือข่ายอินเทอร์เน็ตมีการเข้ารหัสข้อมูลเพื่อป้องกันการดักฟังข้อมูลระหว่างทาง เป็นแนวทางในการป้องกันความลับระหว่างเครือข่าย (Confidentiality) อย่างไรก็ตามในเมื่อมีการปกป้องความลับไม่ให้รั่วไหลแล้วเหตุยังมีภัยคุกคามในระบบคอมพิวเตอร์ อธิบายข้อสงสัยนี้ โดยใช้ตัวอย่าง ผู้ประสงค์ร้ายไม่สามารถ แปลงข้อมูลได้ทางตรง แต่อาจใช้เทคนิคไวรัสติดต่อสัมภาระสู่เครือข่ายได้ หากกูญแจรหัสลับที่เลือกใช้นั้นสามารถคาดเดาได้ ง่าย หรือแม้กระทั่งการใช้ระบบคอมพิวเตอร์เดาสุ่มข้อมูลจำนวนหลายรอบ ซึ่งการป้องกันข้อมูลไม่ให้รั่วไหล อาจไม่เพียงพอ และหากข้อมูลตกอยู่ในมือผู้ร้ายและเกิดการแก้ไขข้อมูลทำให้สูญเสียความปลอดภัยได้ ในการ ปกป้องข้อมูลจึงให้ความสำคัญกับการตรวจสอบความถูกต้องของข้อมูล (Integrity) ด้วย

ภัยคุกคามอย่างอื่น เช่นผู้ใช้งานไม่ทราบว่าเว็บไซต์ที่ตน用เชื่อมต่อนั้นเป็นเว็บไซต์จริงหรือเว็บไซต์偽サイト หรือผู้บุกรุกสามารถตั้งเว็บไซต์偽site หรือโดยการเปลี่ยนแปลงค่า ดีอีนเอล(DNS) ให้มีลักษณะใกล้เคียงกับเว็บไซต์ จริงซึ่งลักษณะการโจมตีแบบนี้เรียกว่า พิชชิ่ง(phishing)

เจ้าของเว็บไซต์สามารถตรวจสอบการถูกแฮกได้โดยง่าย ในบางเว็บไซต์ อาจมีการนำไฟล์ไปปิดไฟล์เพื่อ ให้ผู้บุกรุกเข้าควบคุมเครื่องได้ในภายหลัง ปัญหานี้เกิดจากปัญหา access control นอกจากนี้ ผู้ให้บริการ เว็บไซต์อาจโดนผู้บุกรุกโคลบด์เว็บไซต์เพื่อให้เว็บไซต์นั้นไม่อาจให้บริการได้ตามปกติ เรียกการโจมตีนี้ว่า denial

of service (DoS) แนวทางของการรักษาความมั่นคงปลอดภัยเพื่อให้ระบบยังสามารถให้บริการได้เรียกว่า availability

นอกจากนี้ปัญหาที่พบกับการใช้งานเครือข่ายอินเทอร์เน็ต มาจากซอฟต์แวร์ประสงค์ร้าย (malicious software) หรือเรียกว่า malware ลักษณะของมาแวร์เกิดจากการพัฒนาซอฟต์แวร์ที่มีวัตถุประสงค์ในการเจาะระบบหรือเจาะช่องโหว่ที่เกิดขึ้นในระบบตัวอย่างมาแวร์เช่น worms virous และ botnets เป็นต้น

6.1 ความไว้วางใจ และ ภัยคุกคาม

ก่อนจะเข้าเนื้อหาในการตอบคำถาม ทำไม และ อย่างไร ของการสร้างความมั่นคงปลอดภัยให้เครือข่ายคอมพิวเตอร์ ความไว้วางใจ และ ภัยคุกคาม เปรียบได้กับเครื่องมือด้าน ภัยคุกคาม เป็นโอกาสที่ระบบจะสูญเสียความมั่นคงปลอดภัย เช่นการวางแผนของมิค่าในที่สาธารณะ สำหรับความไว้วางใจให้อธิบายความไว้ใจในสิ่งๆหนึ่ง ยกตัวอย่างเช่น การถูกขโมยของมิค่าถือเป็นภัยคุกคาม แต่หากของมิค่านั้นถูกวางในที่ๆไว้ใจเช่นในห้องมีแค่คน ที่ไว้ใจ ทำให้มีต้องคำนึงระบบป้องกันความปลอดภัยเท่ากับวางในที่สาธารณะ เป็นต้น

การสร้างความมั่นคงปลอดภัยจะดำเนินไปด้านค่าใช้จ่ายควบคู่กับสิ่งที่ได้รับ ยกตัวอย่างเช่น การสร้างความมั่นคงปลอดภัยให้การสื่อสารผ่านแลนไร์สาย ด้วยการเข้ารหัสข้อมูล ซึ่งการเข้ารหัสข้อมูล ต้องการกุญแจเข้ารหัสที่เป็นความลับเมื่อใช้งานผ่านเครือข่ายไร์สายที่เป็นเพียงการใช้งานคนเดียวจะถือกุญแจรหัสรับเพียงคนเดียวแต่เมื่อมีการใช้งานเครือข่ายนั้นจำนวนหลายคนนั่นหมายถึงกุญแจรหัสลับ จะต้องแจกจ่ายให้กับทุกคน ทำให้มีคำถามกลับมาว่าสามารถถูกความไว้วางใจทุกคนได้หรือไม่ เมื่อไม่สามารถถูกความไว้วางใจได้กับทุกคนจึงต้องใส่ระบบที่ทำให้แต่ละคนมีกุญแจรหัสลับแตกต่างกัน ซึ่งส่งผลกระทบต่อค่าใช้จ่ายสำหรับนำระบบบันทึกมาใช้งาน หรือตัวอย่างการส่งข้อมูลภายใน data center ที่ต้องการการสื่อสารความเร็วสูงหากนำกระบวนการเข้ารหัสมาใช้งานทำให้ประสิทธิภาพลดลงเพียง 10% ก็อาจส่งผลต่อประสิทธิภาพรวมของระบบได้ ซึ่งจะต้องคำนึงถึงพื้นที่ใช้งานนั้นสามารถถูกความไว้วางใจได้หรือไม่ หากเป็น data center ในระบบปิดสามารถถูกความไว้วางใจได้จะลดค่าใช้จ่ายด้านการนำระบบความปลอดภัยมาใช้งานได้และยังเป็นการเพิ่มประสิทธิภาพเครือข่าย

การคำนวณความคุ้มค่าในการลงทุนเป็นสิ่งที่ทำอยู่ตลอด อาจเปลี่ยนแปลงตามเทคโนโลยีที่เปลี่ยนไป ตัวอย่างเช่นมีการใช้งานเครือข่ายคอมพิวเตอร์และตลอดระยะเวลา 10 ปีแต่เมื่อถึงเวลาหนึ่งมีการรายงานคนพบช่องโหว่ของการทำงานในระบบคอมพิวเตอร์ทำให้มีโอกาสถูกจัดระบบดังนั้นในการคำนวณความคุ้มค่าในการลงทุนสำหรับทรัพยากรที่อยู่ภายใต้ระบบคอมพิวเตอร์นั้นจะถูกนำมาคำนวณหากถูกเจาะระบบแล้วจะมีความสูญเสียในจุดที่ยอมรับได้หรือไม่ แนวทางการคำนวณนี้เรียกว่า “risk assessment”

6.2 กระบวนการเข้ารหัส

หัวข้อนี้จะอธิบายถึงเทคโนโลยีรหัสลับ โดยอธิบายถึงขั้นตอนวิธีการเข้ารหัส เทคโนโลยีรหัสไซเฟอร์(cipher) และ กระบวนการแฮช(hash) ซึ่งจะกล่าวเป็นลำดับในหัวข้อถัดไปเทคโนโลยีรหัสไซเฟอร์เป็นฟังก์ชันของกุญแจรหัสลับ ซึ่งการทำงานเกี่ยวข้องกับการแจกจ่ายกุญแจ (distributed keys) ในช่องทางที่ปลอดภัย ขณะ

ที่แข็งแกร่งนำไปใช้แตกต่างจาก iPhone เทคโนโลยีแข็งไม่ต้องใช้กุญแจรหัสลับ เป็นการเข้ารหัสที่ไม่ต้องการการถอดรหัส นิยมนำไปใช้บันทึกรหัสผ่าน หรือใช้ตรวจสอบความสมบูรณ์ของไฟล์

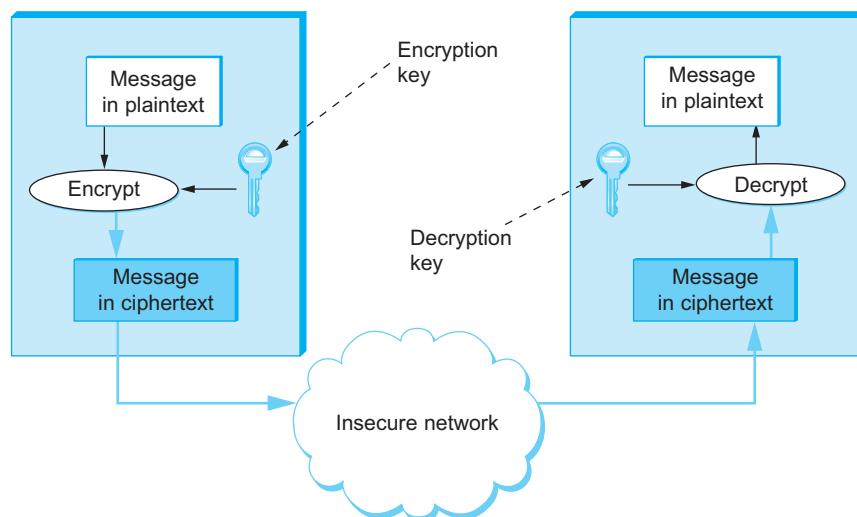
6.2.1 พื้นฐานของ iPhone

การเข้ารหัศคือการแปลงข้อความเป็นรูปแบบอื่น มีเป้าหมายเพื่อให้ไม่ผู้อื่นที่ไม่เป็นคู่สื่อสารสามารถแปลงความได้ กระบวนการในการแปลงรหัสเกี่ยวข้องกับตัวแปรสามตัว C ใช้แทนข้อความรหัสลับเป็นผลจากการแปลงข้อมูล และ $E(K,P)$ ใช้แทนเทคโนโลยีรหัสลับที่ใช้ K เป็นกุญแจรหัสลับ และ P แทนข้อความที่ยังไม่ผ่านการเข้ารหัส

$$C = E(K, P)$$

เมื่อได้รับข้อมูลที่ผ่านการเข้ารหัส (cipher : C) และจะสามารถนำไปส่งต่อผ่านช่องทางสื่อสารที่ไม่ความไว้วางใจได้อย่างปลอดภัย และเมื่อข้อมูลถึงปลายทางหากเป็นผู้รับตัวจริงจะมีกุญแจรหัสลับตรงกับ K ของเครื่องส่งที่สามารถถอดรหัส C เป็น P เมื่อตอนนั้นทางได้เรียกกระบวนการการเข้ารหัสว่า “encryption” และเรียกกระบวนการถอดรหัสว่า “decryption” ซึ่งเป็นตามรูปที่ 6.1

$$P = D(K, C)$$

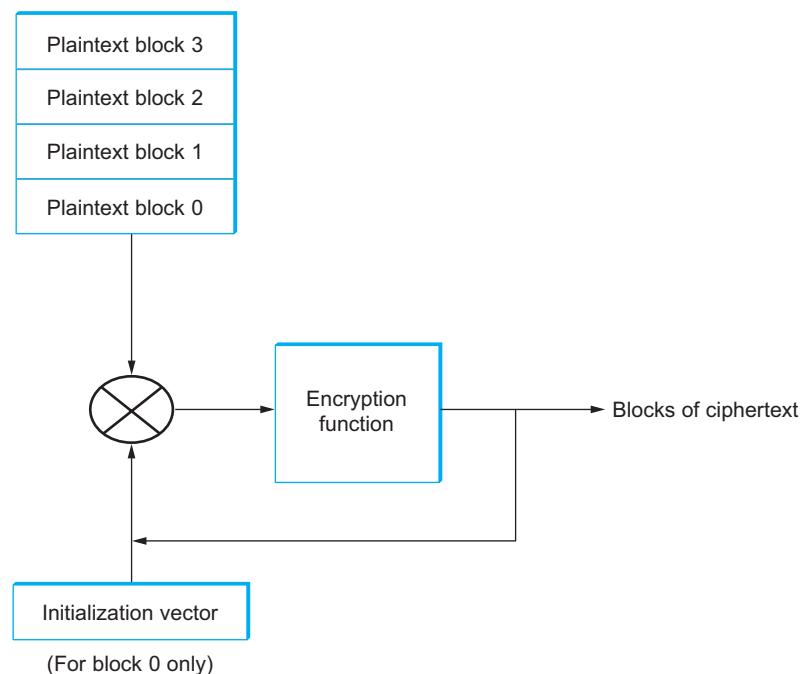


รูปที่ 6.1: ใช้กุญแจรหัสลับในการเข้ารหัสและถอดรหัส
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

จากรูปที่ 6.1 เครื่องส่งมีข้อมูล plaintext (Message in plaintext) ต้องการส่งข้อมูลผ่านช่องสัญญาณที่ไม่ปลอดภัย อันดับแรกเครื่องส่งจะเข้ารหัสโดยใช้กุญแจรหัสลับ (Encryption key) ทำให้ได้ผลลัพธ์เป็น ciphertext (Message in ciphertext) โดย ciphertext นี้จะส่งผ่านช่องสัญญาณที่ไม่ปลอดภัยไปถึงปลายทางเมื่อปลายทางได้รับแล้วจะทำการถอดรหัสโดยใช้กุญแจรหัสลับ (Decryption keys) ตัวเดียวกันทำให้ได้ผลลัพธ์เป็น plaintext เมื่อตอนนั้นฉบับ

ในเทคโนโลยีการเข้ารหัสสมัยใหม่ ข้อมูล plaintext ซึ่งกันเมื่อผ่านกระบวนการเข้ารหัสจะให้ผลลัพธ์ไม่ซ้ำกัน เพื่อป้องกันการคาดเดารหัสลับได้ ทำได้โดยการสร้าง session keys และการใช้กุญแจรหัสลับหลักเข้ารหัสทุก กระบวนการสร้าง session keys สามารถแบ่งได้เป็น stream cipher และ block cipher สำหรับ stream cipher จะสร้าง session keys ที่มีขนาดความยาวต่อเนื่องไปเรื่อยๆ โดยเข้ารหัสจากข้อมูลที่ไม่มีการแบ่งเป็นท่อน ขณะที่ block cipher จะแบ่งข้อมูลออกเป็นท่อนแล้วนำมาเข้ารหัส ด้วยว่า เทคโนโลยีเข้ารหัส เช่น RC4 และ AES สำหรับ stream cipher และ block cipher ตามลำดับ

รูปที่ 6.2 อธิบายถึงการแปลงข้อมูลเป็นท่อนจากรูปแบ่งข้อมูลออกเป็นสี่ท่อน และจึงนำข้อมูลแต่ละท่อนเข้ากระบวนการเข้ารหัสข้อมูลก่อนส่งออกไปยังปลายทางโดยกุญแจที่ใช้ในการเข้ารหัสนั้นเครื่องปลายทาง จะทราบลำดับของกุญแจจากการอ่านค่า IV(initialization vector) ที่ปลายทางจะถึงการสร้าง session key โดยใช้วิธีเดียวกับเครื่องต้นทาง และจึงนำมาถอดรหัส ถึงแม้ข้อมูลจะเดินทางถึงปลายทางไม่เป็นลำดับ เครื่องปลายทางจะใช้ IV เป็นตัวระบุกุญแจรหัสลับที่จะนำมายถอดรหัส



รูปที่ 6.2: การแบ่งข้อมูลเป็นท่อนก่อนเข้ารหัส
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

6.2.2 ไซเฟอร์แบบ Secret-Key

แนวทางของ secret-key คือการให้เครื่องส่งและเครื่องรับคือกุญแจรหัสลับ เดียวกัน หรือบางครั้งเรียกว่า “symetric-key” ขณะที่อีกแนวทางหนึ่งคือการให้เครื่องส่งและเครื่องรับถือกุญแจรหัสลับที่ไม่เหมือนกัน (แต่ยังมีความสัมพันธ์ระหว่างกัน จะกล่าวถึงในหัวข้อต่อไป)

หน่วยงาน NIST(U.S. National Institute of Standards and Technology) ได้ออกข้อกำหนดมาตรฐานในการรับ secret-key มาใช้งานในหน่วยงานภาครัฐ โดย DES(Data Encryption Standard) เป็น

มาตรฐานเทคโนโลยีการเข้ารหัสลับแรกที่หน่วยงานได้กำหนดไว้ซึ่งในขณะที่ออกข้อกำหนดมาตรฐานนี้เชื่อกันว่าเทคโนโลยีการเข้ารหัสลับนี้จะไม่สามารถถอดรหัสโดยเนื่องจากเป็นการเข้ารหัสลับที่มีความยาวกุญแจรหัสลับทั้งหมด 56-บิตไม่เขียนต่อ กัน (ใช้บิตท้ายเป็น parity bit) ทำให้มีกุญแจที่เป็นไปได้ทั้งหมด 2^{56} หรือกล่าวได้วามีกุญแจรหัสลับเท่ากับ $2^{55} = 3.6 \times 10^{16}$ ซึ่งการคาดเดากุญแจรหัสรับที่มีจำนวนมากขนาดนี้เป็นเรื่องยากในอดีต อย่างไรก็ตามเทคโนโลยีการประมวลผลทางคอมพิวเตอร์มีความก้าวหน้าอย่างรวดเร็ว ในปี 1990 มีผลการศึกษาที่สามารถถอดกุญแจรหัสลับของระบบ DES ได้ภายในระยะเวลาไม่กี่ชั่วโมงโดยใช้ระบบคอมพิวเตอร์ทำงานแบบขนาน ทำให้ NIST ได้ยกเลิกคำแนะนำการใช้ DES และเรียกระบบที่ใช้งานว่าเป็นระบบเก่า (legacy)

ต่อมา NIST ได้ออกมาตรฐานใหม่เรียกว่า TripleDES (3DES) ซึ่งนำ DES เพิ่มกุญแจรหัสลับให้มีความยาวขึ้นสามเท่า เป็น 168-บิต ($= 3 \times 56$) โดยมีกุญแจรหัสลับทั้งหมดสามกุญแจดังนี้ DES-key1 DES-key2 และ DES-key3 โดยการทำงาน 3DES ขั้นตอนแรก เป็นการเข้ารหัส เช่นเดียวกับ DES ดังเดิมโดยใช้กุญแจ DES-key1 นำผลลัพธ์ที่ได้มาถอดรหัสด้วย DES-key2 แล้วนำผลลัพธ์จากการถอดรหัสไปเข้ารหัสอีกครั้งด้วย DES-key3 ทำให้ได้ผลลัพธ์เป็น ciphertext ที่เข้ารหัสด้วย 3DES สำหรับการถอดรหัสทำได้โดยการย้อนกลับได้แก่ถอดรหัสด้วย DES-key3 แล้วเข้ารหัสด้วย DES-key2 และ ถอดรหัสด้วย DES-key1 ทำให้ได้ plaintext

เหตุผลของการทำงาน 3DES ในขั้นตอนถอดรหัสด้วย DES-key2 เพื่อให้ระบบยังสามารถทำงานได้กับเทคโนโลยี DES รุ่นเก่า อธิบายได้ดังนี้ ถ้า legacy-DES เข้ารหัสด้วย single-key แล้วเข้าสู่ระบบ 3DES จะถูกถอดรหัสด้วย DES-key2 ที่มี DES-key2=DES-key1 ทำให้กลับมาเป็น plaintext อีกครั้ง และขั้นตอนท้ายเข้ารหัส DES-key3=DES-key2=DES-key1 จึงทำให้เป็นการเข้ารหัสเหมือนระบบ DES การออกแบบนี้มีข้อดีคือ ยังคงใช้งานได้กับเทคโนโลยีเข้ารหัส DES มาตรฐานเก่า

มาตรฐาน 3DES นำมาแก้ไขปัญหา DES ในด้านความยาวกุญแจรหัสลับของ DES ด้วยการเพิ่มจำนวนกุญแจรหัสลับ สิ่งที่ตามมาก็คือการประมวลผลที่เพิ่มขึ้นโดยเทคโนโลยีคอมพิวเตอร์ที่สามารถประมวลผลได้ครั้งละเพียง 64 บิตทำให้การประมวลผล 3DES ทำงานได้ช้า

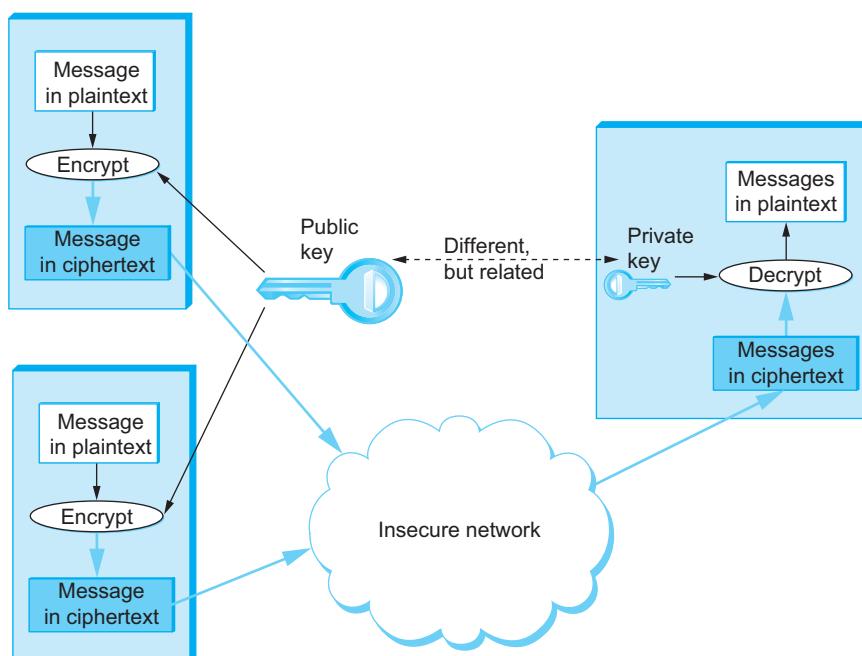
เทคโนโลยีใหม่ที่ NIST ออกคำแนะนำเพื่อใช้ทดแทน 3DES ในปัจจุบันได้แก่ AES(Advanced Encryption Standard) โดยมีเทคนิคเบื้องหลังจาก Rijndael (ออกเสียงว่า “Rhine dahl”) คิดขึ้นโดย Daemen และ Rijmen (1999) เทคโนโลยี AES มีความยาวกุญแจรหัสลับ 128 192 และ 256-บิต และมีการแบ่งท่อนข้อมูลท่อนละ 128-บิต เทคโนโลยี AES สามารถพัฒนาด้วยซอฟต์แวร์หรือฮาร์ดแวร์ก็ได้ การทดสอบความปลอดภัยทางคณิตศาสตร์ (วิธีหนึ่งที่ใช้ทดสอบด้วยการดูพื้นที่กุญแจรหัสลับ ที่ใหญ่พอสำหรับป้องกันการโจมตีด้วยวิธีคาดเดารหัสผ่าน) พบร่วมกับคุณสมบัติตรงกับการถอดรหัสผ่านอย่างมีนัยสำคัญ

6.2.3 ไซเฟอร์แบบ Public-Key

กุญแจรหัสลับ ลำดับต่อมาได้แก่ กุญแจรหัสรับที่เครื่องส่งและเครื่องรับมีกุญแจรหัสลับแตกต่างกันเรียกว่า public-key ใน การส่งข้อมูลด้วยเทคโนโลยีรหัสลับแบบ Public-key จะใช้กุญแจคู่กันสำหรับเข้ารหัสข้อมูล และถอดรหัสข้อมูล หรือบางครั้งเรียกว่า “asymmetric-key” หรือ “pair-key” ประกอบด้วย private-key (กุญแจส่วนตัว) และ public-key (กุญแจสาธารณะ) ข้อดีของการใช้กุญแจส่วนตัว คือแก้ปัญหาความไม่ปลอด-

ภัยจากการแจกจ่ายกัญชาหรือสลับ โดยเจ้าของกัญชาหรือสลับจะถือกัญชาส่วนตัว เพียงผู้เดียวและสามารถแจกจ่าย กัญชาสำหรับคนให้ผู้อื่นได้หลายคน อธิบายการทำงานได้ในรูปที่ 6.3

รูปที่ 6.3 มีผู้เกี่ยวข้องอยู่สามคน กล่องด้านบนสุดเป็นผู้ที่ต้องการส่งข้อมูลความลับไปยังกล่องด้านขวาเมื่อ และที่กล่องซ้ายมีล่างก็มีข้อความของตนเองที่ต้องการส่งไปยังด้านขวาเมื่อ โดยกล่องทั้งสองด้านซ้ายมีจะไม่สามารถถอดรหัสข้อมูลระหว่างกันได้จะมีเพียงกล่องด้านขวาเมื่อที่สามารถถอดรหัสข้อมูลที่ส่งมา ตนเองได้ มีหลักการทำงานดังนี้ กล่องด้านบนมีข้อความต้องการส่งไปถึงกล่องขวาเมื่อ จะใช้กุญแจสาธารณะของกล่องขวาเมื่อเข้ารหัสข้อมูลแล้วส่งผ่านช่องทางที่ไม่ปลอดภัยเมื่อข้อมูลเดินทางถึงปลายทางกล่องขวาเมื่อจะนำกุญแจส่วนตัวของตนเองทำการถอดรหัสเช่นเดียวกับกล่องด้านซ้ายมีล่างเมื่อต้องการส่งข้อมูลไปยังกล่องขวาเมื่อจะนำกุญแจสาธารณะมาเข้ารหัสข้อมูลแล้วจึงส่งผ่านช่องทางที่ไม่ปลอดภัยเมื่อข้อมูลเดินทางถึงกล่องขวาเมื่อทำการถอดรหัสโดยใช้ private-key

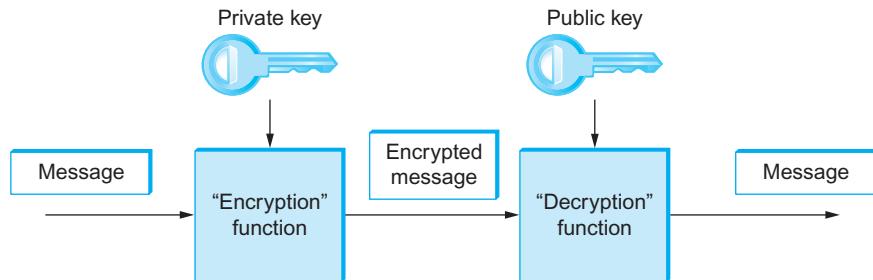


รูปที่ 6.3; การเข้ารหัสด้วยกุญแจสาธารณะ
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

การทำงานลักษณะนี้ทำความเข้าใจได้ยาก เพราะขัดกับความเข้าใจพื้นฐาน ที่คิดว่ากัญแจเข้ารั้งควรเป็นกัญแจถอดรหัส ทำให้เข้าใจจะรับรองความปลอดภัยไม่ได้ แนวคิดการเลือกตัวเลขมาเป็นกัญแจส่วนตัว และกัญแจสาธารณะ มาจากการเลือก coprime number ที่มีช่วงข้อมูลขนาดใหญ่มาก ปัจจุบันเลือกข้อมูลมากกว่า 2048-บิต ซึ่ง composite นี้เป็นเลขจำนวนเฉพาะที่คุณกัน ทำให้ได้ผลลัพธ์เพียงค่าเดียว การสูญเสียผ่านจะต้องคาดเดาเลขจำนวนเฉพาะในช่วง $[2^{2048}]$ ถือว่ามีขนาดใหญ่มาก จนระบบคอมพิวเตอร์ปัจจุบันไม่อาจได้คำตอบในระยะเวลาอันสั้น (มากกว่าสิบปี)

รูปที่ 6.4 อธิบายขั้นตอนการใช้กุญแจสาธารณะ(public-key)ทำหน้าที่ พิสูจน์ตัวจริง โดยการพิสูจน์ตัวจริงนี้จะเกิดจากผู้ที่ถือ กุญแจส่วนตัว เป็นตัวจริงเข้ารหัสข้อมูลด้วย กุญแจส่วนตัว แล้วส่งข้อมูลไปยังปลาย

ทางเครื่องบัญชาจะสามารถพิสูจน์ตัวจริงของผู้ส่งได้จากการใช้ กุญแจสาธารณะ ที่ผู้ส่งได้แจกจ่ายไว้บนอินเทอร์เน็ต



รูปที่ 6.4: การพิสูจน์ตัวจริงโดยใช้กุญแจส่วนตัว
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

กล่าวถึงที่มาของ กุญแจสาธารณะ มีความน่าสนใจอยู่ จุดเริ่มต้นของ public มาจากการเผยแพร่ งานวิจัยของ Diffie และ Hellman (1976) งานวิจัยนี้ สองท่านให้ Britain's Communications Electronics Security Group คิดค้น public-key ขึ้นในปี 1970 และ U.S. National Security Agency (NSA) ยืนยันว่ามีการศึกษาบันทึกตั้งแต่กลางปี 1960

จากที่กล่าวมา public-key หรือที่เรียกว่า RSA (อาร์-เอส-เอ) เป็นชื่อของผู้ประดิษฐ์ได้ Rivest, Shamir และ Adleman Rivest และคณะ (1978) การคำนวณ RSA(ริเวสต์(Ron Rivest) ชาเมร์(Adi Shamir) และแอดเลมาน(Len Adleman)) ต้องใช้ทรัพยากร่นว่าประมาณผลมากกว่าการคำนวณแบบ symetric key นักวิจัยไม่สามารถหาวิธีคำนวณที่มีประสิทธิภาพได้ จึงมีการใช้กุญแจรหัสลับความยาวเพียง 1024-บิต ซึ่งถือว่าปลอดภัยแล้วตลอดระยะเวลา 10 ปี

กุญแจสาธารณะ มีความมั่นคงปลอดภัยสูง แต่ความต้องการในการใช้ประมาณผลมากกว่า secret-key หรือกล่าวได้ว่าด้วยขนาดข้อมูลเท่ากัน การคำนวณด้วยกุญแจสาธารณะ ใช้เวลานานกว่า secret-key มาก จึงทำให้ในการใช้งานส่วนใหญ่จะพบรากурсิ่งงานที่มากกว่า

6.2.4 พิสูจน์ตัวจริง

การเข้ารหัสเพียงอย่างเดียวไม่อาจตรวจสอบความถูกต้องสมบูรณ์(integrity)ของข้อมูลได้ตัวอย่างเช่นถ้ามีการส่งข้อมูลได้ ciphertext และส่งไปปลายทาง เครื่องบัญชาจะใช้กุญแจรหัสลับถอดข้อมูลได้ ซึ่งข้อมูลนั้นอาจจะถูกแปลงความหมายเป็นอย่างอื่นโดย เครื่องบัญชาไม่รู้ได้ว่าข้อมูลนั้นถูกเปลี่ยนแปลงไป ถึงขั้นตอนทำให้ทราบได้ว่าข้อมูลนั้นมีความสมบูรณ์สิ่งที่จำเป็นต้องทราบได้แก่ข้อมูลนั้นถูกส่งตัวจริงและข้อมูลนั้นไม่ถูกแก้ไขจากต้นฉบับจริงมีอยู่สองส่วนที่ระบบที่ต้องการความสมบูรณ์ต้องทำได้เรียกวินัันนว่า “ตัวพิสูจน์ตัวจริง(authenticators)”

ตัวพิสูจน์ตัวจริง เป็นค่าหนึ่งที่ส่งมาพร้อมข้อมูล เพื่อใช้เป็นข้อมูลสำหรับที่ศูนย์ตัวจริงซึ่งข้อมูลนั้นจะประกอบไปด้วยข้อมูลที่เข้าบ่งบอกตัวผู้ส่งและข้อมูลที่เข้าบ่งบอกความสมบูรณ์ของเพย์โหลด

ในการทดสอบความสมบูรณ์ของเพย์โหลด เป็นแนวคิดเดียวกับขั้นตอน checksum หรือ CRC(Cyclic redundancy check) ซึ่งหากเกิดการแก้ไขข้อมูลจะทำให้ CRC เปลี่ยนไป สำหรับการตรวจสอบความสมบูรณ์ของเพย์โหลดในเทคโนโลยีรหัสลับ ใช้กระบวนการเรียกว่าการ แฮช

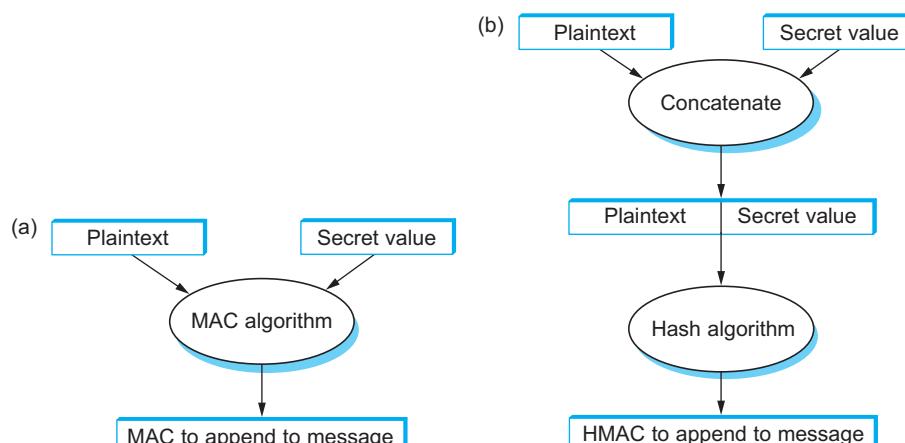
กระบวนการแฮชเป็นกระบวนการทางเทคโนโลยีรหัสลับเพื่อให้ได้ข้อมูลที่มีขนาดคงที่และใช้เป็นตัวอ้างถึงตัวข้อมูล plaintext ได้ ข้อมูลภายหลังจากการ แฮช ไม่สามารถถอดกลับไปเป็น plaintext ได้ จึงนิยมนำ แฮช เข้ารหัสพางส์เวิร์ดก่อนบันทึกลงฐานข้อมูล

เทคโนโลยีรหัสลับที่เกี่ยวข้องกับแฮชกู้ใช้มาหลายปีถึงปัจจุบันที่นิยมแพร่หลายในช่วงหลายปีที่แล้วได้แก่ Message Digest 5 (MD5) และ Secure Hash Algorithm (SHA) การศึกษาพบว่า MD5 มีความอ่อนแองซึ่งเป็นการโจมตีด้วยวิธี Hash collision ทำให้กระยะหลังมีการใช้งานเอกสารห้าม MD5 ปัจจุบัน NIST แนะนำให้ใช้ SHA-3 นับตั้งแต่ 2015

ผลลัพธ์ที่ได้จากการ แฮช ในบางครั้งเรียกว่า digest encryption ซึ่งวิธีนี้เป็นการใช้กุญแจส่วนตัวในการเข้ารหัสผลลัพธ์ที่ได้เรียกว่า “ลายเซ็นดิจิทัล” ซึ่งผลจากการได้เจสันทำให้เอกสารอิเล็กทรอนิกส์สามารถลงนามได้ เช่นเดียวกับการลงนามผ่านลายเซ็น(signature)

การพิสูจน์ตัวจริงอิกรูปแบบได้แก่ การใช้เข้ารหัส(encryption) โดยใช้กุญแจรหัสลับ แต่ประยุกต์การทำงานเหมือน แฮช อธิบายได้ตามรูปที่ 6.5 ผลลัพธ์ที่ได้มีสองรูปแบบได้แก่ แบบแรกในรูปที่ 6.5(a) message authentication code (MAC) เป็นกระบวนการนำ plaintext มาเข้ารหัสด้วย กุญแจรหัสลับ จะได้ผลลัพธ์เป็น MAC และแบบ MAC ไปร่วมเพย์โหลด แบบที่สอง ตามรูปที่ 6.5(b) เรียกว่า hashed message authentication code (HMAC) เป็นการนำเพลส์ร์จกีเข้ารหัสด้วยกุญแจรหัสลับ และนำผลลัพธ์นั้นมาต่อท้าย plaintext และนำข้อมูลที่ได้จากการต่อ กันนี้มาเข้า แฮช อีกครั้ง

ซึ่งข้อมูลที่ส่งไปปลายทางอาจจะเป็นข้อมูลที่ไม่ต้องเข้ารหัส แต่มีออบข้อมูลส่วนที่เป็น MAC หรือ HMAC ไปด้วยจะทำให้ผู้รับทราบว่าข้อมูลนั้นถูกแก้ไขระหว่างทางหรือไม่ การไม่เข้ารหัสข้อมูลยังมีเป็นความจำเป็นสำหรับสิ่งที่ต้องการเผยแพร่ให้คนจำนวนมากทราบ ขณะที่สิ่งที่จำเป็นคือการยืนยันได้ว่าข้อมูลนั้นไม่ได้ถูกเปลี่ยนแปลงไปจากต้นฉบับ



รูปที่ 6.5: การประมวลผล MAC (a) เทียบกับการประมวลผล HMAC (b)
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

6.3 ขั้นตอนแลกเปลี่ยนกุญแจรหัสลับ

การใช้เฟอร์ และ ตัวพิสูจน์ตัวจริง ผู้สื่อสารต้องมีกุญแจรหัสลับ ในกรณีที่ใช้เฟอร์แบบ secret-key คู่สื่อสารจะต้องมีกุญแจรหัสลับตรงกันทั้งสองฝ่าย คำตามมา คู่สื่อสารจะส่งกุญแจรหัสลับให้กันอย่างไร ? คำตอบคือคู่สื่อสารไม่ได้ส่งกุญแจรหัสลับ แต่ใช้ session keys สำหรับงานเข้ารหัสที่ใช้เวลาไม่นาน และใช้ predistributed keys สำหรับงานข้ารหัสที่ใช้เวลานาน

สำหรับ session key ใช้เพื่อเข้ารหัสการสื่อสารแค่บางช่วงเวลา เมื่อใช้เสร็จแล้ว session key จะถูกลบไป เมื่อต้องการสื่อสารครั้งต่อจะสร้างจาก session key ใหม่ โดยที่ session key ถูกสร้างขึ้นจาก กุญแจรหัสลับร่วมกับการสู่มารหัส

เหตุผลที่แบ่งการทำงานระหว่าง session key และ predistributed key มีดังนี้

- ช่วงเวลาจำกัดเวลาใช้งานของกุญแจรหัสลับทำให้หน่วยประมวลผลใช้ระยะเวลาไม่นานทำให้ลดความเสี่ยงจากการถูกโจมตีได้
- public key เป็นใช้เฟอร์ที่มีความปลอดภัยสูง มีประโยชน์ในการใช้การพิสูจน์ตัวจริง แต่เมื่อนำมาใช้ส่งข้อมูลทำได้ช้า

ในหัวข้อนี้จะกล่าวถึงมีการแจกจ่ายกุญแจรหัสลับ (predistributed keys) อย่างไรให้ปลอดภัย จะกำหนดให้มีชื่อตัวละครสำคัญสองคนได้แก่ “อลิช” และ “บ๊อบ” ซึ่งการสื่อสารนี้ไม่ได้เป็นการสื่อสารกันโดยตรง อาจเป็นการส่งต่อคอมมูนิเคชันกันก่อน อลิช ต้องการสื่อสารกับ บ๊อบ อาจไม่เป็นการสนทนาระหว่าง อลิช กับ บ๊อบ โดยตรงแต่เป็นการส่งข้อมูลโดย อลิช ฝากไปยังผู้อื่นที่อยู่ระหว่างทางและผู้อื่นนั้นส่งต่อข้อมูลไปเรื่อยเรื่อยจนกว่าจะถึง บ๊อบ ซึ่งระหว่างทางอาจเป็นการส่งผ่านผู้ประสงค์ร้ายได้

