บทที่ 1

บทนำ

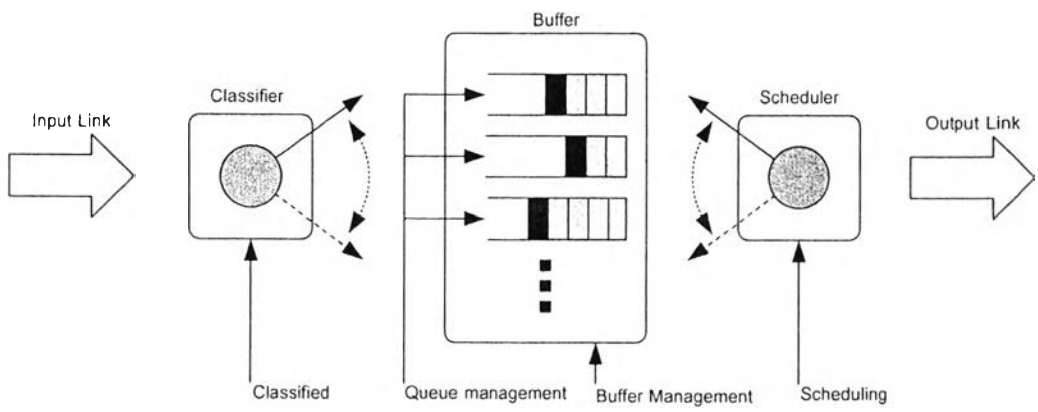
สำนักงานวิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากว่าเครือข่ายอินเทอร์เน็ตในปัจจุบันได้มีการพัฒนาอย่างรวดเร็ว ส่งผลให้มีแอปพลิเคชัน (application) ต่างๆเกิดขึ้นอย่างมากมาย ดังนั้นจึงมีข้อมูลที่วิ่งผ่านโครงข่ายมีหลากหลายประเภทตามไปด้วย ซึ่งแต่ละประเภทนั้นมีความต้องการใช้ทรัพยากรของโครงข่ายที่ไม่เท่ากัน และเนื่องจากว่าในปัจจุบันทางด้านผู้ให้บริการ (provider) ไม่ได้พิจารณาว่าข้อมูลแต่ละประเภทนั้นต้องการใช้ทรัพยากรที่ต่างกันทำให้ทางด้านผู้ให้บริการ ทำการจัดสรรแบนด์วิดท์ (bandwidth) สำหรับข้อมูลทุกประเภทเท่าๆกัน (ไม่แบ่งแยกตามประเภท) ซึ่งทำให้ผู้ให้บริการบางรายใช้งานแบนด์วิดท์อย่างไม่คุ้มค่า

เพื่อแก้ปัญหาตรงจุดนี้จึงได้มีการกำหนดการให้บริการที่แตกต่าง (differentiated service) [1] ขึ้นมา โดยจะทำการแบ่งประเภทของข้อมูลออกเป็นลำดับชั้น (class) และใช้กรรมวิธี เช่น การจัดความสำคัญในการจัดสรรลำดับแบบแถวคอย (priority queuing) ที่เราเตอร์หลัก (core router) เพื่อทำการจัดสรรลำดับแบบแถวคอย (queue) ที่เราเตอร์ ซึ่งเราเตอร์จะทำการส่งข้อมูลออกไปยังโครงข่ายต่อไปดังรูปที่ 1.1 และในระบบโครงข่าย TCP/IP คุณสมบัติสำคัญที่เกี่ยวข้องกับประสิทธิภาพของโครงข่ายคือ ความเท่าเทียม (fairness) โดยความเท้าเทียมจะมีความสำคัญกับโครงข่ายก็ต่อเมื่อเกิดความแออัด (congestion) ในโครงข่ายบริเวณส่วนที่แคบที่สุดซึ่งเรียกว่าคอขวด (bottleneck) จึงจำเป็นต้องมีการควบคุมให้แต่ละโฟลว์ (flow) ไหลผ่านไปอย่างราบรื่น โดยคอยควบคุมไม่ให้แต่ละโฟลว์ แย่งชิงแบนด์วิดท์ที่มีอยู่จำกัดเป็นเหตุให้เกิดโฟลว์อื่นๆ ได้ใช้แบนด์วิดท์ได้น้อยกว่าที่ควร ซึ่งส่งผลให้ประสิทธิภาพโดยรวมทั้งโครงข่ายต่ำลงดังนั้นจึงจำเป็นต้องควบคุมให้แต่ละโฟลว์แบ่งกันใช้แบนด์วิดท์ที่เท่าๆกัน ดังนั้นเมื่อพิจารณา differentiated service ร่วมกับความเท่าเทียมแล้ว จึงทำให้เกิดการวิจัยทางด้านการถ่วงน้ำหนักอย่างเท่าเทียมแบบสัดส่วน (weighted proportional fairness) ขึ้นมาโดยการถ่วงน้ำหนัก (weighted) จะขึ้นอยู่กับราคา (price) [2, 8] ในบทความ [10] ได้กำหนดให้อัตราการส่งข้อมูลไม่ได้ขึ้นกับค่าของความเท่าเทียมแต่ขึ้นอยู่กับราคาที่จ่ายต่อหน่วยของอัตราความเร็ว (price amount paid per rate) โดยที่ผู้ใช้ (user) ที่จ่าย 2 หน่วยจะได้อัตราการส่งข้อมูลเท่ากับในกรณีของที่ผู้ใช้ 2 คนโดยที่แต่ละคนจ่ายเพียง 1 หน่วยรวมกัน แต่ในงานวิจัย [2] ที่กล่าวไว้ข้างต้นไม่ได้พิจารณาถึงความแออัดในโครงข่ายขณะนั้นเลยจึงส่งผลให้ต้องทำการส่งซ้ำข้อมูลที่ถูกทิ้งเนื่องจากความแออัดของโครงข่ายอยู่เสมอ

