

การจำแนกการสูญเสียแพคเกจซึ่งเกิดจากความแออัด

อนุวัตร สมบุญ, บุญชัย งามวงศ์วัฒนา

ผู้ช่วยนักวิจัย, นักวิจัย หน่วยปฏิบัติการวิจัยเทคโนโลยีเครือข่าย

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ โทรศัพท์ 02-564-6900 ต่อ2585

Email: {anuwat.sombun, boonchai.ngamwongwattana} AT nectec DOT or DOT th

ABSTRACT – Packet loss has widely been used as a means to detect the state of the network. Because the probability of random loss on wired networks is very low, packet loss is assumed to be caused by network congestion. Detecting packet loss is very simple by observing out-of-sequence packets. However, detecting packet loss provides very limited information about the state of the network. Several studies have confirmed that there is a correlation between delay and packet loss on the Internet, which can be very useful to get more information about the state of the network. In this paper, we attempt to use observations of the delay and loss correlation to characterize congestive packet loss. Specifically, the proposed scheme attempts to distinguish between packet loss due to a limited bandwidth bottleneck and packet loss that is more random-like due to a high level of statistical multiplexing. Such a distinction would lead a control mechanism to be more intelligent to respond to packet loss and to optimize the performance.

KEY WORDS -- Bottleneck link, congestive packet loss, delay and loss correlation, packet loss characteristics.

บทคัดย่อ – การสูญเสียแพคเกจ (packet loss) อาจกล่าวได้ว่าเป็นปัจจัยที่นำมาใช้อย่างกว้างขวางในการตรวจสอบสถานะเครือข่าย เพราะว่าบนเครือข่ายแบบสายโอกาสที่จะเกิดการสูญเสียแบบสุ่ม (random loss) มีน้อยมาก การสูญเสียแพคเกจจึงถูกสันนิษฐานว่าเกิดจากความแออัดบนเครือข่าย (network congestion) การตรวจสอบการสูญเสียแพคเกจทำได้ง่ายมากโดยการสังเกตลำดับที่หายไปของแพคเกจที่รับมา แต่อย่างไรก็ตามการตรวจสอบเพียงการสูญเสียแพคเกจอย่างเดียวเป็นข้อมูลที่จำกัดสำหรับแสดงสถานะของเครือข่าย มีการศึกษาจำนวนมากแสดงความสัมพันธ์ระหว่างระยะเวลาที่ใช้ในการส่งแพคเกจจนถึงผู้รับ (delay) และแพคเกจที่สูญเสียบนอินเทอร์เน็ต (Internet) ซึ่งเป็นข้อมูลที่มีประโยชน์มากที่ใช้แสดงสถานะของเครือข่าย บทความนี้เราพยายามสังเกตความสัมพันธ์ของ delay และการสูญเสีย เพื่อจำแนกการสูญเสียแพคเกจซึ่งเกิดจากความแออัด เราได้เสนอวิธีการจำแนกระหว่างการสูญเสียแพคเกจซึ่งเกิดจากจุดติดขัดที่มีแบนด์วิดท์จำกัด (limited bandwidth bottleneck) และการสูญเสียแพคเกจแบบสุ่ม เพราะมีการใช้ร่วมของเส้นทางส่งข้อมูลจำนวนมาก (high level of statistical multiplexing) จุดเด่นคือทำให้สามารถนำกระบวนการควบคุมที่มีประสิทธิภาพมากขึ้นมาทำการรองรับแพคเกจที่สูญเสียและปรับปรุงสมรรถนะการทำงานได้อย่างเหมาะสมที่สุด

คำสำคัญ -- Bottleneck link, congestive packet loss, delay and loss correlation, packet loss characteristics.

1. บทนำ

การสูญเสียแพคเกจเกิดขึ้นเมื่อแพคเกจเดินทางบนเครือข่ายแล้วไม่สามารถไปยังปลายทางได้ การสูญเสียแพคเกจ

สามารถแบ่งได้ 2 ชนิดคือ การสูญเสียแบบสุ่ม (random loss) และการสูญเสียจากความแออัด (congestion loss) การสูญเสียแพคเกจแบบสุ่มส่วนมากเกิดขึ้นอย่างอิสระ เมื่อสัญญาณไฟฟ้าบน

ช่องสัญญาณที่ใช้งานมีคุณภาพต่ำลงทำให้เกิดบิต (bit) ผิดพลาด ด้วยสาเหตุนี้แพ็คเกจที่มีปัญหา (corrupted packet) จะถูกทิ้ง (dropped) ระหว่างการขนส่งโดยอุปกรณ์จัดเส้นทาง (routers) หรือเกตเวย์ (gateways) ถึงแม้ว่าแพ็คเกจที่มีปัญหาจะสามารถเดินทางมาถึงปลายทาง แต่แพ็คเกจเหล่านั้นจะถูกทิ้งก่อนที่จะส่งไปยังโปรแกรมประยุกต์ (application) สาเหตุของการสูญเสียแพ็คเกจแบบสุ่มเกิดขึ้นได้ทั้งฮาร์ดแวร์ ซอฟต์แวร์และอุปกรณ์ภายในเครือข่าย เช่น ลิงค์ (link) อุปกรณ์สวิตช์ (switch) router เมื่อเกิดการจัดเส้นทางผิดพลาด ในระบบเครือข่ายแบบสาย (wired network) การสูญเสียแพ็คเกจแบบสุ่มทำความเสียหายเพียงเล็กน้อยและอัตราการการส่งบิตผิดพลาดมีค่าน้อยมาก [1] การสูญเสียแพ็คเกจซึ่งเกิดจากความแออัดเกิดขึ้นเมื่อลิงค์บนเครือข่ายมีการใช้งานหนักมากเกินไป (over-utilized) กล่าวคือแพ็คเกจเดินทางมาถึงคิวภายในเครือข่าย (network queue) ด้วยอัตราเร็วกว่าความสามารถของลิงค์ (link capacity) ด้วยสาเหตุนี้คิว (queue) ถูกทำให้เต็มแพ็คเกจที่ตามมาทีหลังจะถูกทิ้งไปจากคิว การสูญเสียแพ็คเกจซึ่งเกิดจากความแออัดสามารถพบได้ทั่วไปในเครือข่ายแพ็คเกจแบบสวิตช์ (packet-switched network) เพราะเครือข่ายจะรับแพ็คเกจทั้งหมดจากผู้ใช้งานโดยปราศจากการควบคุมการรับแพ็คเกจเข้ามา (admission control) ทำให้ลิงค์ที่ใช้งานหนักมากเกินไปเกิดการสูญเสียแพ็คเกจ กล่าวในอีกประเด็นคือเครือข่ายไม่มีการจัดเตรียมหรือจองแบนด์วิดท์ (bandwidth) ไว้ล่วงหน้าสำหรับการใช้งาน โปรแกรมประยุกต์ต้องพยายามครอบครองแบนด์วิดท์ที่จำเป็นต่อการใช้งาน กรณีของโปรโตคอล TCP จะพยายามเพิ่มหน้าต่างการส่ง (transmission window) จนกระทั่งพบการสูญเสียแพ็คเกจ เช่นเมื่อพบการสูญเสียแพ็คเกจ TCP จะลดหน้าต่างการส่งและทำการประมาณแบนด์วิดท์ที่สามารถใช้งานได้และค่อยเพิ่มหน้าต่างการส่งอย่างระมัดระวังเพื่อหลีกเลี่ยงการเกิดความแออัด