6.3.1 Predistribution of Public Keys

อัลกอริทึมในการสร้างกุญแจรหัสลับของกุญแจสาธารณะที่ประกอบด้วย กุญแจสาธารณะ และ กุญแจส่วนตัว ที่สัมพันธ์กันนั้น มีซอฟต์แวร์ที่ทำงานนี้ให้เช่น像 เช่น RSA อัลช ต้องการใช้งานใช้เฟอร์แบบ public key สามารถสร้างคู่กุญแจกุญแจส่วนตัว และ กุญแจสาธารณะสำหรับตัวเองได้ โดยเก็บกุญแจส่วนตัวไว้กับตัว และเผยแพร่กุญแจสาธารณะสู่สาธารณะ แล้วจะเผยแพร่กุญแจสาธารณะได้อย่างไร—แล้วอัลชจะยืนยันความเป็นตัวจริงอย่างไร—คนอื่นๆ จะมั่นใจได้ว่ากุญแจรหัสลับนั้นเป็นของเจ้าของอย่างไร ที่ไม่ใช่ทางอีเมลหรือเว็บเนื่องจากผู้ประสงค์ร้ายสามารถปลอมแปลงการอ้างสิทธิ์ที่นาเชื่อถือพอกๆ กันว่ากุญแจรหัสลับ x เป็นของอัลช เมื่อ x เป็นของผู้ประสงค์ร้ายจริงๆ เป็นคำเตือนพื้นฐานที่มิตรระบบการแลกจ่ายกุญแจสาธารณะ ซึ่งจะกล่าวในลำดับต่อไป

ระบบที่ทำหน้าที่รับรองการทำงานที่เชื่อมโยงข้อมูลระหว่างกุญแจสาธารณะกับข้อมูลระบุตัวตน¹ มีชื่อว่า พีเคไอ(Public Key Infrastructure) ซึ่งมีความสามารถยืนยันตัวตนและที่ผูกไว้กับระบบศูนย์กลาง ยก

¹ระบบบริหารจัดการที่ทำให้ทราบว่ากุญแจเป็นของใคร

ตัวอย่างเช่น หากอலิซกับบ๊อบรู้จักกัน มาพบกันแล้วอัลิซมอบกุญแจ กุญแจสาธารณะ ให้บ๊อบกับมีอีกตัว อาจอยู่บนนามบัตร แต่ถ้าบ๊อบเป็นองค์กร แต่อลิซเป็นบุคคล การพิสูจน์ตัวจริงอาจเป็นการแสดงบัตรประจำตัว ซึ่งอาจรูปถ่ายหรือลายเซ็นมือ หากอลิซ และบ๊อบ เป็นคอมพิวเตอร์ของบริษัทเดียวกัน ผู้ดูแลระบบสามารถยืนยันตัวจริงอัลิซกับ บ๊อบ ด้วยกุญแจสาธารณะของ อัลิซ

ในเมื่อบ๊อบรู้ว่ากุญแจสาธารณะของอัลิซคือ x ซึ่งสามารถเผยแพร่ได้ ทำให้ระบบสามารถขยายเป็นใช้ทำหน้าที่ลายเซ็นดิจิทัลได้ ซึ่งเป็นแนวคิดเรื่องความไว้วางใจได้ ตัวอย่างเช่น สมมติว่ามอลลี่(Mally)ได้รับกุญแจสาธารณะของบ๊อบ จากภายนอกเครือข่าย และมอลลี่รู้ว่าบ๊อบนี้เป็นตัวจริง บ๊อบสามารถส่งข้อความถึงมอลลี่เพื่อยืนยันว่ากุญแจรหัสลับของอัลิซคือ x และเนื่องจากมอลลี่ทราบกุญแจสาธารณะของบ๊อบว่าเป็นตัวจริงอยู่แล้ว มอลลี่จึงตรวจสอบความถูกต้องของข้อความได้ว่ามาจากบ๊อบตัวจริง (การลงนามแบบดิจิทัล ซอฟต์แวร์จะกำหนดให้บ๊อบ สร้างแซชเข้ารหัสที่เข้ารหัสโดยใช้กุญแจส่วนตัวของบ๊อบ) เนื่องจากมอลลี่ได้ให้ความไว้วางใจกับบ๊อบแล้ว ทำให้เข้าใจว่าบ๊อบบอกความจริง จึงทำให้รู้ว่ากุญแจของอัลิซ คือ x แม้ว่าบ๊อบไม่เคยพabolizel

การใช้ลายเซ็นดิจิทัล บ๊อบ สามารถรับรองกุญแจของอัลิซได้ โดยเผยแพร่ด้วยการลงนามแบบดิจิทัล (ถือว่า กุญแจสาธารณะของอัลิซเป็นตัวจริงโดยผ่านการรับรองของบ๊อบ) บ๊อบสามารถส่งสำเนาไปรับรองให้กับอัลิซ หรือโพสต์บนเว็บไซต์ก็ได้ เมื่อมีคนต้องการตรวจสอบกุญแจสาธารณะของอัลิซ สามารถทำได้โดยคัดลอกสำเนาไปรับรอง ซึ่งอาจได้จากอัลิซ โดยตรง หรือจากเว็บไซต์ใดๆ ก็ได้ ตราบใดที่ยังเชื่อถือบ๊อบ และรู้รหัสสาธารณะของบ๊อบ สามารถดูได้ว่าการเริ่มต้นจากการมีผู้ที่นาเชื่อถือเป็นผู้รับรองกุญแจรหัสลับของผู้อื่น (ในกรณีนี้ คือบ๊อบ) สามารถสร้างกุญแจรหัสลับที่เชื่อถือได้จำนวนมากได้อย่างไร ในกรณีนี้บ๊อบ มีบทบาทที่มักเรียกว่า “ผู้มีหน้าที่ออกใบรับรอง” หรือเรียกว่า CA(Collision Avoidance) และการรักษาความปลอดภัยทางอินเทอร์เน็ตในปัจจุบันส่วนใหญ่ขึ้นอยู่กับ CA VeriSign ซึ่งเป็น CA(Certificate Authority) เชิงพาณิชย์ที่มีชื่อเสียงในปัจจุบัน

หนึ่งในมาตรฐานสำคัญสำหรับใบรับรองดิจิทัลเรียกว่า X.509 มาตรฐานนี้มีรายละเอียดอยู่ แต่โครงสร้างพื้นฐานของใบรับรองประกอบด้วยรายการสำคัญต่อไป:

- มีการระบุตัวตนของบุคคลที่ได้รับการรับรอง
- มีการระบุ public-key สำหรับผู้ที่ได้รับการรับรอง
- สามารถยืนยันตัวตนของผู้ลงนาม
- ใช้ทำลายเซ็นดิจิทัล
- มีตัวระบุอักษรที่มีใช้ทำลายเซ็นดิจิทัล (แยกการเข้ารหัสได้และใช้เฟอร์ที่ใช้)

นอกจากนี้จากนี้มีวันหมดอายุของใบรับรอง จะเห็นการใช้งานคุณลักษณะนี้ในลำดับถัดไป

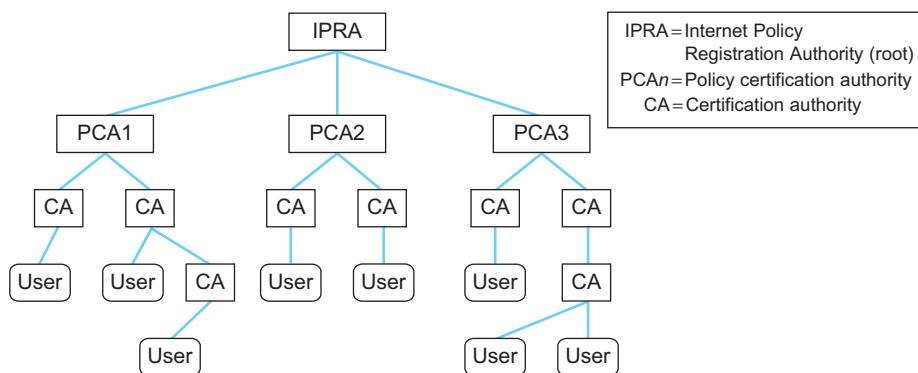
เนื่องจากใบรับรองสร้างการเชื่อมโยงระหว่างข้อมูลระบุตัวตนและกุญแจสาธารณะ จึงควรทำความเข้าใจความหมายของ “identity” ให้ละเอียดขึ้น ตัวอย่างเช่น ใบรับรองที่ระบุว่า “กุญแจสาธารณะนี้เป็นของ John Smith” อาจไม่มีความหมายหาก ไม่สามารถระบุ John Smiths เป็นคนใดในจำนวนหลายพันคน ดังนั้น ใบรับรองจะต้องใช้พื้นที่ระบุชื่อที่กำหนดໄວ่อย่างดี สำหรับอธิบายตัวตนของผู้ที่ได้รับการรับรอง ตัวอย่างเช่น ใบรับรองมักจะออกให้คู่กับอีเมล์และโดเมนที่ผู้ใช้งานนั้นสังกัดอยู่

6.3.2 ระบบออกใบรับรองดิจิทัล

ในรูปแบบการทำงานของระบบความไว้วางใจนี้ ความไว้วางใจมีผลลัพธ์เป็นใบหน้า หมายถึงคู่สื่อสารจะความไว้วางใจในสักคนนั้นเป็นการ ความไว้วางใจโดยไม่สงสัยหรือไม่เกี่ยวกับความไว้วางใจเลย โดยเดลแคนวิคิดนี้นำมาใช้กับระบบใบรับรองดิจิทัล ซึ่งช่วยให้สามารถสร้างห่วงโซ่ความไว้วางใจได้ หาก X รับรองว่ากุญแจสาธารณะบางอันเป็นของ Y จากนั้น Y ก็รับรองว่ากุญแจสาธารณะอื่นเป็นของ Z แสดงว่ามีใบรับรองที่เชื่อถือได้หลายชุดตั้งแต่ X ถึง Z แม้ว่า X และ Z จะไม่เคยเจอกันเลย หากรู้กุญแจสาธารณะของ X และไว้วางใจ X และ Y จะเชื่อใบรับรองดิจิทัลที่ได้รับจาก Z ได้ กล่าวอีกนัยหนึ่ง สิ่งที่ต้องมีคือลำดับของใบรับรองทั้งหมดนามโดยหน่วยงานที่เชื่อถือได้ ทราบได้ที่ยังสามารถติดตามไปถึงหน่วยงานที่ออกใบรับรองนั้นได้

ผู้ออกใบรับรองหรือ CA ซึ่งเป็นระบบซอฟต์แวร์หนึ่งที่ทำหน้าที่ตรวจสอบข้อมูลประจำตัวและการออกใบรับรองกุญแจสาธารณะ มีผู้ให้บริการ CA เชิงพาณิชย์ หรือเป็น CA ของภาครัฐ รวมถึงสามารถตั้งตัวเป็น CA ได้เอง ซึ่งหากต้องการใช้ CA ต้องรู้กุญแจรหัสลับของตัวเอง สิ่งแรกที่ CA ทำคือทำให้ผู้ใช้มอบรับ กุญแจสาธารณะของ CA หลังจากนี้ CA จะเป็นผู้รับรองใบรับรองดิจิทัล(Digital Certificate)จำนวนมากโดยที่ผู้ใช้งานเพียงรู้จักว่า CA เป็นตัวจริง จะถือว่าใบรับรองดิจิทัลทั้งหมดที่ออกให้โดย CA นั้นเป็นตัวจริงโดยปริยาย

วิธีปกติในการสร้างลำดับใบรับรองดิจิทัล มีการเรียงตามโครงสร้างแบบต้นไม้ ดังแสดงในรูปที่ 6.6 หากทุกคนมีกุญแจสาธารณะของ Root CA ผู้ที่เป็นสมาชิกคนใดก็ได้สามารถออกใบรับรองให้กับสมาชิกรายอื่น ได้โดยอาศัยการสร้างลำดับของความไว้วางใจ



รูปที่ 6.6: โครงสร้างต้นไม้ของระบบใบรับรองอิเล็กทรอนิกส์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

มีปัญหาสำคัญประการหนึ่งเกี่ยวกับการสร้างความไว้วางใจ คือแม้ว่าจะแนใจว่ามีกุญแจสาธารณะของ Root CA และ ก็ยังต้องแน่ใจว่า CA ทุกแห่งตั้งแต่บนสุดลงมานั้น ยังทำงานอย่างถูกต้อง หากมี CA เพียงจุดเดียวในระบบออกใบรับรองดิจิทัลโดยไม่ตรวจสอบตัวตน จะกลายเป็นว่าลำดับใบรับรองดิจิทัลที่ถูกต้องจะหมดความหมายไปโดยทันที ตัวอย่างเช่น Root CA อาจออกใบรับรองให้กับ CA ระดับที่สองและตรวจสอบอย่างละเอียดว่าซื่อสัมภានในรับรองตรงกับชื่อรุกิจของ CA อื่นที่ได้ออกใบรับรองดิจิทัลไปแล้ว แต่ CA ระดับที่สองนั้นอาจขายใบรับรองให้กับทุกคนที่ขอซื้อ โดยไม่ยืนยันตัวตน ปัญหานี้ส่งผลกระทบลำดับของความไว้วางใจ ทำให้เกิดใบรับรอง X.509 ที่กำหนดให้มีส่วนควบคุมคุณสมบัติบางประการของ CA

ผู้ให้บริการ CA สามารถมีได้มากกว่าหนึ่งรุ่ห์ ซึ่งเรื่องปกติในการรักษาความปลอดภัยธุกรรมบนเว็บ ในปัจจุบัน เช่น ในเบราว์เซอร์ Firefox และ Internet Explorer จะติดตั้ง CA มาพร้อมกับซอฟต์แวร์ โดยผู้ผลิตเบราว์เซอร์ได้เลือกติดตั้งกุญแจสาธารณะของ CA เพราะผู้ผลิตได้ความไว้วางใจ CA นั้นแล้ว และส่งต่อความความไว้วางใจนั้นไปให้กับผู้ที่เบราว์เซอร์ ผู้ใช้จะใช้ CA เป็นเครื่องมือตรวจสอบตัวจริงของเว็บไซต์ ผ่านพอร์ตโคล Secure Socket Layer (SSL)/Transport Layer Security (TLS) ซึ่งเป็นพอร์ตโคลที่ใช้บอยที่สุดในการรักษาความปลอดภัยธุกรรมทางเว็บ

6.3.3 การเพิกถอนใบรับรอง

CRL (รายการเพิกถอนใบรับรอง) ปัญหาหนึ่งที่เกิดขึ้นกับใบรับรองดิจิทัลคือวิธีการเพิกถอน ทำไมเรื่องนี้จึงสำคัญ? สมมติว่าบื้อบ ถูกเจ้ารหัสผ่านหรือสังสัยว่ามีคนรู้กุญแจรหัสลับของตนเอง คนนั้นอาจมีใบรับรองที่ยืนยันว่าตนเองเป็น บื้อบ หรือเรียกได้ว่าเป็นเจ้าของกุญแจสาธารณะและกุญแจส่วนตัวของบื้อบ เพื่อแก้ปัญหานี้ การเพิกถอนใบรับรองที่ถูกกับกุญแจรหัสลับเก่าที่ถูกกับข้อมูลระบุตัวตนของบื้อบออก เพื่อป้องกันไม่ให้ผู้รายนำไปใช้หลอกลวงผู้อื่นได้

วิธีแก้ปัญหาเบื้องต้นนั้นทำได้่ายโดย CA สามารถอกรายการเพิกถอนใบรับรองได้ (รายการเพิกถอนใบรับรอง(Certificate revocation list)) ซึ่งเป็นรายการใบรับรองดิจิทัลที่ลงนามแบบดิจิทัลสำหรับใบรับรอง ดิจิทัลที่ถูกเพิกถอน โดยที่ รายการเพิกถอนใบรับรอง อัปเดตเป็นระยะและเผยแพร่สู่สาธารณะ เนื่องจากมีการลงนามแบบดิจิทัลจึงสามารถโพสต์บนเว็บไซต์ได้ ยกตัวอย่างเช่น เมื่อลิชต์ได้รับใบรับรองดิจิทัลของบื้อบโดยที่ ที่ ลิชต์ต้องการตรวจสอบ จะตรวจสอบใน CRL ที่เผยแพร่ล่าสุด ซึ่งที่ออกโดย CA เสียก่อน ทราบได้ที่ใบรับรองยังไม่ถูกเพิกถอนก็ถือว่าใบรับรองดิจิทัลของบื้อบนั้นเป็นตัวจริง โดยที่ต้องคำนึงด้วยว่าหากใบรับรองดิจิทัลทั้งหมดมีอายุ CRL ก็จะมีขนาดเพิ่มขึ้นเรื่อยๆ เนื่องจากไม่สามารถนำใบรับรองออกจาก CRL ได้ เนื่องจากต้องบันทึกใบรับรองดิจิทัลที่ถูกเพิกถอน ด้วยเหตุนี้ จึงมีการแนะนำวันหมดอายุกับใบรับรองเมื่อมีการออกใบรับรอง ช่วยให้จำกัดระยะเวลาที่ใบรับรองที่ถูกเพิกถอนต้องอยู่ใน CRL ทันทีที่หมดอายุ

6.3.4 Predistribution of Secret Keys

ในระบบ Public Keys หากอุปกรณ์ต้องการสื่อสารข้อมูลความลับกับบื้อบ โดยไม่มีกุญแจรหัสลับของบื้อบได้โดยใช้กุญแจสาธารณะของบื้อบ แต่สำหรับ Secret key จำเป็นต้องมีแผนการจ่ายกุญแจรหัสลับล่วงหน้า ซึ่งการกำหนดการแจกจ่ายกุญแจรหัสลับล่วงหน้านั้นทำได้ยากกว่า Public key ด้วยเหตุผลสองประการ:

- กุญแจรหัสลับสำหรับใช้ติดต่อสื่อสารจะเป็นกุญแจรหัสลับที่ไม่ซ้ำกับคุณสื่อสารอื่นเพื่อเป็นประโยชน์ต่อการตรวจสอบสิทธิ์และการรักษาความลับ แต่ด้วยการนี้ทำให้ต้องมีกุญแจรหัสลับแต่ละคู่ที่ต้องการสื่อสารจำนวนมาก หากมีคู่สื่อสาร N เครื่อง แสดงว่าต้องมีกุญแจรหัสลับ $N(N-1)/2$
- กุญแจลับต้องเก็บเป็นความลับซึ่งไม่เหมือนกับกุญแจสาธารณะ

โดยสรุปแล้วจะต้องมีกุญแจรหัสลับจำนวนมากเพื่อใช้ในการสื่อสาร และไม่สามารถใช้รูปแบบกุญแจรหัสลับที่ให้ทุกคนอ่านได้ วิธีแก้ปัญหานี้คือการใช้ Key Distribution Center (KDC) ทำหน้าที่แฮร์กุญแจรหัสลับ

ลับกับผู้ที่ต้องการแลกเปลี่ยนกุญแจรหัสลับ ซึ่งทำให้จำนวนกุญแจรหัสลับลดลงเหลือ N-1 ที่จัดการได้มากขึ้น เมื่ออธิบายต้องการสื่อสารกับบ่อ จะไม่เป็นการสื่อสารผ่าน KDC แต่ KDC จะมีส่วนร่วมในโพรโทคอลที่รับรองความถูกต้องของอธิช และบ่อ โดยใช้กุญแจรหัสลับที่ KDC แชร์กับแต่ละกุญแจรหัสลับเรียบร้อยแล้ว และสร้าง session key ใหม่เพื่อให้อธิช และบ่อใช้ จากนั้นอธิชและบ่อจะสื่อสารโดยตรงโดยใช้ session key ที่ได้มา ซึ่ง เคอร์เบрос คือระบบที่ทำหน้าที่ดังกล่าว มีอธิบาย เคอร์เบрос ในส่วนต่อไป

6.3.5 Diffie-Hellman Key Exchange

วิธีหนึ่งในการสร้างกุญแจรหัสลับสำหรับใช้ร่วมกันคือการใช้โพรโทคอลการแลกเปลี่ยนกุญแจรหัสลับแบบ Diffie-Hellman (Diffie และ Hellman, 1976) ซึ่งไม่ต้อง แจกจ่ายกุญแจรหัสลับล่วงหน้า สำหรับระบบนี้เปิดให้สามารถถักฟังข้อมูลความที่แลกเปลี่ยนกันระหว่างกันได้ เช่น อธิชกับบ่อสามารถอ่านได้ แต่ผู้รอบฟังเมื่ออาจจู่ๆ กุญแจรหัสลับที่อธิชและบ่อใช้

Diffie-Hellman ไม่ได้ออกแบบให้รับรองความถูกต้องสมบูรณ์ เนื่องจากหากไม่แน่ใจว่ากำลังสื่อสารกับใครอยู่จะไม่มีประโยชน์ในการสื่อสารอย่างปลอดภัย โดยปกติแล้ว Diffie-Hellman จะความปลอดภัยด้วยวิธีเดียวกันนี้เพื่อให้สามารถครอบครองความถูกต้องสมบูรณ์ การใช้งานหลักของ Diffie-Hellman คือโพรโทคอล Internet Key Exchange (IKE) ซึ่งเป็นศูนย์กลางของสถาปัตยกรรม IP Security (IPsec)

โพรโทคอล Diffie-Hellman มีพารามิเตอร์สองตัวได้แก่ คือ p และ g ทั้งคู่เป็นกุญแจแบบ public key ซึ่งผู้ใช้ทุกคนในระบบสามารถนำไปใช้ได้ พารามิเตอร์ p เป็นเลขจำนวนเฉพาะ และเลขจำนวนเต็มซึ่งได้จาก mod p (ยกเว้น modulo p) มีค่าอยู่ระหว่าง 0 ถึง $p-1$ จากที่ $x \bmod p$ จะได้เศษเหลือเป็น เรียกว่า group g เรียกว่าเป็น generator ถือเป็น primitive root² ของ p

ตัวอย่างเช่น ถ้า p เป็นจำนวนเฉพาะ 5 (ระบบใช้งานจริงจะใช้จำนวนใหญ่กว่านี้มาก) อาจเลือก 2 เป็น g เนื่องจาก:

$$\begin{aligned} 1 &= 2^0 \pmod{p} \\ 2 &= 2^1 \pmod{p} \\ 3 &= 2^2 \pmod{p} \\ 4 &= 2^3 \pmod{p} \end{aligned}$$

สมมติว่าอธิชและบ่อต้องการใช้กุญแจรหัสลับร่วมกันอธิช และบ่อ และคนอื่นๆ รู้ค่าของ p และ g และ อธิช สร้างจำนวนโดยการสุ่มสุ่มได้ a และบ่อ สุ่มค่าขึ้นมาค่าหนึ่งได้ b ทั้ง a และ b เป็นเซตของเลขจำนวนเต็ม $\{1, \dots, p-1\}$ อธิชและบ่อ ติดต่อระหว่างกันผ่านช่องทางสาธารณะ—ค่าที่ส่งให้กันโดยไม่มีการเข้ารหัส—มีขั้นตอนดังต่อไปนี้: อธิช มีข้อมูลสำหรับอภิหารณ์ตั้งนี้

$$g^a \pmod{p}$$

²ทุกจำนวน n ใดๆ จาก 1 ถึง $p-1$ ที่มีบางค่า k ทำให้ $n = g^k \pmod{p}$

บັນຍາມີຂໍ້ມູນສໍາຫຼັບສ່ວນອອກສາຮາຣະນະດັ່ງນີ້

$$g^b \mod p$$

ທັງສອງແລກເປີ່ຍືນຂໍ້ມູນສາຮາຣະນະຮ່ວ່າງກັນ ໂດຍ ອລິຍະ ປະມວລຜລຂໍ້ມູນດັ່ງນີ້

$$g^{ab} \mod p = (g^b \mod p)^a \mod p$$

ແລະບັນຍາມີປະມວລຜລຂໍ້ມູນດັ່ງນີ້

$$g^{ba} \mod p = (g^a \mod p)^b \mod p$$

ຈຶ່ງ ອລິຍະ ແລະ ບັນຍາມີ $g^{ab} \mod p$ ($\Rightarrow g^{ab} == g^{ba}$) ເປັນກຸ່ມແຈຣ້ສລັບທີ່ໃຊ້ຮ່ວມກັນ

ຢັກຕ້ວາຍ່າງເຊັ່ນ $p=5$ ແລະ $q=2$ ສມມຕິອລິຍະສຸ່ມໄດ້ເລີຂ $a=3$ ແລະບັນຍາມີປຸ່ມໄດ້ເລີຂ $b=4$ ຈາກນັ້ນອລິຍະມີຂໍ້ມູນສໍາຫຼັບສ່ວນອອກສາຮາຣະນະເປັນ

$$2^3 \mod 5 = 3$$

ແລະບັນຍາມີສມກາຮສໍາຫຼັບສ່ວນຂໍ້ມູນອອກສາຮາຣະນະເປັນ

$$2^4 \mod 5 = 1$$

ເນື່ອອລິຍະ ໄດ້ຮັບຂໍ້ມູນຈາກບັນຍາ ຈະຄຳນວນໄດ້

$$g^{ab} \mod p = (2^b \mod 5)^a \mod 5 = 1$$

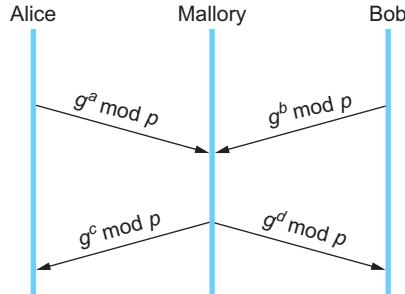
ໂດຍກາຮແທນຄ່າຂໍ້ມູນຈາກ ບັນຍາ $(2^b \mod 5)$ ຂອນເຕີຍວັນກັນກາຮຄຳນວນຂອງບັນຍາ

$$g^{ba} \mod p = (g^a \mod 5)^b \mod 5 = (3)^4 \mod 5 = 1$$

ບັນຍາຄຳນວນໄດ້ຈາກຂໍ້ມູນທີ່ຮັບຈາກອລິຍະ $(2^a \mod 5)$ ເහັນໄດ້ວ່າ ອລິຍະ ແລະບັນຍາ ໄດ້ຮັບກຸ່ມແຈຣ້ສລັບຄ່າເຕີຍວັນກັນ (ຈາກຕໍ່ວອຍາງຄືອ 1) ໂດຍໄມ້ຕ້ອງສ່ວນຄ່ານັ້ນໃນກາຮສື່ອສາຮ

ກາຮໃຊ້ Diffie-Hellman ໄມ່ຮອງຮັບຄວາມຖຸກຕ້ອງສມບູຮົນ ເປັນຫຼຸດທັງໃຫ້ເກີດກາຮໂຈມຕີໄດ້ຕື່ອກາຮ ໂຈມຕີ man-in-the-middle ສມມຸດວ່າມອລີ່ ເປັນຜູ້ຮ້າຍທີ່ມີຄວາມສາມາຮຄວບຄຸມເສັ້ນທາງຂໍ້ມູນໄດ້ ມອລີ່ຮູ້ຄ໏າ p ແລະ q ອູ່ຢູ່ແລ້ວ ໙ີ້ອັກເປົ້າເປັນສາຮາຣະນະ ແລະໄດ້ສຸ່ມຄ່າ c ແລະ d ຈຶ່ງເພື່ອໃຊ້ກັບອລິຍະແລະບັນຍາ ຕາມລຳດັບ ເນື່ອອລິຍະແລະບັນຍາ

อบส่งค่าให้กันและกัน มอลลีตักฟังและสกัดข้อมูลไม่ให้ส่งต่อ พร้อมกับเปลี่ยนเป็นข้อมูลที่มอลลีสร้างขึ้นเอง ดังในรูปที่ 6.7 ผลที่เกิดขึ้นคืออลิชและบ้อบต่างก็เข้าใจว่าตนเองได้สื่อสารกับตัวจริงทั้งที่เป็นการสื่อสารกับมอลลี



รูปที่ 6.7: การโจมตีแบบ man-in-the-middle
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

6.4 โพรโทคอลพิสูจน์ตัวจริง

หัวข้อนี้จะกล่าวถึงวิธีเข้ารหัสข้อมูล กระบวนการตัวพิสูจน์ตัวจริง การแจกจ่ายกุญแจรหัสลับ จากรายการที่กล่าวมา มีวัตถุประสงค์เพื่อทำให้โพรโทคอลมีความปลอดภัย ซึ่งจะต้องทำกับทุกแพ็กเก็ตที่ต้องการรักษาความลับ โดยการเข้ารหัสข้อมูล

มีสาเหตุหลักสองประการที่ทำให้การพัฒนาไม่ง่าย ประการแรก มีปัญหาของการโจมตีด้วยวิธีส่งข้อมูลซ้ำ(replay attack) : ผู้ประสงค์ร้าย(adversary) ส่งข้อมูลซ้ำที่ส่งเคยส่งสำเร็จก่อนหน้านี้ ตัวอย่างเช่น แพ็กเก็ตล็อกอิน(login) เข้าเว็บไซต์ที่เคยล็อกอินสำเร็จ เมื่อนำแพ็กเก็ตนั้นมาส่งซ้ำ จะทำให้ล็อกอินเข้าเว็บไซต์ได้โดยไม่จำเป็นต้องทราบรหัสผ่าน แม้ว่าแพ็กเก็ตจะไม่ใช่รูปแบบเดิมของข้อมูล แต่ตัวตรวจสอบความถูกต้องก็ยังใช้ได้ เพราะข้อมูลนั้นเคยล็อกอินผ่านได้แล้วไม่ได้แก้ไขข้อมูลนั้น จะเห็นได้ว่าการพิสูจน์ตัวจริงต้องการการปรับปรุงเพื่อแก้ปัญหาการโจมตีด้วยวิธีส่งข้อมูลซ้ำ

ของการโจมตีประเภทการโจมตีด้วยวิธีส่งข้อมูลซ้ำ อีกตัวอย่างหนึ่งเรียกว่าการโจมตีด้วยวิธีห่วงเวลา แล้วส่งข้อมูลซ้ำ(suppress-replay attack) ผู้ประสงค์ร้ายทำให้แพ็กเก็ตเดินทางช้า โดยการสกัดกั้นและการโจมตีด้วยวิธีส่งข้อมูลซ้ำในภายหลัง เพื่อให้ระบบการทำงานไม่อยู่ในรูปแบบที่ถูกต้อง ตัวอย่างเช่น ผู้ประสงค์ร้ายทำให้คำสั่งซึ่งผ่านเว็บออนไลน์ช้าลงกว่าที่ควรเป็น ไปจนถึงช่วงเวลาที่ปกติที่สามารถส่งคำสั่งซึ่งได้ ซึ่งข้อมูลนั้นได้ส่งออกไปแล้วและเป็นข้อมูลที่มีความสมบูรณ์เพียงพอต่อกำลังซึ่งแต่คำสั่งต่อไปยังเดินทางไม่ถึงปลายทาง ผู้ประสงค์ร้ายนำแพ็กเก็ตนั้นมาแก้ไขข้อมูล เช่น ที่อยู่ผู้รับสินค้าแล้วส่งไปยังปลายทาง หากระบบออกแบบไม่ดีก็จะทำให้เกิดการซื้อขายผิดพลาดได้ ปัญหานี้อาจแก้ไขด้วย session key แต่ในการออกแบบโพรโทคอล ทำได้ไม่ง่ายนัก

สิ่งที่ปัญหาทั้งสองนี้มีเหมือนกันคือการพิสูจน์ตัวจริง หากแพ็กเก็ตไม่เป็นต้นฉบับและไม่ได้ส่งทันเวลา โพรโทคอลจะต้องสามารถระบุได้ว่าแพ็กเก็ตนั้นไม่อาจนำมาใช้ได้ ไม่ใช่ยอมรับแพ็กเก็ตตามข้อมูลที่ผู้ส่งทำการส่งมา เมื่อระบบสร้าง session key ใหม่ ระบบควรทราบว่า session key นั้นใช้เพื่อสื่อสารกับใคร ระบบจะทำการพิสูจน์ตัวจริงและสร้าง session key ไปพร้อมกัน ดังนั้นมีโพรโทคอลทำขั้นตอนเสร็จ อลิช และ บ้อบ

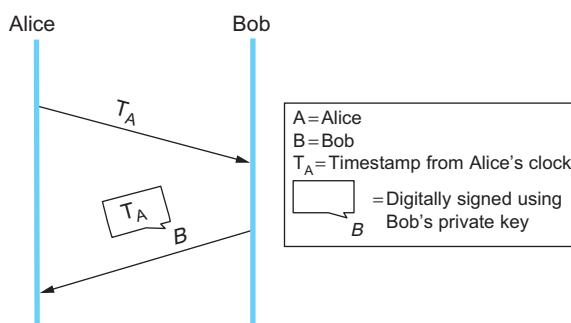
ได้รับการพิสูจน์ตัวจริงซึ่งกันและกันแล้ว และได้รับ session key ใหม่ หากไม่ได้รับ session key ใหม่ โพรโทคอลจะตรวจสอบความถูกต้องของ อลิช และ บ๊อบ ก่อน ซึ่ง session key จะช่วยให้สามารถตรวจสอบความถูกต้องของแพ็กเก็ตได้ โดยที่ว่าไป โพรโทคอลสำหรับสร้าง session key จะมีขั้นตอนการพิสูจน์ตัวจริง ยกเว้นโพรโทคอล Diffie-Hellman

มีเทคนิคพื้นฐานสำหรับตรวจสอบข้อมูลว่ามีความเป็นต้นฉบับ(Originality)และทันเวลา(timeliness)ใน โพรโทคอลพิสูจน์ตัวจริง ที่จะอธิบายก่อนที่จะกล่าวถึงโพรโทคอลที่มีลักษณะเฉพาะด้านในลำดับต่อไป

6.4.1 เทคนิควิธีทางการตรวจสอบ ต้นฉบับ และ ทันเวลา

ระบบตัวพิสูจน์ตัวจริงเพียงอย่างเดียวไม่สามารถตรวจสอบแพ็กเก็ต ที่มีคุณสมบัติต้นฉบับ หรือ ทันเวลา ได้ วิธีหนึ่งที่นำมาใช้คือการเติม timestamp มา กับแพ็กเก็ต ซึ่งจะเห็นได้ว่าเพิ่ม timestamp จะต้องตรวจสอบการแก้ไข timestamp ได้ด้วย ดังนั้นตัวพิสูจน์ตัวจริงจึงต้องทำงานได้ครอบคลุม ข้อจำกัดของ timestamp คือต้อง มีการซิงโครไนซ์กับเวลา เนื่องจากระบบจะขึ้นอยู่กับการซิงโครไนซ์ การซิงโครไนซ์เวลา ที่ต้องได้รับการป้องกัน จากภัยคุกคามด้านความปลอดภัยด้วยเช่นกัน นอกเหนือจากโจทย์ที่พบในการซิงโครไนซ์นาฬิกา อีกปัจจัยหนึ่ง คือนาฬิกาการมีหลักการทำงานจะถูกซิงโครไนซ์กับเครื่องซึ่งมีความแม่นยำ เช่น นาฬิกาอะตอม ที่ต้องการ ประทับเวลาจึงเป็นเรื่องของระดับความถูกต้องของการซิงโครไนซ์

อีกวิธีหนึ่งคือการใส่ตัวเลขสุ่ม nonce ให้ใช้เพียงครั้งเดียว แต่รวมกับแพ็กเก็ต ผู้สื่อสารสามารถ ตรวจสอบการโอนติดวยวิธีส่งข้อมูลซ้ำโดยตรวจสอบว่าเคยมีการใช้ nonce ก่อนหน้านี้หรือไม่ ข้อจำกัดของเรื่อง นี้คือ nonces ที่เคยใช้มาแล้วจะใช้อีกไม่ได้ วิธีแก้ปัญหานี้คือการรวมการใช้ timestamp และ nonces ร่วมกัน ทำให้มั่นใจได้ว่าจะมี nonces เพียงค่าเดียว ที่สามารถจัดการได้ในขณะที่ต้องการเพียงการซิงโครไนซ์ นาฬิกาแบบไม่ต้องเทียบตรงสูงก็ได้



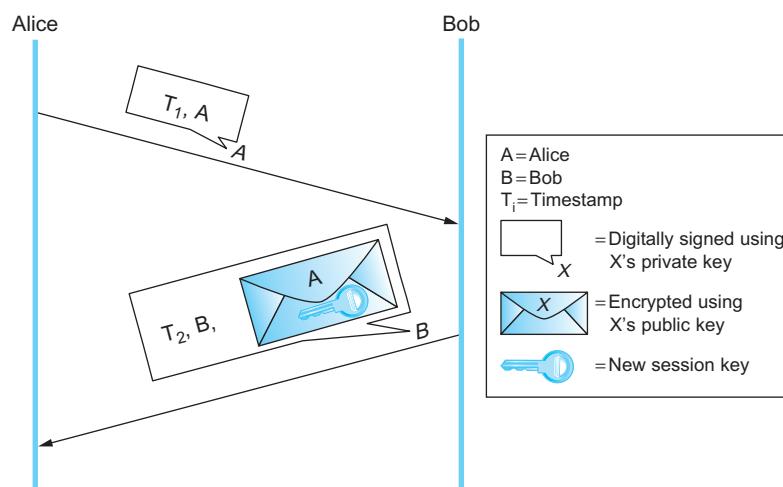
รูปที่ 6.8: โพรโทคอล challenge-response
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

อีกวิธีหนึ่งในการใช้ timestamp และ nonce คือการใช้อย่างได้อย่างหนึ่งหรือทั้งสองอย่างใน โพรโทคอล challenge-response สมมติว่าใช้ timestamp ในโพรโทคอล Challenge-Response อลิชส่ง timestamp ของ บ๊อบ โดยขอให้ บ๊อบ เข้ารหัสในข้อความที่ตอบกลับ (ซึ่งได้ใช้กุญแจรหัสลับร่วมกัน) หรือใช้ ลายเซ็นดิจิทัลในแพ็กเก็ตที่ตอบกลับ (หาก บ๊อบ มีกุญแจสาธารณะ ดังในรูปที่ 6.8 การใช้ timestamp ที่เข้ารหัสเป็นเหมือนการพิสูจน์ตัวจริงไปในตัว อลิชสามารถตรวจสอบความมีทันเวลาของ timestamp ในจังหวะ

การตอบกลับจาก บ๊อบ ได้รับเนื่องจาก timestamp มาจากสัญญาณน้ำพิกาของ อลิช ทำให้ไม่จำเป็นต้องมีการซิงค์โครในช่วงน้ำพิกา

6.4.2 โพร์โทคอลพิสูจน์ตัวจริงด้วย Public-key

ในขั้นตอนการสื่อสารต่อไปนี้ เป็นการใช้กุญแจสาธารณะของอลิช และบ๊อบที่ได้นัดแลกเปลี่ยนกุญแจระหว่างกันไว้แล้วล่วงหน้า ด้วยวิธีการบางอย่าง เช่น พีเคไอ รวมถึงกรณีที่ อลิชรวมใบรับรองดิจิทัล ไว้ในแพ็คเก็ตแรกที่ส่งถึงบ๊อบ และกรณีที่ บ๊อบ สือคันใบรับรองดิจิทัลเกี่ยวกับอลิช ได้ ในครั้งที่รับแพ็คเก็ตครั้งแรก

