

# หลักการวัดประสิทธิภาพในชั้น Transport Layer (TCP)

การตัดสินว่าแพ็กเก็ตใดคือ Request และ Response

เพื่อวัดประสิทธิภาพของเป้าหมายในชั้น Transport Layer ระบบควรจำแนกได้ว่าแพ็กเก็ตใดคือ Request จาก Client ที่มายังเป้าหมายและแพ็กเก็ตใดคือ Response จากเป้าหมายไปยัง Client

โดยในโปรโตคอล TCP นั้นจะตรวจสอบได้จากทิศทางการไหลของข้อมูลและ Payload

- Request: มักจะเป็นแพ็กเก็ตแรกที่ส่งข้อมูล (Payload > 0) หลังจากทำ Three-way Handshake เสร็จสิ้น โดยส่งจากฝั่ง Client ไปยัง Server (เช่น HTTP GET, Database Query)
- Response: คือแพ็กเก็ตที่ส่งข้อมูลสวนทางกลับมาจาก Server ไปยัง Client (เช่น HTTP 200 OK, ผลลัพธ์จากการ Query) โดยมักจะมีเลข Acknowledgment Number ที่ตรงกับเลข Next Sequence Number ของฝั่ง Request

การตัดสินว่า Request และ Response อยู่ในการสนทนาเดียวกัน

เป็นขั้นตอนที่สำคัญเพื่อใช้ในการจำแนกว่าแพ็กเก็ต Response ที่เจอนั้นอยู่ในการสนทนาเดียวกันกับแพ็กเก็ต Request ก่อนหน้าหรือไม่ โดยมีวิธีการจำแนกดังนี้

1. นำข้อมูล Source IP, Destination IP, Source Port, Destination Port, Protocol จากทั้ง Request และ Response มาตรวจสอบ
2. สร้าง hash ที่มี Key คือ Source IP, Destination IP, Source Port, Destination Port, Protocol ตามข้อมูลแพ็กเก็ต ซึ่งเป็น Key แบบสองทิศทาง เช่น A ไป B และ B ไป A ต้องระบุได้ว่าเป็นค่าเดียวกัน
3. เพื่อให้ได้ Key ตามข้อ 2 จะนำ IP ทั้ง 2 มาหา Min-Max และทำแบบเดียวกับ Port ทั้ง 2 เช่นกัน จากนั้นนำมาจัดทำให้อยู่ในรูปแบบนี้

[Address\_Low][Address\_High][Port\_Low][Port\_High][Protocol]

4. นำผลลัพธ์จากข้อ 3 เข้าสู่ฟังก์ชัน Jenkins Hash จะได้ค่า Hash ออกมา (เช่น 0x1A2B3C4D)

หากแพ็กเก็ตที่ตรวจสอบมีค่า Hash ตรงกับค่าที่มีอยู่ในระบบ สามารถระบุได้ว่าแพ็กเก็ตดังกล่าวอยู่ในสายการสนทนา (Flow) เดียวกัน แต่หากไม่พบข้อมูล จะทำการสร้าง รายการใหม่ในตารางสถานะเพื่อรองรับแพ็กเก็ตถัดไปที่มีค่า Hash ตรงกัน

การตัดสินว่า Response นั้นเป็นการตอบกลับของ Request ก่อนหน้าหรือไม่

เมื่อทำการยืนยันแล้วว่าแพ็กเก็ตทั้งคู่นั้นอยู่ในการสนทนาเดียวกัน ขั้นตอนต่อมาคือการตรวจสอบว่า Response ที่มาถึงนั้นคือ Response ที่ตอบกลับ Request ก่อนหน้าจริงหรือไม่ โดยมีวิธีจับคู่ดังนี้

1. เมื่อตรวจพบแพ็กเก็ต Request วิ่งผ่านจะบันทึกค่า Acknowledgment Number ที่คาดหวังตามสูตรดังนี้

$\text{Expected ACK} = \text{Sequence Number} + \text{TCP Payload Length}$

2. เมื่อมีแพ็กเก็ตที่มี Flag ACK หรือ Response วิ่งสวนทางกลับมา จะดูที่ฟิลด์ Acknowledgment Number ของแพ็กเก็ตนั้นและตรวจสอบว่าตรงกับค่าที่คำนวณในข้อ 1 (Expected ACK) หรือไม่ หากตรงจะถือว่าแพ็กเก็ตนั้นมีลำดับการทำงานที่ถูกต้อง

เงื่อนไขอื่นในการจับคู่

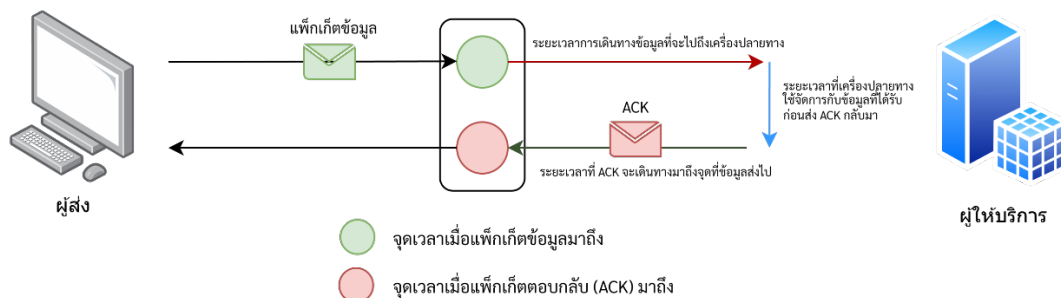
ในหลายครั้งที่การจราจรของเครือข่ายไม่เป็นไปตามลำดับและหลักการ จึงมีเงื่อนไขเพิ่มเติมสำหรับการจับคู่ Request และ Response ดังนี้