การตรวจสอบสถานะของเครือข่ายเป็นงานที่สำคัญสำหรับกระบวนการควบคุมความแออัด (congestion control mechanism) การตรวจสอบการสูญเสียแพ็คเกจถูกนำมาใช้อย่างกว้างขวางเนื่องจากสามารถแสดงความแออัดของเครือข่ายได้เพราะโอกาสที่จะเกิดการสูญเสียแพ็คเกจแบบสุ่มบนเครือข่ายแบบสายมีน้อยมากจึงตั้งสมมุติฐานได้ว่าการสูญเสียแพ็คเกจเกือบทั้งหมดเกิดจากความแออัด ซึ่งกลายเป็นสมมุติฐานเบื้องต้นในการออกแบบการควบคุมความแออัดในโปรโตคอล TCP [1] การตรวจสอบการสูญเสียแพ็คเกจทำได้ง่ายมากที่ตัวรับ (receiver) โดยการสังเกตลำดับที่หายไปของแพ็คเกจที่รับมา การตรวจสอบการสูญเสียแพ็คเกจสามารถทำได้ที่ตัวส่งเช่นกัน โดยการสังเกตลำดับที่หายไป

ของแพ็คเกจที่ตอบกลับ (acknowledgement packet) แต่วิธีนี้มีความซับซ้อนมากกว่าเพราะแพ็คเกจที่ตอบกลับสามารถเกิดการสูญเสียระหว่างการขนส่งได้เช่นกัน การสังเกตการสูญเสียแพ็คเกจเพียงอย่างเดียวเป็นข้อมูลที่จำกัดมากสำหรับการตรวจสอบสถานะเครือข่าย ถึงแม้ว่าแพ็คเกจที่มีการสูญเสียแบบสุ่มหรือการสูญเสียที่เกิดจากความแออัด ระบบควบคุมไม่สามารถจำแนกได้ว่าเป็นการสูญเสียแบบใด ตัวอย่างเช่น TCP จะปฏิบัติกับทุกการสูญเสียแพ็คเกจเหมือนกัน TCP จะทำการลดขนาดหน้าต่างการส่ง โดยมีสมมุติฐานว่าการสูญเสียแพ็คเกจแสดงความแออัดของเครือข่ายซึ่งแตกต่างกับเครือข่ายแบบไร้สายสมมุติฐานข้างต้นไม่สามารถใช้ได้ เพราะบนเครือข่ายแบบไร้สายการสูญเสียแพ็คเกจจำนวนมากเป็นแบบสุ่มซึ่งเกิดจากความไม่แน่นอนของช่องสัญญาณไร้สาย TCP ไม่สามารถปฏิบัติกับทุกการสูญเสียแพ็คเกจด้วยวิธีการเดิมเพราะการลดหน้าต่างการส่งเพื่อตอบสนองต่อการสูญเสียแบบสุ่มทำให้สมรรถนะการทำงานของเครือข่ายลดลง การศึกษาจำนวนมากได้นำเสนอกระบวนการใหม่ๆที่ทำให้ TCP สามารถแยกสาเหตุของการสูญเสียแพ็คเกจได้ [2,3,4]

ดังที่ได้กล่าวมาแล้ว แสดงให้เห็นความจำเป็นที่จะต้องมีความควบคุมที่สามารถจำแนกชนิดการสูญเสียแพ็คเกจได้ ขณะที่มีการศึกษาจำนวนมากศึกษาเกี่ยวกับการจำแนกการสูญเสียแบบสุ่มซึ่งเกิดจากความไม่แน่นอนของช่องสัญญาณไร้สาย [2,3,4] บทความนี้ได้มองอีกประเด็นหนึ่งคือการจำแนกการสูญเสียแพ็คเกจซึ่งเกิดจากความแออัดบนเครือข่ายแบบสาย เราพยายามจำแนกการสูญเสียแพ็คเกจซึ่งเกิดจากความแออัดและพิจารณากรณีเกิดขึ้นที่จุดติดขัดที่มีแบนด์วิดท์จำกัด (limited bandwidth bottleneck) ต่อไปเราจะอธิบายถึงคุณลักษณะของการสูญเสียแพ็คเกจที่เกิดจากความแออัดและแสดงให้เห็นว่าเราสามารถจำแนกได้อย่างไรเพื่อนำระบบควบคุมที่ชาญฉลาดทำการตอบสนองการสูญเสียแพ็คเกจ เราได้เสนอกระบวนการอย่างง่ายที่เรียกว่าการควบคุมที่ตระหนักถึงจุดติดขัดที่จำกัด (limited-bottleneck aware control) สุดท้ายเราอธิบายแบบจำลองและวิเคราะห์ผลการจำลอง