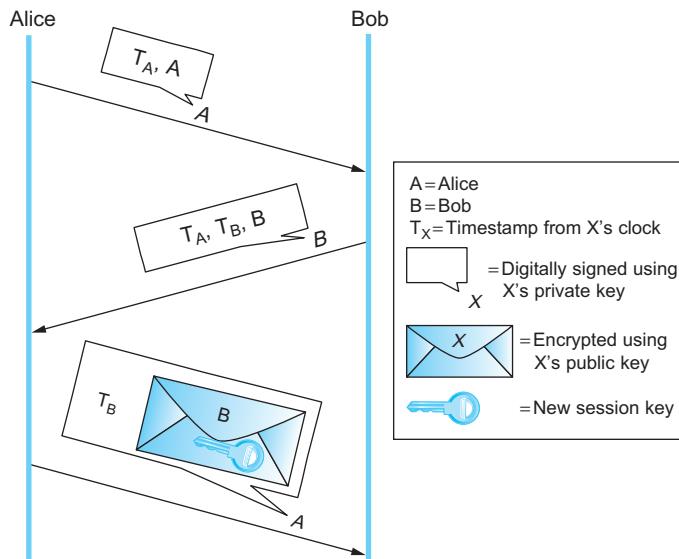


รูปที่ 6.9: การใช้ public-key พิสูจน์ตัวจริงที่ต้องการ synchronization
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

โพร์โทคอลในรูปที่ 6.9 ใช้การซิงค์โครในช่วงน้ำพิกาของอลิช และ บ๊อบ โดยอลิชส่งข้อความถึงบ๊อบพร้อม timestamp และได้พิสูจน์ตัวจริงข้อความที่ส่งพร้อมแบบลายเซ็นดิจิทัลมาด้วย เมื่อบ๊อบได้รับจึงใช้ลายเซ็นดิจิทัลตรวจสอบความถูกต้องของแพ็คเก็ตและทำ timestamp เป็นการยืนยัน เสร็จแล้วบ๊อบ ตอบกลับพร้อม timestamp และข้อมูลพิสูจน์ตัวจริงของตนเอง รูปแบบ plaintext รวมถึงส่ง session key ตัวใหม่ ที่เข้ารหัส (เพื่อรักษาความลับ) โดยใช้กุญแจสาธารณะของ อลิช ซึ่งขั้นตอนทั้งหมดนี้ได้รับการลงนามด้วยลายเซ็นดิจิทัล อลิชสามารถตรวจสอบความถูกต้องและความใหม่ของข้อความได้ ดังนั้นเรอเจ็งรู้ว่าเธอสามารถความไว้วางใจ session key ที่ได้รับมาใหม่นี้ได้ และเพื่อจัดการกับการซิงค์โครในช่วงน้ำพิกา timestamp จะเพิ่มใน nonce

โพร์โทคอลต่อมา (รูปที่ 6.10) คล้ายกับก่อนหน้านี้ แต่ไม่ออาศัยการซิงค์โครในช่วงน้ำพิกา อลิชส่งข้อความที่ลงนามด้วยลายเซ็นดิจิทัลไปให้บ๊อบ พร้อม timestamp และข้อมูลที่ใช้ระบุตัวจริง เนื่องจากนาฬิกาของทั้งสองคนยังไม่ซิงค์โครในช่วง บ๊อบ จึงไม่อาจแน่ใจได้ว่าข้อความนั้นเป็นข้อความใหม่ บ๊อบส่งข้อความที่ลงนามด้วยลายเซ็นดิจิทัลกลับไปที่อลิช พร้อมแบบ timestamp ที่อลิชส่งมา ก่อนหน้านี้ พร้อมกับแบบ timestamp ใหม่ของตนและข้อมูลยืนยันตัวตนไปให้อลิช เมื่ออลิชได้รับแพ็คเก็ต สามารถตรวจสอบความใหม่ของการตอบกลับของบ๊อบโดยเทียบเวลาปัจจุบันของตนกับ timestamp ที่มาจากบ๊อบ จากนั้นอลิชก็ส่งข้อความที่มีลายเซ็นดิจิทัลให้กับ บ๊อบ โดยมี timestamp เวลาเดิมและ session key ใหม่โดยเข้ารหัสด้วยกุญแจสาธารณะขอ

งบอป เมื่อบอปได้รับแพ็กเก็ตสามารถตรวจสอบความใหม่ของข้อความได้เนื่องจากการ timestamp มาจาก นาฬิกาของบอปเอง ดังนั้นบอปสามารถความไว้วางใจ session key ที่ได้มาใหม่นี้ได้



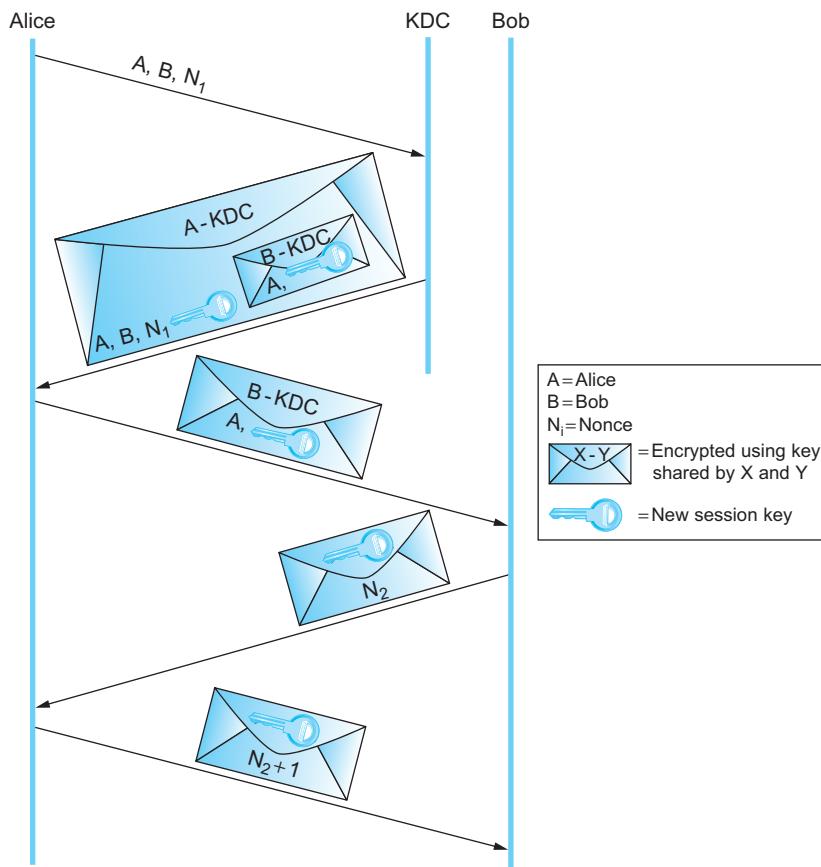
รูปที่ 6.10: การใช้ public-key พิสูจน์ตัวจริงแบบไม่ต้องการ synchronization
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

6.4.3 โพรโทคอลพิสูจน์ตัวจริงด้วย Secret-key

เฉพาะในระบบขนาดเล็กที่จะใช้ประโยชน์จากการแลกจ่ายกุญแจรหัสลับได้กับทุกเครื่องที่ต้องการสื่อสารได้ กุญแจรหัสลับแตกต่างกันทั้งหมด ในที่นี่จะสนใจระบบขนาดใหญ่ ที่สามารถขยายการให้บริการได้ โดยที่แต่ละ เอ็นทีจะมีแม่กุญแจ(master key) เป็นของตัวเองที่ สามารถแชร์กับ Key Distribution Center (KDC) ได้ เท่านั้น ในกรณีนี้ โพรโทคอลสำหรับการพิสูจน์ตัวจริง แบบใช้ secret key จะเกี่ยวข้องกับสามฝ่าย: อลิช บอป และ เคดีซี(Key Distribution Center) ผลลัพธ์สุดท้ายของโพรโทคอลได้ความสามารถตรวจสอบสิทธิ์คือ session key ที่ใช่ว่ามีกันระหว่าง อลิช และ บอป ซึ่งใช้ในการสื่อสารกันได้โดยตรง

โพรโทคอลการตรวจสอบสิทธิ์ Needham-Schroeder แสดงไว้ในรูปที่ ch7:10 มี เคดีซีที่ไม่สามารถ ตรวจสอบความถูกต้องของข้อความเริ่มต้นของ อลิช และไม่ได้แลกเปลี่ยนข้อมูลกับ บอปมาก่อน ในทางกลับ กัน เคดีซี ได้ใช้ข้อมูลที่ได้รับจากแม่กุญแจของอลิช และ บอป สร้างคู่อินพุตไม่ได้ ยกเว้นอลิช (มีเพียง อลิช เท่านั้นที่สามารถถอดรหัสได้)

nonce ในสองแพ็กเก็ตแรกคือการรับรอง อลิช ข้อมูลจาก เคดีซีไม่ใช่ข้อมูลช้า แพ็กเก็ตที่สองและสาม ประกอบด้วย session key และข้อมูลระบุตัวจริงของอลิช ซึ่งเข้ารหัสไว้ด้วยกันโดยใช้แม่กุญแจของบอป เป็น ใบรับรองดิจิทัลแบบ secret-key ทำให้สามารถลงนามโดย เคดีซี ที่มี session key ที่แนบมาของ อลิช และ บอป



รูปที่ 6.11: พิสูจน์ตัวจริงโดยโพรโทคอลแบบ Needham-Schroeder
ลิขสิทธิภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

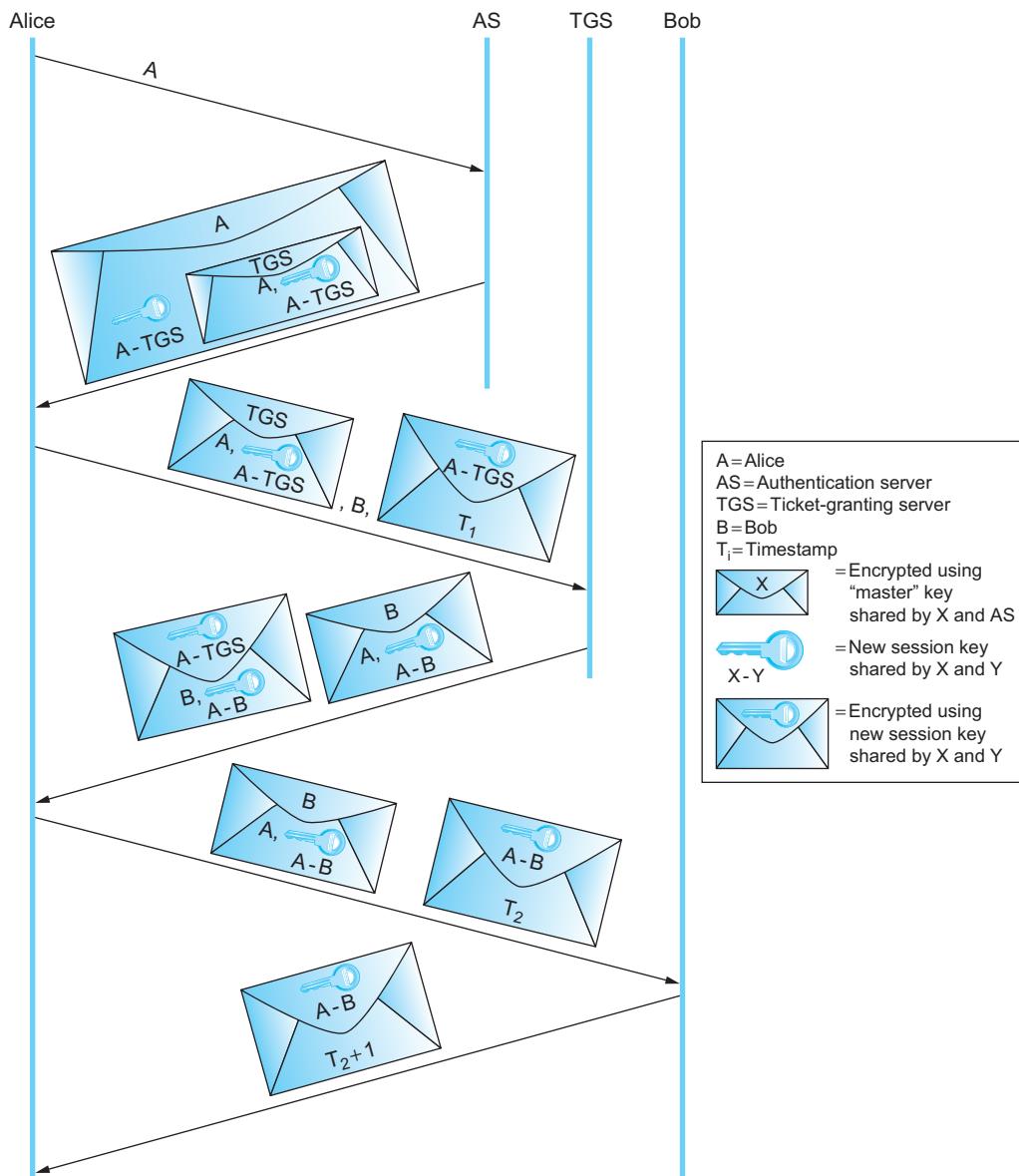
เคอร์เบอส

หัวข้อนี้จะกล่าวถึงส่วนน่าสนใจในมุมเทคโนโลยีของ เคอร์เบอส เคอร์เบอสเป็นระบบพิสูจน์ตัวจริงที่ใช้โพรโทคอล Needham-Schroeder ออกแบบสำหรับการทำงานแบบไฮโลเอนต์/เซิร์ฟเวอร์ พัฒนาขึ้นครั้งแรกที่ MIT โดยได้รับมาตรฐานจาก IETF ซอฟต์แวร์มีแบบโอเพ่นซอร์สและเชิงพาณิชย์

ไฮโลเอนต์ของเคอร์เบอส โดยทั่วไปมีผู้ใช้เป็นมนุษย์ และมีการตรวจสอบสิทธิ์ตนเองโดยใช้รหัสผ่าน ยกตัวอย่างเช่น มีแม่กุญแจของ อลิช ที่ใช้ร่วมกับ เคดีซี ซึ่งได้จากการหัสร่าน รหัสผ่านใช้ประมวลผลกุญแจรหัส ลับได้ เคอร์เบอส ถือว่าทุกคนสามารถเข้าถึงเครื่องไฮโลเอนต์ได้ทางภายนอก ดังนั้นจึงต้องปิดบังรหัสผ่านหรือ แม่กุญแจของอลิช ไม่เฉพาะในเครื่องที่เท่านั้น แต่ยังรวมถึงในเครื่องใดๆ ที่อลิชเข้าสู่ระบบด้วย ในการทำงานของ เคอร์เบอส ใช้หลักการทำงานของ Needham-Schroeder เมื่อ อลิช ต้องการใช้รหัสผ่านจะถอดรหัส การตอบกลับจาก เคดีซี ซอฟต์แวร์ผู้ใช้ไฮโลเอนต์ เคอร์เบอส จะรู้จักกว่า เคดีซี การตอบกลับ แล้วค่อยแจ้งให้ อลิช ป้อนรหัสผ่าน และคำนวนแม่กุญแจและถอดรหัสในลำดับต่อไป

รูปที่ 6.12

การทำงาน Needham-Schroeder เป็นการแลกเปลี่ยนข้อมูลของ เคดีซี กับอลิชมีการทำงานสอง ประการ: ทำให้อลิชมีวิธีการที่จะพิสูจน์ตัวจริง (มีเพียงอลิชที่สามารถถอดรหัสแพ็กเก็ตได้) และให้บริบรอง



รูปที่ 6.12: ระบบพิสูจน์ตัวจริงแบบเครือร์เบอส
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

กุญแจรหัสลับหรือ “ตั๋ว(ticket)” เพื่อส่งให้บ๊อบ —session key และข้อมูลระบุตัวจริงของอลิช ที่เข้ารหัสด้วยแม่กุญแจของบ๊อบ เครือร์เบอสมีการทำงานทั้งสองฝั่งกัน — และเดดลิซ ตั้งการทำงานในรูปที่ 6.12 มีเชิร์ฟเวอร์เรียกว่า Authentication Server (AS) หน้าที่เป็น เคดีซี ขั้นตอนแรกอลิชได้บันทึกข้อมูลไว้บน AS เพื่อใช้พิสูจน์ตัวจริง สำหรับเชิร์ฟเวอร์ที่สองเรียกว่า Ticket Granting Server (TGS) หน้าที่ตอบกลับอลิช ด้วยตั๋วของอลิชเพื่อส่งต่อให้บ๊อบได้ หากอลิชจำเป็นต้องสื่อสารกับเชิร์ฟเวอร์หลายที่ อลิชสามารถรับตั๋วจาก TGS จากแต่ละเชิร์ฟเวอร์ได้โดยไม่ต้องไปขอ กับ AS

6.5 ตัวอย่างการใช้งานซอฟต์แวร์ด้านความปลอดภัย

อ่านนาได้ก่อตัวถึงระบบที่จำเป็นในการสร้างความมั่นคง ปลอดภัยซึ่งได้ก่อตัวถึงระบบที่ครอบคลุมไม่เพียงแค่การเข้ารหัสข้อมูลด้วยเทคโนโลยีรักษาความปลอดภัย แต่ได้ก่อตัวถึงองค์ประกอบที่ทำให้เกิดความมั่นคงปลอดภัยด้านอื่น ด้วยในหัวข้อนี้จะกล่าวถึงการประยุกต์ใช้ซอฟต์แวร์ที่มีการนำองค์ประกอบด้านความมั่นคงปลอดภัยมาใช้กับระบบจริง

ซอฟต์แวร์ที่จะกล่าวถึงนี้ได้แก่ Pretty Good Privacy (PGP) ซึ่งให้การรักษาความปลอดภัยอีเมล และ Secure Shell (SSH) ใช้สำหรับควบคุมเครื่องแม่ข่ายจากระยะไกล มีการออกแบบให้เข้ารหัสข้อมูลในชั้น Transport Layer Security (TLS) โดย IETF ทดสอบ Secure Socket Layer (SSL) เป็นโพรโทคอลที่เก่ากว่า ต่อไปกล่าวถึง โพรโทคอล IPsec (IP Security) ทำงานที่สร้างบันทึกที่มีความมั่นคงปลอดภัยสูงเป็นการเชื่อมต่อแบบ end-to-end และสุดท้ายกล่าวถึงมาตรฐานความปลอดภัยสำหรับแลนไร้สาย 802.11i ให้การรักษาความปลอดภัยที่ชั้นลิ้งค์ของเครือข่ายไร้สาย

ผู้อ่านอาจจะสงสัยว่าแล้วทำไม จึงต้องมีการสร้างความมั่นคงปลอดภัยในหลายเลเยอร์ เป็นไปได้หรือไม่ที่จะนำความมั่นคงปลอดภัยไปใช้ในเลเยอร์เดียวแล้วสร้างความปลอดภัยได้ เหตุผลหนึ่งก็คือมีการคุกคามเข้ามายังรูปแบบแตกต่างกัน และมักจะเป็นคุณลักษณะเฉพาะของแต่ละเลเยอร์ ตัวอย่างเช่น หากใช้งานเน็ตบุ๊ก ในห้องบรรพสินค้าอาจจะกังวลว่าคนข้างข้างดักฟังข้อมูลของตนหรือไม่ สิ่งที่จะสร้างความปลอดภัยคือการเข้ารหัสข้อมูลที่ส่งออกจากโน้ตบุ๊กไปยังแก๊สพอยต์ แต่ถ้าหากใช้อินเทอร์เน็ตแบงค์กิ้งอาจจะกังวลว่าจะมีผู้ใช้งานที่อยู่บนอินเทอร์เน็ตสามารถเข้าถึงเราเตอร์และดักฟังข้อมูลได้ สิ่งนี้ทำให้จำเป็นจะต้องเข้ารหัสข้อมูลในชั้นขนส่ง ดังนั้นการสร้างความมั่นคงปลอดภัยจะขึ้นอยู่ กับความเหมาะสมของข้อมูลและความคุ้มค่า

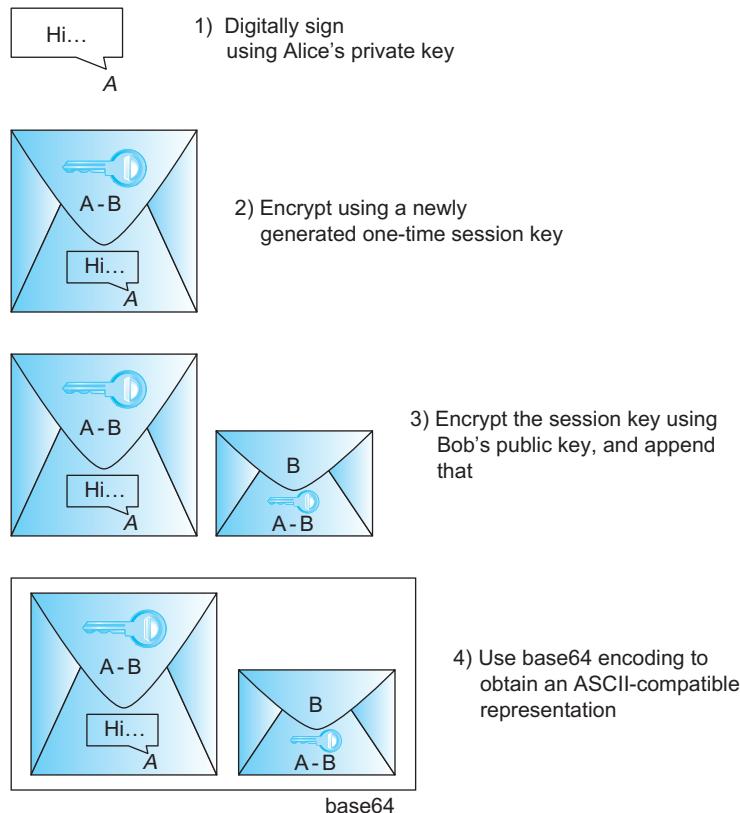
6.5.1 Pretty Good Privacy(PGP)

PGP (Pretty Good Privacy) เป็นแนวทางที่ใช้กันอย่างแพร่หลายในการรักษาความปลอดภัยให้กับอีเมล โดยการรับรอง การรักษาความลับ(confidentiality)ได้ ความถูกต้องสมบูรณ์ของข้อมูล และการทำให้ไม่สามารถการปฏิเสธ(nonrepudiation)ได้ PGP คิดขึ้นโดย Phil Zimmerman และพัฒนาเป็นมาตรฐาน IETF ที่รู้จักกันในชื่อ OpenPGP ดังที่กล่าวในหัวข้อก่อนหน้านี้ PGP เด่นในด้านการใช้โมเดล "web of trust" เพื่อแจกจ่ายกุญแจรหัสลับแทนที่จะเป็นการทำงานแบบตั้นไม้

การรักษาความลับและการตรวจสอบผู้รับของ PGP ทำได้โดยผู้รับได้รับข้อมูลที่มีกุญแจสาธารณะของผู้ส่ง ทำให้สามารถตรวจสอบตัวจริงของผู้ส่งและตรวจสอบความสมบูรณ์ของข้อมูลได้พร้อมกัน สำหรับกุญแจสาธารณะจะมีการจัดเตรียมไว้บนเครื่องแม่ข่ายล่วงหน้าแล้ว โดยใช้บริบูรณ์ดิจิทัล และ พีเคไอ ของเร็บที่เขื่อถือได้ PGP รองรับการทำงานแบบ RSA และ DSS โดยใบอนุญาตจะระบุว่าใช้อัลกอริทึมใด ในการเข้ารหัส

พิจารณาตัวอย่างต่อไปนี้ เพื่อทำความเข้าใจขั้นตอนการตรวจสอบผู้ส่งและการรักษาความลับ สมมติว่าอุปกรณ์มีข้อมูลที่จะส่งถึงบีบอ卜 และพลิกเซชัน PGP ที่เครื่องอุปกรณ์ทำขั้นตอนที่แสดงในรูปที่ 6.13 ขั้นแรก ลงนามข้อมูลโดยใช้ลายเซ็นดิจิทัลของอุปกรณ์ พลิกกับทำแฮชข้อมูล โดยใช้ MD5, SHA-1 หรือ SHA-2

Hi... = The plaintext message



รูปที่ 6.13; การใช้ลายเซ็นดิจิทัล ด้วยวิธี PGP
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

แอปพลิเคชัน PGP สร้าง session key สำหรับใช้ส่งข้อความโดยเข้ารหัสด้วย AES หรือ 3DES ข้อความที่ลงนามไปแล้วนั้นจะเข้ารหัสด้วย session key จากนั้นตัว session key เองถูกเข้ารหัสโดยใช้ กุญแจสาธารณะของบออบอีกครั้ง ผนวกเข้ากับข้อความแล้ว แต่เนื่องจากต้องส่งข้อความอีเมลมีรูปแบบ ASCII (หรือ Unicode) จึงแปลงข้อความเป็นรหัส base64 แล้วจึงส่งออก

เมื่อบอําได้รับข้อมูล จะทำการถอดรหัสโดยใช้กุญแจสาธารณะของอลิ๊ฟในการถอดรหัส ตัวข้อมูลนั้นเทียบกับแฮชที่แนบมากับข้อมูลนั้น

6.5.2 Secure Shell(SSH)

โพรโทคอล SSH(secure shell) ออกแบบเพื่อให้สามารถควบคุมเครื่องคอมพิวเตอร์ได้จากระยะไกลซึ่งนำมาทดแทนตัวซอฟต์แวร์ telnet ซึ่งมีการใช้งานมายาวนาน telnet เป็นซอฟต์แวร์ที่ไม่เข้ารหัสข้อมูลขณะสื่อสารทำให้เกิดความไม่ปลอดภัยหากมีการตักฟังข้อมูลตามเส้นทางที่มีการเชื่อมต่อ SSH และมีการใช้งานแพร่หลายสำหรับใช้ควบคุม โคลอนต์/เซิร์ฟเวอร์ โดย SSH มีความสามารถสร้างเส้นทางที่ปลอดภัยและสามารถพิสูจน์ตัวจริงผู้ใช้งานได้

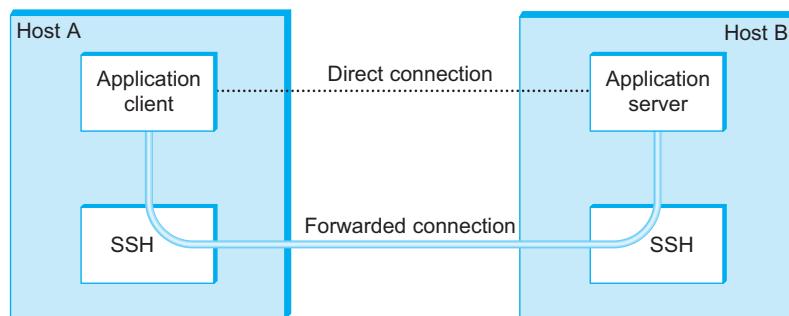
ระบบเครื่องแม่ข่ายที่ให้บริการจากเครือข่ายอินเทอร์เน็ต ถูกติดตั้งใน data center เพื่อให้เป็นศูนย์กลางการติดต่อสื่อสารความเร็วสูง เมื่อผู้ดูแลระบบต้องการควบคุม จะใช้โปรแกรมประเภท remote terminal สำหรับควบคุมเครื่องแม่ข่ายจากระยะไกล เช่น จากสำนักงาน โปรแกรม telnet ถูกใช้ควบคุมเครื่องแม่ข่ายมาเป็นระยะเวลานาน ซึ่งข้อมูลที่สื่อสารผ่าน telnet เป็นข้อมูลไม่เข้ารหัส เปรียบได้กับการร่วงหละข้อมูลภายใน data center ของมาสู่ภายนอก SSH จึงพัฒนาขึ้นมาเพื่อทดแทนการทำงาน telnet และกลายเป็นซอฟต์แวร์ที่ได้รับความนิยมในปัจจุบัน

โปรโตคอล SSH รุ่นล่าสุดเป็น version 2 ประกอบด้วยสามprotoคือ ดังนี้

- SSH-TRANS, โปรโตคอลสำหรับการส่งข้อมูล
- SSH-AUTH โปรโตคอลพิสูจน์ตัวจริง
- SSH-CONN โปรโตคอลสำหรับเชื่อมต่อ

สองอันดับแรกทำหน้าที่เป็นลีกอินระยะใกล้ สำหรับ SSH-CONN จะกล่าวถึงในท้ายบท

SSH-TRANS ทำหน้าที่เข้ารหัสข้อมูลตลอดเส้นทางจากต้นทางถึงปลายทาง โดยโปรโตคอลรันผ่าน TCP ทุกครั้งที่มีการใช้งาน SSH จะสร้าง SSH-TRANS เป็นห่อสื่อสารที่ปลอดภัยระหว่างคู่สื่อสาร เทคโนโลยีรหัสลับที่ใช้ในการพิสูจน์ตัวจริงคือ RSA เมื่อผ่านการพิสูจน์ตัวจริงแล้ว คู่สื่อสารจะเริ่มต้นแลกเปลี่ยน session key เพื่อใช้เป็นกุญแจในการเข้ารหัสข้อมูลที่จะสื่อสารกัน session key นี้จะเป็น กุญแจรหัสลับ ที่เหมือนกันทั้งสองฝั่ง สำหรับการเข้ารหัสข้อมูลที่ใช้ในการสื่อสารเพื่อให้ทำได้เร็วกว่า RSA จะใช้เทคโนโลยีรหัสลับแบบ secret-key ซึ่งโปรโตคอล SSH ใช้เทคโนโลยี AES มีรูปแบบการทำงานเป็นตามรูปที่ 6.14



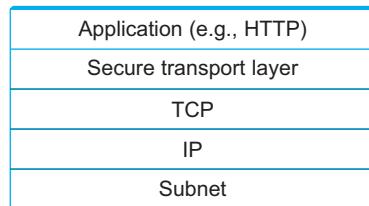
รูปที่ 6.14: ใช้ SSH สร้างทันเนตที่ปลอดภัยในการสื่อสาร
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

6.5.3 Transport Layer Security (TLS, SSL, HTTPS)

ในการทำความเข้าใจเป้าหมายของการออกแบบมาตรฐาน TLS(Transport Layer Security) และ SSL(Secure Socket Layer) ซึ่งมีพื้นฐานจาก TLS จะให้พิจารณาปัญหาหลักที่โปรโตคอลพัฒนาขึ้นเพื่อแก้ปัญหานั้น เริ่มต้นจากเทคโนโลยี WWW(World Wide Web) ได้รับความนิยมแพร่หลายทั่วโลก เช่น สำนักบุคคลและใช้ภายในองค์กร เทคโนโลยีเว็บก็คือเว็บไซต์ที่ให้บริการผ่านระบบอินเทอร์เน็ตในยุคเริ่มต้นของเว็บไซต์นั้นเครื่องเซิร์ฟเวอร์จะไม่มีการเข้ารหัสข้อมูลทำให้เกิดความเสี่ยงที่มีผู้ประสงค์ร้ายดักฟังข้อมูลที่ใช้สื่อสารกันระหว่างคู่สื่อสารคือเว็บ

เบราว์เซอร์(browser)และเบ็บเซิร์ฟเวอร์จากปัญหานี้ทำให้มีความต้องการพัฒนาระบบที่สามารถเข้ารหัสข้อมูลจากเว็บเบราว์เซอร์ไปจนถึงเบ็บเซิร์ฟเวอร์ กลยุทธ์ที่มาของการพัฒนามาตรฐาน TLS

นักออกแบบ SSL และ TLS มองวิธีแก้ปัญหาความไม่ปลอดภัยของเว็บแบบไม่ต้องแก้ไขมาตรฐานเว็บแต่เลือกวิธีการสร้าง เลเยอร์ ให้เข้ารหัสส่วนเพย์โหลดของเว็บ(web)อีกชั้นเรียกว่า “secure transport layer” ตามรูปที่ 6.15



รูปที่ 6.15: โพร์โทคอล secure transport layer เพิ่มตระกล่างระหว่างชั้นแอปพลิเคชันกับ TCP
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

เมื่อใช้งาน HTTP(HyperText Transport Protocol) ติดต่อผ่านเว็บตามปกติจะไม่มีการเข้ารหัส แต่เมื่อติดต่อด้วยช่องทาง secure transport layer จะเข้าขั้นตอนเข้ารหัสข้อมูล ในการเชื่อมต่อนั้นจะทราบว่า เป็น secure transport layer จากการระบุหมายเลขพอร์ตที่เชื่อมต่อ ซึ่งพอร์ตมาตรฐานคือ 443 เรียกโพร์โทคอลนี้ว่า HTTPS(Secure HTTP)

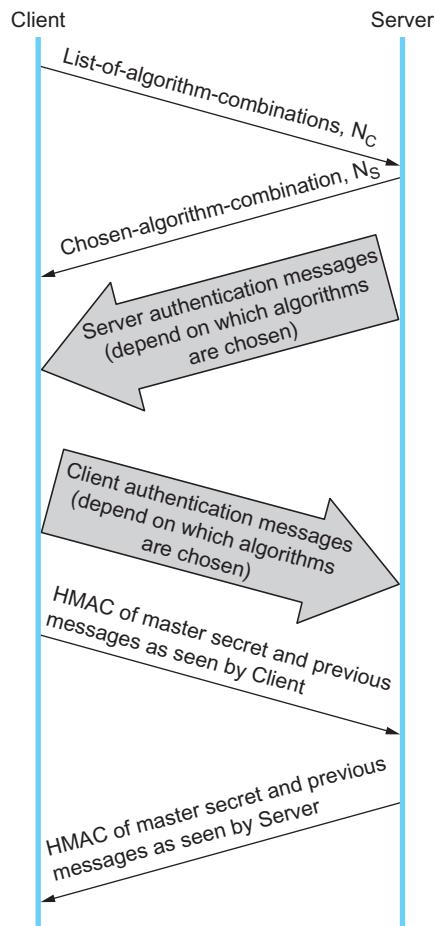
รูปที่ 6.16 แสดงโพร์โทคอล Handshake ขั้นตอนแรกไคลเอนต์ส่งข้อมูลบอกรายละเอียดอัลกอริทึม การเข้ารหัสที่ตนรองรับ ต่อมาก็รับตอบกลับโดยเลือกอัลกอริทึมจากรายการที่ไคลเอนต์ส่ง พร้อมกับ ส่ง nonce ของไคลเอนต์ และ nonce ของเซิร์ฟเวอร์ ตามลำดับ แล้วจึงได้ session key สำหรับใช้เข้ารหัส ข้อมูลที่เป็นความลับ

6.5.4 Wireless Security (IEEE802.11i)

การสื่อสารผ่านเครือข่ายไร้สายมีภัยคุกคามด้านความมั่นคงปลอดภัยจากการแพร่สัญญาณรอบทิศทางและ สามารถทะลุสิ่งกีดขวางได้ ขณะที่การออกแบบแลนไร้สายมีเป้าหมายเพื่อให้สามารถใช้งานได้ง่าย ทำให้ต้อง ผ่อนปรนการรักษาความปลอดภัย ตัวอย่างเช่น พนักงานสามารถเชื่อมต่อเครือข่ายภายในองค์กรได้ทุกที่ที่ โดยไม่ต้องมองหาจุดเชื่อมต่อสายแลน เพราะมีการส่งสัญญาณจาก ออกเซสพอยต์ ทั่วบริเวณสำนักงาน ปัญหา ด้านความปลอดภัยจึงตามมา จากการควบคุมพื้นที่ให้บริการได้ยาก ทำให้สัญญาณแพร่ออกนอกพื้นที่ได้ ซึ่ง เสมือนกับมีจุดเชื่อมต่อสายแลนอยู่ในสำนักงาน

ผลกระทบที่ตามมาทำให้ต้องเข้ารหัสข้อมูลที่ส่งทางไร้สาย มาตรฐานการเข้ารหัสในช่วงแรกได้เสนอ วิธีอย่างง่าย ซึ่งว่า WEP(Wired Equivalent Privacy) และถูกยกเลิกหลังนักวิจัย Borisov และคณะ (2001) พบปัญหาด้านความปลอดภัยร้ายแรง

มาตรฐาน IEEE 802.11i จึงพัฒนาขึ้นโดยมุ่งศึกษาการสร้างความมั่นคงปลอดภัยในการสื่อสารให้แก่ แลนไร้สาย และได้พัฒนาถึง WPA3 (Wi-Fi Protected Access 3) ซึ่งเป็นระบบปลอดภัยล่าสุดของเทคโนโลยี แลนไร้สาย เพื่อให้ระบบเข้ารหัสยังคงใช้งานร่วมกับมาตรฐานเดิมได้ ทำให้ระบบยังคงรองรับการทำงานแบบ



รูปที่ 6.16: กระบวนการ Handshake ของ TLS
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

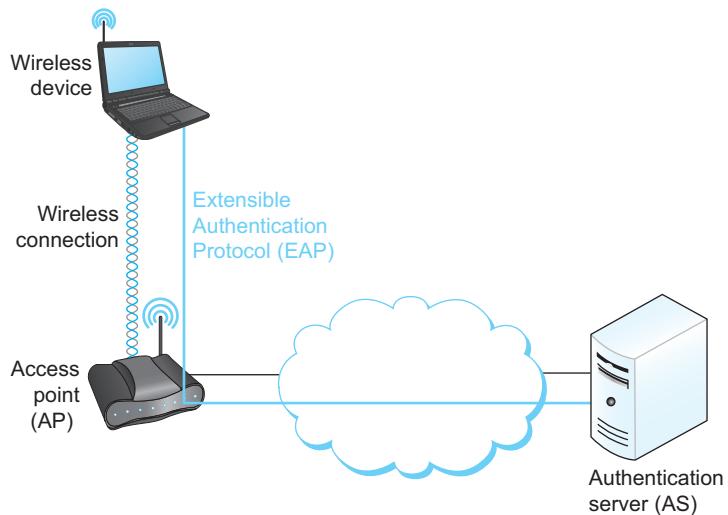
WEP โดยการปรับให้ปลอดภัยขึ้นจากการเปลี่ยนเป็น Pre-Shared Key (PSK) โดยภายในใช้ cipher RC4 เช่นเดิม

สำหรับมาตรฐาน 802.11i สำหรับใช้ในสำนักงาน ได้ออกแบบให้มีระบบ back-end ที่สามารถควบคุมผู้ใช้งานให้ใช้กุญแจรหัสลับแตกต่างกันผ่านการส่งกุญแจด้วยระบบ Authentication Server (AS) ดังแสดงในรูปที่ 6.17 เครื่องโนํตบุ๊กจะเชื่อมต่อเครือข่ายภายในองค์กรโดยผ่านทางไผ่ซึ่งออกเซสพอยต์นี้จะตรวจสอบผู้ใช้งานผ่านทางฐานข้อมูลที่ติดตั้งใน AS

802.11i ออกแบบให้ยืดหยุ่นต่อการทำงาน โดยสามารถเลือกโพรโทคอลมาใช้ในการพิสูจน์ตัวจริงได้หลากหลาย อาทิเช่น EAP-PEAP EAP-TLS EAP-TTLS EAP-SIM และ EAP-AKA เป็นต้น

6.5.5 ไฟล์วอลล์

กล่าวมาทั้งหมดเป็นการใช้เทคโนโลยีเพื่อรักษาความมั่นคงปลอดภัยสองประการได้แก่ การรักษาความลับ และความถูกต้องสมบูรณ์ ซึ่งประเด็นของความมั่นคงปลอดภัยยังหรือประการสำคัญที่น่าสนใจจากการเข้ารหัส

