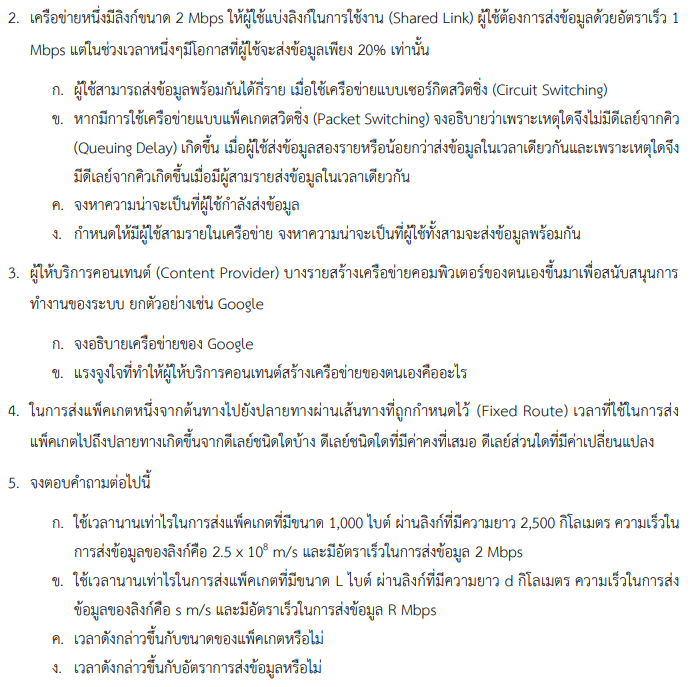
# ในหนังสือ

## ท้ายบท 1

1. L/R1 + L/R2
2. Link bandwidth = 2Mbps
   1. 2
   2. เนื่องจากการส่งทีละ 2 คน จะใช้ full bandwidth capacity พอดี หากอีกคนนึงต้องการส่งด้วย จะต้องรอให้คนใดคนหนึ่งในสองคนแรกส่งให้เสร็จก่อน = queuing delay
   3. (0.2)^(k) (0.8)^(n-k) when k = max sender in a time = 2
   4. (20%)^3 = 0.008
3. 
   1. ไม่รู้ ใครรู้ก็เก่งมาก
   2. เดา: internal use, multi-purpose, break some limitations, less complexity, more compatibility
4. Processing delay คงที่  
   Queuing delay ขึ้นอยู่กับ arrival rate  
   Transmission delay คงที่  
   Propagation delay คงที่

ก. 8\*(1000)/(2\*10^6) + (2500\*10^3)/(2.5\*10^8) = 0.014 seconds

ข. 8L/(R\*10^6) + (d\*10^3)/s วินาที

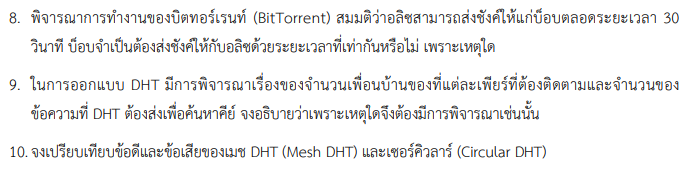
ค. ขึ้น

ง. ขึ้น

* 1. min(R1,R2,R3)
  2. L/R1 + L/R2 + L/R3  
     4 M bytes \* 8 bits / byte = 32 M bits  
     Time = 4/0.5 + 4/2 + 4/1 = 14 s

1. (-) การตั้งค่า/กำหนดความสามารถต่างๆ อาจจะซ้ำซ้อน จนทำให้เกิด overhead เช่น ทำให้ app layer มี reliability ในการส่งข้อมูล แต่ใช้ TCP protocol ใน network layer  
   (-) จะต้องใช้ Protocol ที่ตรงกันอย่างเคร่งครัด เพื่อให้สามารถติดต่อกันได้รู้เรื่อง
2. .
   1. ข้อมูลจากใน Application ที่ต้องการส่งไปยังอีก host หนึ่ง
   2. Packet + Transport layer header
   3. Segment + Network layer header
   4. Datagram + Link layer header
3. Router = Network layer  
   Switch = Link layer  
   Host = App layer and transport layer
4. ใครจะไปทำ เสียเวลา