2. คุณลักษณะของแพ็คเกจสูญเสียซึ่งเกิดจากความแออัด

การสูญเสียแพ็คเกจเนื่องจากอัตราบิตผิดพลาด (bit error rate) เป็นที่รู้กันว่าไม่ค่อยเกิดบนเครือข่ายแบบสาย กระบวนการควบคุมการขนส่ง (transport control mechanisms) ถูกออกแบบโดยสมมุติฐานว่าการสูญเสียแพ็คเกจเป็นตัวแสดงว่าเครือข่ายแออัด ผลคือการดำเนินการโดยทันทีเพื่อหลีกเลี่ยงความแออัดเช่นลด

การส่งข้อมูลอย่างรวดเร็ว นี่เป็นลักษณะเฉพาะอันหนึ่งที่เกิดขึ้นในกรณีที่มีการใช้ร่วมของเส้นทางข้อมูลจำนวนน้อย (low level of statistical multiplexing) เช่นมีจำนวนตัวส่งไม่มากที่ใช้เส้นทางร่วมกันในลิงค์เดียวกัน เนื่องจากตัวส่งมีส่วนทำให้เกิดความแออัดโดยตรง อย่างไรก็ตาม มีการศึกษาจำนวนมากแสดงให้เห็นว่าการใช้งานร่วมกันบนเครือข่ายแบบ best-effort network ในกรณีที่มีการใช้ร่วมของเส้นทางข้อมูลจำนวนมาก (high level of statistical multiplexing) พฤติกรรมของแพ็คเกจที่สูญเสียถูกกำหนดโดยปริมาณการใช้เส้นทางข้อมูลรวมทั้งหมด (aggregate traffic) รูปแบบการสูญเสียแพ็คเกจนั้นมีลักษณะใกล้เคียงพฤติกรรมแบบสุ่ม [5,6] ผลการทดลองโดย Bolot [6] แสดงให้เห็นว่าการสูญเสียของแพ็คเกจจะมีลักษณะเป็นแบบสุ่มเมื่อมีปริมาณการใช้เส้นทางข้อมูลร่วมกันมาก กรณีนี้ถึงแม้ว่าแพ็คเกจที่สูญเสียเกิดจากความแออัด แต่ระบบอาจจะไม่ต้องตอบสนองโดยทันทีเนื่องจากตัวส่งอาจจะไม่ใช่สาเหตุสำคัญที่มีผลต่อความแออัด ดังนั้นการจำแนกการสูญเสียแพ็คเกจจากความแออัดจะทำให้กระบวนการควบคุมมีความชาญฉลาดมากขึ้นและตอบสนองกับแพ็คเกจที่สูญเสียได้ดีขึ้น ตัวอย่างเช่น โพรโทคอล TCP Reno [7] เติร์ยวิธีการ Fast Retransmit และ Fast Recovery ซึ่งเป็นวิธีการจำแนกการสูญเสียแพ็คเกจที่เกิดจากความแออัดแบบหนึ่งซึ่งทำการตอบสนองแพ็คเกจที่สูญเสียได้อย่างมีประสิทธิภาพมากสำหรับวิธี Fast Retransmit ถ้ามีการส่งสัญญาณตอบกลับ (acknowledgement) ของแพ็คเกจที่สูญเสียช้ามากกว่าสามครั้งขึ้นไปแสดงว่าเกิดแพ็คเกจสูญเสีย แพ็คเกจต่อมาซึ่งจะได้รับที่ตัวรับ ดังนั้นแพ็คเกจสูญเสียน่าจะเกิดจากสาเหตุความแออัดชั่วคราวหรือความแออัดไม่รุนแรง ตัวส่งจะทำการส่งซ้ำของส่วนที่สูญหายทันทีโดยข้ามกระบวนการส่งข้อมูลซ้ำปกติ (normal retransmit process) สำหรับวิธี Fast Recovery ตัวส่งทำการส่งแพ็คเกจอย่างต่อเนื่องโดยใช้วิธี Congestion Avoidance แทนวิธี Slow Start ทำให้สามารถเพิ่มการส่งข้อมูลได้เร็วกว่าและมีประสิทธิภาพมากขึ้น

2.1 การจำแนกการสูญเสียแพ็คเกจซึ่งเกิดจากความแออัดที่จุดตัดขั้วที่มีแบนด์วิดท์จำกัด

เราพยายามจำแนกการสูญเสียซึ่งเกิดจากความแออัดกรณีเกิดขึ้นที่จุดตัดขั้วที่มีแบนด์วิดท์จำกัด ด้วยเหตุนี้การสังเกตแพ็คเกจที่สูญเสียอย่างเดียวนั้นเป็นข้อมูลที่จำกัดมากในการแสดงสถานะเครือข่าย แพ็คเกจที่สูญเสียบนจุดตัดขั้วที่มีแบนด์วิดท์จำกัดแสดงถึงความแออัดของเครือข่ายที่ต้องทำการแก้ไขโดยทันที เนื่องจากตัวส่งมีส่วนทำให้เกิดความแออัดโดยตรง ในทางตรงกันข้ามแพ็คเกจที่สูญเสียจากความแออัดซึ่งมีลักษณะแบบสุ่มแสดง