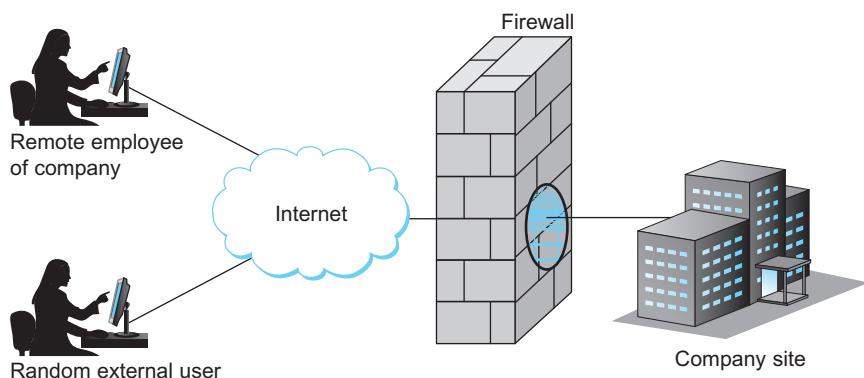


รูปที่ 6.17: ใช้ Authentication server สำหรับเครือข่ายแลนไร้สาย
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ข้อมูลคือ การทำให้ระบบพร้อมให้บริการ หรือเรียกว่า การรักษาความพร้อมให้บริการ(availability) อุปกรณ์เครือข่ายที่มีส่วนสำคัญต่อการควบคุมตรงเส้นทางเข้าออกข้อมูลได้แก่ “ไฟร์วอลล์”

ไฟร์วอลล์ทำหน้าที่ควบคุมการเข้าออกของข้อมูลชั้นไอพีและชั้นบนส่ง สามารถควบคุมโดยอ่านต้นทางและปลายทางโดยอ่านข้อมูลจากแพ็กเก็ตเดอร์รวมถึงการอ่าน พาวร์ตต้นทางและพาวร์ปลายทาง ซึ่งข้อมูลทั้งสี่ประการนี้เป็นข้อมูลที่พบใน แพ็กเก็ตเดอร์ และทำให้ไฟร์วอลล์สามารถควบคุมการเดินทางเข้าออกของข้อมูลได้

จากรูปที่ 6.18 แสดงถึงการควบคุมข้อมูลอินเทอร์เน็ตที่เดินทางเข้าสู่สำนักงานและข้อมูลเดินทางออก จากสำนักงาน จากรูปใช้สัญลักษณ์กำแพงใช้เปรียบเทียบการกันไม่ให้ข้อมูลจากอินเทอร์เน็ตเข้าถึงเครือข่ายภายในสำนักงานและเปรียบเป็นกำแพงกันไม่ให้ออกจากสำนักงานสู่อินเทอร์เน็ตได้โดยตรง กำแพงดังกล่าวมีรูปแบบการทำงานของอุปกรณ์เครือข่ายที่ทำหน้าที่อ่านและคัดกรองข้อมูล โดยสามารถควบคุมข้อมูลที่ส่งผ่านทางโปรโตคอล IP TCP และ UDP อุปกรณ์นั้นเรียกว่า ไฟร์วอลล์



รูปที่ 6.18: ไฟร์วอลล์ทำหน้าที่กรองแพ็กเก็ต
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ยกตัวอย่าง เช่น ต้องการควบคุมการสื่อสารของคู่สื่อสารที่ประกอบด้วยต้นทางมีหมายเลข IP เป็น 192.12.13.14 มีพอร์ตต้นทางเป็น 1234 และมี IP ปลายทางเป็น 128.7.6.5 ที่มี口号ปลายทางเป็น 80 กำหนดค่าให้ไฟร์วอลล์ สรุปเป็นข้อมูลสำคัญ 4-tuple ต่อไปนี้

(192.12.13.14, 1234, 128.7.6.5, 80)

ซึ่งรูปแบบนี้อธิบายได้ว่า ไม่อนุญาตให้แพ็กเก็ตที่เดินทาง ออกจากพอร์ตหมายเลข 1234 ที่ส่งจาก โฮสต์หมายเลข IP 192.12.13.14 ไปยังโฮสต์ มี口号ปลายทาง 128.7.6.5 พอร์ต 80 สื่อสารได้

อีกหนึ่งตัวอย่างต่อไปนี้ เครื่องหมาย * 代表หมายถึงทุกหมายเลข IP ต้นทาง และ * ที่สองหมายถึงทุกหมายเลขพอร์ตต้นทาง ไม่อนุญาตให้ติดต่อกับโฮสต์ อินเทอร์เน็ตโพรโทคอล 128.7.6.5 พอร์ต 80 (หรือเขียนว่า 128.7.6.5:80)

(*, *, 128.7.6.5, 80)

การใช้เครื่องหมาย * ทำให้ประหยัดบรรทัดในการกำหนด rule ไฟร์วอลล์ ได้ ในเครือข่ายใช้งานจริง จะมีการกำหนด rule จำนวนมาก หากไฟร์วอลล์ต้องตรวจสอบทุกแพ็กเก็ตกับทุก rule จะทำให้ระบบทำงานช้า จึงมีการคิดค้น stateful firewall ขึ้นโดยใช้ state ที่ได้จาก TCP เป็นตัวตรวจสอบเพียงครั้งแรก ทำให้ไฟร์วอลล์ทำงานได้เร็วขึ้น

บทที่ 7

ขั้นตอนแอปพลิเคชัน

ปัญหาของแอปพลิเคชัน: เหตุผลการมีขั้นตอนแอปพลิเคชัน

บทที่ 1 กล่าวถึงโปรแกรม ตั้งแต่เว็บเบราว์เซอร์ไปจนถึงซอฟต์แวร์สำหรับประชุมทางออนไลน์ ที่ต้องใช้ผ่านเครือข่ายคอมพิวเตอร์ และได้กล่าวถึงซอฟต์แวร์สำหรับใช้ด้านโครงสร้างพื้นฐานเครือข่ายที่ช่วยสนับสนุนให้โปรแกรมต่างๆ สื่อสารผ่านอินเทอร์เน็ตได้ สำหรับหนึ่ง จำกัดมาที่แอปพลิเคชันเครือข่ายอีกครั้ง แอปพลิเคชันที่จะกล่าวถึงนี้เป็นส่วนหนึ่งของโปรโตคอลเครือข่าย (ในแบบแลกเปลี่ยนข้อมูลกับคู่สื่อสาร) และเป็นส่วนหนึ่งของโปรแกรมแอปพลิเคชันตั้งเดิม (ในแบบที่ต้องตอบกับระบบหน้าจอ ระบบไฟล์ และผู้ใช้งาน) บทนี้มีกล่าวถึงแอปพลิเคชันเครือข่ายที่ได้รับความนิยมในปัจจุบัน

7.1 แอปพลิเคชันตั้งเดิม

ก่อนกล่าวถึงรายละเอียดแอปพลิเคชัน มีพื้นฐานสามประการที่ควรทำความเข้าใจ ประการแรกความแตกต่างระหว่าง “แอปพลิเคชันโพรโทคอล(application protocol)” และ “แอปพลิเคชันโปรแกรม(application program)” ตัวอย่างเช่น HTTP เป็นแอปพลิเคชันโพรโทคอล มีซอฟต์แวร์ที่เขียนขึ้นโดยทำงานภายใต้พื้นฐานHTTP จำนวนมาก เช่น Chrome Safari Firefox Opera Edge เป็นต้น เรียกซอฟต์แวร์นี้ว่า “แอปพลิเคชันโปรแกรม”

บทนี้จะกล่าวถึงแอปพลิเคชันโพรโทคอล ที่พบทั่วไปดังนี้

- SMTP(Simple Mail Transfer Protocol) ใช้สำหรับรับ-ส่งอีเมล
- HTTP ใช้สำหรับสื่อสารผ่านเว็บเบราว์เซอร์และเว็บเซิร์ฟเวอร์(web server)

ประการสอง สังเกตได้ว่าแอปพลิเคชันโพรโทคอลจำนวนมาก รวมทั้ง HTTP และ SMTP มีโพรโทคอลร่วมมีข้อกำหนดครุภัณฑ์รูปแบบการแลกเปลี่ยนระหว่างกันได้ ซึ่งเป็นเหตุผลที่โปรโตคอลทำงานร่วมกันได้ง่าย: ความซับซ้อนของการทำงานส่วนใหญ่ได้แก้ไขในโดยใช้การทำงานแบ่งหน้าที่กัน ตัวอย่างเช่น SMTP เป็นโปรโตคอลสำหรับการรับ-ส่งอีเมล โดยมีแต่ ([RFC822, 1982](#)) และ MIME(Multipurpose Internet Mail Extensions) สำหรับกำหนดรูปแบบของข้อมูล ในทำนองเดียวกัน HTTP เป็นโปรโตคอลสำหรับแลกเปลี่ยนข้อมูลหน้าเว็บ แต่ HTML(HyperText Markup Language) เป็นข้อกำหนดร่วมที่กำหนดรูปแบบของหน้าเว็บ

ประการสุดท้าย เนื่องจากแอปพลิเคชันโพรโทคอลที่อธิบายในหัวข้อนี้เป็นไปตามการสื่อสารแบบ request/reply ซึ่งจะเรียกใช้บริการการสื่อสาร TCP หรือ UDP ก็ได้ ขึ้นอยู่กับการทำงานของแอปพลิเคชัน เช่น แอปพลิเคชันโพรโทคอล HTTP ใช้การสื่อสาร TCP แต่ แอปพลิเคชันโพรโทคอล DNS ใช้ UDP เป็นต้น

7.1.1 จดหมายอิเล็กทรอนิกส์ (SMTP, MIME, IMAP)

อีเมลเป็นหนึ่งในแอปพลิเคชันโปรแกรมที่เก่าแก่ที่สุด (สังเกตได้จากลำดับเลข RFC) ในยุคเริ่มต้นการสื่อสารระยะไกลจะมีอะไรดีไปกว่าได้ส่งข้อความถึงเพื่อนที่อยู่ไกลอีกประเทศ แต่น่าแปลกใจที่ผู้ก่อตั้ง ARPANET (Advanced Research Projects Agency Network) ไม่ได้มองอีเมลจะเป็นแอปพลิเคชันหลัก แต่มองการเข้าถึงทรัพยากรคอมพิวเตอร์จากระยะไกลเป็นเป้าหมายการออกแบบหลักอย่างไรก็ตาม อีเมลกล้ายเป็นแอปพลิเคชันหลักแทน แอปพลิเคชันที่ผู้ก่อตั้งตั้งใจให้อินเทอร์เน็ตเป็น

สำหรับการทำความเข้าใจการทำงาน ตามที่ได้กล่าวไว้ข้างต้นได้แบ่งออกเป็นสองส่วน (1) แยกแยะส่วนต่อประสานผู้ใช้ (เช่น โปรแกรมอ่านอีเมล) จากโทรศัพท์และการแลกเปลี่ยนแฟกติกีติที่บรรจุข้อมูลใดๆ (เช่น SMTP หรือ IMAP) และ (2) แยกความแตกต่างระหว่างโทรศัพท์และการแลกเปลี่ยนข้อมูลกับมาตรฐานที่ใช้ออกจากกัน ([RFC822 \(1982\)](#) และ MIME)

Message Format

[RFC822 \(1982\)](#) กำหนดรูปแบบข้อความไว้สองส่วน: ส่วนหัวและเนื้อหา ทั้งสองส่วนจะแสดงเป็นข้อความเป็นรหัส รหัสแอลกอริทึม เช่น โปรแกรมอ่านอีเมล เป็นค่าปกติ เนื้อหาถือว่าเป็นข้อความธรรมดา ถึงแม้ว่าได้กำหนด MIME ใน [RFC822 \(1982\)](#) สำหรับระบุเนื้อหาข้อความให้สามารถส่งข้อมูลได้หลายประเภท ข้อมูลนี้ยังคงถูกแปลงเป็นข้อความ รหัสแอลกอริทึม รูปภาพJPEG ตัวข้อมูลไม่มีอยู่ในรูปแบบรหัสแอลกอริทึม แต่เมื่อส่งทางอีเมล ผ่าน MIME จะแปลงเป็นรูปแบบรหัสแอลกอริทึม วิธีแปลงไฟล์ใบนาทีที่ได้รับความนิยมได้แก่ BASE64

เมื่อร่วมทั้งหมดนี้แล้ว ประกอบด้วยข้อความธรรมดา รูปภาพ JPEG และไฟล์ PostScript จะมีดังนี้:

```
MIME-Version: 1.0
Content-Type: multipart/mixed;
boundary="-----417CA6E2DE4ABCAFBC5"
From: Alice Smith <Alice@npu.world>
To: Bob@mail.npu.ac.th
Subject: promised material
Date: Mon, 07 Sep 1998 19:45:19 -0400
```

```
-----417CA6E2DE4ABCAFBC5
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
```

Bob,

Here are the jpeg image and draft report I promised.

--Alice

```

-----417CA6E2DE4ABCAFBC5
Content-Type: image/jpeg
Content-Transfer-Encoding: base64
... unreadable encoding of a jpeg figure
-----417CA6E2DE4ABCAFBC5
Content-Type: application/postscript; name="draft.ps"
Content-Transfer-Encoding: 7bit
... readable encoding of a PostScript document

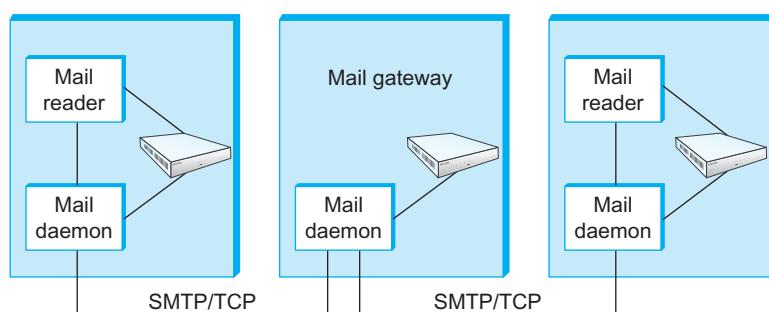
```

จากตัวอย่าง บรรทัดในส่วนหัวของข้อความระบุว่าข้อความนี้ประกอบด้วยส่วนต่างๆ ซึ่งแต่ละส่วนแสดงด้วยตัวอักษร รวมอักษรที่ไม่แสดงออกหน้าจอ รวมกันภายใต้ Content-Type และ Content-Transfer-Encoding

Message Transfer

โทรศัพท์เคลื่อนที่ คอมพิวเตอร์ พัฒนาหลังจากเกิดระบบอินเทอร์เน็ตไม่นาน ก่อนหน้านี้ข้อความอีเมลส่วนใหญ่ส่งจากโฮสต์ไปไว้ในบนเครื่องแม่ข่ายผ่านโทรศัพท์เคลื่อนที่ SMTP เท่านั้น แม้ว่า SMTP จะยังคงมีเป็นโทรศัพท์เคลื่อนที่สำหรับแลกเปลี่ยนอีเมล แต่ตอนนี้ไม่ได้เป็นโทรศัพท์เคลื่อนที่ที่ทำหน้าที่เกี่ยวข้องกับอีเมล โดยอีเมลมีโทรศัพท์เคลื่อนที่สำหรับอีเมล จำนวนหนึ่งสำหรับแบ่งหน้าที่ออกเป็นงานย่อย ยกตัวอย่างเช่น Internet Message Access Protocol (IMAP) และ Post Office Protocol (POP) เป็นโทรศัพท์เคลื่อนที่สำหรับการดึงข้อความอีเมล จะกล่าวถึงโทรศัพท์เคลื่อนที่ IMAP และ SMTP ในลำดับต่อไป

รูปที่ 7.1 อธิบายลำดับการส่งอีเมลนับจากผู้ใช้เปิดแอปพลิเคชันโปรแกรม Mail read สำหรับใช้รับส่งอีเมล เมื่อผู้ใช้ต้องการส่งอีเมลได้เขียนข้อความผ่านทางโปรแกรม Mail reader หลังจากนั้นโปรแกรมจะนำข้อความนั้นส่งผ่าน Mail daemon เพื่อทำหน้าที่ติดต่อผู้รับผ่านโทรศัพท์เคลื่อนที่ SMTP ซึ่งใช้วิธีส่งข้อมูลแบบ TCP ไปยัง Mail gateway และ Mail gateway ทำหน้าที่ส่งต่อไปยัง Mail daemon เครื่องปลายทางผ่านทางโทรศัพท์เคลื่อนที่ SMTP เมื่อ Mail daemon ได้รับข้อมูลจะส่งข้อมูลต่อไปยัง Mail reader เครื่องปลายทาง



รูปที่ 7.1: ลำดับการส่งอีเมลใช้หลักการ store-and-forward
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

Mail daemon เป็นโปรแกรมทำงานอยู่บนแต่ละไฮสต์ มีกันพื้นที่สำหรับเก็บอีเมล์ (หรือเรียกว่า Mailbox) โดย Mail daemon เรียกอีกอย่างว่าตัวแทนถ่ายโอนข้อมูล (MTA) เนื่องจาก SMTP เป็นโปรแกรมโดยรวมที่คร่า ก็นำไปใช้ได้ในทางทฤษฎีแล้ว Mail daemon อาจมีการใช้งานที่แตกต่างกันมากมาย แต่มีความใช้งานที่ได้รับความนิยมเพียงไม่กี่รายการ โดยมี sendmail โปรแกรมนิยมในอดีต และ postfix เป็นที่แพร่หลายมากที่สุด

แม้ว่า MTA บนเครื่องผู้ส่งสามารถสร้างการเชื่อมต่อแบบ SMTP/TCP กับ MTA บนแมล์เซอร์ฟเวอร์ (mail server) ของผู้รับได้โดยตรง แต่ในหลายกรณี MTA ไม่ได้เป็นคู่ดิตต่อโดยตรงแต่เป็นผู้รับแล้วส่งต่อจนกว่าจะถึงแมล์เซอร์ฟเวอร์ปลายทาง หน้าที่ของ MTA ที่เป็นการส่งต่อเนี้ยก็คือ “แมล์เกตเวย์ (Mail gateway)” ทำหน้าที่เหมือนเราเตอร์ที่ค่อยค้นหาเส้นทางเหมาะสมจนถึงปลายทาง จากรูปที่ [7.1](#) แมล์เกตเวย์ได้แก่กล่องตรงกลาง

สาเหตุที่ต้องมีแมล์เกตเวย์ (mail gateway) คือ มีสถานการณ์ที่ผู้รับไม่ได้อยู่ในเครือข่ายที่สามารถติดต่อ แมล์เกตเวย์ได้ เช่นการเดินทางไปต่างประเทศแล้วต้องการใช้อีเมล์สำนักงานในการสื่อสาร ดังนั้นพอไปติดต่อ SMTP อนุญาตให้ผู้ส่งอีเมล์สามารถฝ่ากังวลกับ แมล์เกตเวย์ ได้ ประการสอง มีการแบ่งหน่วยงานย่อยภายในองค์กร ไม่ต้องการใช้กล่องรับอีเมล์(e-mail box)ร่วมกัน ตัวอย่างเช่น อีเมล์ส่งถึง bob@cpe.npu.ac.th จะถูกส่งไปยังแมล์เกตเวย์ในสาขาวิชาชีวกรรมคอมพิวเตอร์ (CPE) ที่มหาวิทยาลัยนครพนมก่อน (นั่นคือไปยังไฮสต์ชื่อ cpe.npu.ac.th) แล้วส่งต่อ ซึ่งเป็นการเชื่อมต่อครั้งที่สอง ไปยังเครื่องที่มีกล่องรับอีเมล์ของ Bob โดย เกตเวย์ที่ทำหน้าที่ส่งต่อไม่จำเป็นต้องมีกล่องรับอีเมล์ของ Bob

```

HELO npu.ac.th
250 Hello daemon@mail.npu.ac.th [216.58.196.19]

MAIL FROM:<Alice@npu.ac.th>
250 OK

RCPT TO:<Bob@cpe.npu.ac.th>
250 OK

RCPT TO:<Tom@npu.world>
550 No such user here

DATA
354 Start mail input; end with <CRLF>.<CRLF>
Blah blah blah...
...etc. etc. etc.
<CRLF>.<CRLF>
250 OK

```

QUIT

221 Closing connection

ตัวอย่างด้านบนเป็นข้อมูลที่เห็นใน SMTP เชิร์ฟเวอร์ มีลำดับการแลกเปลี่ยนระหว่างโคลเอนต์และเชิร์ฟเวอร์ ในการแลกเปลี่ยนแต่ละครั้ง โคลเอนต์จะโพสต์คำสั่ง (เช่น QUIT) และเชิร์ฟเวอร์ตอบสนองด้วยรหัส (เช่น 250, 550, 354, 221) เชิร์ฟเวอร์ยังส่งคืนคำอธิบายที่มุนุชย์สามารถอ่านได้ เช่น โคลเอนต์ส่งคำสั่งระบุตัวเองไปยังเชิร์ฟเวอร์ด้วยคำสั่ง HELO ตามด้วยชื่อโดเมนเป็นอาร์กิวเม้นต์ เมื่อเชิร์ฟเวอร์ตรวจสอบว่าชื่อนี้ตรงกับ IP ที่ข้อติดต่อ เชิร์ฟเวอร์จะตอบกลับโดยระบุ IP กลับไปยังโคลเอนต์ จากนั้นโคลเอนต์จะถามเชิร์ฟเวอร์ว่ามีอะไรรับ อีเมล์สำหรับใช้ส่งสองคนหรือไม่ เชิร์ฟเวอร์ตอบกลับด้วยข้อความ "250 OK" หมายถึงยอมรับ กับอีเมล์ "550 No such user here" หมายถึงไม่รู้จักผู้รับ จากนั้นผู้ใช้จะส่งข้อความซึ่งสิ้นสุดด้วยบรรทัดที่มีจุดเดียว (".") เพื่อเป็นการหยุดการสนทนากับเชิร์ฟเวอร์

มีคำสั่งและรหัสส่งคืนอื่น ๆ อีกมากmany ตัวอย่างเช่น เชิร์ฟเวอร์สามารถตอบสนองต่อคำสั่ง RCPT ด้วย 251 ซึ่งบ่งบอกว่าผู้ใช้มีอีเมล์ของคนนี้แล้ว เชิร์ฟเวอร์จะส่งต่ออีเมล์ไปยัง Mail daemon อื่นให้ กล่าวอีกนัยหนึ่ง โคลเอนต์ที่ทำงานเป็นเมล์เกตเวย์ สามารถเป็นกลางข่ายได้และสามารถตรวจสอบคำสั่ง VRFY เพื่อยืนยันที่อยู่อีเมล์ของผู้ใช้ โดยไม่ต้องส่งข้อความถึงผู้ใช้จริงๆ (เพราะฝากรส่ง)

Mail Reader

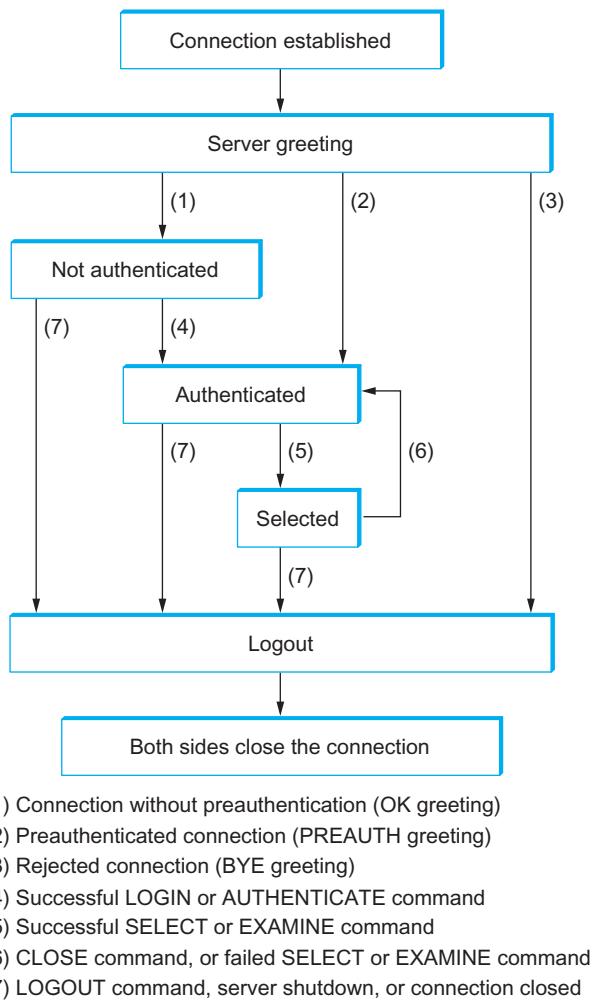
ขั้นตอนสุดท้ายคือให้ผู้ใช้อ่านอีเมล์จากกล่องรับอีเมล์ เดิมที่โปรแกรมอ่านอีเมล์เป็นเพียงโปรแกรมที่ทำงานบนเครื่องเชิร์ฟเวอร์เดียวกับกล่องรับอีเมล์ ซึ่งผู้ใช้สามารถอ่านและเขียนไฟล์ตามต้องการบันเกล่องรับอีเมล์ได้ วิธีนี้ เป็นกรณีที่ร่วมกันในอดีต ทุกวันนี้ผู้ใช้ส่วนใหญ่เข้าถึงกล่องจดหมายของตนจากเครื่องระยะไกลโดยใช้proto協定 ที่ชื่อ POP(Post Office Protocol) หรือ IMAP(Internet Message Access Protocol) ปัจจุบันนิยมใช้ IMAP แพร่หลายกว่า POP

รูปที่ 7.2 อธิบายลำดับการทำงานภายใต้ proto協定 IMAP โดยอธิบายเป็นสถานะของอีเมล์เมื่อเข้าสู่ ขั้นตอนการทำงาน IMAP

เมื่อมีอีเมล์เข้าสู่ระบบ จะตรวจสอบตัวจริงผู้ใช้ (1) หมายถึงสถานะผู้ใช้ยังไม่ผ่านการพิสูจน์ตัวจริง (2) หมายถึงสถานะผู้ใช้เข้าสู่ขั้นตอนผ่านการพิสูจน์ตัวจริง (3) อยู่สถานะปิดการเชื่อมต่อ เมื่อผู้ใช้อยู่ในสถานะ (1) จะต้องเข้าขั้นตอนพิสูจน์ตัวจริง เมื่อผ่านการพิสูจน์ตัวจริงจะสามารถไปสถานะ (5) และ (7) ได้ โดยที่ (7) คือ ดำเนินการเสร็จแล้วและยกเลิกการเชื่อมต่อ ขณะที่ (5) จะดำเนินการคำสั่งด้านอีเมล์ และหากการดำเนิน การนั้นต้องการการพิสูจน์ตัวจริงอีกครั้งจะกลับไปพิสูจน์ตัวจริงในสถานะ (6)

7.1.2 World Wide Web (HTTP)

proto協定 WWW(World Wide Web) ประสบความสำเร็จอย่างมากและทำให้ผู้คนจำนวนมากสามารถเข้าถึงข้อมูลผ่านเครือข่ายอินเทอร์เน็ตอินเทอร์เน็ตได้โดยง่าย ซึ่งเหมือนว่า WWW เป็นสิ่งเดียวกับกับอินเทอร์เน็ต การออกแบบระบบที่กล้ายมาเป็นเว็บนั้นเริ่มเมื่อปี 1989 หลังจากที่อินเทอร์เน็ตกล้ายเป็นระบบที่มีการใช้งาน



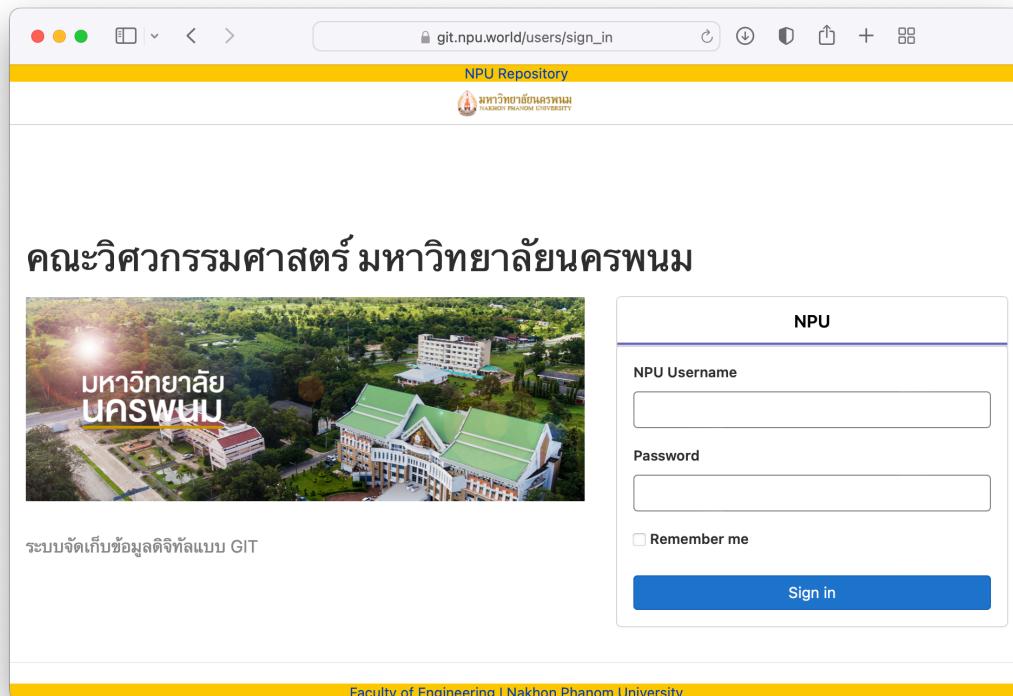
รูปที่ 7.2: การเปลี่ยนสถานะของโปรโตคอล IMAP
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

อย่างก้าวข้างหน้า วัตถุประสงค์ตั้งต้นของการพัฒนาเว็บขึ้นมาเพื่อต้องการหาวิธีจัดระเบียบและดึงข้อมูล โดยใช้แนวคิดด้าน HyperText (ไฮเปอร์เทกซ์)¹ ซึ่งแนวคิดนี้มีมานานตั้งแต่ช่วงทศวรรษ 1960² แนวคิดหลักของไฮเปอร์เทกซ์คือเอกสารสามารถเชื่อมโยงไปยังเอกสารอื่นได้ ทำให้เกิดการพัฒนาโปรโตคอล HTTP และภาษาเอกสาร HTML ได้รับการออกแบบมาเพื่อทำให้เป้าหมายบรรลุตามวัตถุประสงค์

สิ่งหนึ่งที่ทำให้เว็บได้รับความนิยมคือ การทำงานแบบคลิกเอนต์-เชิร์ฟเวอร์ที่เป็นมาตรฐานและได้รับการพัฒนาซอฟต์แวร์ภายใต้ข้อกำหนดอย่างเคร่งครัด ซึ่งทั้งหมดใช้ภาษาเดียวได้แก่ HTTP ผู้ใช้เปิดเว็บผ่านโปรแกรมคลิกเอนต์ที่สามารถแสดงผลแบบกราฟิก เรียกว่า “เว็บเบราว์เซอร์” ตัวอย่างซอฟต์แวร์ เช่น Safari, Chrome, Firefox หรือ Internet Explorer รูปที่ 7.3 แสดงเบราว์เซอร์ Safari ที่ใช้งานอยู่ แสดงหน้าข้อมูลเว็บ git ของสาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยนครพนม

¹เอกสารที่เชื่อมโยงระหว่างกัน

²World Wide Web consortium



รูปที่ 7.3: เว็บเบราว์เซอร์ Safari

เห็นได้ ถ้าต้องการจัดระบบเอกสารหรือเชื่อมโยงเอกสารกัน จะต้องมีเอกสารหนึ่งเป็นไฟล์เริ่มต้น ดังนั้น เว็บเบราว์เซอร์ใดๆ จะมีฟังก์ชันที่อนุญาตให้ผู้ใช้รับป้อนที่อยู่เอกสารโดยป้อนผ่านช่อง ยูอาร์แอล (Uniform Resource Locators) ซึ่งใช้ทั่วไปคุณเคยติดใจบ้อนที่อยู่ของเอกสารดังนี้

<https://git.npu.world>

หากป้อน URL บนเว็บเบราว์เซอร์ โปรแกรมจะเริ่มสร้างการเชื่อมต่อให้ไว้ในส่วน TCP ไปยังเว็บไซต์ที่เครื่อง git.npu.world และค้นหาและแสดงไฟล์ชื่อ index.html โดยไฟล์ส่วนใหญ่บนเว็บประกอบด้วยรูปภาพและข้อความ และหลายไฟล์มีวัตถุอื่นด้วย เช่น คลิปเสียงและวิดีโอ รวมถึงโค้ดบางส่วน นักจักนี้ยังมี ยูอาร์แอล ที่ชี้ไปยังไฟล์อื่นๆ หรือชี้ไปเครื่องอื่น ซึ่งนี่เป็นหลักการทำงานของ “ไฮเปอร์เทกซ์” ที่ใช้ใน โปรแกรม kol HTTP และ HTML ทำให้การย้ายไปเอกสารอื่นทำได้ง่ายเพียงเลือกลิงค์ที่เชื่อมไปยังเอกสารนั้น

เว็บเบราว์เซอร์ทำหน้าที่เป็นโคลเลกเตอร์ จะดึงหน้าเริบจากเซิร์ฟเวอร์ผ่านทางพอร์ต HTTP ซึ่งสื่อสารทาง TCP มีหลักการเช่นเดียวกับ SMTP ข้อความผ่านพอร์ต HTTP มีรูปแบบทั่วไปดังนี้

START LINE <CRLF>

MESSAGE HEADER <CR> F>

<CRLF>

MESSAGE BODY <CR>F>

ในอดีต <CRLF> หมายถึง carriage-return+line-feed บรรทัดแรก START_LINE ระบุว่าเป็นข้อความร้องขอ (request) หรือข้อความตอบกลับ (response) เพื่อต้องการระบุ “ขั้นตอนวิธีสื่อสารระยะไกล (remote procedure)” (ในกรณีของข้อความร้องขอ) หรือกำหนดสถานะของคำขอ (ในกรณีของข้อความตอบกลับ) บรรทัดต่อมาจะระบุพารามิเตอร์ที่ตรงกับคำขอหรือการตอบสนอง มีค่าเป็น 0 หรือมากกว่าตามจำนวนบรรทัดใน MESSAGE_HEADER หลังจากนั้นสิ้นสุดคำสั่งด้วย <CRLF> HTTP กำหนดข้อความส่วนເheads เดอර์ได้ hely คำสั่ง บางส่วนเกี่ยวกับคำขอข้อความ บางส่วนสำหรับข้อความตอบกลับ และบางส่วนใช้ประกอบกับข้อมูลเนื้อหา เสร็จแล้วกำหนดบรรทัดว่างด้วย <CRLF> สุดท้ายหลังจากบรรทัดว่าง เป็นเนื้อหาของข้อความที่ร้องขอ MESSAGE_BODY ซึ่งเป็นข้อความเป็นที่เชิร์ฟเวอร์ตอบกลับมาเป็นหน้าเว็บ ส่วนนี้จะว่างหากเป็นคำสั่งประเภทร้องขอ

กล่าวถึงสาเหตุที่นักออกแบบ HTTP เลือก TCP นั้น ความจริงแล้วนักออกแบบไม่จำเป็นต้องใช้ TCP เสมอไป แต่ TCP ทำงานร่วมกับ HTTP ได้ดีโดยเฉพาะการจัดส่งข้อมูลที่มีความน่าเชื่อถือ (ไม่มีใครต้องการเข้าหน้าเว็บหายไปบางส่วน) มีการควบคุมการไหล และการควบคุมความแออัด อย่างไรก็ตาม มีปัญหาเล็กน้อยที่อาจเกิดขึ้นจากการใช้โปรโตคอล request/response ที่เลเยอร์อยู่บนโพรโทคอล TCP ได้หากนักพัฒนาไม่สนใจรายละเอียดปลีกย่อยของการติดต่อบรระหว่างชั้นแอปพลิเคชันและชั้นขนส่ง

Request Messages

บรรทัดแรกของคำขอ HTTP ระบุข้อมูลประกอบด้วยสามส่วน: คำสั่งการดำเนินการ หน้าเว็บที่ต้องการดำเนินการ และเวอร์ชันของ HTTP ที่ใช้ แม้ว่า HTTP จะกำหนดคำขอได้มากกว่านี้ แต่คำสั่งที่พบบ่อยที่สุดสองประการคือ GET(ดึงข้อมูลหน้าเว็บที่ระบุ) และ HEAD(ดึงข้อมูลสถานะเกี่ยวกับหน้าเว็บที่ระบุ) เห็นได้ว่าเป็นคำสั่งที่ใช้เมื่อเบราว์เซอร์ต้องการดึงข้อมูลและแสดงเว็บเพจ ตารางที่ 7.1 แสดงชุดคำสั่งที่พบใน HTTP

จากตารางที่ 7.1

ตารางที่ 7.1: คำสั่งร้องขอ ของโพรโทคอล HTTP

Operation	Description
OPTIONS	ขอข้อมูลจากเซิร์ฟเวอร์ว่ารองรับคำสั่งแบบไหนบ้าง
GET	รับเอกสารจาก ยูอาร์แอล ที่ป้อน
HEAD	รับข้อมูลส่วนເheads (meta information) จาก ยูอาร์แอล ที่กำหนด
POST	ส่งข้อมูลให้ เชิร์ฟเวอร์
PUT	จัดเก็บเอกสารตามที่ได้รับจาก ยูอาร์แอล
DELETE	ลบข้อมูลตามระบุใน ยูอาร์แอล
TRACE	สำหรับใช้ติดตามปัญหา ยูอาร์แอล
CONNECT	สำหรับเชื่อมทาง พรีอคชี(proxy)

ตัวอย่างคำสั่งประเภท START_LINE ดังนี้

GET http://git.npu.world/index.html

HTTP/1.1

เพื่อบอกว่าผู้ใช้ต้องการให้เซิร์ฟเวอร์ส่งข้อมูลจากไฟล์ index.html กลับมา นอกจากนี้ยังสามารถระบุชื่อไอ索ตใน MESSAGE_HEADER ตัวอย่างเช่น

GET index.html HTTP/1.1

Host: git.npu.world

ข้อความระบุไอโซต ใน MESSAGE_HEADER สามารถใช้ If-Modified-Since ซึ่งทำให้ผู้ขอสามารถขอหน้าเว็บโดยกำหนดเงื่อนไขได้ โดยเซิร์ฟเวอร์จะคืนข้อมูลหน้าเว็บกลับมา

Response Messages

ข้อความตอบกลับจะเริ่มต้นด้วยการส่ง START_LINE ใช้ระบุเวอร์ชันของ HTTP ที่ต้องการติดต่อ ตัวอย่างเช่น

HTTP/1.1 202 Accepted

แสดงว่าเซิร์ฟเวอร์สามารถตอบสนองคำขอได้ในขณะที่การตอบกลับ

HTTP/1.1 404 Not Found

แสดงว่าเซิร์ฟเวอร์ไม่สามารถตอบสนองคำขอได้ เนื่องจากไม่พบหน้าดังกล่าว รหัสตอบกลับทั่วไปมีห้าประเภท โดยหลักแรกของรหัสใช้ระบุประเภท ตารางที่ 7.2