1. หากพบแพ็กเก็ตที่มี Sequence Number เดิมส่งมาอีกครั้ง (Retransmission) จะไม่นำไปจับคู่เพื่อตามกฎของ Karn เพื่อป้องกันค่า RTT ที่ผิดเพี้ยน
2. หากมีแพ็กเก็ตข้อมูลส่งมากกว่า 1 ต้องคำนวณค่า ACK Number ที่คาดหวังจากแพ็กเก็ตข้อมูลตัวสุดท้ายที่มาถึง เร็วขึ้นนี้จะรวมถึงการข้ามลำดับ ซึ่งสามารถตรวจสอบที่ Sequence Number ของแพ็กเก็ตว่าการไล่ลำดับหรือไม่

## ค่าดัชนีการวัดประสิทธิภาพ (Metrics)

### ACK Response Time

ค่าดัชนีนี้คือส่วนต่างของเวลาระหว่าง แพ็กเก็ตข้อมูลที่ส่งไป กับ แพ็กเก็ตยืนยัน (ACK) ที่ตอบกลับมา ซึ่งบอกระยะเวลาการเดินทางของข้อมูลที่จะไปถึงเครื่องปลายทาง ระยะเวลาที่เครื่องปลายทางใช้จัดการกับข้อมูลที่ได้รับก่อนส่ง ACK กลับมา และระยะเวลาที่ ACK จะเดินทางมาถึงจุดที่ข้อมูลส่งไป



### การคำนวณ

ตรวจหาแพ็กเก็ตที่มีข้อมูล จากนั้นตรวจหาแพ็กเก็ตที่อยู่ในสายการสนทนาเดียวกัน (Flow) ที่เป็นการยืนยัน (ACK) ว่าได้รับข้อมูลนี้ แล้วนำ Timestamp ของทั้ง 2 มาหาส่วนต่างตามสูตรด้านล่าง

$$\text{ACK Response Time} = T_{\text{ACK}} - T_{\text{Segment}}$$

- $T_{\text{Segment}}$ : เวลาที่ส่ง TCP Segment ที่มีข้อมูล (Payload) ออกไป
- $T_{\text{ACK}}$ : เวลาที่ได้รับ ACK Packet ที่ระบุหมายเลข Acknowledgment Number ตรงกับ Next Sequence Number ของ Segment นั้นพอดี

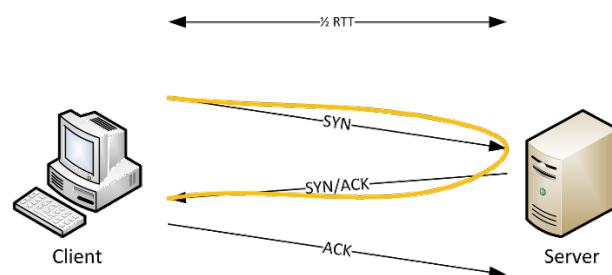
## Handshake Time/Initial Round Time Trip

คือ ระยะเวลาในการทำ TCP 3-way Handshake ซึ่งมีขนาดแพ็กเก็ตที่เล็กและคงที่ (ปกติไม่เกิน 74 bytes) ทำให้ค่า Transmission Delay ต่ำ และยังไม่มีการประมวลผลของแอปพลิเคชัน ทำให้ไม่มี Delay ในส่วนนี้ ดังนั้นจึงบอกได้ว่าเป็นค่า Latency พื้นฐานของเส้นทางนั้นๆ (Baseline Latency) สามารถใช้เป็นเกณฑ์อ้างอิงในการเปรียบเทียบเมื่อเกิดปัญหาได้

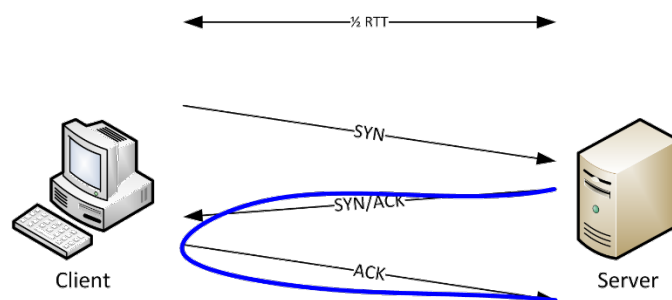
### การคำนวณ

วัดช่วงเวลาไปและกลับของการทำงาน TCP 3-way Handshake โดยมีเงื่อนไขดังนี้

หากเราวัดจาก Client จะเป็นการวัดระยะเวลาระหว่าง SYN ถึง SYN/ACK



หากเราวัดจาก Server จะเป็นการวัดระยะเวลาระหว่าง SYN/ACK ถึง ACK



โดยการกำหนดว่าใครเป็น Server หรือ Client จะดูว่าฝั่งไหนเป็นผู้เริ่มสื่อสาร ในบริบทนี้คือการส่งแพ็กเก็ต SYN เพื่อเริ่ม TCP 3-way Handshake

ขอบคุณรูปภาพจาก <https://blog.packet-foo.com/2014/07/determining-tcp-initial-round-trip-time/>