## ท้ายบท2

1. Web HTTP
2. Bittorrent
3. TCP เนื่องจากการทำ RPC ต้องการ reliability ซึ่ง UDP ไม่มีสิ่งนี้
4. ตรวจสอบว่า application นั้น ต้องการความเชื่อถือของข้อมูลหรือไม่ ต้องการความเร็วในการรับส่งข้อมูลมากแค่ไหน
5. เนื่องจาก HTTP แต่เดิมแล้ว มีไว้เพียงเพื่อแสดงผลหน้าเว็บเท่านั้น ไม่ต้องจดจำว่า user เป็นใคร เลื่อนอะไรไปที่หน้าไหน ในปัจจุบันนั้นสามารถทำ application over HTTP ได้หลากหลาย และใช้ Cookie เป็นตัวจดจำ state แทน เมื่อพัฒนาแยกกันทำให้เกิด scalability ได้ดี
6. เพราะเป็น application ที่ต้องการเรื่อง reliability ซึ่ง UDP ไม่สามารถให้ได้
7. สามารถจดจำพฤติกรรมของลูกค้าได้ เช่น สินค้าที่อยู่ในตะกร้า / หน้าที่เปิดล่าสุด เป็นการเพิ่ม User Experience ได้ดี
8. ไม่จำเป็น เพราะ Upload ของแต่ละ Peer ไม่เท่ากัน (ขึ้นอยู่กับ Internet provider ที่เชื่อมต่ออยู่)
9. เดาว่า load balancing  
   
10. Mesh = ถามใครก็ได้ อาจจะถามมั่ว เจอคนที่รู้บ้างไม่รู้บ้าง แต่มี load distribution ทำให้มีโอกาสที่จะถามเจอคนที่รู้เลยสูง ประมาณ traffic จึงระบุได้ยาก  
    Circular = ถามแค่ตัวที่ติดกัน ทำให้เกิด traffic แบบคาดเดาได้ แต่การถามเป็นระบบมากกว่า

## ท้ายบท3

1. Source port = 61, dest port = 37
2. UDP เนื่องจากต้องการความ realtime ซึ่งไฟล์วิดีโอเป็นไฟล์ขนาดใหญ่ ทำให้รับส่งนาน อีกทั้งข้อมูลประเภทเสียงและวิดีโอนั้น ยังสามารถยอมรับ packet loss ได้ จึงสามารถเลือกใช้ UDP ได้
3. TCP overhead > UDP overhead
   1. ผิด สามารถส่ง ACK flag ได้ และควรส่ง เพราะไม่งั้น sender จะ timeout และ retransmission ไปเรื่อย ๆ
   2. ผิด เพราะว่าใน TCP Header ก็มี rwnd อยู่เพื่อ Indicate ความว่างของ Socket หน้า Buffer ตอนนั้น
   3. ถูก
   4. ไม่ถูก เพราะ Sequence # กล่าวถึงตำแหน่ง Byte, ไม่เกี่ยวข้องกับลำดับที่ของ Segment
   5. ถูก
   6. ถูก
   7. ไม่ถูก เหตุผลเหมือนข้อ d. + เพราะ ack ขึ้นอยู่กับ sequence number ที่"ได้รับ" ไม่เกี่ยวกับอันที่กำลังส่งออกไป
4. ก. 27 bytes

ข. 65 (Segment ที่ 2 จะถูกเก็บไว้ใน Buffer <https://stackoverflow.com/a/21897275>)