ตารางที่ 7.2: คำสั่งร่องตอบกลับ ของโพรโทคอล HTTP

รหัส	ประเภท	ตัวอย่าง
1xx	เกี่ยวกับข้อมูล	request received, continuing process
2xx	เกี่ยวกับความสำเร็จ	action successfully received, understood, and accepted
3xx	เกี่ยวกับการเปลี่ยนเส้นทาง	further action must be taken to complete the request
4xx	เกี่ยวกับข้อผิดพลาดเกิดกับไคลเอนต์	request contains bad syntax or cannot be fulfilled
5xx	เกี่ยวกับข้อผิดพลาดเกิดกับเซิร์ฟเวอร์	server failed to fulfill an apparently valid request

บางครั้งข้อความตอบกลับ ถูกนำมาใช้ในรูปแบบอื่น ตัวอย่าง เช่น คำขอเปลี่ยนเส้นทาง (โดยระบุรหัส 302) กลายเป็นวิธีที่มีใช้ใน CDN(Content Distribution Networks) โดยเปลี่ยนเส้นทางคำขอไปยังแคช (cache) ที่อยู่ใกล้เคียง

ข้อความตอบกลับเริ่มตั้งแต่ MESSAGE_HEADER มีหนึ่งบรรทัดขึ้นไป สามารถเปลี่ยนตำแหน่งของข้อมูลได้ ตัวอย่างเช่น Location ระบุว่า URL ที่ร้องขออยู่ในตำแหน่งอื่น ดังนั้น หากหน้าเว็บของสาขาวิศวกรรมคอมพิวเตอร์ได้ย้ายจาก <http://en.npu.ac.th/cpe/index.html> ไปยัง <http://git.npu.world/index.html> เซิร์ฟเวอร์อาจตอบสนองดังต่อไปนี้

HTTP/1.1 301 Moved Permanently

Location: <http://git.npu.world/index.html>

ในกรณีที่ว่าไป ข้อความตอบกลับจะแสดงหน้าที่ร้องขอด้วย หน้านี้เป็นเอกสาร HTML แต่เนื่องจาก อาจมีข้อมูลที่ไม่ใช่ข้อความ (เช่น รูปภาพ GIF) จึงเข้ารหัสโดยใช้ MIME บาง MESSAGE_HEADER ได้กำหนด เนื้อหาเว็บโดยระบุ Expires(เวลาที่เนื้อหาเว็บหมดอายุ) และ เวลาที่เว็บได้รับการแก้ไขล่าสุดที่เซิร์ฟเวอร์

Uniform Resource Identifiers

URL ถือเป็น *URI*(Uniform Resource Identifier) ประเภทหนึ่ง URI คือสตริงที่ระบุตำแหน่งทรัพยากร โดยที่ ทรัพยากรสามารถเป็นอะไรก็ได้ที่สามารถอ้างถึง เช่น เอกสาร รูปภาพ หรือบริการต่างๆ

รูปแบบของ URI ส่วนแรกเรียกว่า “scheme” ใช้กำหนดชื่อสำหรับระบุทรัพยากรใดๆ เช่น mailto สำหรับบอกว่าทรัพยากรที่กำลังอ้างถึงนั้นคือ อีเมล หรือ file สำหรับใช้อ้างถึงชื่อไฟล์ ยกตัวอย่างเช่น

`mailto:santa@northpole.org`

และ

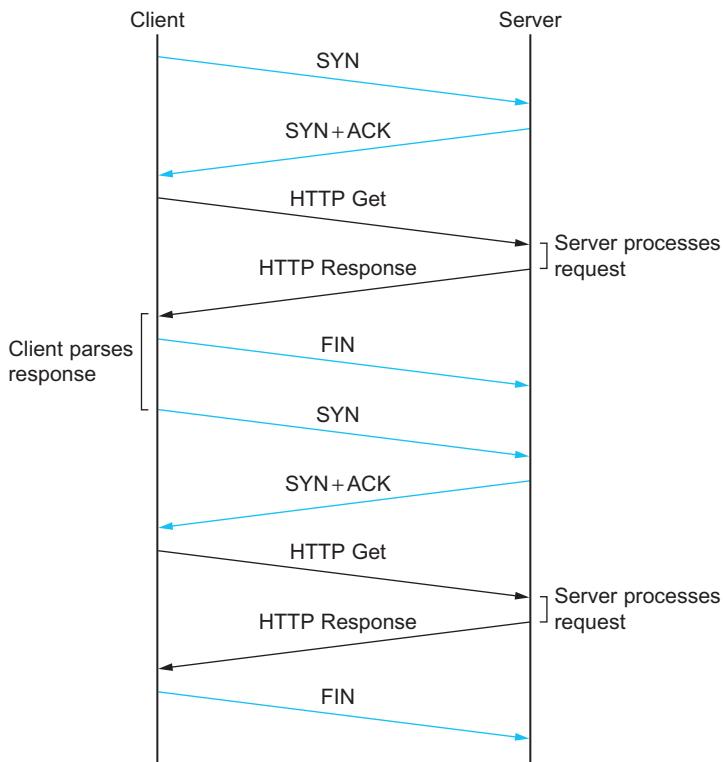
`file:///C:/foo.html`

ทรัพยากรไม่จำเป็นต้องสามารถเรียกคืนหรือเข้าถึงได้ เช่นในภาษาマークアップ(extensible markup language) จะระบุผ่านทาง URI ที่ดูเหมือน URL ถ้ากล่าวให้ถูกตามหลักแล้ว ภาษามาร์กอัปไม่ใช่ตัวระบุตำแหน่ง เพราะไม่ได้บอกถึงวิธีค้นหาทรัพยากร แต่เป็นตัวกำหนดชื่อทรัพยากรเพื่อให้มีชื่อกันภายในแมสเปชันๆ โดยไม่มีข้อกำหนดว่าจะสามารถตั้งชื่ออย่างไรก็ได้ที่ URI ที่กำหนด

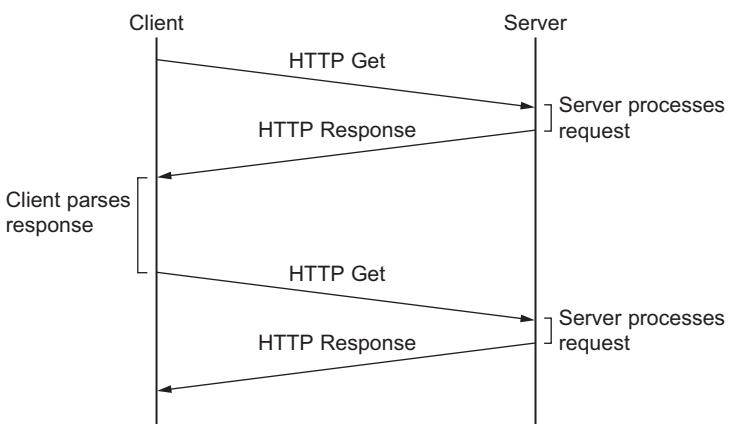
TCP Connections

HTTP รุ่นแรก(1.0) เป็นการเชื่อมต่อ TCP เพื่ออ่านทรัพยากรจากเว็บเพจที่ล็อกอินแล้ว เช่น อ่านไฟล์ index.html จะส่งข้อมูลให้ TCP สร้างการเชื่อมต่อเมื่อได้ข้อมูลแล้วจะปิดการเชื่อมต่อ ก่อนแล้วจึงสร้างการเชื่อมต่อใหม่กับ ทรัพยากรใหม่ เป็นลำดับต่อไป จึงเรียกว่า “Non-persistent connection” สังเกตได้ว่า HTTP/1.0 มีปัญหา ในการสร้างการเชื่อมต่อ TCP ทุกครั้งกับทุกทรัพยากร ซึ่งการสร้างการเชื่อมต่อของ TCP แต่ครั้งมีการใช้ ทรัพยากรในการสร้างการเชื่อมต่อและปิดการเชื่อมต่อ ซึ่งหากหน้าเว็บเพจมีทรัพยากรมากเช่น มีรูปจำนวน มาก จะทำให้เกิดการสร้างการเชื่อมต่อมาหลายครั้ง รูปที่ 7.4 อธิบายการเชื่อมต่อ HTTP โดยอ่านข้อมูลหนึ่งชิ้นจะ สร้าง TCP หนึ่งไฟล์

วิธีปรับปรุงประสิทธิภาพทำได้จ่ายคือลดจำนวนของการสร้างการเชื่อมต่อ เพราะเป็นการติดต่อใน เว็บไซต์เดียว หรือเว็บเพจเดียวที่เคยเชื่อมต่อได้ จึงไม่จำเป็นต้องสร้างการเชื่อมต่อใหม่ทุกครั้งเป็นที่มาของการ กำหนดใน โพรโทคอล HTTP/1.1 โดยการให้มีการสร้างการเชื่อมต่อใหม่ในครั้งแรกครั้งเดียว และปิดการเชื่อม ต่อเมื่อได้ทรัพยากรครบหมดแล้ว วิธีนี้เรียกว่า “persistent connections” มีการเชื่อมต่อตามรูปที่ 7.5



รูปที่ 7.4: HTTP Non-persistent อ่านข้อมูลครั้งละชุด
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>



รูปที่ 7.5: ปรับปรุง HTTP ด้วยวิธี persistent connections
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

อย่างไรก็ตาม การเชื่อมต่อแบบ persistent connections ไม่ได้มีข้อดีเพียงอย่างเดียว แต่ต้องแลกกับปัญหาที่ทั้งโคลอนต์และเซิร์ฟเวอร์จะไม่รู้ว่าต้องเปิดการเชื่อมต่อ TCP ไว้นานเท่าไหน การเปิดรอการเชื่อมนี้สำคัญอย่างยิ่งกับเครื่องเซิร์ฟเวอร์ เพราะหมายถึงการสำรองหน่วยความจำ สำรองทรัพยากรคอมพิวเตอร์ ไว้สำหรับการเชื่อมต่อนั้น ซึ่งอาจถูกขอให้เปิดการเชื่อมต่อไว้เป็นพันการเชื่อมต่อ วิธีแก้ไขคือเซิร์ฟเวอร์ต้องกำหนดเวลาสูงสุดก่อนปิดการเชื่อมต่อโดยอัตโนมัติหากไม่ได้รับคำขอทั้งโคลอนต์และเซิร์ฟเวอร์ต้องตรวจสอบว่าอีกฝ่ายปิดการเชื่อมต่อหรือไม่ และคงอย่างสังสัญญาณเป็นระยะว่าყังคงเชื่อมต่อ

Caching

วิธีการที่ทำให้เว็บใช้งานได้อย่างมีประสิทธิภาพขึ้นคือการบันทึกข้อมูลหน้าเว็บที่ถูกเรียกบ่อยๆไว้ในหน่วยความจำที่เข้าถึงได้เร็วกว่าปกติ เรียกวิธีนี้ว่า “การแคช(caching)” การทำ แคชมีประโยชน์หลายด้าน อาทิ เช่น ในมุมมองของโกลเอน์ต์ สามารถดึงข้อมูลจากแคชที่อยู่ในลิสต์ให้แสดงผลออกหน้าจอได้เร็วกว่าต้องดึงมาจากทั่วโลกสำหรับประโยชน์ในมุมมองของเซิร์ฟเวอร์ การมีแคชเป็นการสกัดกั้นและตอบสนองคำขอสามารถลดภาระงานบนเซิร์ฟเวอร์ได้

การทำแคชสามารถทำได้หลายจุด ตัวอย่างเช่น เว็บเบราว์เซอร์ของผู้ใช้สามารถแคชเว็บเพจที่เคยเข้าถึงล่าสุด เมื่อเรียกเพ็จซ้ำเพียงแค่แสดงหน้าเพจที่สำเนาที่แคชไว้อีกตัวอย่างหนึ่ง เว็บไซต์สามารถรองรับแคชหมดทั้งไซต์ได้ ซึ่งช่วยให้ผู้ใช้สามารถได้ใช้ประโยชน์จากหน้าเพจหรือไฟล์ที่ดาวน์โหลดบ่อย โดยผู้ใช้รายอื่นก่อนหน้านี้ยังได้ใช้ประโยชน์ร่วมด้วย

ถึงแม้หน้าเพจแค่ได้ทั้งหมดไม่ว่าข้อมูลใด แต่เรื่องที่ควรให้ความสำคัญคือแคชต้องตรวจสอบให้แน่ใจว่าไม่ตอบด้วยหน้าเว็บตัวเก่า(แคชเก็บไฟล์เก่า แต่เว็บจริงได้เปลี่ยนเป็นข้อมูลใหม่แล้ว) วิธีแก้ปัญหาทำได้ง่าย ตัวอย่างเช่น เชิร์ฟเวอร์กำหนดวันอายุให้กับข้อมูล (Expires ในไฟล์ส่วนเซ็ตเดอร์) เมื่อข้อมูลหมดอายุจะลบข้อมูลในแคชแล้วรอข้อมูลปรับข้อมูลใหม่

7.1.3 Web Services

วิวัฒนาการการพัฒนาเทคโนโลยีเว็บได้มุ่งเน้นไปที่การโต้ตอบระหว่างมนุษย์กับเว็บไซต์ฟิเวอร์ ด้วยร่างเข่นมนุษย์ใช้เว็บเบราว์เซอร์เพื่อโต้ตอบกับเซิร์ฟิเวอร์ และการโต้ตอบจะตอบสนองตามอินพุตที่รับจากผู้ใช้ (เช่นโดยการคลิกลิงก์) อย่างไรก็ตาม สำหรับการโต้ตอบระหว่างคอมพิวเตอร์กับคอมพิวเตอร์โดยตรงมีความต้องการมากขึ้น เช่นเดียวกับแอปพลิเคชันที่อธิบายข้างต้นจำเป็นต้องมีโทรศัพท์เคลื่อนที่สื่อสารถึงกันโดยตรงเช่นกัน

ในการสื่อสารระหว่างแอปพลิเคชันกับแอปพลิเคชันนั้นมาจากการแนวคิดทางธุรกิจ ในอดีต การติดต่อระหว่างองค์กร ต้องทำด้วยตนเอง เช่น การกรอกแบบฟอร์มคำสั่งซื้อหรือโทรศัพท์เพื่อตรวจสอบว่ามีสินค้าอยู่ในสต็อกหรือไม่ ถึงแม้มุ่งหมายในองค์กรเดียวกันก็ตาม ซึ่งในอดีตหากต้องการอ่านข้อมูลจากแผนกอื่นจะเป็นเรื่องปกติที่จะต้องทำเองด้วยตนเอง ในมุ่งมองการออกแบบเว็บเซอร์วิส(service) ให้เกิดการแบ่งบันทรัพยากรแก่กันได้สะดวกขึ้น

ยกตัวอย่างประโยชน์จากการแบ่งบันข้อมูล สมมติว่าซื้อหนังสือจากร้านขายออนไลน์อย่างเช่น ซึ่งหนังสือผ่าน Amazon และบริษัท Amazon ได้รับคำสั่งซื้อแล้วจะจัดส่งผ่านบริษัทขนส่ง เช่น <http://www.fedex.com> เมื่อบริษัทขนส่งนำพัสดุเข้าระบบแล้วจะมอบหมายเลขติดตามพัสดุให้ Amazon และ Amazon สามารถใช้หมายเลขนี้ส่งต่อให้ลูกค้าได้ การแยกเปลี่ยนข้อมูลเหล่านี้ทำผ่านเครือข่ายอินเทอร์เน็ต โดยพนักงานไม่ต้องเดินทางไปบริษัท FedEx ซึ่งต้องมีโทรศัพท์เคลื่อนในการแยกเปลี่ยนข้อมูลที่จำเป็นในการติดตามพัสดุภัณฑ์เรียกว่า “Package Tracking Protocol”

7.2 แอปพลิเคชันประเภทโครงสร้างพื้นฐาน

มีโพรโทคอลบางประเภทที่จำเป็นต้องมีเพื่อทำให้อินเทอร์เน็ตทำงานได้ราบรื่น แต่ไม่ได้จำกกลุ่มเข้ากับตัวแบบที่มีการแบ่งชั้น หนึ่งในโพรโทคอลนี้คือระบบชื่อดomen (DNS) ซึ่งไม่ได้เป็นแอปพลิเคชันปกติสำหรับให้ผู้ใช้ได้ใช้งานโดยตรง แต่เป็นบริการที่แอปพลิเคชันอื่นเก็บทั้งหมดต้องพึ่งพา สาเหตุเนื่องจากบริการซึ่งมีเพื่อแปลงชื่อโฮสต์เป็นหมายเลขไอพี การมี DNS ทำให้ผู้ใช้แอปพลิเคชันอื่นสามารถอ้างถึงโฮสต์จากระยะไกลโดยใช้ชื่อแทนการจดจำหมายเลขไอพี

7.2.1 Name Service (DNS)

เนื้อหาภายในหนังสือเล่มนี้ส่วนใหญ่กล่าวถึงวิธีระบุตำแหน่งของไซต์โดยใช้หมายเลขไอพี แม้ว่าเราเตอร์ประมวลผลไอพีได้ง่าย แต่สำหรับมนุษย์การจดจำหมายเลขไอพีเป็นเรื่องยาก ด้วยเหตุนี้จึงมีการออกแบบให้มีการอ้างถึงหมายเลขไอพีโดยใช้ชื่อแทนด้วยอักษรภาษาอังกฤษชื่นการใช้บริการ HTTP โดยใช้ชื่อเช่น www.npu.ac.th ซึ่งระบบ DNS จะจับคู่ระหว่างหมายเลขไอพีกับชื่อที่จดจำได้ง่าย บริการซึ่งบางครั้งเรียกว่า “มิดเดิลแวร์” เนื่องจากเป็นตัวทำหน้าที่อยู่ตรงกลางระหว่างแอปพลิเคชันและเครือข่ายพื้นฐาน

ชื่อโฮสต์ แตกต่างจากที่อยู่โฮสต์ ทั้งสองใช้งานแตกต่างกัน ชื่อโฮสต์มีไว้สำหรับมนุษย์อ่านทำความเข้าใจได้รวดเร็ว โดยไม่มีข้อมูลช่วยเหลือใดๆ ให้ระบบคอมพิวเตอร์ใช้ค้นหาตำแหน่งของไซต์ สำหรับที่อยู่โฮสต์มีข้อมูลที่ระบบคอมพิวเตอร์สามารถใช้ติดตามค้นหาไปถึงโฮสต์ได้ซึ่งที่อยู่โฮสต์ต้องไม่ซ้ำกัน

ก่อนที่จะลงรายละเอียดว่าระบบ DNS ทำงานอย่างไร ตั้งชื่อโฮสต์อย่างไรในระบบอินเทอร์เน็ต ขั้นแรกจะแนะนำคำศัพท์พื้นฐานซึ่งเกี่ยวข้องกับระบบ DNS คำศัพท์แรกคือขอบเขตในการอ้างถึงซึ่ง เทียบกับเอกภพสัมพาร์ท(Universal Set) ในระบบเซต ในระบบโดเมนเนมเรียกว่า “เนมสเปช” มีเพื่อใช้กำหนดขอบเขตที่มีทั้งหมดในระบบ ซึ่ง เนมสเปชอาจจะกำหนดเป็นระนาบเดียว หรืออาจจะแบ่งเป็นเนมสเปชย่อย (เนมสเปชภายในเนมสเปชก็ได้ ต่อมาก็จะเรียกว่า “bindings”) เป็นการกำหนดค่าให้กับส่วนที่ต้องการอ้างถึง ค่าที่ว่านี้ส่วนใหญ่อ้างถึงหมายเลขไอพี และสุดท้ายคือกระบวนการในการอ่านค่าจากข้อมูลจำนวนมากเรียกว่า “resolution mechanism” กระบวนการทั้งหมดนี้เรียกเป็นระบบ “name server”

เปรียบเทียบการทำงานระบบ DNS ได้เหมือนการอ้างถึงวัตถุบางอย่างด้วยชื่อ เช่น อ้างถึงรูปของคนในประเทศต่างๆ ในโลก จุดอ้างถึงบนสุดคือโลกใบเดียว ก่อนจะแบ่งสมาชิกจากโลกใบเดียวออกเป็น 195 ประเทศ (ข้อมูลปี 2022) การแบ่งประเทศจัดเป็นกลุ่มตามลำดับถัดจากโลก ซึ่งภายใต้แต่ละประเทศอาจจัดกลุ่มเป็นรัฐ หรือ จังหวัด แบ่งการจัดกลุ่มตามประเทศนั้นๆ และลดขนาดลงจนถึงระบบครอบครัว จนถึงตัวบุคคล

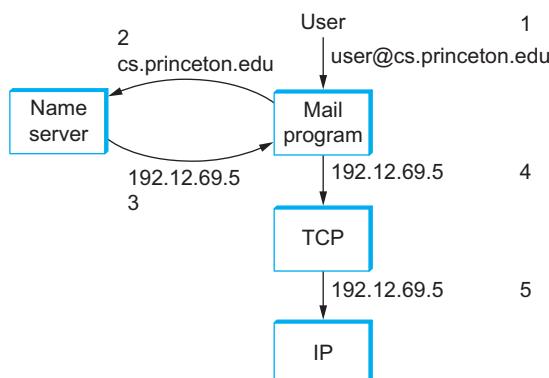
ในอดีตยกเริ่มต้นอินเทอร์เน็ตมีคอมพิวเตอร์และอุปกรณ์เครือข่ายไม่กี่เครื่อง มีระบบกลางที่ทำหน้าที่จดจำชื่อเครื่องและอุปกรณ์ทั้งหมดเรียกว่า “Network Information Center (NIC)” โดยบันทึกชื่อโฮสต์คู่กับหมายเลขไอพี บันทึกเป็นตารางจัดเก็บในไฟล์ชื่อ HOST.TXT³ เมื่อมีโฮสต์ใหม่ ผู้ดูแลระบบจะส่ง อีเมล์ถึง NIC

³ปัจจุบันยังมีพิมพ์หนังสือปึกใหญ่บันทึกชื่อโฮสต์กับหมายเลขไอพีหรืออีเมล์ อยู่เป็นระยะ (คล้ายสมุดหน้าเหลือง)

เพื่อขอเพิ่มข้อมูล แล้ว NIC จะนำข้อมูลใหม่ต่อท้ายตาราง หลังจากนั้น NIC จะส่งข้อมูลให้กับผู้ใช้เครือข่าย ทราบผ่านทาง อีเมล ซึ่งอาจต้องใช้เวลาหลายวันกว่าที่ระบบจะปรับปรุงครบ

ไม่น่าแปลกใจแต่อย่างใด วิธีที่กล่าวมาข้างต้นไม่เป็นผลต่อระบบอินเทอร์เน็ตที่มีการขยายตัวอย่างรวดเร็ว ช่วงกลางปี 1980 ได้เกิดระบบ DNS เพื่อทำหน้าที่ทดสอบการบันทึกชื่อผ่าน NIC ระบบ DNS มีการทำงานเป็นลำดับชั้น แตกต่างจาก NIC ที่บันทึกแบบตารางແ霎วเดียว ทำให้ตารางถูกจัดเป็นกลุ่มย่อยโดยแต่ละกลุ่มย่อยมี เนมสเปชเป็นของตนเอง

ผู้ใช้สามารถเรียกใช้ชื่อโฮสต์จากโปรแกรมแอปพลิเคชัน (อาจฝังอยู่ในโปรแกรม(program) เช่น อีเมล หรือ บูร์แออล) และโปรแกรมจะใช้ชื่อโฮสต์แปลเป็นที่อยู่ของโฮสต์ しながらแอปพลิเคชันจะเชื่อมต่อไปยังโฮสต์ ผ่านทางชั้นขนส่ง (เช่น TCP) โดยมีหมายเลขไอพี ของโฮสต์ มีการทำงานเป็นไปตามรูปที่ 7.6 รูปภาพนี้ได้ตัดรายละเอียดออกเพื่อให้ทำความเข้าใจง่าย ในชั้ntonการทำงานจริงมีรายละเอียดเบื้องหลังอยู่มากซึ่งจะกล่าวถึงในลำดับต่อไป



รูปที่ 7.6: การแปลงชื่อโฮสต์เป็นที่อยู่โฮสต์ ตั้งแต่ลำดับ 1 ถึง 5
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

จากรูปเมื่อ User ต้องการส่งอีเมล ไปยัง user@cs.princeton.edu สิ่งแรกที่ระบบเมล์เซิร์ฟเวอร์ทำคือการแปลงชื่อโฮสต์ เป็นหมายเลขไอพี ด้วยการส่งข้อมูล cs.princeton.edu ไปสอบถามกับ Name server (ระบบ DNS) และได้คำตอบเป็น 192.168.69.5 เสร็จแล้ว Mail program จะติดต่อไปยังเครื่องปลายทางผ่านการขนส่งแบบ TCP ที่หมายเลขไอพีเป็น 192.168.69.5

ลำดับขั้นของโดเมนเนม

ระบบอินเทอร์เน็ตเป็นระบบประกอบจากโพรโทคอลจำนวนมาก มีการเชื่อมโยงเป็นเครือข่ายขนาดใหญ่ ระบบภายในประกอบขึ้นจากการที่ได้รับการพัฒนามาเป็นอย่างดี ระบบ DNS เป็นระบบหนึ่งที่พัฒนามาอย่างดี ระบบหนึ่งที่สนับสนุนให้อินเทอร์เน็ตขยายวงกว้างอย่างไร้ขีดจำกัด

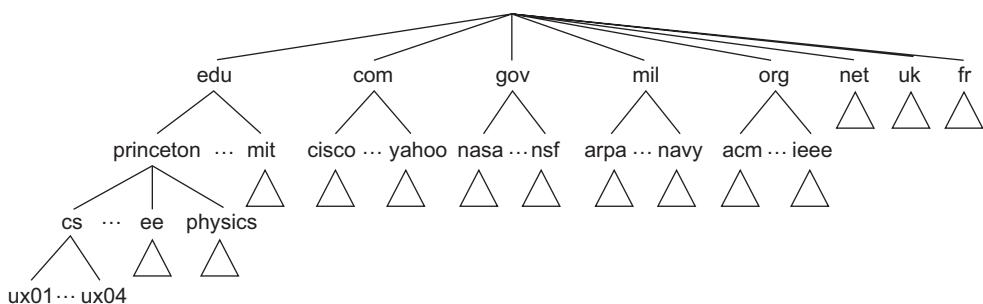
DNS ใช้เนมสเปชแบบลำดับชั้น แตกต่างจากระบบที่ใช้ไฟล์ในระบบปฏิบัติการยูนิกซ์(Unix) ที่ประมวลผลจากช้ายไปขวา โดยมีเครื่องหมาย slash (/) คั่นระหว่างชื่อ สำหรับชื่อ DNS จะอ่านจากขวาไปซ้ายและใช้จุดเป็นตัวคั่นชื่อ แต่ในชั้นยังคงอ่านชื่อโดเมนจากซ้ายไปขวา ตัวอย่างชื่อโดเมนสำหรับโฮสต์ <git.npu.world>

คือ สังเกตว่าชื่อโดเมนใช้เพื่อตั้งชื่อ “วัตถุ (object)” ได้ในระบบอินเทอร์เน็ต ไม่ได้ใช้เพื่อจับคู่ชื่อโฉสต์กับหมายเลขไอพีเพียงอย่างเดียว จะซัดเจนกว่าถ้ากล่าวว่า DNS จับคู่ชื่อโดเมนกับค่าต่างๆ เพียงแต่ที่พบส่วนใหญ่เป็นการรับค่าระหว่างชื่อโดเมนกับหมายเลขไอพี

รูปแบบการอ่านโดเมนจะอ่านจากขวาไปซ้ายมือ ยกตัวอย่างเช่น

mit.edu

โดเมน mit.edu. มีรากคือ “.” ซึ่งอยู่ทางขวาเมื่อสุดหลัง edu (ปกติไม่จำเป็นต้องเขียน . ที่เป็น root หลังโดเมน) ต่อมาเป็นโดเมน edu และสามารถแยกได้ edu คือ mit



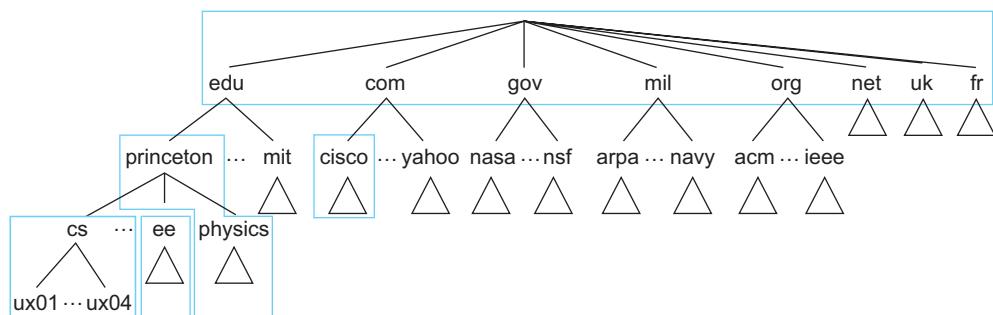
รูปที่ 7.7: การแบ่งลำดับชั้นของระบบโดเมนเนม
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ในระบบโดเมนเนมตำแหน่งบนสุดคือ root ทำหน้าที่เป็นเป็นจุดเริ่มต้นของระบบโดเมนเนม อธิบายจากรูปที่ 7.7 ถัดจาก root จะเป็นโดเมนที่อยู่ภายใต้ root นอกจากชื่ออย่างอ้างถึงประเทศต่างแล้ว จะมีโดเมนใหญ่ 6 โดเมนเรียกว่า “big six” ได้แก่ edu com gov mil org และ net เป็นต้น และยกตัวอย่างเช่นถัดจากโดเมน edu จะมีโดเมนย่อยลงมาตามลำดับ ระบบ DNS พัฒนาขึ้นโดยมีพื้นฐานจากประเทศสหรัฐอเมริกา ซึ่งโดเมนในอดีตจะเกี่ยวข้องกับหน่วยงานในสหรัฐฯ เช่น .edu .gov .mil เป็นต้น ต่อมาอินเทอร์เน็ตมีการขยายตัวไปยังประเทศอื่นทั่วโลกทำให้โดเมนระดับบน (top-level domain) ได้เพิ่มจำนวนขึ้นมาก ในปัจจุบันมีโดเมนระดับบนสุดมากกว่า 1200 โดเมน

Name Servers

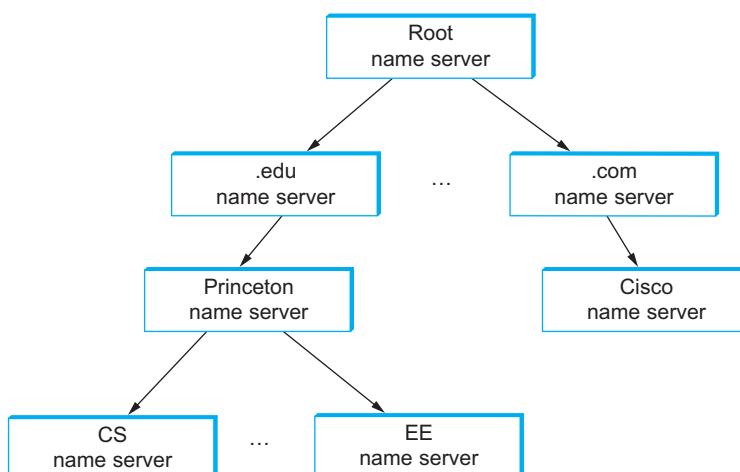
หัวข้อนี้กล่าวถึงวิธีทำงานแบบลำดับชั้นของโดเมน ซึ่งใช้อธิบายพื้นฐานการทำงานของซอฟต์แวร์ DNS จากที่กล่าวมาข้างต้นระบบ DNS แบ่งการทำงานเป็น เนมสเปช ย่อย สำหรับแต่ละโดเมน โดยโดเมนอ้างอิงด้วยชื่อและคั่นด้วยเครื่องหมาย “.” สำหรับแบ่งเนมสเปช จากรูปที่ 7.8 สังเกตจากบันลงล่างได้แบ่งเนมสเปชจาก top-level domain ประกอบด้วย edu com gov mil org net uk และ fr เป็นเนมสเปชเป็นระดับเดียวกัน และเรียกสมาชิกย่อยที่เรียงเป็นลำดับลงล่างเรียกว่า “โซน” ตัวอย่าง โซนของ edu ได้แก่ princeton และ mit

ม่องค์กรสากลที่ทำหน้าที่บริหารจัดการชื่อโดเมนได้แก่ ICANN(Internet Corporation for Assigned Names and Numbers) ทำหน้าที่บริหารจัดการ top-level domain โดยแจกหน้าที่ดูแลให้แก่ผู้รับผิดชอบในแต่ละโซน



รูปที่ 7.8: การแบ่งเนมสเปซออกเป็นโฉนด
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

โฉนดคือเป็นหน่วยพื้นฐานของระบบโดเมนเนม ซึ่งตรงกับเนมเชิร์ฟเวอร์ใช้งานทึกข้อมูลที่เกี่ยวข้องกับชื่อโดเมนนั้นๆ ข้อมูลที่บันทึกแต่ละโฉนดนำไปใช้ในซอฟต์แวร์เนมเชิร์ฟเวอร์ โดยเนมเชิร์ฟเวอร์จะทำหน้าที่ค่อยตอบคำถามที่เกี่ยวกับชื่อโฉนดที่รับผิดชอบ การทำงานเป็นไปตามลำดับดังที่แสดงในรูปที่ 7.9



รูปที่ 7.9: ลำดับชั้นของเนมเชิร์ฟเวอร์
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

ระบบบันทึกโฉนดจะเก็บไว้กับเครื่องเซิร์ฟเวอร์จำนวนสองเครื่อง เรียกว่า primary และ secondary สำหรับ primary จะเป็นเครื่องหลักในการตอบข้อมูลขณะที่ secondary ทำหน้าที่สำรองกรณีเครื่อง primary ไม่ทำงาน ดังนั้นในการป้อนข้อมูลโดเมนใหม่จะทำให้เครื่อง primary เพียงเครื่องเดียว โดยเครื่อง secondary จะคัดลอกข้อมูลไปโดยอัตโนมัติ ข้อมูลที่บันทึกในแต่ละโฉนดประกอบด้วย

(Name, Value, Type, Class, TTL)

โดย Name และ Value เป็นส่วนที่จะนำไปใช้หาก มีการขอข้อมูล Name ระบบจะคืนค่าเป็น Value ที่จดคู่กับ Name นั้น ลำดับต่อมาได้แก่ Type ทำหน้าที่ระบุประเภทวัตถุ(object)ของสิ่งที่อ้างถึง เช่น Type=A หมายถึง ต้องการอ้างถึง Address (A) ซึ่งคือหมายเลขไอพี ประเภทของวัตถุมีหลากหลายดังรายการต่อไปนี้

- NS และ Value ใช้กำหนดชื่อ เนมเชิร์ฟเวอร์

- CNAME ใช้กำหนดชื่อเรียกอีกชื่อ (นามแฝง)
- MX ใช้กำหนดชื่อโฮสต์ที่ทำหน้าที่เป็น เมล์เซอร์ฟเวอร์

ฟิลด์ Class มีเพื่อให้ระบบอื่นที่ไม่ใช่ NIC กำหนดประเภทเรกคอร์ดขึ้นเองได้ จนถึงปัจจุบัน มีเพียงระบบอินเทอร์เน็ตระบบเดียวที่มี ดังนั้น Class จึงมีเพียง Class=IN เท่านั้น สุดท้าย ฟิลด์ time-to-live (TTL) ใช้จะแสดงระยะเวลาที่ข้อมูลในโซน (record) ยังคงใช้ได้ เมื่อ TTL หมดอายุ เซิร์ฟเวอร์ต้องลบข้อมูลออกจากแคช

การบันทึกข้อมูลของโซนทำความเข้าใจได้จากตัวอย่างต่อไปนี้ การดึงมาจากลำดับขั้นของโดเมนในรูปที่ 7.7 ในที่นี่จะเห็น TTL ซึ่งยังไม่ถูกใช้ในส่วนนี้ เพื่อลดความซับซ้อนของการอธิบาย ได้แบ่งข้อมูลดังนี้

```
(edu, a3.nstld.com, NS, IN)
(a3.nstld.com, 192.5.6.32, A, IN)
(com, a.gtld-servers.net, NS, IN)
(a.gtld-servers.net, 192.5.6.30, A, IN)
...
```

จากข้อมูลบรรทัดแรก (edu, a3.nstld.com, NS, IN) อธิบายได้ว่าโดเมน edu (Name=edu) จะมีหมายเลขไอพีเป็น a3.nstld.com (Value=a3.nstld.com) ทำหน้าที่เป็น เนมเซอร์ฟเวอร์ (Type=NS) โดยใช้ในระบบอินเทอร์เน็ต (Class=IN)

มาตรฐานลำดับย่อของโดเมน edu โดยมีเครื่องหมายเป็นตัวอักษรภาษาไทย ได้แก่

```
(princeton.edu, dns.princeton.edu, NS, IN)
(dns.princeton.edu, 128.112.129.15, A, IN)
...
```

จากลำดับย่อของโดเมนด้านบนสังเกตได้ว่า princeton.edu อยู่ภายใต้โซน edu จากข้อมูล (princeton.edu, dns.princeton.edu, NS, IN) บอกถึง โดเมน princeton.edu มี value=dns.princeton.edu ซึ่งเป็นค่าของข้อมูลที่ใช้สำหรับไปยังเซิร์ฟเวอร์ (Type=NS) ข้อมูลนี้ใช้ในระบบอินเทอร์เน็ต ข้อมูลต่อมา (dns.princeton.edu, 128.112.129.15, A, IN) เป็นการให้ข้อมูลหมายเลขไอพีของโดเมน dns.princeton.edu โดยการกำหนดให้มี Type=A ซึ่งหมายถึง Address (ไอพี แบบ IPv4)

ในกรณีนี้ Type=NS และ A ซึ่งใช้อ้างถึง princeton.edu เมื่อมีการใช้ระบบเว็บเซิร์ฟเวอร์ ซึ่งเป็นเครื่องหมายหมายเลขไอพีอื่น เช่น www.princeton.edu สามารถกำหนดหมายเลขไอพีโดยกำหนด Type=A และระบุหมายเลขไอพีที่ต้องการ ตัวอย่างเช่น

```
(www.princeton.edu, 128.112.198.35, A, IN)
(penguins.cs.princeton.edu, dns1.cs.princeton.edu, NS, IN)
(dns1.cs.princeton.edu, 128.112.136.10, A, IN)
...
```