ความแออัดที่เกิดขึ้นชั่วคราวหรือความแออัดที่ไม่รุนแรง ตัวส่งไม่ค่อยมีส่วนในความแออัดนั้น การจำแนกแพ็คเกจที่สูญเสียจากความแออัดทำให้กระบวนการควบคุมไม่ต้องตอบสนองทันทีต่อความแออัดที่ไม่รุนแรงทำให้การรองรับมีประสิทธิภาพมากขึ้นมีการศึกษาจำนวนมากแสดงให้เห็นว่ามีความสัมพันธ์ระหว่าง delay และการสูญเสียแพ็คเกจบนอินเทอร์เนต [8,9,10,11] ซึ่งมีประโยชน์มากในการจำแนกการสูญเสียแพ็คเกจที่เกิดจากความแออัด เราใช้โมเดลอย่างง่ายแสดงความเป็นไปได้ในการใช้ความสัมพันธ์ระหว่าง delay และแพ็คเกจที่สูญเสียจำแนกการสูญเสียแพ็คเกจบนจุดตัดขั้วที่มีแบนด์วิดท์จำกัด กำหนดให้เส้นทางจากตัวส่งถึงตัวรับประกอบด้วยลิงค์จำนวน H ลิงค์ โดยที่ $i = 1, \dots, H$ สมมติให้ลิงค์ทำงานแบบ FIFO (First-In-First-Out) และการรออยู่ในคิวของเราเตอร์ (queuing) แบบ drop-tail ระยะเวลาที่ใช้ในการส่งแพ็คเกจจนถึงผู้รับ (one-way network delay) (D) ของแพ็คเกจที่ได้รับสามารถเขียนได้ดังสมการที่ (1)

$$D = \sum_{i=1}^H (D_{prop}^i + D_{tran}^i + D_{queue}^i) \quad (1)$$

D_{prop}^i คือ เวลาที่ใช้ในการเดินทางบนสาย cable (propagation delay) ที่ลิงค์ i

D_{tran}^i คือ เวลาที่ใช้ในการส่งของ router (transmission delay) ที่ลิงค์ i

D_{queue}^i คือ เวลาที่รออยู่ในคิวของ router (queuing delay) ที่ลิงค์ i

โดย D_{prop}^i และ D_{tran}^i โดยปกติมีค่าคงที่ตลอดทั้งเส้นทาง ในขณะที่ D_{queue}^i โดยปกติมีค่าไม่แน่นอน การสูญเสียแพ็คเกจเกิดขึ้นเมื่อลิงค์ทำงานจนเต็มขีดความสามารถและแพ็คเกจเริ่มถูกทิ้งจากคิว โดยให้ m -th เป็นลิงค์ที่แพ็คเกจถูกทิ้ง กำหนดให้ D_{loss} เป็น delay ของแพ็คเกจที่รับมาก่อนแพ็คเกจที่สูญเสียซึ่งเขียนได้ดังสมการที่ (2)

$$D_{loss} = \sum_{i=1}^H (D_{prop}^i + D_{tran}^i) + \sum_{i=1, i \neq m}^H D_{queue}^i + \hat{D}_{queue}^m \quad (2)$$

สมการที่(2) แสดงให้เห็นว่า D_{queue}^i อาจจะมีค่ามากหรือน้อยได้ ในขณะที่ \hat{D}_{queue}^m เป็นค่า queuing delay สูงสุดของลิงค์ลำดับที่ m หาได้จากขนาดของคิวหารด้วยปริมาณแบนด์วิดท์ของลิงค์ (C^i)

$$D_{queue}^i = \frac{QueueSize}{C^i} \quad (3)$$

จะเห็นได้ว่า queuing delay เป็นสัดส่วนกลับกันกับปริมาณแบนด์วิธของลิงค์ ถ้าปริมาณแบนด์วิธของลิงค์น้อย queuing delay จะมีค่ามาก จากสมการที่(2) เราพยายามสังเกตค่า D_{loss} และจำแนกแพ็คเกจที่ไปถึงที่เกิดการสูญเสียบนจุดติดขัดที่มีแบนด์วิธจำกัด ถ้าเป็นกรณีนี้ในสมการที่(2) \hat{D}_{queue}^m จะมีค่ามากกว่า $\sum_{i=1, i \neq m}^H D_{queue}^i$ และเราสังเกตเวลาที่ใช้ในการส่งของแต่ละแพ็คเกจที่ใช้เทียบกับแพ็คเกจอื่น (peak delay) ในทางตรงกันข้าม ถ้าการสูญเสียแพ็คเกจเกิดขึ้นบนลิงค์ที่ไม่ติดขัดคือปริมาณแบนด์วิธที่มีขนาดใหญ่ เราจะสังเกตได้ว่า delay มีค่าต่ำ อย่างไรก็ตาม D_{loss} มีขอบเขตดังต่อไปนี้

$$\sum_{i=1}^H (D_{prop}^i + D_{tran}^i) + \hat{D}_{queue}^m \leq D_{loss} \quad (4)$$

และ

$$D_{loss} \leq \sum_{i=1}^H (D_{prop}^i + D_{tran}^i) + \sum_{i=1, i \neq m}^H D_{queue}^i + \hat{D}_{queue}^m \quad (5)$$

D_{loss} มีค่าต่ำสุดดังสมการที่(4) เมื่อแพ็คเกจที่รับมามีค่า queuing delay น้อยๆบน i -th ลิงค์ แต่มีค่า queuing delay สูงสุดบน m -th ลิงค์ D_{loss} มีค่าสูงสุดดังสมการที่ (5) เมื่อแพ็คเกจที่รับมามี queuing delay สูงสุดบนทุกลิงค์ ที่จริงขอบเขตเหล่านี้มีความเป็นไปได้น้อยมากที่สุด เนื่องจาก queuing delay ปกติมีการเปลี่ยนแปลง (delay variation) สูงมาก D_{loss} จะแปรผันอย่างมากเช่นกันจากสมการที่ (2) $\sum_{i=1, i \neq m}^H D_{queue}^i$ อาจจะมีส่วนในค่า delay มากกว่า D_{queue}^m ดังนั้นไม่ยากที่จะจำแนกการสูญเสียแพ็คเกจซึ่งเกิดจากความแออัดจากค่า delay ได้อย่างแม่นยำ ถึงกระนั้นข้อจำกัดนี้ไม่ใช่ปัญหาในระบบจัดการของการควบคุมที่ตระหนักถึงจุดติดขัดที่มีแบนด์วิธจำกัด (limited-bandwidth aware control) ของเราซึ่งจะได้ทำการอธิบายต่อไป