1. เพราะเป็น protocol สำหรับกำหนดการสื่อสารบน host เท่านั้น (network layer) ซึ่ง switch อยู่ในระดับ Link layer
2. R/2
3. AI = ทุกคนค่อยๆเพิ่มขึ้นทีละนิด  
   MD = หากมีข้อมูลใน traffic มากเกินไป ให้ลดกำลังการส่งของทุกคนลงครึ่งหนึ่ง เพื่อให้เกิดที่ว่างใน network มากเพียงพอ (หากลดลงแค่ 1 ก็จะมีบางอันที่เพิ่ม 1 ซึ่งสุดท้ายก็จะกลับมาเต็มอยู่ดี)
4. ใช่
5. ก. t = 0-6 และ 23-26

ข. t = 6-16 และ 17-22

ค. Duplicate ACK

ง. Timeout

จ. 32 segments

ฉ. 42/2=21 segments

ช. 26/2=13 segments

ซ. 1+2+4+8+16+32 = 63 (ครั้งที่ 6 ส่งได้ถึงเซกเมนต์ที่ 63 (เริ่มจาก 1))   
 1+2+4+8+16+32+33 = 96 (ครั้งที่7)

ฌ. Congestion window = ssthresh = 4 segments

ญ. ssthresh = 21 segments, congestion windows = 4 segments (1 -> 2 -> 4)

ฎ. 1+2+4+8+16+32 = 63?