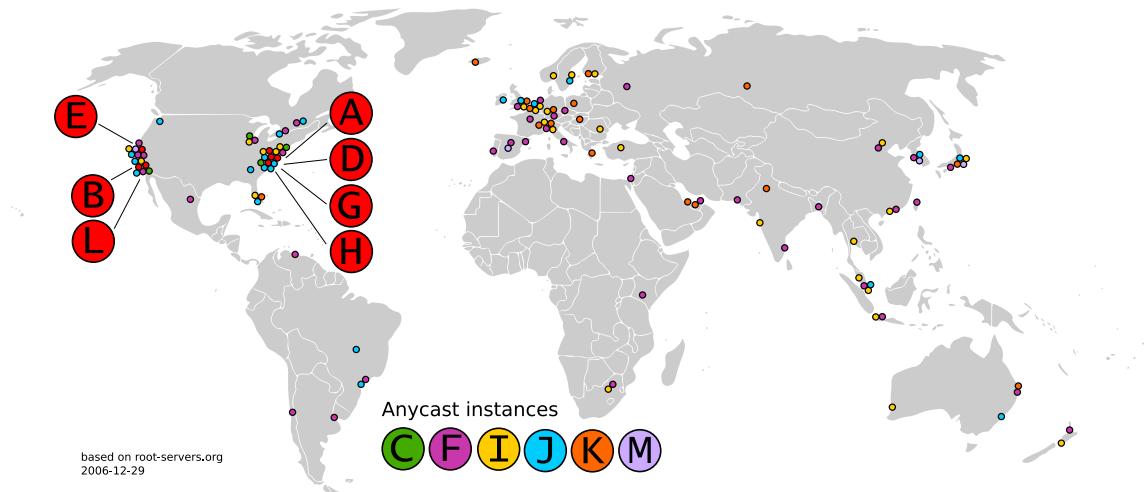
จากที่กล่าวมาเนมเซิร์ฟเวอร์จะรับหน้าที่ดูแลและแต่ละไซนโดย เนมเซิร์ฟเวอร์ตามมาตรฐานกำหนดให้มีสองเครื่องได้แก่ primary และ secondary โดยการสังเกตเนมเซิร์ฟเวอร์ดูได้จากการค้นระหว่างชื่อ เช่น mit.edu. หมายถึงมีเนมเซิร์ฟเวอร์สำหรับดูแลไซน edu และอีกกลุ่มคือเนมเซิร์ฟเวอร์ที่ดูแลไซน mit.edu การทำงานของเนมเซิร์ฟเวอร์สามารถจ่ายย่อยลงได้เรื่อยๆ เช่น ชื่อโดเมน cs.princeton.edu จะมีการตั้งเนมเซิร์ฟเวอร์ที่ทำหน้าที่ดูแลไซน cs.princeton.edu เป็นระดับที่สาม การกำหนดชื่อโดเมนสามารถใช้นามแฝง (CNAME) ได้ เช่น www.cs.princeton.edu ใช้นามแฝงเป็น coreweb.cs.princeton.edu หมายถึงเป็นการใช้ข้อมูลร่วมกันซึ่งในที่นี้คือ 128.112.136.35 ลำดับต่อมาเป็นการอ้างถึงเนมเซิร์ฟเวอร์ ของ cs.princeton.edu โดยชี้ไปที่ mail.cs.princeton.edu และระบุว่าชื่อดังกล่าวคือที่อยู่ของ เมลเซิร์ฟเวอร์ และบรรทัดสุดท้ายได้ระบบหมายเลขไอพีของ mail.cs.princeton.edu เป็น 128.112.136.72

```
(penguins.cs.princeton.edu, 128.112.155.166, A, IN)
(www.cs.princeton.edu, coreweb.cs.princeton.edu, CNAME, IN)
(coreweb.cs.princeton.edu, 128.112.136.35, A, IN)
(cs.princeton.edu, mail.cs.princeton.edu, MX, IN)
(mail.cs.princeton.edu, 128.112.136.72, A, IN)
...
```

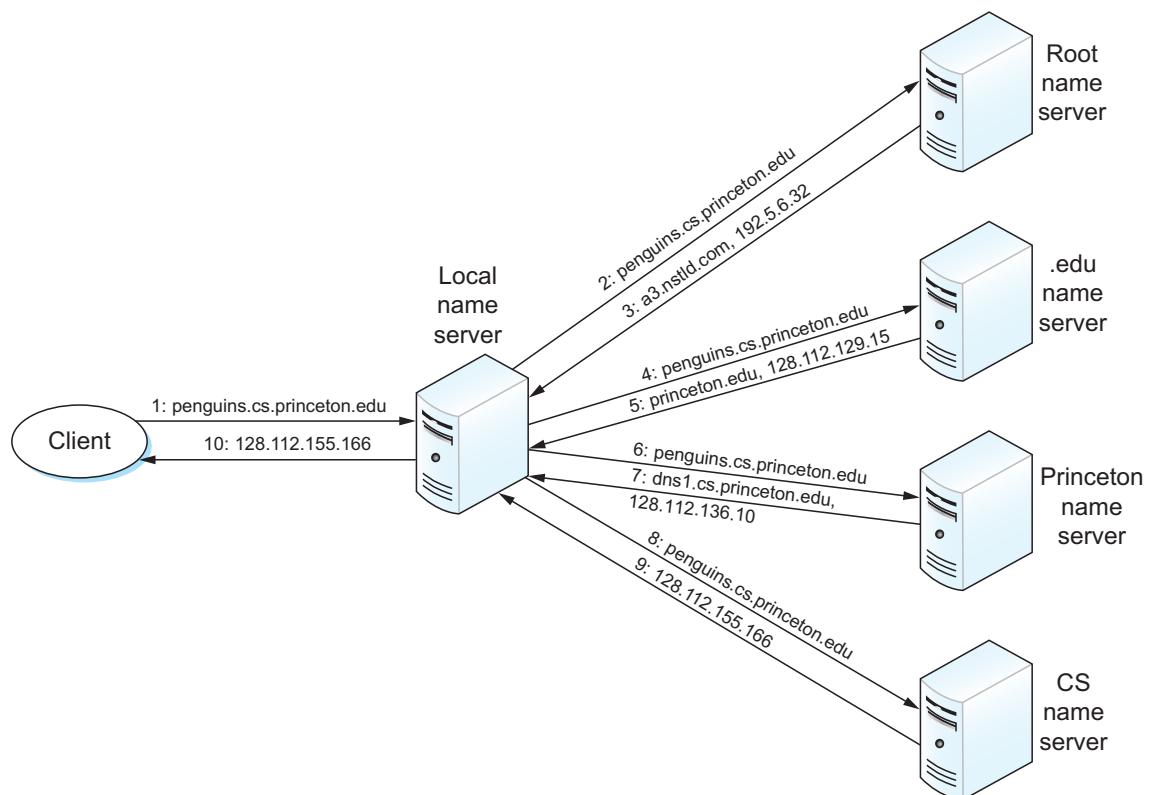
Name Resolution

หัวข้อนี้กล่าวถึงขั้นตอนนับตั้งแต่เครื่องลูกข่ายมีชื่อโดเมนและต้องการทราบข้อมูลจากชื่อโดเมนนั้นจะเกิดขึ้น ตอนสื่อสารอะไรขึ้นบ้าง สมมติลูกข่ายต้องการติดต่อกับเครื่อง penguins.cs.princeton.edu ดังอธิบายในรูปที่ 7.11 สิ่งแรกที่ขอฟ์เวอร์กายนในเครื่องลูกข่ายทำคือส่งคำถามไปสอบถาม Root name server เพื่อสอบถามว่า ถ้าต้องการได้ข้อมูลโดเมน penguins.cs.princeton.edu จะต้องถามเครื่องใด โดยเครื่อง Root name server จะบันทึกอยู่ในระบบปฏิบัติการมาตั้งแต่แรกแล้ว เครื่อง root name server บนโลกนี้มีทั้งหมด 16 เครื่อง [a-m].root-servers.net (ยกเว้น root-servers.net ไม่อยู่ในระบบโดเมน) มีที่ตั้งตามรูปที่ 7.10

ข้อมูลส่งไป root ทุกเครื่อง (a ถึง m) โดยข้อมูลที่ตอบกลับเร็วสุดจะนำมาใช้ต่อ สมมติได้ข้อมูล ตอบกลับมาระบุว่า เครื่อง Root ไม่ได้รับผิดชอบโดยตรง จึงส่งต่อให้ไปสอบถามกับ เครื่องที่รับผิดชอบโดเมน edu คือ a3.nstld.com ลำดับต่อมาเครื่องลูกข่ายส่งคำสั่งขอหมายเลขไอพีของโดเมน a3.nstld.com มีหมายเลขไอพีเด ทำให้ได้คำตอบเป็น 192.5.6.3 เมื่อลูกข่ายได้หมายเลขไอพีเครื่องที่รับผิดชอบโดเมน edu จึงส่งข้อมูลไปถามว่าเครื่องได้รับผิดชอบ penguins.cs.princeton.edu ได้คำตอบว่า ให้ไปสอบถาม กับเครื่องที่รับผิดชอบโดเมน princeton.edu ซึ่งได้หมายเลขไอพีเป็น 128.112.129.15 ลูกข่ายใช้หมายเลขไอพีดังกล่าวสอบถามอีกรอบ ด้วยส่วนคำถาม(query) ไปยังเครื่อง 128.112.129.15 ว่าทราบหมายเลขไอพีของ penguins.cs.princeton.edu หรือไม่ ซึ่งได้คำตอบว่า princeton.edu ไม่ได้รับผิดชอบโดยตรง แต่ส่งต่อให้ไปถาม cs.princeton.edu ซึ่งมีหมายเลขไอพี 128.112.136.10 ลูกข่ายจึงส่งข้อมูลไปถามเครื่อง cs.princeton.edu (128.112.136.10) ว่าหมายเลขไอพีของ penguins.cs.princeton.edu มีหมายเลขไอพีเด เมื่อ cs.princeton.edu ตรวจสอบแล้วพบว่ามีข้อมูลบันทึกอยู่ในระบบ DNS ที่ตนดูแลจึงตอบกลับด้วยหมายเลขไอพี 128.112.155.166 ถือเป็นสิ้นสุดการสอบถาม



รูปที่ 7.10: Root เนมเซิร์ฟเวอร์
ลิขสิทธิ์ ภาพ CC-2.5 แหล่งที่มา https://en.wikipedia.org/wiki/Root_name_server#/media/File:Root-current.svg



รูปที่ 7.11: ขั้นตอนการสอบถามหมายเลขอีพีในระบบ DNS
ลิขสิทธิ์ ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

อย่างไรก็ตามขั้นตอนการทำงานนี้เป็นขั้นตอนสมบูรณ์แบบซึ่งในการทำงานจริง ระบบDNS มีการบันทึกใน แคช ทำให้การตรวจสอบไม่ได้สอบถามจนครบขั้นตอนตามที่กล่าวมา

7.3 เทคโนโลยีコンเนนเนอร์

ปัจจุบันการพัฒนาซอฟต์แวร์เว็บขนาดใหญ่ได้ประยุกต์ใช้ความรู้ด้านโครงสร้างเครือข่ายพื้นฐานเพื่อสนับสนุนให้ซอฟต์แวร์ทำงานได้อย่างมีประสิทธิภาพ โดยจำเป็นต้องพัฒนาการทำงานทั้งหมดภายในแอปพลิเคชันเลเยอร์ ตัวอย่างเช่น การให้บริการเว็บเซอร์วิสเพื่อรับลูกค้าข่ายจำนวนมาก ต้องการการพัฒนาแอปพลิเคชันสำหรับแบ่งเบาการประมวลผลไปยังเครื่องแม่ข่ายอื่น ซึ่งอาจจะติดตั้งอยู่คุณละพื้นที่ เช่น เทคโนโลยีคลาวน์ เป็นต้น หากพัฒนาโดยใช้มุมมองแอปพลิเคชันเพียงอย่างเดียวจะทำให้เกิดข้อจำกัด เช่น การฝากข้อมูลผ่านเทคโนโลยีเข้ารหัส TLS ที่ต้องการการเชื่อมต่อแบบ end-to-end ทำให้ไม่สามารถแบ่งโหลดไปในชั้นอื่นได้ จึงจำเป็นต้องปรับแอปพลิเคชันให้รองรับเทคโนโลยีการเข้ารหัสสำหรับรองรับการกระจายได้

การพัฒนาซอฟต์แวร์ขนาดใหญ่จัดกลุ่มได้สองแนวคิดได้แก่ สถาปัตยกรรมโมโนลิธ (monolith architecture) และ สถาปัตยกรรมไมโครเซอร์วิส การขยายตัวของผู้ให้บริการเว็บแอปพลิเคชัน เพื่อตอบสนองความต้องการของผู้ใช้งานที่เพิ่มมากขึ้น โดยมีการให้บริการในหลากหลายรูปแบบแต่ทุกรูปแบบจะใช้เว็บแอปพลิเคชันเป็นสื่อกลาง อาทิเช่น การจำหน่ายสินค้าหรือบริการ วิธีชำระสินค้าผ่านทางออนไลน์ ซึ่งทั้งหมดล้วนแล้วแต่ทำได้ผ่านทางเว็บแอปพลิเคชัน ดังนั้นเว็บแอปพลิเคชันจึงเป็นเป้าหมายในการโจมตีของผู้ประสงค์ร้ายไปโดยปริยาย เมื่อผู้ให้บริการสูญเสียความมั่นคงปลอดภัยของข้อมูลจากอุบัติเหตุความเสียหายต่อภาระงานส่งผลต่อธุรกิจ ทำให้ไม่มีการดำเนินกิจการต่อไปได้

บทที่ 8

เครือข่ายไร้สายประเภทประยุกต์พลังงาน

มีการแบ่งกลุ่มเทคโนโลยีสื่อสารไร้สายตามระยะทางของการส่งสัญญาณไว้ดังนี้

- WWAN (Wireless WAN)
- WLAN (Wireless LAN)
- WPAN (Wireless PAN)
- WBAN (Wireless BAN)

ตัวอย่างเทคโนโลยีในกลุ่ม WWAN เช่น LTE-A (5G) LTE(4G) UMTS (3G) EDGE (2.5G) หรือ WiMAX รวมถึงโครงข่ายดาวเทียม (Satellite) ในการใช้งานส่วนใหญ่เป็นความถี่ที่ต้องได้รับอนุญาตจากหน่วยงานกำกับดูแลด้านความถี่ (Regulator) เป็นต้น เทคโนโลยี WLAN เป็นกลุ่มที่มีการส่งสัญญาณได้ระยะทางใกล้ๆ กัน คือ WWAN เป้าหมายเพื่อสื่อสารภายในอาคาร หรือระยะทางไม่เกิน 5km ซึ่งมาตรฐานส่วนใหญ่ใช้ย่านความถี่ ISM (industrial, scientific and medical) เช่น IEEE802.11 ([IEEE Computer Society LAN/MAN Standards Committee, 2021](#)) Hiperlan ([Committee และคณะ, 1996](#)) เป็นต้น เทคโนโลยี WPAN มีระยะให้บริการในบริเวณรอบๆ ตัวผู้ใช้งาน(หรืออุปกรณ์ตัวส่ง) โดยทั่วไปกำหนดให้มีระยะไม่เกิน 15 เมตร ตัวอย่างเทคโนโลยี WPAN ได้แก่ บลูทูธ และ ชิกบี(ZigBee®)

เทคโนโลยีการสื่อสารภายใน IoT(Internet of Things) เป็นแนวทางการเชื่อมเครือข่ายยุคใหม่ ที่กำลังได้รับการกล่าวถึง แนวคิดคือต้องการให้อุปกรณ์ดิจิทัลเขื่อมต่อโครงข่ายได้ทั้งหมด จะทำให้เกิดโครงข่ายขนาดใหญ่ขึ้น เทคโนโลยีพื้นฐานของ IoT เป็นการเชื่อมผ่านเครือข่ายไร้สายแบบประเภทประยุกต์พลังงาน ซึ่งปัจจุบันมีหลายเทคโนโลยีที่มุ่งเน้นด้านการประยุกต์พลังงานซึ่งจะกล่าวถึงดังต่อไปนี้

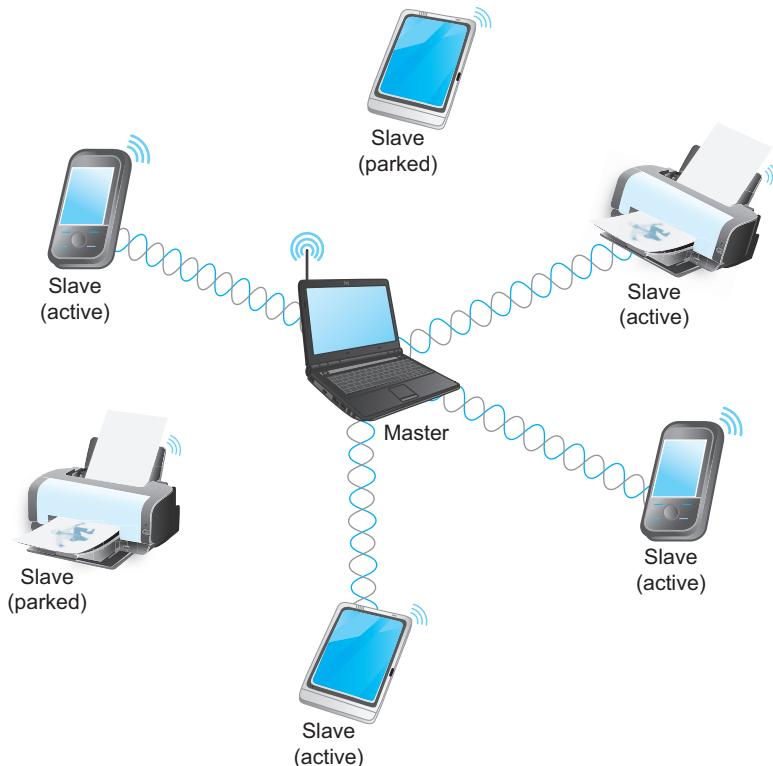
8.1 บลูทูธ (IEEE802.15.1)

เทคโนโลยีบลูทูธออกแบบได้ดีและมีการใช้งานอย่างแพร่หลายในอุปกรณ์พกพา เช่น โทรศัพท์มือถือ ที่ต้องการแลกเปลี่ยนข้อมูลระยะใกล้ไม่ต้องการความเร็วสูง และ ประยุกต์พลังงานเทคโนโลยีบลูทูธถูกใช้ในการส่งสัญญาณเสียงและแลกเปลี่ยนข้อมูลอย่างง่าย เช่น การแลกเปลี่ยนคอนแทคหมายเลขโทรศัพท์ เป็นต้น

บลูทูธทำงานบนความถี่สากลในย่าน ISM (2.45GHz) ซึ่งได้รับยกเว้น บลูทูธมีความเร็วในการสื่อสารอยู่ระหว่าง 1 ถึง 3Mbps ส่งข้อมูลได้ระยะทาง 10 เมตร

บลูทูธได้รับความร่วมมือในการออกแบบจากองค์กรชื่อว่า Bluetooth Special Interest Group ทำหน้าที่ในการกำหนดรูปแบบโทรศัพท์ กำหนดคุณสมบัติทางฮาร์ดแวร์ และแอพพลิเคชันที่ใช้งานผ่านบลูทูธ ซึ่งปัจจุบันทุกอุปกรณ์โทรศัพท์เคลื่อนที่รองรับการทำงานเทคโนโลยีบลูทูธ

โครงข่ายบลูทูธถูกเรียกว่าเป็น piconet เครื่องมีพื้นที่ให้บริการขนาดเล็กกว่าเครือข่ายแลนด้วยสายมี อุปกรณ์เข้มต่อในโครงข่ายได้สูงสุด 7 เครื่องดังรูปที่ 8.1 อุปกรณ์บลูทูธไม่สามารถสื่อสารกันได้โดยตรง ทุก การสื่อสารเป็นการส่งไป อุปกรณ์หลัก(primary) ก่อนแล้วจึงส่งให้รอง(secondary) ต่อไป



รูปที่ 8.1: เครือข่ายพิโคนेटเทคโนโลยีบลูทูธ
ลิขสิทธิ์ภาพ CC-4.0 แหล่งที่มา <https://github.com/SystemsApproach/book>

จากที่เทคโนโลยีบลูทูธทำงานช่วงความถี่ ISM ทำให้ต้องใช้เทคโนโลยีการแฟสเปกตรัม เพื่อลบการ รบกวน สำหรับข้อกำหนดการแฟสเปกตรัม ของ บลูทูธ ใช้วิธีFHSS โดยมีความถี่ทั้งหมด 79 ความถี่โดยถือ โครงสร้างแต่ละความถี่เป็นเวลา $625 \mu\text{s}$

อุปกรณ์สามารถเข้า荷มด 荷มดหลับ(parked) เพื่อประหดพลังงานได้ ทำให้มีลูกข่ายได้ 255 อุป กรณ์ นอกจากรบกวนที่ใช้ความถี่ 2.45GHz แล้วยังมี เทคโนโลยีประหดพลังงานมาตรฐานอื่นที่ถูกใช้ในความถี่ เดียวกันนี้ เช่น ชิกบี (Safaric และ Malaric, 2006) ซึ่งกำหนดในมาตรฐาน IEEE 802.15.4 ซึ่งประหดพลังงาน มากกว่าบลูทูธแต่ส่งข้อมูลได้ช้า

8.2 เทคโนโลยีลอรา

ลอรา เป็นเทคโนโลยีการสื่อสารไร้สายจัดอยู่ในกลุ่ม LPWAN(Low-Power Wide Area Networks)¹ เทคโนโลยีนี้ถือลิขสิทธิ์โดยบริษัท Semtech เป้าหมายการพัฒนาเทคโนโลยีเพื่อให้สามารถส่งสัญญาณได้ระยะทางไกล และประหยัดพลังงาน เพื่อใช้ในงานประมวลอยู่ห่างไกลจะจำเป็นต้องใช้พลังงานอย่างจำกัด ตัวอย่างเช่น พื้นที่ป่า พื้นที่ขนาดใหญ่และเป็นอันตรายต่อมนุษย์ สิ่งที่เทคโนโลยีนี้ต้องแลกเพื่อให้ส่งสัญญาณได้ระยะทางไกล และประหยัดพลังงานคือความเร็วในการสื่อสาร ซึ่งการสื่อสารผ่านเทคโนโลยีลอรา จะทำให้น้อยที่สุดเพื่อการประหยัดพลังงาน

บริษัทผู้ถือลิขสิทธิ์ลอราได้เปิดเผยแพร่เทคโนโลยีบางส่วนและมีการศึกษาเพร่หلامภัยหลัง Knight และ Seeber (2016) ได้เปิดเผยขอสโคดที่พัฒนาในรูปแบบซอฟต์แวร์ GNURadio ทำให้เกิดการวิจัยกว้างขวางขึ้น

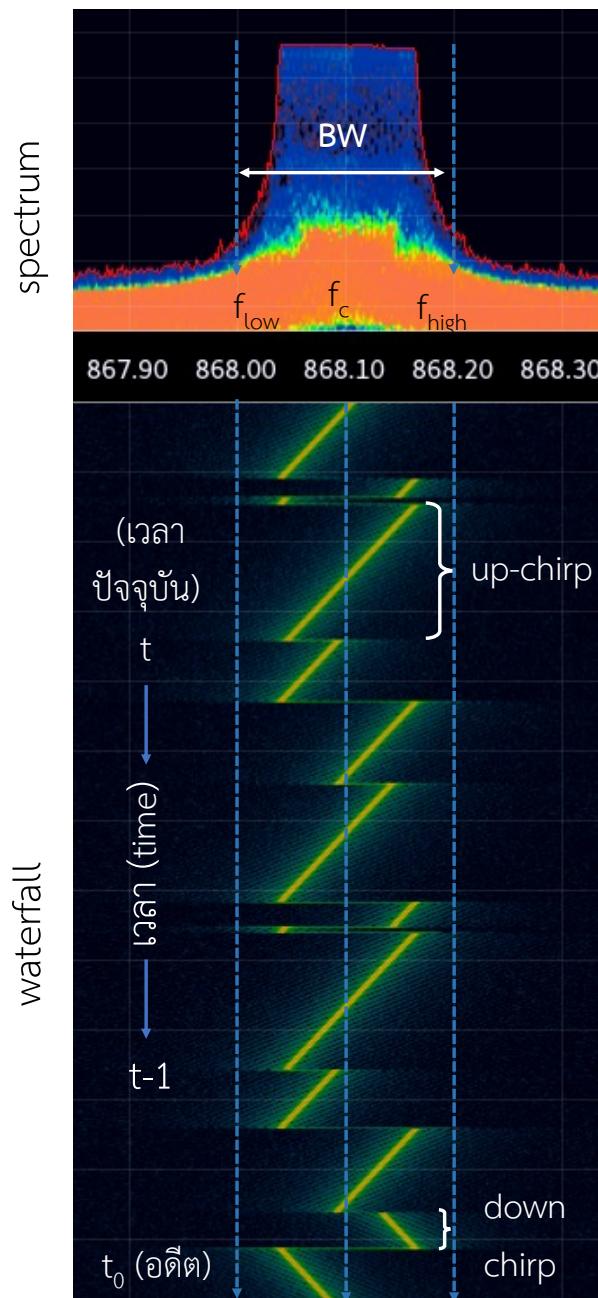
การmodulation(modulation) สัญญาณทางกายภาพของลอราใช้เทคโนโลยีชีร์พ² เทคโนโลยีชีร์พ มีการทำงานแบบ กວาดความถี่(frequency sweep) จากความถี่ต่ำไปความถี่สูงแล้วกวดจากสูงกลับมาต่ำ รูปแบบสัญญาณมีลักษณะแบบฟังเสียงรูปที่ 8.2(revspace.nl, 2020) เป็นสัญญาณที่วัดจากเครื่อง spectrum analyzer จากภาพส่วนด้านบนเป็นการวัดค่าโดยใช้ฟังก์ชัน แสดงผลสัญญาณที่ค่าสูงสุด(max hold) เพื่อให้สัญญาณหยุดนิ่งที่ค่าสูงสุด(เส้นสีแดง) แกน x แทนความถี่และแกน y แทนค่าความแรงสัญญาณ กราฟลักษณะนี้เรียกว่า “スペクトル” และภาพด้านล่างอธิบายโดยให้แกน x แทนความถี่ และแกน y แทนเวลา (time) โดยที่เวลาที่ผ่านไปแล้วอยู่ด้านล่างสุด เวลาล่าสุดอยู่บนสุด เรียกการแสดงภาพนี้ว่า “waterfall” วิธีอ่านกราฟ waterfall อ่านแกน y จากล่างขึ้นบน จากรูปส่วน “up-chirp” คือการเปลี่ยนจากความถี่ต่ำไปความถี่สูง และสัญญาณที่เปลี่ยนความถี่จากสูงลงมาต่ำ(ลดลงจากซ้ายไปขวา) เรียกว่า “down-chirp”

รูปที่ 8.3 อธิบายスペクトรัมของสัญญาณลอราโดยแกน x แทนค่าเวลา และแกน y แทนความถี่โดยมีความถี่กลาง (center frequency (f_0)) อยู่ตรงกลาง อ่านสัญญาณจากซ้ายไปขวา สัญญาณแรกเริ่มที่ f_0 มีความถี่เพิ่มขึ้นจนถึงความถี่สูงสุดของแบบดิวิเดอร์ แล้วกลับไปเริ่มต้นที่ความถี่ต่ำสุดของแบบดิวิเดอร์แบบกระแทน หันแล้วยับความถี่เพิ่มขึ้นไปจนถึงความถี่สูงสุดมีทั้งหมด เรียกสัญญาณที่เริ่มจากความถี่ต่ำไปความถี่สูงเรียกว่า “up-chirp” และมีชิมเบล(symbol) สุดท้ายของ preamble ได้ปรับจากความถี่สูงลงมาความถี่ต่ำ(แบบไม่กระแทนหัน) เรียกว่า “down-chirp” จากรูปมี 8 up-chirp และมี 2 down-chirp ก่อนที่จะเริ่มต้นส่งส่วนที่เป็นสัญญาณข้อมูล การทำ up-chirp และ down-chirp มีเพื่อให้เครื่องรับได้กำหนดสัญญาณนาฬิกาให้ตรงกับเครื่องส่งโดยการคำนวนโดยใช้ข้อมูลจากส่วน preamble ตัวอย่างเช่นรูปที่ 8.4 มีส่วน preamble ประกอบด้วย 5 up-chirp และ 2 down-chirp ก่อนเข้าสู่สัญญาณส่วนข้อมูล

เทคนิคการmodulationลอราถือเป็นคุณสมบัติเด่นทำให้สามารถส่งสัญญาณได้ระยะทางไกลแม้ใช้พลังงานน้อย เทคนิคนี้มาจากพื้นฐานจากการส่งสัญญาณแบบ เรดาห์(radio detection and ranging) ใช้วิธีการวัดความถี่จากต่ำไปสูงและจากสูงลงมาต่ำ เรียกว่าการแปลงสัญญาณแบบชีร์พ สำหรับวิธีการวัด

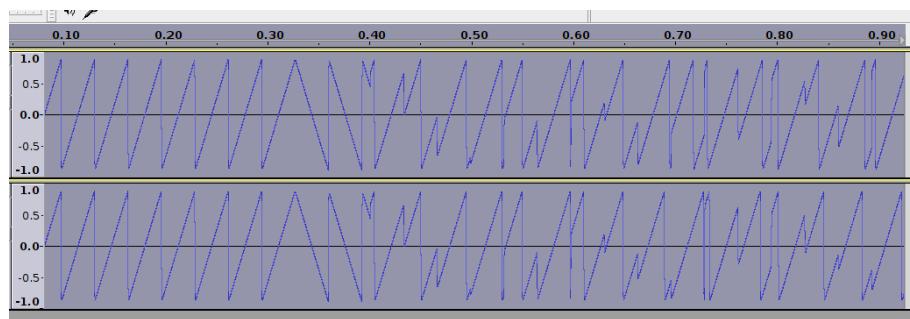
¹โครงการที่อยู่ต่ออุปกรณ์สื่อสารบริเวณกว้างกำลังส่งต่ำ เหมาะสำหรับการใช้ในต่ออุปกรณ์จำนวนมากซึ่งมีการส่งข้อมูลขนาดเล็กที่มีแบบต่อรีนตัวเอง มีการเริ่มต่อโดยทั้งหมดโครงข่ายโทรศัพท์เคลื่อนที่ และ สถานีฐานที่ใช้คลื่นความถี่ unlicensed(สมาคมโทรศัพท์มือถือไทยฯ, 2017)

²เทคโนโลยีชีร์พ

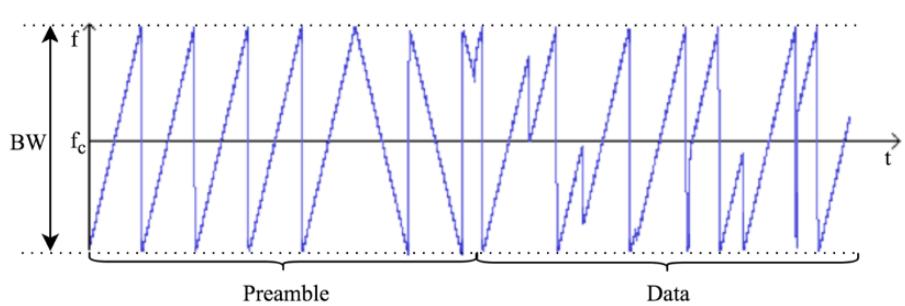


รูปที่ 8.2: สัญญาณคลื่นวิทยุเทคโนโลยีเชิร์พ ส่วนด้านบนอธิบายด้วยสเปกตรัม ส่วนด้านล่างอธิบายด้วย waterfall

ความถี่ที่ใช้กับโลราเรียกว่า “CSS(Chirp Spread Spectrum)” ทำให้มีคุณสมบัติทนทานต่อการถูกกรบกวนที่ต้องปรากฏการณ์ดอปเบอร์(doppler effect) และมีภาคส่วนสัญญาณให้กำลังต่ำที่ทำงานไม่ซับซ้อน เนื่องจากไม่ต้องใช้ตัวสร้างสัญญาณนาฬิกาที่แม่นยำสูง สำหรับภาครับสามารถแปลงสัญญาณได้ทั้งที่สัญญาณมีความเบา (-130 dBm) หรือเรียกได้ว่าเครื่องรับมีความไวต่อการรับสัญญาณสูง (high sensitivity) ลิ่งที่ทำให้ภาครับสามารถรับสัญญาณเบาได้ส่วนหนึ่งเกิดจากการแก้ไขความผิดพลาด(FEC(Forward Error Correction))ของสัญญาณได้ดี ซึ่งการใช้ FEC ถือเป็นพารามิเตอร์หนึ่งในการกำหนดคุณลักษณะของเครือข่าย



รูปที่ 8.3: ส่วน preamble ของ เซิร์ฟสเปกตรัม
ลิขสิทธิ์ภาพ CC BY-SA 3.0 แหล่งที่มา <https://revspace.nl/DecodingLora>



รูปที่ 8.4: ส่วน preamble และส่วนข้อมูล
ลิขสิทธิ์ภาพ CC BY-SA 3.0 แหล่งที่มา <https://revspace.nl/DecodingLora>

ยลอรา ที่ส่งผลต่ออัตราเร็วในการส่งข้อมูล การกำหนดพารามิเตอร์ที่ส่งผลกระทบต่อการสื่อสารเทคโนโลยีล้อราจะกล่าวถึงในหัวข้อต่อไป

8.2.1 ถอดสัญญาณล้อรา

การส่งสัญญาณแบบได้ส่องรูปแบบ รูปแบบแรกเป็นการส่งออกจากเกตเวย์ (GW) เรียกว่า “ดาวน์ลิงก์(downlink)” รูปแบบที่สองเป็นการส่งสัญญาณออกจากลูกข่าย(ED(end-devices)) เรียกว่า “อัปลิงก์(uplink)”

สัญญาณคลื่นวิทยุเทคโนโลยีล้อราทั้งอัปลิงก์หรือดาวน์ลิงก์แบ่งเป็น 2 ส่วนได้แก่ ส่วน preamble และส่วนข้อมูล

สำหรับดาวน์ลิงก์กำหนดให้ preamble เป็นสัญญาณ up-chirp และ ให้ข้อมูลเป็นสัญญาณ down-chirp ขณะที่สัญญาณแบบอัปลิงก์กำหนดให้ preamble เป็น down-chirp และข้อมูลเป็น up-chirp

เซิร์ฟในเทคโนโลยีล้อราคือชิมโบล ซึ่งชิมโบล³ ในการสื่อสารอ่านนั้นมีแบบดิจิตอล หากทำให้ระบบสามารถส่งได้หลายเซิร์ฟ จะทำให้สามารถส่งข้อมูลหลายบิตได้ภายในได้แบบดิจิตอลที่จำกัด เป็นผลให้สื่อสารได้เร็ว

³เทคนิคที่นำพาข้อมูลดิจิทัลผ่านคลื่นวิทยุ

หนึ่งเครื่องจะแบ่งออกเป็นส่วนๆ ตามจำนวนบิตที่ใช้ในการ模ดูเลชัน ค่าจำนวนบิตต่อเครื่อง (หรือเรียกว่า บิตต่อชิมโบล) แปรผันตามค่าพารามิเตอร์ของการ模ดูเลชัน ซึ่งจะกล่าวถึงในหัวข้อต่อไป

8.2.2 พารามิเตอร์การ模ดูเลชันเครือข่าย lorra

การกำหนดพารามิเตอร์ด้านการ模ดูเลชันสัญญาณในเครือข่าย lorra ผลการสื่อสารของ lorra ในด้านอัตราเร็วของการส่งข้อมูล (Data Rate) ค่าความไวต่อการรับสัญญาณ(sensitivity) ซึ่งส่งผลต่อระยะห่างระหว่างเครื่องส่งและเครื่องรับ ค่าพารามิเตอร์ที่สำคัญมีดังนี้ ค่ากำลังส่ง(Transmission Power:TP) ค่าความถี่(Carrier Frequency : CF) ค่าแบบดิวิดิธ(BW) สเปรดแฟกเตอร์ และ อัตราส่วนข้อมูลกับรหัสแก้ความผิดพลาด(Coding Rate) ค่าเริ่มต้นเป็นไปตามตารางที่ 8.1

ตารางที่ 8.1: ค่าพารามิเตอร์ของ lorra

พารามิเตอร์	ค่าเริ่มต้น	ช่วงตัวเลขที่เป็นไปได้
TP	50mW	0-50mW
CF	925MHz	137-1020MHz
SF	7	7-12
BW	125kHz	7.8 - 500kHz
CR=4/(4+n)	4/5	$n \in \{1 - 4\}$

TP คือพลังงานภาคส่งสัญญาณที่ภาคส่งสัญญาณมีเพื่อจำกัดไม่ให้เกิดการส่งเกินจากข้อกำหนด (เมื่อส่งเกินข้อกำหนดจะขัดต่อกฎหมาย) ข้อกำหนดพื้นฐานเทคโนโลยี lorra ในประเทศไทย โดยการกำหนดของ กสทช. กำหนดให้ใช้ความถี่อยู่ใน 920-925MHz (มีแบบดิวิดิธ 5MHz) สำหรับใช้สื่อสารในงานด้านไอโอที(อินเทอร์เน็ตของสรรพสิ่ง) ได้โดยไม่ต้องมีใบอนุญาตวิทยุคมนาคม ภายใต้ข้อจำกัดให้มีกำลังส่งไม่เกิน 500mW e.i.r.p(Effective Isotropic Radiated Power) (สำหรับผู้ค้าที่ไม่มีใบอนุญาตฯ ส่งได้ไม่เกิน 50mW e.i.r.p) อุปกรณ์ส่วนใหญ่สามารถส่งสัญญาณได้ไม่เกิน 50mW

CF คือค่าความถี่ที่ใช้ (Carrier Frequency) ซึ่งประเทศไทยสามารถใช้งานความถี่ได้ในช่วง 920-925MHz (คณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ, 2017)

SF ใช้กำหนดเทคนิคิวิธีในการ模ดูเลชันคลื่นวิทยุ โดยสเปรดแฟกเตอร์เป็นปัจจัยสำคัญในการอัตราเร็วของการสื่อสาร lorra ซึ่งสเปรดแฟกเตอร์มีทั้งหมด 6 ค่าดังนี้ SF7 - ถึง SF12 ค่าสเปรดแฟกเตอร์เพิ่มขึ้นหมายถึงระยะเวลาในการส่งข้อมูล (เพิ่ม symbol duration (D_s)) โดย SF7 เป็นการกำหนดให้ส่งข้อมูลได้เร็วที่สุดและ SF12 ส่งข้อมูลได้ช้าที่สุด นอกจากสเปรดแฟกเตอร์มีความสัมพันธ์กับความเร็วในการส่งข้อมูลแล้วขนาด แบบดิวิดิธ (BW) ก็เป็นปัจจัยสำคัญเช่นกัน

BW ใช้กำหนดช่วงความถี่ที่ทำการ模ดูเลชัน ($BW = f_{\max} - f_{\min}$) หากมีแบบดิวิดิธกว้างจะมีผลให้มีการ模ดูเลชันสัญญาณได้จำนวนบิตมาก

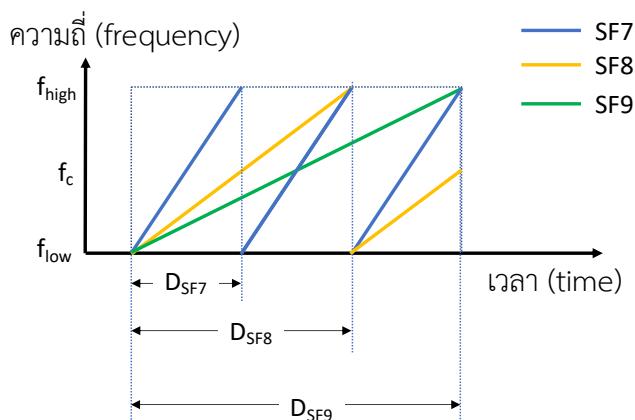
ชิมโบลใช้เรียกสัญญาณหนึ่งชุดที่บรรจุจำนวนบิตได้ตามแต่เทคโนโลยีมี模ดูเลชันนั้นจะทำได้ ยกตัวอย่างการแปลงข้อมูลดังนี้ เทคโนโลยีมี模ดูเลชัน 16-QAM จะสามารถส่งบิตได้ 4-บิตต่อหนึ่งชิมโบล ขณะที่