2.2 การควบคุมที่ตระหนักถึงจุดติดขัดที่มีแบนด์วิธจำกัด

ดังที่ได้กล่าวมาแล้วระบบต้องพยายามหา peak delay เพื่อจำแนกการสูญเสียแพ็คเกจที่เกิดขึ้นบนจุดติดขัดที่มีแบนด์วิธจำกัด โดยการตรวจค่า D_{loss} ระบบไม่มีทางทราบค่า peak delay ค่าต่อไปว่าจะเป็นอะไร ดังนั้นการตัดสินใจว่าเกิดการสูญเสียแพ็คเกจบนจุดติดขัดที่มีแบนด์วิธจำกัดหรือไม่ สามารถทำได้โดยใช้ D_{loss} ก่อนหน้าเท่านั้น เราใช้กฎนี้เพื่อจำแนก peak delay ดังต่อไปนี้

$$D_{loss} \geq D_{thres}; D_{thres} = \alpha \max(D_{loss}) \quad (6)$$

จะเห็นได้ว่า D_{loss} ต้องเท่ากับหรือมากกว่า D_{thres} ซึ่งทำการหาจาก peak ของ D_{loss} ที่บันทึกไว้ในอดีต โดยที่ α กำหนดให้มีค่าเท่ากับ 0.5 การทดลองของเราแสดงให้เห็นว่าค่านี้ของ α สามารถจัดการกับ delay variation ที่เกิดจาก $\sum_{i=1, i \neq m}^H D_{queue}^i$ ในสมการที่ (2) ได้ด้วยวิธีนี้ตัวควบคุมจะสามารถจำแนกการสูญเสียแพ็คเกจที่เกิดจากจุดติดขัดที่มีแบนด์วิธจำกัดได้ ถ้าการสูญเสียแพ็คเกจเกิดขึ้นบนลิงค์ที่ไม่ติดขัดและมีแบนด์วิธขนาดใหญ่ตัวควบคุมมีทางเลือกได้หลายทางที่จะจัดการกับความแออัดและปรับปรุงสมรรถนะการทำงานให้เหมาะสมที่สุด ค่าถามที่สำคัญคือจะเป็นอย่างไรถ้าตัวควบคุมจำแนกค่า D_{loss} ไม่ถูกต้อง คำตอบคือเป็นไปได้

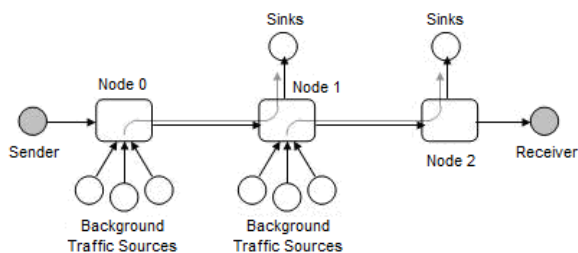
2 กรณีคือ กรณีแรกจากสมการที่ (2) ถ้า $\sum_{i=1, i \neq m}^H D_{queue}^i$ มีส่วนเกี่ยวข้องกับ peak delay มาก ตัวควบคุมจะจำแนกการสูญเสียแพ็คเกจผิดพลาดโดยจำแนกการสูญเสียแบบสุ่มเป็นการสูญเสียแพ็คเกจที่จุดติดขัดที่มีแบนด์วิธจำกัด ถึงแม้ว่าจะไม่ถูกต้องก็ตาม ระบบยังคงรักษากระบวนการตอบสนองโดยทันทีไว้ กรณีที่สองถ้า D_{thres} มีค่าสูงมากและ peak delay มีค่าต่ำกว่า D_{thres} ตัวควบคุมจะจำแนกผิดพลาดโดยจำแนกการสูญเสียแพ็คเกจที่จุดติดขัดที่มีแบนด์วิธจำกัดเป็นการสูญเสียแบบสุ่ม กรณีนี้ต้องหลีกเลี่ยงเพราะระบบจะตอบสนองความแออัดที่เกิดขึ้นช้ามากดังนั้นค่า α จึงเป็นตัวแปรที่สำคัญ การศึกษาในอนาคตเกี่ยวกับ การแจกแจงค่า delay variation และการทดสอบหลายๆครั้งจะช่วยให้ค่า α ที่เหมาะสม

3. แบบจำลองและวิเคราะห์ผลการจำลอง

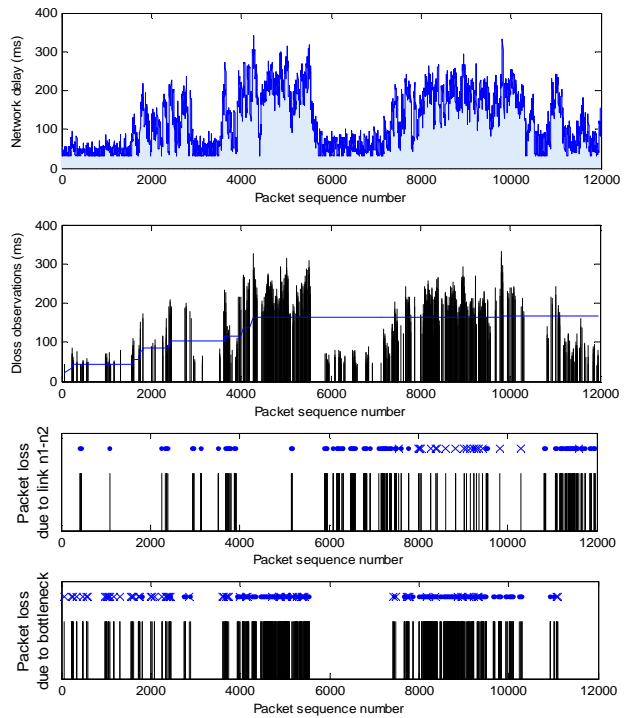
เราได้เสนอแบบจำลอง (simulation) เพื่อศึกษาวิธีการควบคุมที่ตระหนักถึงจุดติดขัดที่มีแบนด์วิธจำกัดและแสดงให้เห็น