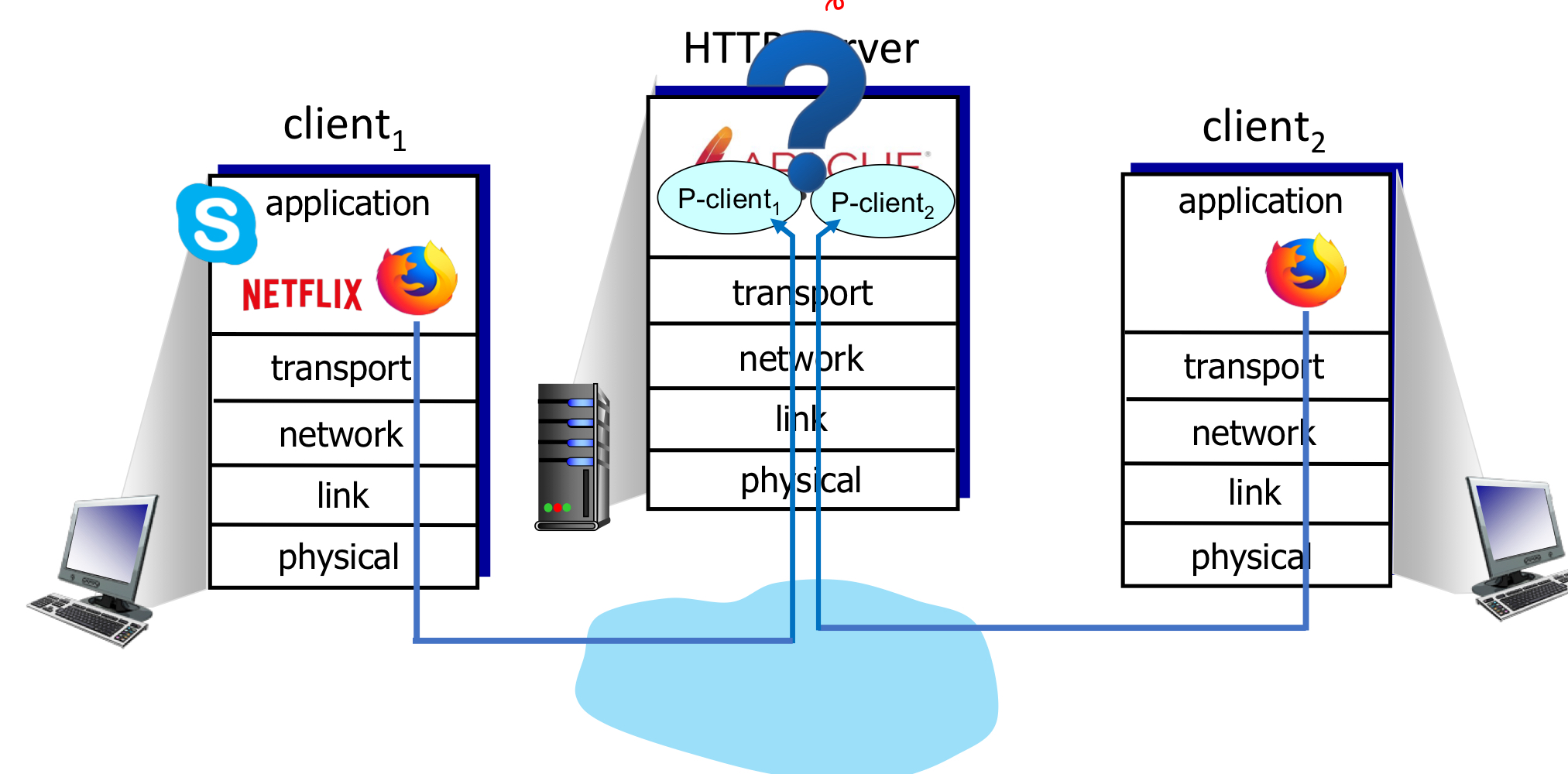
# CP37 midterm

1. ราคา: Twisted pair < Coaxial cable < Fiber Optic  
   ความเร็ว: Twisted pair < Coaxial cable < Fiber Optic  
   ความผิดพลาดในการส่งข้อมูล: -  
   ความยากง่ายในการติดตั้ง: -  
   ระยะทาง: -  
   ไม่ออกหรอก
2. ไม่ เนื่องจากปัจจุบันการเชื่อมต่อบน internet เป็นไปอย่างแพร่หลาย หากใช้ Circuit switching จะต้องคำนึงถึงเรื่อง resource allocation ซึ่งอาจเกิด overhead เป็นจำนวนมาก อีกทั้ง Packet switching ยังสามารถทำ routing ไปบนเส้นทางใหม่เพื่อควบคุมความหนาแน่นของ traffic ได้อีกด้วย
3. เลขหาย เขียนเท่าที่มีละกันนะ
   1. min(TP1,TP2,TP3)
   2. File size = 4 MBytes \* (8 bits/Byte) = 32000 kb  
      Time = 32000kb/TP1 + 32000kb/TP2 + 32000kb/TP3
   3. Throughput = min(TP) = 100kbps  
      Time = 32000 kb / 500kbps + 32000 kb / 100kbps + 32000kb / 1000kbps = 416 s
4. คำตอบจากในหนังสือ  
   เนื่องจากบริการของชั้นพรีเซ็นเตชั่นและชั้นเซสชั่นไม่จำเป็นเสมอไปในแอพพลิเคชั่นที่อยู่ในอินเตอร์เน็ต เพราะฉะนั้นหากแอพพลิเคชั่นใดที่ต้องการบริการเหล่านี้ก็สามารถที่จะพัฒนาอยู่ในชั้นแอพพลิเคชั่นได้

* Name service (DNS service) 4 ประเภท
  + Hostname-to-IP-address translation
  + Host aliasing
  + Mail server aliasing
  + Load distribution
* Out-of-band -> อ่าน CP39 Part 2 ข้อ 1
* DNS -> อ่าน CP39 Part 2 ข้อ 5
* Header ของ Email ประกอบด้วย sender name, receiver name, subject