64-QAM สามารถส่งบิตได้ 6-บิตต่อชิมโบล เรียกจำนวนชิมโบลต่อวินาทีว่า อัตราเร็วชิมโบล แทนด้วย R_s ซึ่ง ประพันตามขนาดแบบดิวิดร์และค่า SF ตามสมการที่ (8.1)

$$R_s = \frac{BW}{2^{SF}} \quad (8.1)$$

จากสมการ (8.1) มีการแบ่งแบบดิวิดร์ออกเป็น 2^{SF} ส่วน ก็ตจากการนอตูเลชันด้วยเทคนิค CSS ทำให้ช่วงเวลาของการกว้างความถี่แบ่งผันกับ SF เมื่อ SF มีค่าน้อยจะทำให้ความชันสูงมีผลต่อจำนวนชิร์ พบรรจุในแบบดิวิดร์มาก ขณะที่ SF ค่ามากจะทำให้ความชันของการกว้างสัญญาณน้อยและส่งผลให้มีจำนวนชิร์พน้อยในแบบดิวิดร์เท่ากัน อธิบายจากรูปที่ 8.5 ที่ SF7(เส้นสีน้ำเงิน) ภายใต้แบบดิวิดร์เท่ากันสามารถส่งสัญญาณได้ 3-chirp และ SF-8 (เส้นสีเหลือง) มี 2-chirp และ SF9 สามารถส่งได้เพียง 1-ชิร์พ เมื่อ SF มีค่าสูงขึ้นจะทำให้อัตราเร็วชิมโบลลดลง ถึงแม้อัตราเร็วลดลงแต่ส่งผลดีให้การสื่อสารทำได้ระยะใกล้ขึ้นซึ่งจะกล่าวถึงในลำดับต่อไป

ขยายความของสัญญาณหนึ่งชิมโบลจะประกอบด้วยการแบ่งส่วนย่อยออกเป็นจำนวน 2^{SF} ส่วน การแบ่งส่วนย่อยนี้



รูปที่ 8.5: ความสัมพันธ์ของสเปรดแฟกเตอร์ กับแบบดิวิดร์ และ symbol duration

ในการออกแบบอัตราเร็วในการส่งแต่ละชิมโบล มีพื้นฐานจากขนาดแบบดิวิดร์ โดยมาตรฐานกำหนดให้อัตราเร็วสัมพันธ์กับแบบดิวิดร์ โดยมีหน่วยเป็นชิป(chip) ซึ่งจำนวนชิปในหนึ่งวินาทีเท่ากับความกว้างของแบบดิวิดร์ เช่น กำหนดให้เครือข่ายใช้แบบดิวิดร์เท่ากับ 125 kHz หมายถึงการมี 125,000 ชิปต่อวินาที เป็นไปตามสมการที่ (8.2)

$$BW = R_c = \text{chip rate} \ (\text{chips/s}) \quad (8.2)$$

ยกตัวอย่างเช่น $BW=125 \text{ kHz}$ หมายถึง $R_c = 125000 \text{ chips/s}$ สมมติกำหนดให้ $SF=9$ ทำให้หนึ่งชิมโบลบรรจุจำนวนบิตได้ $2^{SF} = 2^7 = 128 \text{ chips}$ ดังนั้นจำนวนชิมโบล ที่ส่งได้ในหนึ่งวินาทีเท่ากับ

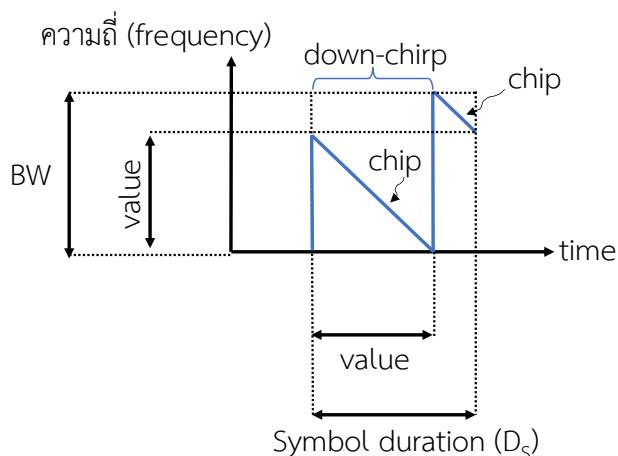
$$R_s = \frac{125000}{128} \\ = 977 \text{ symbols/sec}$$

ค่า chip rate มีค่ามากกว่า symbol rate เสมอ ($R_c > R_s$)

ระยะเวลาการความถี่จากความถี่สูงไปความถี่ต่ำ หรือจากความถี่ต่ำไปความถี่สูงเรียกว่า “symbol duration (D_s)” ซึ่งแบ่งตามค่า SF

การกำหนดความถี่นี้ทำตั้งแต่ความถี่เริ่มต้นไปจนถึงความถี่สูงสุดหรือกล่าวได้ว่ามีระยะห่างเท่ากับความกว้างของแบบดิจิตอล (BW)

แต่ละชิปสามารถแบ่งออกเป็น 2^{SF} ส่วน แต่ละส่วนเรียกว่า “chip” ซึ่งจำนวนชิปขึ้นอยู่กับค่า SF เช่น SF7 มี 128-ชิป ตัวอย่าง down-chirp ในรูปที่ 8.6 ค่าบิตแรกจาก value ซึ่งแบ่งตามจำนวน 2^{SF} ค่า value เป็นการเลื่อนความถี่ จากรูปเห็นได้ว่าหาก BW มีความกว้างมากขึ้นจะทำให้การแบ่งความถี่รองรับความผิดพลาดได้มากขึ้น หรือสามารถเพิ่มจำนวนบิตต่อชิมโบลได้มากขึ้นเช่นกัน และถ้าเพิ่มจำนวนบิตต่อชิมโบลในขนาดแบบดิจิตอลเท่าเดิมจะทำให้ส่งเร็วขึ้นแต่จะเพิ่มโอกาสสัญญาณผิดพลาดได้ง่ายตามมา



รูปที่ 8.6: ตัวอย่างสัญญาณโลรา จำนวน 1 down-chirp

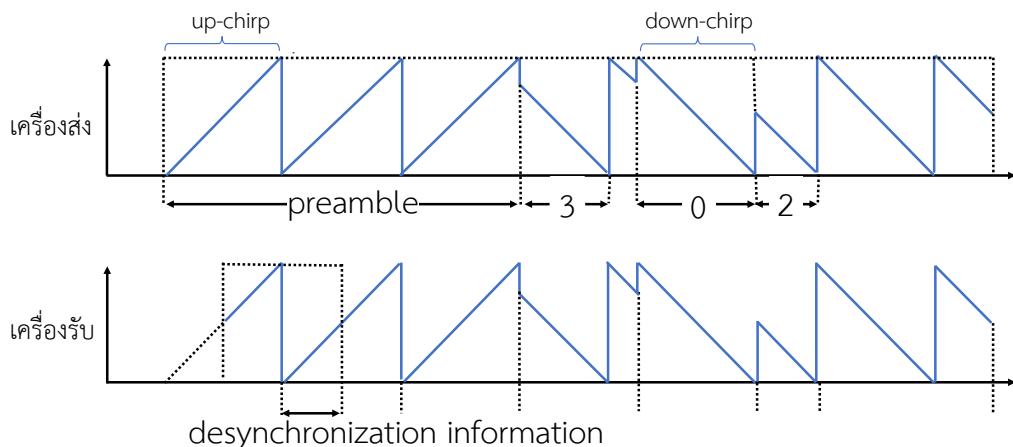
คำนวณอัตราเร็วการสื่อสารได้จากการความสัมพันธ์ของจำนวนบิตที่ส่งได้ต่อหนึ่งวินาที กำหนดให้ R_b แทนอัตราเร็วของการสื่อสาร (Bit rate) เรียกว่า “บิทเรท”

$$R_b = \frac{\text{bit}}{\text{second}} \quad (8.3)$$

การส่งข้อมูลเครือข่ายไร้สายแต่ละครั้งมีความสัมพันธ์กับเทคนิควิธีมอตุเลขัน สัญญาณที่ผ่านการมอดุเลขันเรียกว่าชิมโบล

อธิบายการถอดรหัสสัญญาณตามรูปที่ 8.7 เครื่องส่งเริ่มต้นด้วยการส่ง preamble จากรูปมี preamble เป็น up-chirp จำนวน 3 up-chirp (ปกติมี 8 up-chirp) และถัดไปเป็นการส่งจากเกตเวย์(GW) ไป ED สมมติ

ว่ากำหนดพารามิเตอร์เครือข่ายให้มี $SF=2$ ดังนั้นข้อมูลส่วน down-chirp จะแบ่งได้เป็น $2^{SF} = 2^2 = 4$ บิตต่อชิมโบล ค่าอ้างอิงที่เป็นได้ด้วยในเซ็ต chip $\in \{0, 1, 2, 3\}$ เมื่อผ่านช่วง preamble เป็นส่วน down-chirp สำหรับส่งข้อมูล สังเกตได้ว่า down-chirp จะถูกแบ่งเป็นสัดส่วน 4 ส่วน ข้อมูลส่วนแรกใช้อ้างอิงค่า 3 ส่วนต่อมาเป็นการใช้ความยาวเต็มแบนด์วิดธ์ ถูกเซ็ตค่าเท่ากับ 0 และ



รูปที่ 8.7: ตัวอย่างสัญญาณโลรา จากเครื่องส่งและเครื่องรับ

กำหนดให้ R_b แทนอัตราเร็วในการส่งข้อมูล สำหรับเครือข่ายโลราสามารถคำนวณจากสมการที่ (8.3) เมื่อ จำนวนบิตค้าเป็นความสมพันธ์ของ SF กับ แบนด์วิดธ์ ดังนั้นอัตราเร็วคำนวณได้ตามสมการที่ (8.4) โดยค่า CR ใช้แทน coding rate ที่เหมาะสมกับแต่ละมอดูลเลชัน

$$R_b = SF \times \frac{BW}{2^{SF}} \times \frac{4}{(4 + CR)} \quad (8.4)$$

8.2.3 โลราแวน

เทคโนโลยีลօรากล่าวถึงเทคโนโลยีการส่งคลื่นวิทยุเทียบใน OSI 7 layer ได้กับชั้นกายภาพ มีเป้าหมายให้เครื่องส่งและเครื่องรับสามารถเชื่อมต่อกันได้ แต่มาตรฐานโลรามีโครงสร้างคลุมถึงการเชื่อมต่อหลายเครื่องในกรณีใช้ช่องสัญญาณร่วมกัน

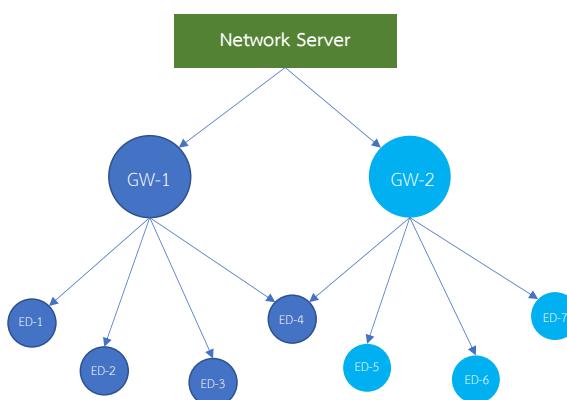
สำหรับมาตรฐานที่ออกแบบให้โลราสามารถเชื่อมกันได้หลายเครื่องมีชื่อว่า โลราแวน ออกแบบมาเพื่อแก้ปัญหาของการใช้ความถี่ร่วมกันเป็นปัญหาเดียวกับการแบ่งใช้สื่อซึ่งพบในการทำงานชั้นลิงค์ หน่วยงานพัฒนาเทคโนโลยีเพื่อให้โลรานำทำงานชั้นลิงค์ หน่วยงาน [LoRa Alliance® \(2022\)](#) มาตรฐานการทำงานชั้นลิงค์เพื่อรับรองรับ multiple access (MA)

โครงข่ายในระบบโลราแวนประกอบด้วย อุปกรณ์ที่ต้องการสื่อสารเรียกว่า “End-device (ED)” อุปกรณ์ทำหน้าที่ผู้ส่งต่อเรียกว่า “Gateway (GW)” การบริหารลูกข่ายจะทำผ่านทางซอฟต์แวร์ที่ใช้ควบคุม GW เรียกว่า “เนตเวิร์กเซิร์ฟเวอร์(Network Server)” ซึ่งการควบคุมการเข้าถึงเครือข่ายในระบบโลราแวนแบ่งตามการทำงานของ ED ออกเป็นคลาสจำนวน 3 คลาสดังนี้

- คลาส A อนุญาตให้ ED สามารถกำหนดช่วงเวลาการส่งได้ตามที่ต้องการ เป็นการสื่อสารได้สองทางโดย การส่งข้อมูลอัปลิงก์จะตามด้วยการสื่อสารดาวน์ลิงก์ สองค์รั้ง กล่าวได้กล่าวการส่งหนึ่งรอบประกอบด้วย หนึ่งอัปลิงก์และสองดาวน์ลิงก์
- คลาส B เป็นการสื่อสารแบบสองทางและมีการส่งสัญญาณเบคอนเพื่อให้ควบคุมจังหวะการส่งข้อมูลให้มีความเรียบง่ายที่ อุปกรณ์ที่ต้องการใช้คลาส B จะเป็นอุปกรณ์ที่ต้องการกำหนดคุณภาพของการส่งข้อมูล
- คลาส C เป็นการสื่อสารแบบสองทาง ที่ต้องการการส่งด้วยความต่อเนื่อง

รูปแบบเฟรมของ ลอร่าเวน

จากมาตรฐาน LoRaWAN 1.0.3 สำหรับคลาส A มีรูปแบบโครงข่ายเป็น Star Topology ดังรูปที่ 8.8 จากรูป GW-1 และ GW-2 จะถูกควบคุมผ่านทาง Network Server โดยอุปกรณ์ ED จะเชื่อมผ่านทาง GW



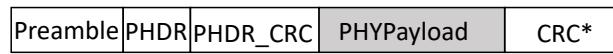
รูปที่ 8.8: โครงข่ายลอร่าเวน เป็นแบบ Star Topology

รูปแบบของเฟรมลอร่าเวน มาตรฐาน 1.0.3 เป็นตามรูปที่ 8.9 เรียงลำดับดังนี้ Preamble ตามด้วย PHDR ตามด้วย PHDR_CRC และตามด้วยส่วนข้อมูล PHYPayload และปิดท้ายด้วย CRC ส่วน PHYPayload เป็นส่วนข้อมูลที่ใช้สำหรับ ลอร่าเวน โดยมีโครงสร้างเฟรมแบ่งได้ 3 แบบได้แก่ MACPayload Join-Request และ Join-Response

PHYPayload ประกอบด้วย MHDR (1-byte) MACPayload (ขนาด 1 ถึง M⁴) และ MAC (4-byte)

MHDR(MAC Header) มีขนาด 1-byte โดย $b_7..b_5$ แทน MType(message type) และ $b_4..b_2$ แทนอาร์ເອີ້ມ່ງ(Reserved for Future Usage) (ไม่ใช้งาน สำรองไว้ใช้ในอนาคต) และ $b_1..b_0$ แทน Major (major version)

⁴M แทนขนาดเฟรม荷载ที่มีขนาดใหญ่ที่สุด



- PHYPayload



หรือ



หรือ



รูปที่ 8.9: โครงข่ายลอราเวน เป็นแบบ Star Topology

การແຍ່ງໃຫ້ຂ່ອງສັນຄູານ

การແຍ່ງໃຫ້ຂ່ອງສັນຄູານໃນຮຽບລອරາແວນ คลາස A ทำงานคล້າຍກັບກາರທຳມານ ອໂລໜາ(ALOHA) ແບ່ງຂ່ອງເວລາ ໃນການສ່ວນສັນຄູານເປັນ 3 ສ່ວນໄດ້ແກ່ Uplink RX1 ແລະ RX2 ເຮີມຈາກ ED ສ່ວນອັປລິງກືໄປຢັ້ງ GW ແລະ Gateway ຈະຕອບສູນອັນໃນໜ່ວງ RX1 ທີ່ມີກຳນົດຈະສ່ວນອັປລິງກືໄປຢັ້ງ ມີອີກຄັ້ງ

บทที่ 9

การออกแบบเครือข่าย

การออกแบบเครือข่ายคอมพิวเตอร์เป็นการวางแผนเครือข่ายเพื่อให้เกิดการเชื่อมต่อทางเครือข่าย และการบริการเครือข่ายมีเพื่อดูแลให้ระบบสามารถทำงานได้ต่อเนื่อง ในการออกแบบและทดสอบการทำงาน เครือข่ายคอมพิวเตอร์สามารถทดสอบกับอุปกรณ์จริง อุปกรณ์ภายในห้องควบคุม จำลองการทำงานทางซอฟต์แวร์ และ จำลองการทำงานผ่านตัวแบบคณิตศาสตร์

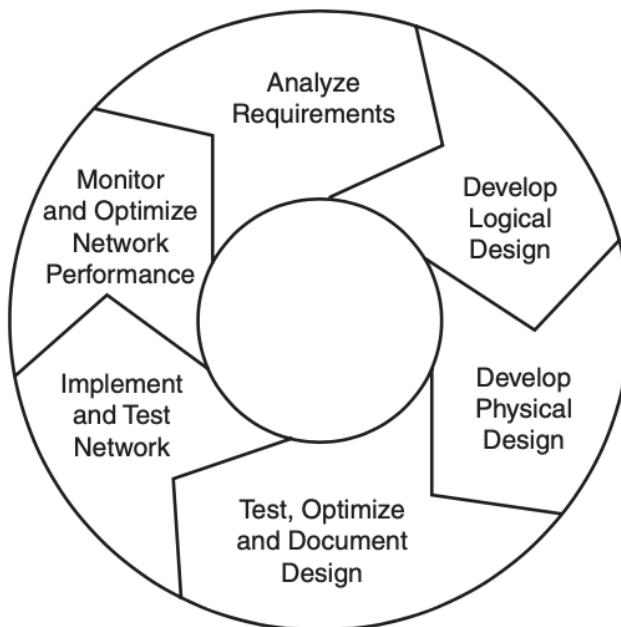
การทดสอบกับอุปกรณ์ในสภาพแวดล้อมจริงทำได้ได้ผลใกล้เคียงกับการทำงานจริงแต่การศึกษาพารามิเตอร์ทำได้ยาก ขณะที่การทดลองในห้องควบคุมสามารถควบคุมพารามิเตอร์ได้แต่ค่าใช้จ่ายสูง ดังนั้นในการทดสอบเครือข่ายส่วนใหญ่จะพึ่งในการจำลองการทำงานทางซอฟต์แวร์หรือผ่านทางการจำลองทางตัวแบบคณิตศาสตร์

9.1 การออกแบบเครือข่าย

บทนี้กล่าวถึงการออกแบบ หลักการออกแบบเป็นการกำหนดวิธีทางใช้งานโดยกำหนดวัตถุประสงค์การใช้งานในการออกแบบแต่ละกระบวนการอาจได้ผลลัพธ์ไม่เหมือนกัน เมื่อลูกค้ามีความต้องการไม่เหมือนกัน วงรอบการออกแบบเครือข่ายสิ่งแรกที่ควรทำได้แก่การทำความเข้าใจความต้องการของลูกค้า (Analyze Requirements) เมื่อเข้าใจความต้องการลูกค้าแล้ว นักออกแบบจะเริ่มออกแบบเครือข่ายทางโลจิคอล(Develop Logical Design) เสร็จแล้วจึงกำหนดรูปแบบของการเชื่อมต่อและระบุพอร์ตที่เกี่ยวข้อง รวมถึงฮาร์ดแวร์และซอฟต์แวร์ที่เลือกใช้ (Develop Physical Design) เปรียบเทียบความคุ้มค่าในการลงทุน การคัดเลือกรูปแบบโครงข่ายที่เหมาะสมและเขียนเอกสารคู่มือที่เกี่ยวข้อง (Test, Optimize and Document Design) เมื่อออกแบบเสร็จแล้ว จึงเข้าขั้นตอนวางแผน (Implement and Test Network) เสร็จสิ้นแล้วขั้นตอนต่อไปเป็นขั้นการบริหารเครือข่ายให้ดำเนินไปอย่างต่อเนื่องเป็นไปตามการออกแบบ(Monitor and Optimize Network Performance) มีลำดับการดำเนินการตามรูปที่ 9.1 จากที่กล่าวมาถือเป็นวิธีในการออกแบบและบริหารเครือข่าย กลับมาที่จุดเริ่มต้นของการออกแบบคือศึกษาความต้องการลูกค้า เพื่อให้ทำความเข้าใจได้แจ่มชัดจะกล่าวถึงพื้นฐานความต้องการของระบบธุรกิจ ซึ่งจะกล่าวถึงในหัวข้อต่อไป

9.2 วิเคราะห์เป้าหมายของการทำธุรกิจและข้อจำกัด

แนวทางการออกแบบเครือข่ายให้ตรงตามความต้องการ จะเริ่มวิเคราะห์จากการใช้งานเป็นหลัก โดยศึกษาแนวทางของธุรกิจมีความเกี่ยวข้องกับเครือข่ายอย่างไร แล้วจึงกำหนดวิธีการใช้งานเครือข่ายไปจนถึงการเลือกใช้อุปกรณ์ที่เหมาะสม เป็นการเสร็จขั้นตอนการออกแบบ แนวทางออกแบบที่ได้รับความนิยมได้แก่การออกแบบโดยใช้แนวคิด Top-Down Network Design Methodology (Priscilla, 2010)



รูปที่ 9.1: การออกแบบเครือข่ายและวิธีการดำเนินการ

การออกแบบเครือข่ายที่ดีควรตอบความต้องการผู้ใช้โดยเข้าใจพอดีรวมของผู้ใช้งานในแต่ละธุรกิจ มีความแตกต่างกัน ความต้องการในการให้บริการเครือข่ายเป้าหมายให้เครือข่ายมีความสามารถ

- ให้บริการได้ต่อเนื่อง(availability)
- สามารถขยายได้ในอนาคต (scalability)
- คุ้มค่า(affordability)
- มีความปลอดภัย(security)
- สามารถบริการจัดการได้ง่าย (manageability)

การลงทุนเพื่อตอบความต้องการทั้งห้าประการอาจไม่เหมาะสมกับบางธุรกิจ มีความสำคัญกับแต่ละธุรกิจแตกต่างกัน เช่น หน่วยงานด้านการทหารให้ความสนใจกับความปลอดภัยมากกว่าการจัดการได้ร่าย หรือภาคธุรกิจให้ความสำคัญกับความสามารถในการขยายได้ในอนาคต เป็นต้น

การใช้แนวคิด top-down วิเคราะห์โดยอ้างอิง OSI model จากชั้นแอปพลิเคชัน ลงไปถึงชั้นกายภาพ

เป้าหมายพื้นฐานของธุรกิจ

หัวข้อนี้กล่าวถึงความต้องการพื้นฐานของการดำเนินธุรกิจ ซึ่งนำไปสู่การออกแบบเครือข่ายให้เหมาะสม ประกอบด้วย

- เพิ่มกำไร (Increase revenue และ profit)

- เพิ่มส่วนแบ่งในการตลาด
- เปิดตลาดใหม่
- เพิ่มความสามารถในการแข่งขัน
- ลดค่าใช้จ่าย
- ให้พนักงานเพิ่มผลผลิต
- ลดระยะเวลาในการผลิต
- ระบบการผลิตแบบทันเวลาพอดี (Just in Time)
- ใช้เวลาเตรียมแผนสั้น
- มีบริการใหม่ให้ลูกค้า
- ให้บริการลูกค้าได้ดี
- สร้างเครือข่ายกับนักลงทุนหลัก
- เลี่ยงการหยุดชะงักจากปัญหาด้านความปลอดภัยทางเครือข่ายคอมพิวเตอร์
- ลดผลกระทบจากการภัยธรรมชาติ
- ปรับตัวทันกับเทคโนโลยีใหม่
- ทำให้ศูนย์ข้อมูลทำงานอย่างมีประสิทธิภาพ เช่น พลังงาน สายนำสัญญาณ ตู้เรซิค หน่วยบันทึกข้อมูล และวงจรสื่อสาร
- มีการดำเนินการตามรูปโถรังร่องไอกทีและบริหารไปร่องใส

ระบุโปรแกรมที่ลูกค้าใช้งานผ่านเครือข่าย

วิธีหนึ่งในการทำความเข้าใจลูกค้าได้แก่ การสำรวจซอฟต์แวร์ที่ลูกค้าใช้งานอยู่เป็นประจำ โดยระบุชื่อโปรแกรมประเภทการทำงานของโปรแกรม เป็นโปรแกรมใหม่หรือไม่ ความสำคัญ และ ความเห็นเพิ่มเติม ผู้ออกแบบระบบ เครือข่ายสามารถเขียนแบบฟอร์มให้ผู้ใช้งานป้อนข้อมูลดังตัวอย่างตารางที่ 9.1

ตารางที่ 9.1: แบบสำรวจโปรแกรม

ชื่อโปรแกรม	ประเภทโปรแกรม	เป็นโปรแกรมใหม่ (ใช่/ไม่)	ความสำคัญ	ความเห็นเพิ่ม
Edge	เว็บเบราว์เซอร์	ใช่	สูง	ใช้เข้าถึงเว็บไซต์

ประเภทโปรแกรม ขึ้นอยู่กับการนำไปใช้งาน รายการต่อไปนี้เป็นตัวอย่างประเภทโปรแกรมที่พบใช้งานทั่วไป

- อีเมล์ (E-mail)
- แลกเปลี่ยนไฟล์ แชร์ไฟล์ และเข้าถึงไฟล์ (File transfer, sharing, and access)
- ฐานข้อมูล (Database access and updating)
- เว็บเบราว์เซอร์ (Web browsing)
- เกมผ่านเครือข่าย (Network game)
- ควบคุมระยะไกล (Remote terminal)
- ปฏิทินออนไลน์ (Calendar)
- รูปทางการแพทย์ (Medical imaging)
- ประชุมวิดีโอทางไกล (Videoconferencing)
- วิดีโอออนดีமานด์ (Video on demand (VoD))
- ระบบบันด์หมายการกระจายภาพวิดีโอ (Scheduled multicast video)
- กล้องวงจรปิด (Surveillance and security camera video)
- โทรศัพท์ผ่านไอพี (Internet or intranet voice (IP telephony))
- อินเทอร์เน็ตและการใช้อิปสำหรับสื่อสาร (Internet or intranet voice (IP telephony))
- ออกออเดอร์ (Sales order entry)
- รายงานการบริหาร (Management reporting)
- ติดตามการขาย (Sales tracking)
- งานใช้คอมพิวเตอร์ออกแบบ (Computer-aided design)
- เอกสารเกี่ยวกับภาพ (Document imaging)
- การควบคุมคลังสินค้า (Inventory control and shipping)
- การวัดและส่งข้อมูลทางไกล (Telemetry)
- ไอวีอาร์ (Interactive Voice Response (IVR))
- การส่งข้อความ (Unified messaging)
- การเผยแพร่เอกสารทางเดสท็อป (Desktop publishing)
- การเผยแพร่เอกสารผ่านเว็บ (Web publishing)
- กระดานอิเล็กทรอนิกส์ (Electronic whiteboard)
- หน้าจอจำลองสำหรับควบคุมระยะไกล (Terminal emulation)
- สมุดโทรศัพท์ (Online directory (phone book))
- เรียนทางไกล (Distance learning)
- พีโอดีส (Point of sales (retail store))
- พานิชย์อิเล็กทรอนิกส์ (Electronic commerce)
- ตัวแบบทางการเงิน (Financial modeling)
- ฝ่ายบริหารทรัพยากรบุคคล (Human resources management)
- การใช้คอมพิวเตอร์ควบคุมการผลิต (Computer-aided manufacturing)
- ควบคุมกระบวนการ (Process control and factory floor)

นอกจากโปรแกรมที่ไปแล้วควรรับโปรแกรมสำหรับระบบเข้ามาด้วยตัวอย่างโปรแกรมสำหรับระบบดังนี้

- ระบบพิสูจน์ตัวจริง (Authentication and authorization)
- โปรแกรมค้นหาชื่อ (Host naming and name resolution)
- โปรแกรมแจกจ่ายไอพี (Dynamic host addressing)
- โปรแกรมบูตระยะไกล (Remote booting)
- โปรแกรมเซตค่าระยะไกล (Remote configuration download)
- ระบบไดเรกทอรี (Directory service)
- ระบบสำรองข้อมูลผ่านเครือข่าย (Network backup)
- ระบบบริหารเครือข่าย (Network management)
- ระบบแจกจ่ายซอฟต์แวร์ (Software distribution)

ความสำคัญของซอฟต์แวร์สามารถจัดลำดับความสำคัญตามระดับได้ โดยทั่วไปกำหนดให้มีสามระดับดังนี้

- สูงสุด (Extremely critical)
- ปานกลาง (Somewhat critical)
- ปกติ (Not critical)

การเมืองภายในบริษัทและข้อกำหนดของบริษัท

การกล่าวถึงประดิษฐ์ด้านการเมืองภายในบริษัทและข้อกำหนดของบริษัทนั้นอาจดูไม่เกี่ยวข้องกับการออกแบบเครือข่ายมากนักแต่เมื่อมอง ลึกลงไปแล้วจะเห็นได้ว่าหากไม่สนใจประเด็นการเมืองภายในบริษัทอาจทำให้โครงการออกแบบเครือข่ายมีความเสี่ยงไม่สำเร็จได้ เช่นมีผลต่อการอนุมัติงบประมาณหรือการกำหนดให้มีหรือไม่มีระบบเครือข่ายภายในบริษัทได้ การทำความเข้าใจประเด็นการเมืองภายในบริษัทถือเป็นเรื่องยากเป็นสิ่งที่ไม่พบในการประชุม หรือพูดกันอย่างตรงไปตรงมา ซึ่งเป็นประเด็นซ่อนเร้นที่ผู้ออกแบบต้องทำความเข้าใจให้ลึกซึ้ง

ให้ความสำคัญกับประเด็นตัวบุคคลที่จะส่งผลกระทบต่อโปรเจค ศึกษาผู้จัดการคนใดที่เป็นคนเสนอโครงการนี้ส่งผลกระทบต่อผู้จัดการฝ่ายอื่นหรือไม่ เมื่อโปรเจคนี้เกิดขึ้นจะเป็นฝ่ายใดได้ประโยชน์และฝ่ายใดเสียประโยชน์แก่การเสียประโยชน์นั้นส่งผลกระทบมากเพียงใดต่อบุคคลและบุคคลนั้นเป็นผู้ที่มีอำนาจการตัดสินใจหรือไม่ การเก็บข้อมูลเหล่านี้ไม่เกี่ยวข้องกับเทคโนโลยี แต่มีผลต่อการดำเนินการของโครงการอย่างมีนัยสำคัญ บริษัทด้วยบุคคลที่มีผลต่อโครงการโดยตรงแล้วให้เข้าสู่กระบวนการทำความเข้าใจและไก่ เกลี่ยข้อพิพาทก่อนที่โครงการจะเริ่ม

งบประมาณและข้อจำกัดด้านบุคคลากร

การออกแบบเครือข่ายที่ดีควรอยู่ภายใต้งบประมาณที่ลูกค้ากำหนด งบประมาณในการออกแบบและติดตั้งควรครอบคลุมถึงการจัดซื้ออุปกรณ์ ลิขสิทธิ์ซอฟต์แวร์ การดูแลรักษา การบริการหลังการขาย การทดสอบ การฝึกอบรม และการจ้างบุคคลากร รวมถึงค่าใช้จ่ายในการให้คำปรึกษา

ปกติแล้วการวิเคราะห์ทักษะความสามารถของบุคคลากรที่มีถือเป็นสิ่งที่ควรทำก่อนเสนอให้มีการจ้างบุคคลากรเพิ่ม ผู้ออกแบบจะประเมินเบื้องต้นเปรียบเทียบระหว่างการฝึกอบรมพนักงานเก่า(เพิ่มภาระงาน) กับการจ้างพนักงานใหม่ มีความเหมาะสมในด้านเวลา ประสิทธิภาพ และความคุ้มค่าของงบประมาณ หากเป็นไปได้ควรวิเคราะห์ return on investment (ROI) ร่วมกับลูกค้า หลังจากใช้งานการออกแบบเครือข่าย เพื่อให้แน่ใจว่าสิ่งที่ลงทุนไปได้ผลลัพธ์ตามความต้องการ

กำหนดขอบเขตการออกแบบ

สิ่งแรกที่ควรทำเมื่อเริ่มโครงการออกแบบเครือข่าย ได้แก่การจำกัดขอบเขตของงาน โดยทั่วไปเครือข่ายสมัยใหม่จะกำหนดให้มีขนาดเล็กก่อน เช่น โครงการออกแบบให้มีพนักงานฝ่ายขายบางคนสามารถเข้าถึงเครือข่ายภายในองค์จากภายนอกสำนักงานผ่านเครือข่าย VPN หรือการออกแบบอีกด้านหนึ่ง สำหรับเครือข่ายที่มีขนาดใหญ่ ผู้ออกแบบจะสอบถามลูกค้าเพื่อทำความเข้าใจความต้องการของลูกค้าสำหรับช่องขาจัดกลุ่มเครือข่ายตามสายงาน ซึ่งมีการวางแผนเครือข่ายแลนและเครือข่ายแวง(wide area networks) รวมถึงการอนุญาตให้เข้าถึงเครือข่ายภายนอกเครือข่ายอินเทอร์เน็ต และการสอบถามถึงความพร้อมของลูกค้าเมื่อมีการใช้งานเครือข่ายในรูปแบบใหม่จากเดิม

การอธิบายลูกค้าในสิ่งที่ควรระวังเมื่อเกิดเหตุและเป็นการปรับความเข้าใจถึงผลกระทบต่อธุรกิจในระหว่างที่ดำเนินการปรับปรุงออกแบบเครือข่ายสิ่งเหล่านี้มีความจำเป็นในการออกแบบเครือข่าย

สอบถามลูกค้าให้แน่ใจว่าได้รับข้อมูลครบถ้วนแล้วที่เกี่ยวข้องกับเครือข่ายและการออกแบบ และให้แน่ใจว่าได้สอบถามเรื่องสำคัญหมวดแล้ว สอบถามความต้องการลูกค้าทราบอีกรอบเพื่อให้เกิดความถูกต้องของข้อมูลเกี่ยวกับ จุดเชื่อมต่อสายสัญญาณและอุปกรณ์เกี่ยวข้องต่างๆที่เคยมีส่งถึงระบบความมั่นคงปลอดภัยที่บริษัทนั้นได้ใช้งานมาก่อนเพื่อให้แน่ใจว่าการออกแบบจะไม่ส่งผลกระทบต่อระบบเดิม

กำหนดระยะเวลาดำเนินงาน

กำหนดกรอบเวลาของการดำเนินการ แบ่งงานออกแบบเป็นส่วนย่อย และกำหนดกรอบเวลาให้แต่ละส่วนย่อย นักออกแบบเครือข่ายควรรายละเอียดการดำเนินการแก่ลูกค้าเป็นระยะตามกรอบเวลาในแต่ละส่วนย่อย การรายงานให้ลูกค้าได้ทราบจะทำให้มีการติดตามและเข้าใจกระบวนการทำงาน ซึ่งบางครั้งอาจมีผลกระทบต่อการทำงานปกติ เช่น เมื่อถึงขั้นตอนการเดินสายไฟฟ้าเพิ่มเติมทำให้ต้องหยุดจ่ายกระแสไฟฟ้า หากผู้ออกแบบได้แจ้งล่วงหน้าตามกรอบเวลาอย่างถูกต้องจะช่วยให้ลูกค้าสามารถบริหารธุรกิจให้ได้รับผลกระทบน้อยลงได้ ดังนั้นการกำหนดกรอบเวลาและกรอบเวลาอย่างถือเป็นเรื่องสำคัญในการทำงานโครงการ

9.3 การวิเคราะห์ด้านเทคนิคและข้อจำกัด

การรองรับการขยายตัว (Scalability)

การรองรับการขยายตัวสนับสนุนอัตราการเติบโตของการใช้งานเครือข่าย บริษัทธุรกิจขนาดใหญ่ถือว่าการออกแบบให้รองรับการขยายตัวเป็นเป้าหมายหลัก การเติบโตของการใช้เครือข่ายมีการขยายตัวอย่างรวดเร็ว หากไม่ออกแบบให้ระบบสามารถรองรับการขยายตัวได้ทัน จะส่งผลต่อการดำเนินธุรกิจได้

สำหรับการวิเคราะห์ อัตราการเติบโตของการใช้งานเครือข่ายจำเป็นต้องได้รับความช่วยเหลือจากลูกค้านักออกแบบระบบควรให้ลูกค้าประเมิน เช่นภายในสองปีข้างหน้าจะมีความต้องการใช้งานเครือข่ายมากน้อยเพียงใดและในอีกห้าปีข้างหน้าบริษัทจะมีแนวโน้มการใช้งานระบบเครือข่ายคอมพิวเตอร์มากขึ้นหรือไม่ซึ่งข้อมูลเหล่านี้จะเป็นประโยชน์ต่อการวางแผนออกแบบเครือข่าย แบบสอบถามที่จะสำรวจประกอบด้วยคำถามต่อไปนี้

- ภายในปีนี้จะขยายสำนักงานกี่แห่ง ? อีกสองปีข้างหน้าจะขยายเพิ่มอีกกี่แห่ง ?
- สำนักงานใหม่ต้องการใช้ทรัพยากรเครือข่ายของบริษัทหรือไม่มากน้อยเพียงใด ?
- ภายในปีนี้จะมีจำนวนเครื่องแม่ข่ายเพิ่มขึ้นเท่าใด ? อีกสองปีหรือมีการขยายกิจการต้องเพิ่มเครื่องข่ายกี่เครื่อง ?