ว่าระบบทำงานได้อย่างไร การจำลองใช้โปรแกรม Network Simulation 2 หรือ ns-2 [12] โครงสร้างเครือข่ายแสดงดังรูปที่ 1 ประกอบด้วยจุดต่อ (node) 3 จุด โดยทุกจุดต่อมีการทำงานแบบ FIFO (First In First Out) และการทำคิวแบบ drop-tail (drop-tail queuing) เช่นเดียวกับเครือข่ายที่ใช้ในปัจจุบันซึ่งเป็นแบบ best effort service ลิงค์ระหว่างจุดต่อ 0 และ 1 มีปริมาณแบนด์วิดท์ขนาด 10 Mbps และสมมุติให้เป็นแกนเครือข่าย (core network) ที่มีการใช้ร่วมของเส้นทางส่งข้อมูลจำนวนมาก ลิงค์ระหว่างจุดต่อ 1 และ 2 มีปริมาณแบนด์วิดท์จำกัดและกำหนดให้เป็นจุดติดขัดเพื่อให้แบบจำลองสมจริงมากขึ้นในแต่ละลิงค์มีจำนวน background traffic ซึ่งเป็นสาเหตุทำให้การสูญเสียแพคเกจเกิด background traffic เป็นข้อมูลแบบ ON/OFFซึ่งมีรูปแบบตามค่าตัวแปรของการแจกแจงแบบพาราได (Pareto distribution) ค่าเท่ากับ 1.5 [13] เราใช้แหล่งจ่าย 9 ตัวบนแต่ละจุดต่อสร้าง background traffic ให้เป็นไปตามการแจกแจงที่ศึกษาจาก [14,15] คือ 60 เปอร์เซ็นต์ของแพคเกจมีขนาด 40 ไบท์ 25 เปอร์เซ็นต์มีขนาด 550 ไบท์ และ 15 เปอร์เซ็นต์มีขนาด 1500 ไบท์ ในส่วนแพคเกจของโหนดมีการกระจายดังนี้ 7 เปอร์เซ็นต์มีขนาด 40 ไบท์ 35 เปอร์เซ็นต์มีขนาด 550 ไบท์และ 58 เปอร์เซ็นต์มีขนาด 1500 ไบท์

ตัวส่งและตัวรับที่จุดต่อแรกและจุดต่อสุดท้ายตามลำดับ โดยข้อมูลเดินทางไปยังทุกจุดต่อ ตัวส่งกำหนดให้ส่งข้อมูลด้วยอัตราคงที่มีค่าเท่ากับ 120 Kbps เราสมมุติให้โปรแกรมประยุกต์มีปริมาณการใช้ข้อมูลเสียงหรือวิดีโอ ตัวรับสามารถวัดค่า one-way network delay โดยใช้เวลา timestamp ของแพคเกจ ปัญหาที่สำคัญอันหนึ่งในการหาค่าเฉลี่ยคือสัญญาณนาฬิการะหว่างตัวส่งกับตัวรับไม่ตรงกัน ดังนั้นตัวส่งและตัวรับจำเป็นต้องกำหนดเวลาให้ตรงกันสำหรับวัดค่า one-way delay เช่นการใช้ระบบ GPS (Global Positioning System) หรือโปรโตคอล NTP (Network Time Protocol) และอัลกอริทึมอื่นๆสามารถนำมาใช้ได้เช่นกัน [16] เราโปรแกรมตัวรับให้มีการใช้งานการควบคุมที่ทะหนักถึงจุดติดขัดที่มีแบนด์วิดท์จำกัดและพยายามจำแนกการสูญเสียแพคเกจที่เกิดจากจุดติดขัดที่มีแบนด์วิดท์จำกัด โดยการสังเกตค่า D_{loss}



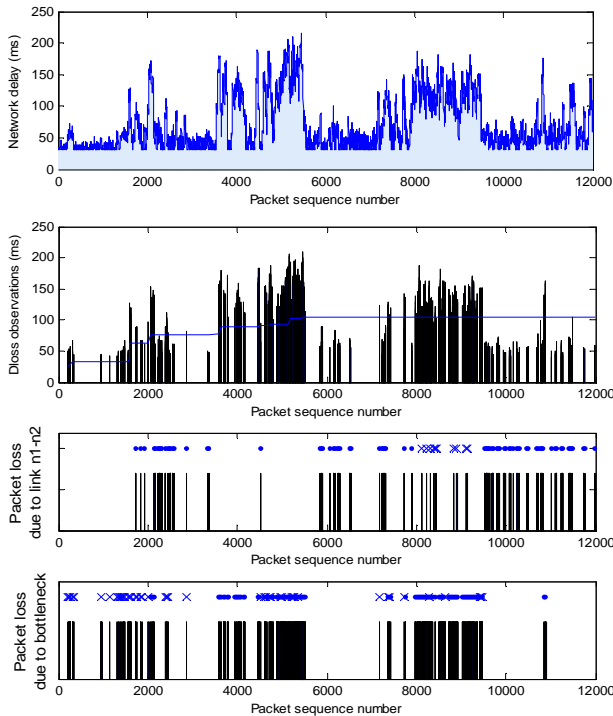
รูปที่ 1. แสดงโครงสร้างแบบจำลองเครือข่าย



รูปที่ 2. แสดงผลการจำลองกับลิงค์จุดติดขัดขนาด 512Kbps

(a) แสดง network delay ของแพคเกจที่รับมา (b) แสดง D_{loss} (c) แสดงการสูญเสียแพคเกจที่เกิดจากลิงค์ n0-n1 (d) แสดงการสูญเสียแพคเกจที่เกิดจากจุดติดขัด