# CP38 midterm

* ทำไมถึงใช้ packet switching อ่าน CP37 ข้อ 2
* ข้อดีและข้อเสียของ model แบบ layer  
  (+) สามารถสร้าง scalability ได้จากการปรับปรุงในบาง layer โดยจะไม่ส่งผลกระทบกับ layer อื่นๆ  
  (+) flexibility  
  (-) การตั้งค่า/กำหนดความสามารถต่างๆ อาจจะซ้ำซ้อน จนทำให้เกิด overhead เช่น ทำให้ app layer มี reliability ในการส่งข้อมูล แต่ใช้ TCP protocol ใน network layer  
  (-) จะต้องใช้ Protocol ที่ตรงกันอย่างเคร่งครัด เพื่อให้สามารถติดต่อกันได้รู้เรื่อง
* Network ห้องคอมโรงเรียนชนบท ใช้สาย TP เนื่องจากราคาถูก ติดตั้งได้ง่าย (เป็นหัวพลาสติกปากจระเข้) มี bandwidth ได้สูงสุด 1 Gbps ซึ่งถือว่าเพียงพอต่อระบบภายในห้องคอม  
   *แต่ข้อนี้น่าจะปลายเปิด ตอบยังไงก็ได้ให้มีเหตุผล*
* A B อยู่ห่างกัน X km ส่งไฟล์ขนาด P bit ผ่านสายขนาด N bps มีpropagation speed C m/s ใช้เวลาเท่าไหร่  
  D trans = P bit / N bps = P/N s  
  D prop = distance/speed = 1000X/C s   
  **Ans** = P/N + 1000X/C
* HTTP request msg if-modified since คือ เป็นสิ่งที่ส่งไปด้วยกับ Conditional GET เพื่อระบุว่าหากไฟล์ถูกแก้ไขก่อนวันใดถึงจะให้ส่ง Object \*อันใหม่\* กลับมาด้วย
* FTP out-of-band อ่าน CP39 Part 2 ข้อ 1
* เห็นด้วยหรือไม่ที่บอกว่า smtp เป็นโปรโตคอลสำหรับส่งเมลล์ระหว่าง client กับ server
* Host A หา www.yahoo.co.jp อ่าน CP39 Part 2 ข้อ 5   
  เปลี่ยนเป็น ask Root > ask .jp > ask .co.jp > return IP of www.yahoo.co.jp   
  ถ้า Host B ถามซ้ำ = local DNS server คืน IP ให้ได้เลยเนื่องจากมี cache ไว้ใน local DNS server แล้ว
* Bittorrent คือ File sharing system แบบ P2P ทำการแบ่งข้อมูลเป็น chunks แล้วไปฝากไว้กับหลาย ๆ hosts
* Distributed Hash Table (DHT) คือตารางที่ใช้สำหรับเก็บข้อมูลว่าข้อมูลที่ต้องการ download นี้ อยู่ที่ host เครื่องใด โดยเก็บเป็น (key: data, value: host)  
  Circular DHT คือการถาม peer เป็นวงกลม โดยถาม peer ที่อยู่ติดกับตนเองว่ามีหรือไม่ หากไม่มีจะทำการถามอีกตัวที่อยู่ติดกันไปเรื่อย ๆ แล้วตัวที่มี (key,value) ดังกล่าว จะคืนคำตอบกลับมาให้เครื่องที่ถามไป
* Browser 2 ตัวจากเครื่องเดียวกันติดต่อ server (ตกแต่งระบายสีให้สวยงาม)



* Why UDP อ่าน CP39 Part 3 ข้อ 1
* TCP segment structure อ่าน CP39 Part 3 ข้อ 2
* ประโยคใดผิด  
  1) ถ้า host a ส่งข้อมูลให้ host b ขนาดใหญ่มาก ๆ แล้ว unacknowledged bit ต้องไม่เกิน received buffer   
  **ถูก**  
  2) piggybacking (จำไม่ค่อยได้)   
  3) ถ้า sequence no. = m + 3 และส่งข้อมูลสำเร็จ sequence no. ต่อไป = m + 4   
  **ผิด เพราะ seg no. ต่อไปคือ ACK number ที่ได้รับจาก receiver**  
  4) ถ้า sequence no. = 38 ขนาดข้อมูล 4 byte ACK No. = 42   
  **ผิด ถ้าพูดถึง segNo กับ ackNo ในข้อความเดียวกัน แต่ถูกถ้าพูดถึง ackNo ของ response**  
  5) ไม่มีทางที่เราจะทำ reliable data transfer ได้โดยอาศัย UDP   
  **ผิด เพราะสามารถทำได้บน QUIC (HTTP/3)**
* กราฟ TCP congestion control (ไม่รู้จะทำยังไง ถ้าเจอก็ สู้ๆ อ่านสไลด์หน้า 125 ช่วยได้ กับแบบฝึกหัดท้ายบท แปะไว้ให้หน้าแรกๆ)