ข้อจำกัดของการขยายตัว เมื่อทราบถึงการเตรียมพร้อมในการขยายตัวของบริษัทของลูกค้าแล้วสิ่งสำคัญสิ่งหนึ่งคือการทำความเข้าใจข้อจำกัดทางเทคโนโลยี ตัวอย่างเช่นลูกค้าต้องการให้เครือข่ายภายในบริษัทสามารถใช้ไฟฟ้าภายในวงเดียวทั่วทุกสาขา ซึ่งเทคโนโลยีและไม่สามารถทำได้ในบางกรณี หรือลูกค้าต้องการใช้งานเครือข่ายความเร็วสูงซึ่งยังไม่ปราฏอยู่ปัจจุบัน และอาจจะต้องรออีกสองปีจึงสามารถนำมาใช้กับบริษัทได้ เป็นต้น

การพร้อมให้บริการ (Availability)

การพร้อมให้บริการมีความเกี่ยวข้องกับเวลาที่เครือข่ายสามารถให้บริการแก่ผู้ใช้งานได้บ่อยครั้งที่เกิดปัญหาด้านเครือข่ายส่งผลกระทบต่อการดำเนินธุรกิจ การวัดความพร้อมให้บริการจะมีหน่วยวัดเป็นเปอร์เซ็นต์ของ การให้บริการ อาจเป็นหนึ่งปี (uptime per year) ต่อเดือน(months) ต่อสัปดาห์(7-days-a-week) ต่อวัน(24-hour) เป็นต้น ตัวอย่าง 165 ชั่วโมง ของเวลาเต็ม 168 ชั่วโมง หมายถึงค่าการพร้อมให้บริการเท่ากับ 98.21 เปอร์เซนต์

ภาคธุรกิจเริ่มกล่าวถึง การภัยคุกคามเมื่อเกิดภัยพิบัติ (Disaster Recovery) ในช่วงระยะเวลาไม่กี่ปีที่ผ่านมาเกิดภัยพิบัติที่ส่งผลกระทบต่อการดำเนินธุรกิจ ตัวอย่างเช่นการเกิดเหตุการณ์คลื่นซินมาในปี ค.ศ.2004 หรือการเกิดเหตุการณ์น้ำท่วมใหญ่ปี ค.ศ.2011 มีหลายธุรกิจได้รับความเสียหาย จึงทำให้เกิดความสนใจในการเตรียมความพร้อมรับกับเหตุการณ์ที่เกิดจากภัยพิบัติทางธรรมชาติ ซึ่งวิธีการดังกล่าวเรียกว่า “Disaster Recovery”

เป็นเรื่องที่ผู้ออกแบบเครือข่ายจะต้องตกลงกับลูกค้าถึงความต้องการในการพร้อมให้บริการ ซึ่งความพร้อมในการให้บริการนั้นมีค่าใช้จ่ายในการออกแบบจัดเตรียมความพร้อมแตกต่างกัน เช่นค่าความพร้อมให้บริการ 99.70% กับ 99.95% หากระบบมีรับการอัพไทม์(uptime) ได้ที่ 99.70% หมายถึงในหนึ่งสัปดาห์สามารถยอมรับให้ระบบหยุดทำงานได้ 30นาที ขณะที่อัพไทม์ 99.95% ยอมให้หนึ่งสัปดาห์หยุดทำงานได้เพียง 5 นาที

โดยปกติบริษัททั่วไปต้องการความพร้อมให้บริการอยู่ในช่วง 99.70% ถึง 99.95% แต่ก็มีบางระบบที่ต้องการความพร้อมให้บริการสูง ถึง 99.999% หรือบางครั้งเรียกว่า “five nines availability” การออกแบบระบบให้ให้รองรับ five nines availability นั้นทำได้ยาก พิจารณาเครือข่ายใช้งาน 24 ชั่วโมงต่อวันตลอด 365 วัน หรือกล่าวได้ว่า 8760 ชั่วโมง ซึ่งเครือข่ายสามารถหยุดให้บริการได้เพียง 0.001% หรือ 5 นาทีต่อปี

การกำหนดค่าความพร้อมให้บริการ พิจารณาได้จากค่าใช้จ่ายที่เสียไปเมื่อเกิดความไม่สงบ(downtime) โดยใช้ค่าความเสียหายนั้นมาพิจารณาในการลงทุนเพื่อเพิ่มค่าความพร้อมในการให้บริการ MTTSR(mean time to service repair) MTBSO(mean time between service outage) การวัดค่าประสิทธิภาพด้านความพร้อมให้บริการสามารถใช้ค่าเฉลี่ยของเวลาอุปกรณ์หยุดทำงาน MTBF(mean time between failure) และ ค่าเฉลี่ยของเวลาในการกู้คืนระบบ MTTR(mean time to repair)

ค่า MTBF มาจากค่าประสิทธิภาพที่ใช้ในอุตสาหกรรมมาก่อน เพื่อใช้ระบุคุณสมบัติคอมพิวเตอร์หรืออุปกรณ์คอมพิวเตอร์จะใช้งานได้นานเท่าใดก่อนอุปกรณ์จะเสีย สามารถนำมาใช้ในการวัดค่าประสิทธิภาพของ การให้บริการเครือข่ายได้เพื่อให้ความหมายของการให้บริการคือเครือข่ายไม่ใช้อุปกรณ์ตามความหมายเดิม หากอธิบายความหมายตรงกับการให้บริการเครือข่ายจะพบการใช้ MTBSO ทดแทน MTBF และใช้ MTTSR แทน MTTR ในที่นี้จะยังคงใช้ MTBF และ MTTR สำหรับอธิบายความพร้อมการให้บริการเครือข่าย

ตามปกติ MTBF เท่ากับ 4000 ชั่วโมง หมายถึงเครือข่ายจะหยุดให้บริการได้ไม่เกิน 4000 ชั่วโมง หรือ 166.67 วัน และในปกติจะมีค่า MTTR เท่ากับ 1 ชั่วโมง หรือกล่าวได้ว่าจะใช้เวลาภัยคืนเสร็จภายใน 1 ชั่วโมง ดังนั้นค่าเฉลี่ยความพร้อมให้บริการเท่ากับ

$$4000/4001 = 99.98 \text{ percent}$$

เป้าหมาย 99.98% เป็นค่าพื้นฐานที่บริษัทส่วนใหญ่ต้องการ

ค่าสมการค่าความพร้อมให้บริการจึงเป็นไปตามสมการที่(9.1)

$$\text{Availability} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} \quad (9.1)$$

ประสิทธิภาพเครือข่าย

รายการค่าประสิทธิภาพที่สำคัญในเครือข่ายมีดังนี้

- ทรัพุต แทนปริมาณบิตที่ส่งได้ภายในหนึ่งวินาที
- Accuracy

- Efficiency
- Delay และ Delay Variation
- Response Time
- Security
- Manageability
- Usability
- Adaptability
- Affordability

ตัวอย่างแบบฟอร์มสำรวจด้านเทคนิคสำหรับการใช้งานแอปพลิเคชันเป็นไปตามตารางที่ 9.2

ตารางที่ 9.2: ตัวอย่างแบบฟอร์มสำรวจด้านเทคนิคของการใช้งานแอปพลิเคชัน

ชื่อโปรแกรม	ประเภทโปรแกรม	โปรแกรมใหม่ (ใช่/ไม่)	ความสำคัญ	ค่าใช้จ่ายดาวน์โหลด	ค่า MTBF ที่ยอมรับได้

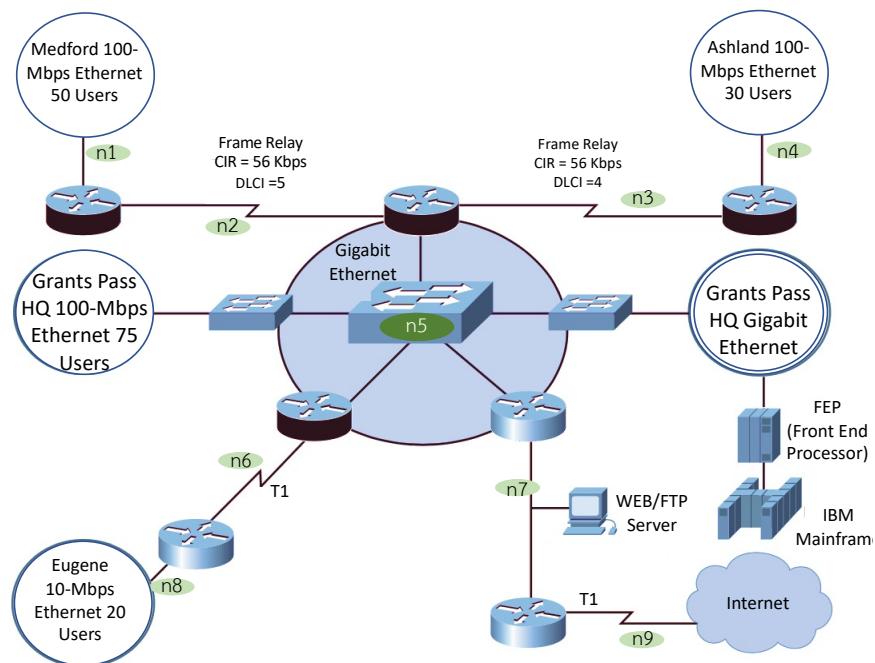
9.4 ทำความเข้าใจเครือข่ายเดิมของลูกค้า

ในการออกแบบเครือข่ายตามปกติแล้วผู้ออกแบบไม่ได้เริ่มนั้นใหม่ทุกครั้ง มีการศึกษาเครือข่ายเดิมของลูกค้า พฤติกรรมการทำงานและความต้องการที่ต้องการปรับปรุงแก้ไขให้ตรงจุด พวกแบบจะตรวจสอบและวิเคราะห์ ถึงจุดสำคัญที่ทำให้เครือข่ายขาดประสิทธิภาพ เช่นปัญหาความชัดของลิงก์

9.5 Physical Network Design

การอธิบายการเชื่อมต่ออุปกรณ์หลักหรือเครื่องคอมพิวเตอร์ต่างๆ ระหว่างกันนั้นผู้ออกแบบเครือข่ายนิยมใช้ ภาพการเชื่อมต่ออุปกรณ์ต่างๆ ซึ่งเรียกว่า Logical Network Diagram ตัวอย่างตามรูปที่ 9.2 ซึ่งประกอบด้วย อุปกรณ์สำคัญได้แก่ สวิตช์ และ เร้าเตอร์ โดยอุปกรณ์อาจเชื่อมต่อกันผ่านสายนำสัญญาณหรือทางไร้สายก็ได้ จากรูปไม่มีวดต่ออุปกรณ์ปลายทาง(ไฮสต์) แต่ใช้รูปวงกลมแทนกลุ่มเครือข่าย โดยกำหนดจำนวนผู้ใช้ในแต่ละ เครือข่าย จำนวนผู้ใช้เป็นประโยชน์ต่อการกำหนด subnet network

ซึ่งเห็นได้ว่าโครงสร้างนี้มีสวิตช์จำนวน 3 เครื่องและเร้าเตอร์จำนวน 7 เครื่อง มีเครือข่ายอยู่ทั้งหมด เท่ากับจำนวนลิงก์ที่เชื่อมกับเร้าเตอร์ จำนวน 9 เครือข่ายย่อย สังเกตที่วงรี n1 ถึง n9 สังเกตได้ว่าสายสัญญาณ ที่เชื่อมกับสวิตช์คือเป็นการเชื่อมลิงก์กันโดยตรง ดังนั้นเครือข่าย Grants Pass HQ 100-Mbps Ethernet 75



รูปที่ 9.2: เนตเวิร์กโดยรวมของผู้ประกอบการด้านผลิตชิ้นส่วนอิเล็กทรอนิกส์

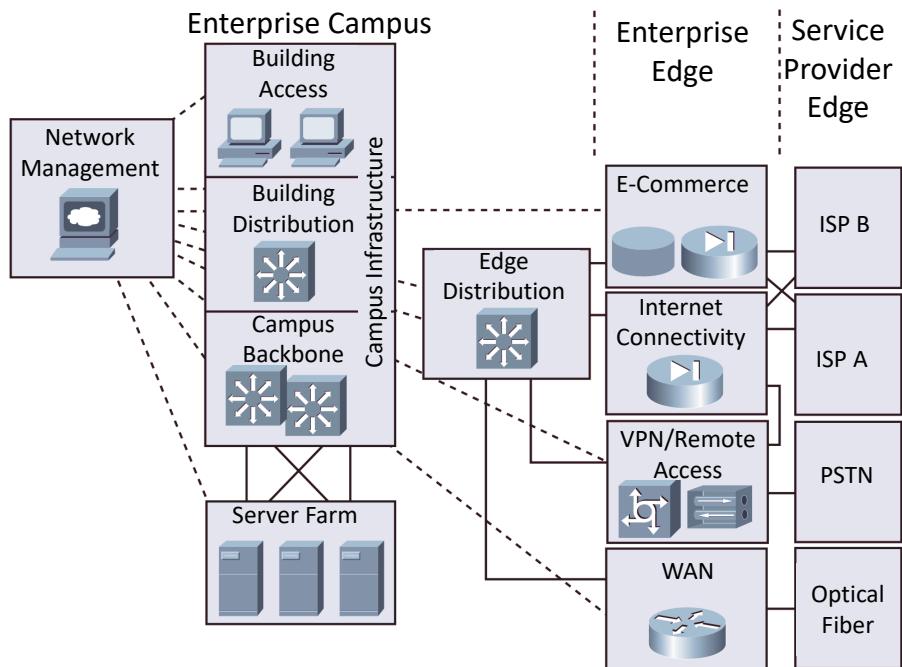
Users ถือเป็นการเชื่อมกันโดยตรง(เครือข่ายวงเดียวgan)กับ Grants Pass HQ Gigabit Ethernet ภาพเครือข่ายแบบนี้ให้ความสำคัญกับการเชื่อมต่อแบบโลจิคัล ระหว่างอุปกรณ์และเชื่อมกับเครือข่าย ซึ่งเป็นประโยชน์ต่อการซื้อขาย

รูปต่อมาในรูปที่ 9.3 เป็นรูปโครงข่ายแบบนามธรรม โดยจัดกลุ่มตามรูปแบบการใช้งาน รูปลักษณะนี้ เป็นประโยชน์ต่อการกำหนดนโยบายการใช้งาน จากรูปเห็นได้ว่ามีการแบ่งเป็นส่วนงาน และมีการเชื่อมโยงความเกี่ยวข้องกันซึ่งไม่เกี่ยวข้องกับการเชื่อมโยงด้วยสายสัญญาณจริง

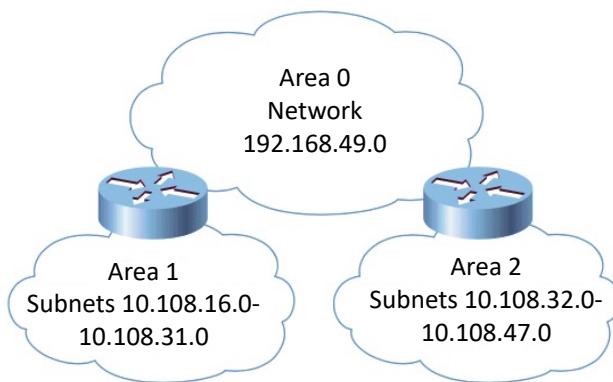
รูปต่อมาในรูปที่ 9.4 ใช้อิบายภาพใหญ่ของเครือข่ายโดยให้ความสำคัญกับ IP address ซึ่งเป็นตัวจัดกลุ่มโครงข่าย จากรูปได้แบ่งชั้นเน็ตเป็นสามกลุ่มใหญ่ได้แก่ 192.168.49.0 10.108.16.0-10.108.31.0 และ 10.108.32.0-10.108.47.0 และมีเร้าเตอร์จำนวนสองเครื่องทำหน้าที่เลือกเส้นทางทั้งสามเครือข่าย

รูปที่ 9.5 เป็นรูปสำหรับอธิบายการติดตั้งจริง ซึ่งให้ความสำคัญกับตำแหน่งการติดตั้งทางกายภาพเห็นได้ว่ามีการกำหนดจุดติดตั้งตั้งแต่บนชั้นของอาคารจริง และสายเชื่อมโยงอ้างถึงสายสัญญาณจริงไปยังแต่ละชั้น เช่น ใจคุณสมบัติสายนำสัญญาณ เพื่อช่วยให้การออกแบบสามารถตอบเบื้องมากยการขยายตัว และการรักษาความพร้อมให้บริการ ได้มีความจำเป็นต้องรู้คุณสมบัติทางกายภาพของสายนำสัญญาณซึ่ง สายนำสัญญาณที่เชื่อมระหว่างชั้นของอาคาร เป็นได้หลายเทคโนโลยี ยกตัวอย่างเช่น

- ใยแก้วนำแสงประเภทซิงเกิล莫ดูล(Single-mode fiber)
- ใยแก้วนำแสงประเภทมัลติ莫ดูล(Multimode fiber)
- สายทองแดงคู่ตีเกลียวแบบมีชีลด์(Shielded twisted-pair (STP) copper)
- สายทองแดงคู่ตีเกลียวแบบไม่มีชีลด์(Unshielded twisted-pair (UTP) copper)



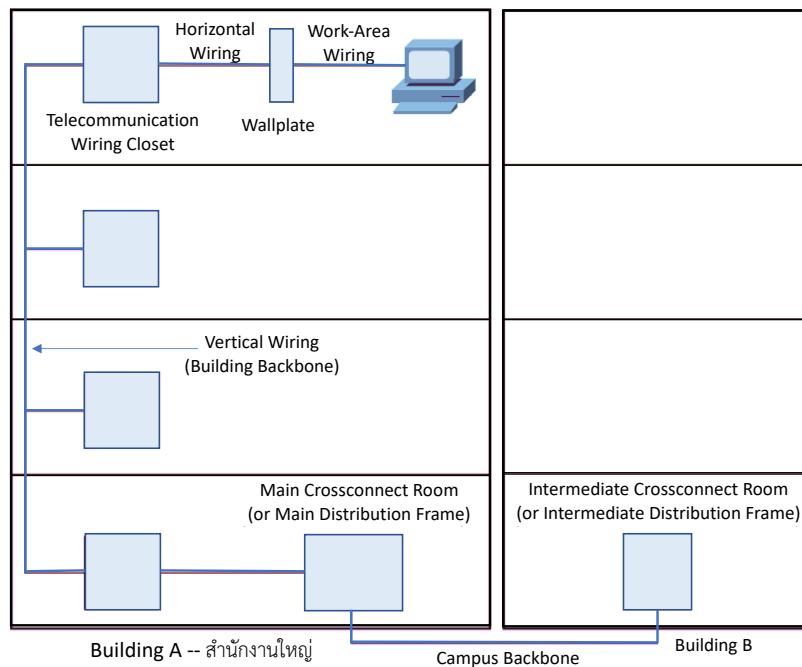
รูปที่ 9.3: ภาพเนตเวิร์กโดยรวมแบบ Modular Block Diagram



รูปที่ 9.4: การแบ่งชั้บเน็ต

- สายตัวภายนอกเชือกเชิง(Coaxial cable)
- ไมโครเวฟ(Microwave)
- เลเซอร์(Laser)
- คลื่นวิทยุ(Radio)
- อินฟราเรด(Infrared)

คุณสมบัติทางกายภาพของสายนำสัญญาณมีข้อจำกัด และมีค่าใช้จ่ายในการติดตั้งแตกต่างกัน สายนำสัญญาณใช้แก้วนำแสงประเภทซิงเกิลโหมดสามารถส่งสัญญาณได้ไกลกว่าไนโตรเจนและประกายมัลติโหมด และ



รูปที่ 9.5: ไดอะแกรมการเชื่อมสายนำสัญญาณ

เมื่อเทียบราคางานสายต่อเมตรไร้แก้วน้ำแสงประเภทชิงเกล่์โลหดมีราคาถูกกว่าสายนำสัญญาณประเภทไฟแก้วน้ำแสงประเภทมัลติโลหด แต่สายนำสัญญาณไฟแก้วน้ำแสงประเภทชิงเกล่์โลหดต้องการความเที่ยงตรงสูงกว่าไฟแก้วน้ำแสงประเภทมัลติโลหดทำให้ราคาอุปกรณ์สูงกว่าไฟแก้วน้ำแสงประเภทมัลติโลหด ในอดีตภายในอาคารนิยมใช้มีเชื่อมต่อด้วยสายนำสัญญาณประเภทไฟแก้วน้ำแสงประเภทมัลติโลหด อย่างไรก็ตามปัจจุบันเทคโนโลยีการผลิตค่าก้าวไปมากทำให้พัฒนาระบบใช้สายนำสัญญาณทั้งระบบเป็นไฟแก้วน้ำแสงประเภทชิงเกล่์โลหด ซึ่งการติดตั้งระบบเดียวกับไฟดูแลอุปกรณ์ภายนอกหลังติดตั้งทำได้ง่าย

สายสายทองแดงคู่ตีเกลียวแบบมีชีล์ด์ใช้กับเครือข่ายที่ต้องการแบบดิวิดิชั่นสูงแต่ราคาต่อเมตรสูงกว่าสายสายทองแดงคู่ตีเกลียวแบบไม่มีชีล์ด์ จะเห็นการใช้งานสายทองแดงคู่ตีเกลียวแบบมีชีล์ด์ ภายใต้พื้นที่จำกัดขณะที่สายสายทองแดงคู่ตีเกลียวแบบไม่มีชีล์ด์ มีราคาถูกกว่าจึงนิยมใช้เป็นสายนำสัญญาณเครือข่ายอีเทอร์เน็ต

สายนำสัญญาณประเภท สายสัญญาณแบบโคงอกเชียล เป็นสายนำสัญญาณที่มีแบบดิวิดิชั่นกว้างนิยมใช้นำสัญญาณประเภทเคเบิลทวีและสายสัญญาณดาวเทียม มีราคาสูงกว่า สายทองแดงคู่ตีเกลียวแบบมีชีล์ด์ และ สายทองแดงคู่ตีเกลียวแบบไม่มีชีล์ด์

การส่งสัญญาณไมโครเวฟเป็นการส่งสัญญาณผ่านคลื่นแม่เหล็กไฟฟ้าซึ่งไม่ต้องการตัวนำสัญญาณ หรือกล่าวได้ว่าเป็นสัญญาณไร้สาย ทำให้สามารถติดตั้งได้ทุกบริเวณโดยไม่ถูกจำกัดทางกายภาพ แต่การใช้งานแม่เหล็กไฟฟ้าต้องเข้าใจปัญหาที่เกิดกับคลื่นแม่เหล็กไฟฟ้าซึ่งมีมากกว่าการเชื่อมต่อโดยใช้สายสัญญาณโดยตรงยกตัวอย่างเช่น การลดตอนของแม่เหล็กไฟฟ้ามีค่าสูงกว่าสายนำสัญญาณอยู่มาก การควบคุมทิศทางของสัญญาณและการใช้ความถี่เดียวกันทำได้ยาก

การใช้ไวริสส์สัญญาณ เลเซอร์ นิยมใช้ในระยะใกล้ซึ่งเลเซอร์มีคุณสมบัติเป็นทั้งแสงและคลื่นแม่เหล็ก-ไฟฟ้า ดังนั้นสามารถส่งสัญญาณผ่านตัวนำสัญญาณ เช่น ไก้วันนำแสง หรือไม่ใช้ตัวนำสัญญาณก็ได้ แต่เมื่อส่งแบบไม่ใช้ตัวนำสัญญาณจะเกิดการลดthon สัญญาณระดับเร็วและไม่อาจผ่านส่งผ่านสิ่งกีดขวางที่พื้นเส้นได้ ดังนั้นในการสื่อสารด้วยเลเซอร์ควรเป็นการสื่อสารแบบสัญญาณแบบตรงมองเห็นกันโดยตรง(Line of sight)

การส่งด้วยคลื่นวิทยุมีความถี่ต่ำกว่าไมโครเวฟ ทำให้นิยมใช้ส่งสัญญาณแอนะล็อก เช่นระบบ AM (Amplitude Modulation) FM(Frequency Modulation) และการสื่อสารที่มีอัตราเร็วต่ำ ข้อดีของการส่งคลื่นวิทยุคือมีความยาวคลื่นมากกว่าไมโครเวฟทำให้ส่งได้ระยะทางไกลๆ ถูกกรอบกว้างได้มาก

อินฟราเรด พบรูปในมาตรฐานการควบคุมอุปกรณ์อิเล็กทรอนิกส์ เช่นรีโมททีวี รีโมทแอร์ เป็นต้น การส่งสัญญาณด้วยอินฟราเรดมีระยะทางใกล้ (ไม่เกิน 10 เมตร) และไม่มีสิ่งกีดขวาง

นอกจากคุณสมบัติต้านราคาแล้ว ข้อจำกัดของสายนำสัญญาณยังถูกกำหนดในการออกแบบ และติดตั้งด้วยเช่นกัน ยกตัวอย่าง เช่น สายนำสัญญาณประเภทสายทองแดงคู่ตีเกลียวแบบไม่มีชีล์ด์ ถูกกรอบกว้างได้ง่ายเมื่อติดตั้งใกล้สายนำไฟฟ้า ดังนั้นในการติดตั้งจะแยกกลุ่มของสายนำสัญญาณออกจากสายนำไฟฟ้า เป็นต้น

9.6 Logical Network Design

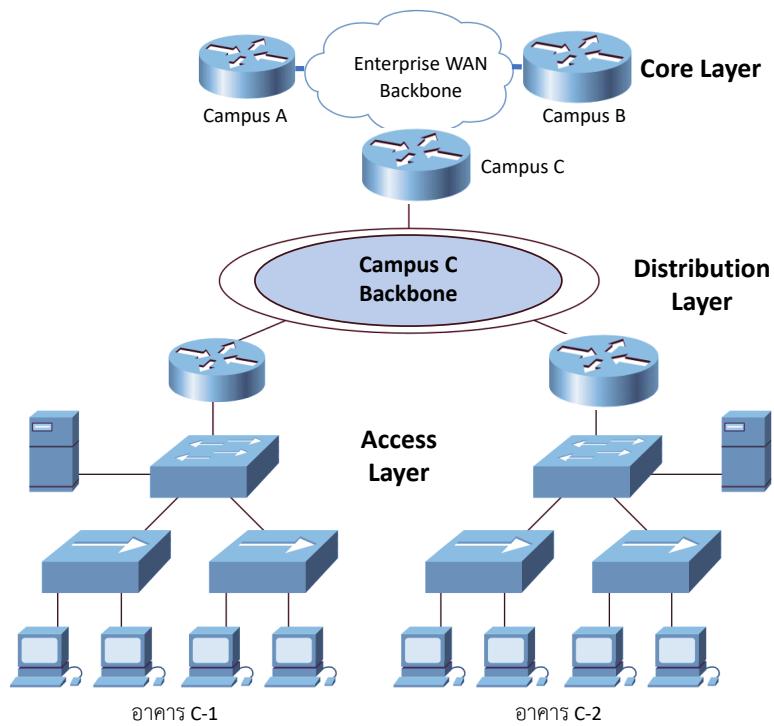
ในการออกแบบ Logical Network นิยมแบ่งกลุ่มของการออกแบบเป็นสามกลุ่มได้แก่

1. คอร์เลเยอร์(Core Layer)
2. ดิสทริบิวชันเลเยอร์(Distribution Layer)
3. แอคเซสเลเยอร์(Access Layer)

การแบ่งลำดับออกแบบเป็นสามกลุ่มนี้ อาจเรียกได้ว่าเป็นวิธีการจัดกลุ่มแบบ “Hierarchical Network Design” โดยกลุ่มนี้คือ คอร์เลเยอร์ มีที่ตั้งอยู่ห่างกันและเชื่อมต่อ กันผ่านโครงข่าย “Enterprise WAN Backbone” ซึ่งอาจจะเป็นไอโอเอสพี หรือเป็นการเช่าช่องสัญญาณก็ได้ ตัวอย่าง Hierarchical Network Design เป็นตามในรูปที่ 9.6 เป็น Network Logical Diagram แบ่งเครือข่ายเป็น 3 กลุ่ม กลุ่มที่มีความเร็วการสื่อสารต่ำสุดคือ คอร์เลเยอร์ ในกลุ่ม คอร์เลเยอร์ จะไม่มีความซับซ้อน ซึ่งขั้นนี้เป็นการเชื่อมต่อของเร้าเตอร์สามเครื่อง เรียกทั้งสามเร้าเตอร์ว่า “คอร์ร้าเตอร์” หน้าที่เชื่อมต่อระหว่าง Campus-A Campus-B และ Campus-C เข้าด้วยกัน เครือข่ายที่เชื่อมกันบน คอร์เลเยอร์

9.7 การออกแบบเส้นทางสำรองทางเครือข่าย

แนวคิดการออกแบบเส้นทางสำรองแบ่งได้สองประเภทตามการใช้งานได้แก่ ทางสำรองในกรณีเส้นทางหลักใช้การไม่ได้ (Backup Path) และประเภทที่สองคือ ทางสำรองในกรณีต้องการแบ่งเบาปริมาณเครือข่าย (Load Sharing)



รูปที่ 9.6: Hierarchical Network Design

ทางสำรองในกรณีเส้นทางหลักใช้การไม่ได้ โดยออกแบบให้มีอุปกรณ์และลิงก์ที่ทำงานหน้าที่เชื่อมต่อ เป็นโครงข่ายหลัก และมีโครงข่ายสำรองทำงานเฉพาะกรณีที่โครงข่ายหลักหยุดทำงาน โดยหลักการนี้สามารถ ประยุกต์ใช้ในเลเยอร์ OSI-7 ได้ เช่นการกำหนดเครื่องฐานข้อมูลหลักและฐานข้อมูลสำรอง สำหรับการกำหนด เส้นทางหลักและเส้นทางสำรองอาจจำเป็นต้องวางแผนด้านการเชื่อมต่อทางกายภาพ เช่นเส้นทางการเชื่อมต่อ ของเส้นทางหลัก กับเส้นทางสำรองควรอยู่คนละเส้นทาง โดยมีความเสี่ยงในการเสียหายไม่เกี่ยวข้องกัน หลัก การเส้นทางสำรองที่ได้ยินบ่อยครั้งได้แก่การออกแบบเครือข่ายให้รองรับ การกู้คืนเมื่อเกิดภัยพิบัติ โดยการ วางให้ศูนย์ข้อมูลหลักและศูนย์ข้อมูลสำรองอยู่ห่างไกลกัน เช่น คนละจังหวัด หรือมีที่ตั้งอยู่ประเทศแตกต่าง กัน ทางสำรองในกรณีต้องการแบ่งเบาปริมาณเครือข่าย เป็นการออกแบบเพื่อแบ่งเบาการทำงานซึ่งเป็นการ ขยายความสามารถของเครือข่ายให้รองรับปริมาณข้อมูลได้เพิ่มขึ้น การออกแบบลักษณะนี้ช่วยให้สามารถ รองรับการขยายตัวและการรักษาความพร้อมให้บริการได้พร้อมกัน

បរណានុក្រម

- Aquegg commonswiki. Pulse-code modulation - Wikipedia. https://en.wikipedia.org/wiki/Pulse-code_modulation#/media/File:Pcm.svg, March 2014. (Accessed on 04/27/2022).
- J. Bang-Jensen และ G. Z. Gutin. *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.
- N. Borisov, I. Goldberg, และ D. Wagner. Intercepting mobile communications: The insecurity of 802.11. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 180–189, 2001.
- P. R. Center. <https://www.pewresearch.org/internet/2021/09/01/the-internet-and-the-pandemic/>, Sep 2021.
- E. S.-R. Committee และ គណន៍. Radio equipment and systems: Hiperlan type 1, functional specifications ets 300-652. *ETSI, June*, 1996.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, และ C. Stein. Introduction to algorithms second edition. *The Knuth-Morris-Pratt Algorithm*, 2001.
- J. Daemen และ V. Rijmen. Aes proposal: Rijndael. 1999.
- W. Diffie และ M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. doi: 10.1109/TIT.1976.1055638.
- B. A. Forouzan. *Data Communications and Networking Global Edition 5e*. McGraw Hill, 2012.
- R. W. Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.
- P. Hyman. In honor of Alan Turing. *Communications of the ACM*, 55(9):20–23, 2012.
- IEEE Computer Society LAN/MAN Standards Committee. IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks—Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pages 1–4379, 2021. doi: 10.1109/IEEESTD.2021.9363693.
- V. Jacobson. Congestion avoidance and control. In *Symposium Proceedings on Communications Architectures and Protocols*, SIGCOMM '88, page 314–329, New York, NY,

- USA, 1988. Association for Computing Machinery. ISBN 0897912799. doi: 10.1145/52324.52356. URL <https://doi.org/10.1145/52324.52356>.
- V. Jacobson. Congestion avoidance and control. In *Symposium Proceedings on Communications Architectures and Protocols, SIGCOMM '88*, page 314–329, New York, NY, USA, 1988. Association for Computing Machinery. ISBN 0897912799. doi: 10.1145/52324.52356. URL <https://doi.org/10.1145/52324.52356>.
- R. Jain, A. Durresi, และ G. Babic. Throughput fairness index: An explanation. In *ATM Forum contribution*, volume 99, 1999.
- M. J. Karels และคณะ. Congestion avoidance and control 3. *ACM Computer Communication Review*, 18(4):314–329, 1988.
- L. Kleinrock. An early history of the internet [history of communications]. *IEEE Communications Magazine*, 48(8):26–36, 2010. doi: 10.1109/MCOM.2010.5534584.
- M. Knight และ B. Seeber. Decoding lora: Realizing a modern lpwan with sdr. *Proceedings of the GNU Radio Conference*, 1(1), 2016. URL <https://pubs.gnuradio.org/index.php/grcon/article/view/8>.
- B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, และ S. Wolff. A brief history of the internet. *ACM SIGCOMM Computer Communication Review*, 39(5):22–31, 2009.
- LoRa Alliance®. Homepage - lora alliance®. <https://lora-alliance.org/>, April 2022. (Accessed on 04/16/2022).
- J. C. Maxwell. *The Scientific Papers of James Clerk Maxwell...*, volume 2. University Press, 1890.
- R. Perlman. An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN. *ACM SIGCOMM Computer Communication Review*, 15(4):44–53, 1985.
- O. Priscilla. Top-down network design, 2010.
- I. S. Reed และ G. Solomon. Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, 8(2):300–304, 1960.
- revspace.nl. Decodinglora - revspace. <https://revspace.nl/DecodingLora>, May 2020. (Accessed on 04/16/2022).

- RFC2001. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. RFC 2001, Jan. 1997. URL <https://www.rfc-editor.org/info/rfc2001>.
- RFC3168. Rfc3168: The addition of explicit congestion notification (ecn) to ip, 2001.
- RFC768. User Datagram Protocol. RFC 768, Aug. 1980. URL <https://www.rfc-editor.org/info/rfc768>.
- RFC792. Internet Control Message Protocol. RFC 792, Sept. 1981. URL <https://www.rfc-editor.org/info/rfc792>.
- RFC793. Transmission Control Protocol. RFC 793, Sept. 1981. URL <https://www.rfc-editor.org/info/rfc793>.
- RFC822. STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES. RFC 822, Aug. 1982. URL <https://www.rfc-editor.org/info/rfc822>.
- RFC896. Congestion Control in IP/TCP Internetworks. RFC 896, Jan. 1984. URL <https://www.rfc-editor.org/info/rfc896>.
- R. L. Rivest, A. Shamir, และ L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, feb 1978. ISSN 0001-0782. doi: 10.1145/359340.359342. URL <https://doi.org/10.1145/359340.359342>.
- S. Safaric และ K. Malaric. Zigbee wireless standard. In *Proceedings ELMAR 2006*, pages 259–262. IEEE, 2006.
- คณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ. หลักเกณฑ์การอนุญาตให้ใช้คลื่นความถี่ย่าน ๘๒๐ - ๘๒๕ เมกะเฮิรตซ์. <http://www.ratchakitcha.soc.go.th/DATA/PDF/2560/E/289/47.PDF>, Nov 2017. (Accessed on 04/18/2022).
- สมาคมโทรคมนาคมแห่งประเทศไทยฯ. Powerpoint presentation. <http://www.tct.or.th/images/article/member/25601214/IoT-in-920-925-MHz-Basic-Info.pdf>, Jan 2017. (Accessed on 04/18/2022).
- สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติ. คู่มืออบรมและสอบเพื่อรับประกาศนียบัตร พนักงานวิทยุสมัครเล่นขั้นต้น. NBTC, December 2017.

ຕັ້ງນີ້

- 802.11i, 168
- ad-hoc, 56
- address translation protocol, 95
- ARP, 95
- Availability, 209
- Backup Path, 215
- bluetooth, 192
- broadcast, 80
- broadcast storm, 78
- CA, 155
- CDN, 180
- Certificate Authority, 155
- cipher, 146
- circuit switching, 61
- count to infinity problem, 101
- CRC, 41
- datagram, 66
- demultiplex, 22
- DHCP, 95
- Diffie-Hellman Key Exchange, 157
- DNS, 184
- Dynamic Host Configuration Protocol, 95
- e-mail, 173
- encoding, 36
- error detection, 39
- ethernet, 52
- FCFS, 137
- fifo, 137
- hash, 147
- hop-by-hop flow control, 72
- HTTP, 176, 177
- hub, 65
- ICMP, 96
- IMAP, 174, 176
- internet protocol, 81
- Introduction to computer networks, 14
- ip, 81
- IPv4, 84
- IPv6, 107
- Jacobson/Karels Algorithm, 125
- Karn/Partridge Algorithm, 125
- KDC, 156
- kerberos, 163
- Key Distribution Center, 156
- layer, 21
- Load Sharing, 216
- Logical Network Design, 215
- LoRa, 194
- LoRaWAN, 200
- M/M/1, 135
- Markovian, 135
- mesh, 56
- MIME, 173
- mpls, 97
- MTBF, 210
- MTTR, 210
- multi-protocol label switching, 97
- multicast, 80
- multiplex, 22

- Nagle's Algorithm, 124
- netmask, 90
- network design, 203
- Network layer, 64
- network security, 145
- OSI 7-layer, 24
- OSPF, 104
- packet switching, 17
- PGP, 165
- physical layer, 30
- Physical Network Design, 211
- poison reverse, 102
- poisson, 135
- POP, 176
- Power of the network, 135
- Pre-Shared Key, 169
- Pretty Good Privacy, 165
- rip, 104
- routing, 98
- Routing Information Protocol, 104
- Scalability, 209
- service model, 83
- Silly Window Syndrome, 123
- SMTP, 174
- source routing, 72
- spanning tree, 76
- split horizon, 101
- stateful firewall, 171
- tail drop, 137
- TCP, 114
- threats, 146
- thrust, 146
- time to live, 84
- Triggering Transmission, 123
- TTL, 84
- UDP, 113
- Uniform Resource Identifiers, 181
- URI, 181
- VC, 68
- virtual circuit switching, 68
- VLAN, 80
- VLAN ID, 80
- wavelength, 34
- WEP, 168
- Wi-Fi, 57
- Wired Equivalent Privacy, 168
- wireless lan, 56
- World Wide Web, 176
- WPA, 168
- WWW, 176
- XML, 181
- การค้นหาเส้นทางแบบบอร์ดคาส, 104
- การค้นหาเส้นทางแบบมัลติคาส, 104
- การจัดเส้นทางแบบพลวัต, 104
- การตรวจสอบความผิดพลาด, 41
- การถอดสัญญาณ, 22
- การพร้อมให้บริการ, 209
- การรวมสัญญาณ, 22
- การรองรับการขยายตัว, 209
- การส่งข้อมูลแบบวงจร, 61
- การส่งข้อมูลแบบแพคเกจ, 37
- การส่งข้อมูลแบบแพ็กเก็ต, 17
- การออกแบบเครือข่าย, 90, 203
- การออกแบบเส้นทางสำรองทางเครือข่าย, 215
- กำลังงานทางเครือข่าย, 135
- ขั้นตอนวิธีเลือกเส้นทาง, 98

- ความปลอดภัยทางเครือข่าย, 145
 ความผิดพลาดระดับบิต, 39
 ชั้นกายภาพ, 30
 ชั้นการซื่ออมต่อ, 30
 ชั้นขนส่ง, 112
 ชั้นเครือข่าย, 64
 ชั้นโปรแกรมประยุกต์, 172
 ซีอาร์ซี, 41
 ดัชนีของเงิน, 136
 ดีอี็นเอส, 184
 ต้นไม้แบบทอดข้าม, 76
 ทางสำรองในกรณีต้องการแบ่งเบาปริมาณเครือข่าย, 216
 ทีซีพี, 114
 บรรดคาสต์, 80
 บลูทูธ, 192
 ปัญหาการนับไม่รู้จบ, 101
 ป้าชง, 135
 มัลติคาส, 80
 มาส, 90
 ยูดีพี, 113
 ระบบเครือข่ายคอมพิวเตอร์เบื้องต้น, 14
 โลรา, 194
 โลราเวน, 200
 วีแลน, 80
 สวิตซ์, 65
 อีเทอร์เน็ต, 52
 อีเมล์, 173
 อัป, 65
 เครือข่ายไร้สาย, 56
 เครือข่ายไร้สายประเภทประยุกต์พลังงาน, 192
 เดต้าแกรม, 66
 เนตเวิร์กสวิตซ์, 65
 เน็ตมาส, 90
 เลือกเส้นทางที่สั้นที่สุด, 99
 เลือกเส้นทางแบบระยะทางเวกเตอร์, 99
 เลเยอร์ริง, 21
 เอชทีทีพี, 176
 เอสเอ็มทีพี, 174
 แบบจำลองโอลีสไอ, 24
 แลนไร้สาย, 56, 57
 ไอพี, 84
 ไอพีรุ่นที่ ๔, 84
 ไอพีรุ่นที่ ๖, 107