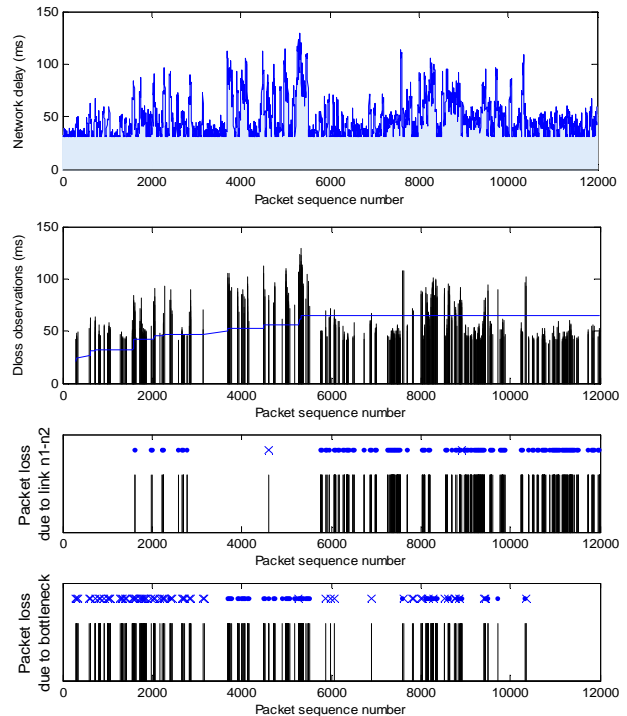
รูปที่ 2, 3 และ 4 เป็นผลการจำลองโดยที่ลิงค์ติดขัดเป็น 512Kbps 1Mbps และ 2Mbps ตามลำดับ แต่ละภาพประกอบด้วยรูป (a) แสดง one-way network delay ของแพคเกจที่รับมา รูป (b) แสดงค่า D_{loss} และ D_{thres} รูป (c) แสดงการสูญเสียแพคเกจจากลิงค์ n0-n1 รูป (d) แสดงการสูญเสียแพคเกจที่ลิงค์จุดติดขัด โดยที่จุดด้านบนแสดงการจำแนกการสูญเสียแพคเกจที่ถูกต้องและภาพทางด้านบนแสดงการจำแนกที่ผิดพลาด ตัวอย่างรูป (c) จุดแสดงการสูญเสียแพคเกจที่เกิดขึ้นบนลิงค์ n0-n1 และภาพบนแสดงการสูญเสียแพคเกจที่ไม่เกิดขึ้นบนลิงค์ n0-n1 แต่เกิดบนจุดติดขัด รูป (d) จุดแสดงการสูญเสียแพคเกจที่เกิดขึ้นที่จุดติดขัดและภาพบนแสดงการสูญเสียแพคเกจที่เกิดขึ้นบนลิงค์ n0-n1 ผลของแบบจำลองแสดงให้เห็นว่า วิธีการควบคุมจุดติดขัดที่มีแบนด์วิดท์จำกัดสามารถจำแนกการสูญเสียแพคเกจได้อย่างถูกต้อง โดยเฉลี่ยมากกว่า 60 เปอร์เซ็นต์สังเกตได้จากรูป (b) เริ่มต้นตัวควบคุมยังไม่ทราบค่า peak delay ดังนั้นตัวควบคุมต้องสร้างตัวตัดสินใจจากค่า D_{loss} ในอดีต ผลในรูป (d) ช่วงแรกตัวควบคุมจำแนกการสูญเสียแพคเกจผิดพลาดโดยมองว่าเป็นการสูญเสียที่เกิดจากจุดติดขัด ถึงแม้ว่าจะไม่ถูกต้องแต่ตัวควบคุมจะมองทุกแพคเกจเป็นการสูญเสียที่ลิงค์จุดติดขัด



รูปที่ 3. แสดงผลการจำลองกับลิงค์จุดติดขัดขนาด 1Mbps
 (a) แสดง network delay ของแพ็คเก็ตที่รับมา (b) แสดง D_{loss}
 (c) แสดง แพ็คเก็ตสูญเสียนี่เกิดจากลิงค์ n0-n1 (d) แสดงแพ็คเก็ตสูญเสียนี่เกิดจากจุดติดขัด

ซึ่งไม่เป็นข้อเสียแต่อย่างใด สิ่งสำคัญคือตัวควบคุมต้องไม่จำกัดการสูญเสียนี่เกิดจากจุดติดขัดเป็นการสูญเสียนี่ไม่เกิดจากจุดติดขัด ผลในรูป (c) แสดงให้เห็นว่าตัวควบคุมสามารถประสบผลสำเร็จและมีข้อผิดพลาดเพียงเล็กน้อยเท่านั้น

เมื่อทำการเพิ่มปริมาณแบนด์วิดท์ที่ลิงค์จุดติดขัดจาก 512Kbps , 1Mbps และ 2Mbps จากผลการจำลองในรูป (d) ตัวควบคุมมีแนวโน้มการจำกัดการสูญเสียนี่บนลิงค์ n0-n1 ผิดพลาดเป็นการสูญเสียนี่เกิดจากจุดติดขัด เมื่อลิงค์จุดติดขัดมีแบนด์วิดท์มากขึ้น เครือข่ายดูเหมือนไม่มีลิงค์แบนด์วิดท์จำกัด ดังนั้นผลของ delay variation ทำให้การจำกัด peak delay ที่เกิดจากจุดติดขัดทำได้ยากขึ้น นี้ไม่ใช่ปัญหาเพราะระบบจะมองทุกการสูญเสียนี่แพ็คเก็ตเป็นการสูญเสียนี่เกิดจากจุดติดขัด ในทำนองเดียวกันเมื่อทุกลิงค์มีแบนด์วิดท์ที่จำกัด ค่า delay variation จะทำให้การจำกัด peak delay ทำได้ยากเช่นกัน ผลการจำลองแสดงว่ากระบวนการที่นำเสนอสามารถทำงานได้อย่างดีและมีประสิทธิภาพเมื่อเครือข่ายมีจุดติดขัดที่มีแบนด์วิดท์จำกัด