# CP39 midterm

## Part1

1. ARPANET
2. สื่อที่ต้องการตัวนำ เช่น Coaxial cable, Fiber Optic  
   สื่อที่ไม่ต้องการตัวนำ เช่น WiFi, Cellular
3. **อ่านคำอธิบายในหนังสือหน้า 10**  
   D trans = 2L/R
   1. 4\*2L/R = 8L/R
   2. (n)L/R
   3. m(n)L/R
4. ไม่มีผล เนื่องจาก propagation delay ขึ้นอยู่กับระยะทางในการส่ง และ ความเร็วของสายที่ใช้ส่ง
5. D prop = m/s  
   D total = L/R + m/s
6. จงยกตัวอย่างการคุกคามทางอินเตอร์เน็ตมา 3 ข้อ  
   DDoS attack, Spoofing attack, Malware, Packet sniffing
7. Throughput = min(TPs) = 500 kbps  
   Delay = 3/500 + 3/10000 + 3/100000 = 6.33 ms
8. ใช้ Circuit switching หาก link path bandwidth มีขนาดใกล้เคียงกับ 200 kbps เนื่องจากเป็นการจอง path ของการส่งข้อมูลเอาไว้เลย โดยหาก application นี้ ส่งข้อมูลทั้งวัน แปลว่า path นี้จะ fully utilize และไม่มี application อื่น ๆ มาแทรกการทำงานได้ และทำให้การเชื่อมต่อนั้น predictable หากใช้ packet switching อาจจะทำให้เกิดการแทรกของข้อมูลจาก application อื่นได้ ซึ่งควรใช้เมื่อ bandwidth มีขนาดมากกว่า 200 kbps มาก จะทำให้ utilization ดีที่สุด

| **Layer** | **Protocol** | **Data** |
| --- | --- | --- |
| App | HTTP / FTP / SMTP | Packet |
| Transport | TCP / UDP | Segment |
| Network | IP | Datagram |
| Link | Ethernet / WiFi / PPP | Frame |

## Part2

1. out of band ของ ftp มีความหมายอะไร

โดยปกติแล้ว FTP จะเปิด 2 Parallel TCP connections ซึ่งอันหนึ่งจะเปิดเพื่อส่งไฟล์จริง ๆ แต่อีกอันหนึ่งจะเปิดเพื่อส่ง Control information เท่านั้น จะได้ว่าการที่ Control information นี้ไม่ได้ถูกส่งไปที่ Connection เดียวกับไฟล์จริง ๆ เป็นการส่งแบบ Out of band

1. ไม่ เนื่องจาก SMTP เป็น protocol สำหรับการเชื่อมต่อกันระหว่าง mail server ถึง mail server อีกตัว ~~โดยจะส่งข้อมูลแบบใดก็ได้ ไม่ใช่เพียงแต่ Multimedia~~ [ไม่ถูกต้อง เนื่องจาก SMTP เป็นระบบที่สามารถส่งได้เพียง ASCII text-based message ขนาดจำกัดเท่านั้น = ไม่สามารถแนบ attachments ได้ However, MIME is another more advanced mail protocol which allows you to send attachments, so called "multimedia"]
2. การ Get ข้อมูล ก็ต่อเมื่อไม่มีข้อมูลใน Proxy server หรือเป็นข้อมูลที่เก่าเกินไปเท่านั้น เพื่อให้ไม่ต้องติดต่อกับ Main server บ่อยๆ ลด traffic ของ network ได้
3. non-Persistent เหมาะกับเว็บไซต์ในปัจจุบันหรือไม่ เพราะเหตุใด

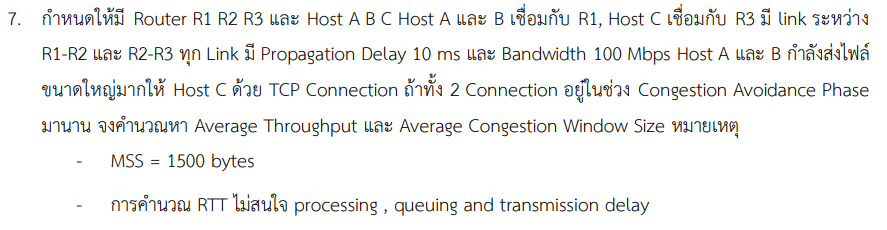
ไม่เหมาะ เพราะในปัจจุบันแต่ละเว็บไซต์ประกอบไปด้วย http objects จำนวนมาก ซึ่งหากใช้ non-persistent TCP connection จะเสีย Overhead ในการ Connect สูงมาก

| 1. Client asks cp.eng.chula.ac.th local DNS server where is www.youtube.com 2. If don't have information of www.youtube.com, local DNS ask Root DNS server 3. Root DNS server replies IP of .com DNS server 4. Local DNS asks .com DNS server 5. .com DNS server replies IP of www.youtube.com 6. Local DNS replies the information to client | 1. Client asks cp.eng.chula.ac.th local DNS server where is www.youtube.com 2. If don't have information of www.youtube.com, local DNS asks Root DNS server 3. If Root DNS doesn’t know where is www.youtube.com, Root DNS asks .com DNS server 4. .com DNS server replies IP to Root DNS server 5. Root DNS server replies IP to local DNS server 6. Local DNS replies the information to client |
| --- | --- |

1. จงอธิบายหลักการของ tit-for-tat   
   Tit-for-tat คือระบบที่จะดึงดูดให้คนที่มีความสามารถในระดับเดียวกันมาอยู่ในกลุ่ม (torrent) เดียวกัน

## Part3

1. เพราะ UDP สามารถติดต่อสื่อสารได้เร็ว ในกรณีที่ไม่จำเป็นจะต้องใช้ reliability
2. TCP header  
   Source port = port ของผู้ส่ง, Dest port = port ของผู้รับ มีไว้เพื่อให้รับส่งได้ถูกคน(?)  
   ACK = flag สำหรับบอกว่าข้อความนี้เป็นข้อความ ACK หรือไม่  
   SYN = flag สำหรับ handshaking  
   FIN = flag เพื่อให้รับทราบว่าต้องการจะ close connection  
   Sequence number = byte number แรกของข้อความปัจจุบันที่กำลังส่งไป เพื่อให้ผู้รับรู้ว่าผู้ส่งส่งอะไรมา  
   ACK number = byte number ของ expected next message เพื่อให้ผู้ส่งรับรู้ว่าผู้รับได้รับถึงข้อความที่เท่าไหร่  
   Checksum = การนำ segment content ทั้งหมดมาบวกกันแบบ bitwise เพื่อ detect flip bit  
   Window size = receiver window buffer เพื่อให้ผู้ส่งรู้ว่าผู้รับยังสามารถรับข้อมูลได้อีกกี่ byte



Assumptions

1. A, B จะแบ่ง Rate กันไปคนละครึ่ง, implies max congestion window = 750 bytes for each.
2. RTT = 80 ms (ไป 40 ms, กลับ 40 ms)
3. Average window size = ¾ W because it’s linearly increasing and cut half

Hence,

1. Average congestion window size = ¾(750) = 562.5 bytes
2. Average TP = 562.5/80\*10^-3 = 7 kb/s