รูปที่ 4. แสดงผลการจำลองกับลิงค์จุดติดขัดขนาด 2Mbps
 (a) แสดง network delay ของแพ็คเก็ตที่รับมา (b) แสดง D_{loss}
 (c) แสดง แพ็คเก็ตสูญเสียนี่เกิดจากลิงค์ n0-n1 (d) แสดงแพ็คเก็ตสูญเสียนี่เกิดจากจุดติดขัด

4. บทสรุป

การสูญเสียนี่แพ็คเก็ตบนเครือข่ายแพ็คเก็ตแบบสวิตซ์เป็นที่รู้จักโดยทั่วไปว่าส่วนใหญ่มีสาเหตุมาจากเครือข่ายเกิดความแออัดแต่อย่างไรก็ตาม การตรวจสอบเพียงการสูญเสียนี่แพ็คเก็ตเพียงอย่างเดียวเป็นข้อมูลที่จำกัดมากที่ใช้แสดงความแออัดบนเครือข่าย บทความนี้ได้พยายามจำกัดการสูญเสียนี่แพ็คเก็ตซึ่งเกิดจากความแออัดเพื่อเรียนรู้สถานะของเครือข่าย โดยการสังเกตความสัมพันธ์ของ delay และการสูญเสียนี่แพ็คเก็ต เราแสดงความเป็นไปได้ในการจำกัดระหว่างการสูญเสียนี่แพ็คเก็ตซึ่งเกิดจากจุดติดขัดที่มีแบนด์วิดท์จำกัดและการสูญเสียนี่มีลักษณะแบบสุ่มเนื่องจากการใช้ร่วมเส้นทางข้อมูลจำนวนมากในลิงค์เดียวกัน จากนั้นเราได้เสนอวิธีที่เรียกว่าการควบคุมที่ตระหนักถึงจุดติดขัดที่มีแบนด์วิดท์จำกัด ผลจากแบบจำลองแสดงให้เห็นว่าวิธีการที่ได้นำเสนอทำงานได้ดีและสามารถจำกัดสาเหตุของความแออัดได้ โดยเฉพาะอย่างยิ่งกรณีเครือข่ายเป็นแบบจุดติดขัดที่มีแบนด์วิดท์จำกัด

วิธีการควบคุมที่ตระหนักถึงจุดติดขัดที่มีแบนด์วิดท์จำกัด (limited-bandwidth aware control) ที่นำเสนอในบทความนี้

สามารถนำไปประยุกต์ใช้ในระบบควบคุมการติดขัดสำหรับการส่งข้อมูลประเภท Streaming media เช่น ข้อมูลเสียง และวิดีโอ ซึ่งโดยปกติมีการส่งผ่าน UDP (User Datagram Protocol) ที่ไม่มีการตอบสนองต่อการติดขัดบนเครือข่าย วิธีที่นำเสนอจะช่วยให้ตัวรับฉลาดขึ้น เพราะถ้าเกิดการสูญเสียแพคเกจบนเครือข่ายที่จุดติดขัดที่มีแบนด์วิดท์จำกัดระบบจะลดการส่งทันที เพื่อหลีกเลี่ยงการสูญเสียแพคเกจมากขึ้น แต่ถ้าไม่ใช่ ตัวส่งอาจไม่จำเป็นต้องลดการส่งข้อมูลทันที ชุดท้ายระบบจะช่วยให้การส่งข้อมูลมีประสิทธิภาพมากขึ้น และช่วยลดการติดขัดบนเครือข่ายได้

เอกสารอ้างอิง

- [1] V. Jacobson, "Congestion Avoidance and Control," Proceedings of ACM SIGCOMM, *Computer Communication Review*, Aug. 1988.
- [2] M. Mathis et al., "TCP Selective Acknowledgement Options," RFC 2018, Oct. 1996.
- [3] C. P. Fu and S. C. Liew, "TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks," *IEEE JSAC*, vol. 21, Feb. 2004.
- [4] K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for Wireless IP Communications," *IEEE JSAC*, vol. 22, May 2004.
- [5] R. Rejaie, M. Handley, and D. Estin, "RAP: An End-to-end Rate-based Congestion Control Mechanism for Real-time Streams in the Internet," *Proc. IEEE INFOCOM*, Mar. 1999.
- [6] J. C. Bolot, "End-to-end Packet Delay and Loss Behavior in the Internet," *ACM SIGCOMM*, 1993.
- [7] K. Fall and S. Floyd, "Simulation-Based Comparisons of Tahoe, Reno and SACK TCP," *Computer Communication Review*, vol. 26, Jul. 1996.
- [8] L.S. Brakmo, S. W. O'Malley, and L. Peterson, "TCP Vegas: New Techniques for Congestion Detection and Avoidance," *Proc. ACM SIGCOMM*, Aug. 1994.
- [9] S. B. Moon, "Measurement and Analysis of End-to-End Delay and Loss in the Internet," Ph.D. Thesis, Univ. Massachusetts, Amherst, 200.
- [10] V. Paxson, "Measurement and Analysis of End-to-End Internet Dynamics," Ph.D. Thesis, Univ. California, Berkeley, 1997.
- [11] L. Roychoudhuri, E. Al-Shaer, H. Hamed, and G. B. Brewster, "On Studying the Impact of the Internet Delays on Audio Transmission," *IEEE Workshop on IP Operations and Management*, 2002.
- [12] The Network Simulator – ns-2, <http://www.isi.edu/nsnam/ns/>.
- [13] Y. R. Yang, M. Kim, and S. Lam, "Transient Behaviors of TCP-Friendly Congestion Control Protocols," *Proc. IEEE INFOCOM*, Apr. 2001.
- [14] K. Claffy, G. J. Miller, and K. Thompson, "The Nature of the Beast: Recent Traffic Management from an Internet Backbone," *Proc. INET'98*, Geneva, Switzerland, Jul. 1998.
- [15] K. Thompson, G. J. Miller, and R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics," *IEEE Network*, Nov. 1997.
- [16] V. Paxson, "On Calibrating Measurements of Packet Transit Times," *Proc. ACM SIGMETRICS*, Jun. 1998.