ซึ่งในงานวิจัยทางด้านการให้บริการที่แตกต่างส่วนมาก [1, 3] เน้นไปทางด้านการจัดลำดับแบบแถวคอยที่เราเตอร์โดยจะแบ่งโฟลว์ที่เข้ามายังเราเตอร์ทั้งหมดออกเป็นกลุ่มๆที่เรียกว่าลำดับชั้น โดยอาศัยฟิลด์ในส่วนของเฮดเดอร์ DS (differentiated service header) เป็นหลักเกณฑ์ในการแบ่งและโฟลว์ที่ถูกแบ่งนั้นจะถูกจัดลำดับการส่งออกไปยังเราเตอร์ที่อยู่ถัดไปตามลำดับตามความสำคัญ (priority) ของลำดับชั้นที่ถูกแบ่งไว้และจำเป็นต้องมีการจัดส่งข้อมูลแบบ per-hop ซึ่งเป็นการจัดการระหว่างคู่โหนดเท่านั้น โดยที่คู่โหนดอื่นๆไม่มีความสัมพันธ์กันเลย จึงจำเป็นต้องอาศัยการควบคุมของส่วนกลาง (CAC) มาคอยจัดการในการส่งผ่านข้อมูลเหล่านั้นไปยัง hop อื่นๆ ดังนั้นจึงได้ศึกษาและทำการวิจัยโดยจัดการในระดับของโฟลว์ (flow) แทน ซึ่งเป็นการควบคุมการไหลแบบปลายถึงปลาย (end-to-end flow control) ซึ่งอาศัยการถ่วงน้ำหนักของแต่ละโฟลว์ โดยจะปรับเปลี่ยนอัลกอริทึมของการเริ่มต้นอย่างช้าๆ (slow start) และช่วงหลีกเลี่ยงความแออัด (congestion avoidance) ของการส่งข้อมูลด้วย TCP แต่ในงานวิจัยที่กล่าวมาไม่ได้พิจารณาถึงเรื่องความเท่าเทียม ดังนั้นจึงได้พิจารณาการแบ่งแบนด์วิดท์อย่างเท่าเทียมกัน ซึ่งขนาดของแบนด์วิดท์ขึ้นอยู่กับจำนวนโฟลว์และขนาดแบนด์วิดท์ของลิงก์ ซึ่งการแบ่งแบนด์วิดท์ตามอัตราส่วนของแต่ละโฟลว์นี้สามารถนำไปสู่การส่งข้อมูลบนโครงข่ายโดยรวมอย่างมีประสิทธิภาพ



รูปที่ 1.1 แสดงการทำงานของวิธีการจัดสรรลำดับแบบแถวคอย (Priority queue management)

จากกรรมวิธีการส่งข้อมูลแบบ Tahoe TCP และ Reno TCP จะเห็นได้ว่าจะไม่มีการจำกัดขนาดของหน้าต่างความแออัด (congestion window : cwnd) ของผู้ใช้แต่ละคน ในช่วงที่ใช้เทคนิคหลีกเลี่ยงความแออัด ที่ผู้ใช้แต่ละคนจะส่งข้อมูลตามขนาดของ cwnd ที่เพิ่มมาทุก รอบของเวลาจริงในการเข้าถึงข้อมูล (round trip time : RTT) ไปเรื่อยๆจนกระทั่งทำให้ข้อมูลที่ส่งมายังเราเตอร์ที่อยู่บริเวณคอขวดมีขนาดเกินกว่า บัฟเฟอร์ที่มีอยู่สามารถรองรับได้ จึงทำให้ต้องทำการ

ทิ้งข้อมูลในส่วนที่มีขนาดเกินกว่าขนาดของบัฟเฟอร์ทิ้งไป ซึ่งส่งผลให้ประสิทธิภาพโดยรวมของโครงข่ายลดลงเนื่องจากต้องทำการส่งซ้ำข้อมูลที่ถูกทิ้งหรือสูญหายใหม่อีกครั้ง ด้วยเหตุนี้แสดงให้เห็นว่าสมควรที่จะมีข้อมูลที่บ่งบอกสถานะปัจจุบันของโครงข่ายขึ้น [9] เพื่อที่จะให้ผู้ใช้ต้นทางสามารถรับรู้สถานะหรือสภาพของเส้นทางที่ส่งข้อมูลไปยังปลายทาง เช่น ขนาดของบัฟเฟอร์ที่เราเตอร์หรือขนาดของแบนด์วิดท์บริเวณคอขวด และเนื่องจากว่าข้อมูลในโครงข่ายมีหลากหลายประเภท ซึ่งแต่ละประเภทมีความต้องการแบนด์วิดท์ หรืออัตราการส่งข้อมูลที่ต่างกัน และมีการวิจัยทางด้านการให้บริการที่แตกต่าง ซึ่งเป็นการจัดหรือแบ่งกลุ่มข้อมูลที่มีคุณลักษณะคล้ายๆกันออกเป็นลำดับชั้นและทำการจัดลำดับแบบแถวคอย โดยให้ความสำคัญของแต่ละลำดับชั้นที่แตกต่างกัน ซึ่งวิธีนี้จำเป็นต้องมีการจับจองหรือจัดการแบนด์วิดท์บนพื้นฐานของลำดับชั้น แทนที่จะต้องมีการจับจองหรือจัดการบนพื้นฐานของโฟลว์ โดยส่วนมากแล้วในการให้บริการที่ระดับสูงๆ หรือบริการที่มีความสำคัญมากๆ (premium service) จำเป็นต้องมีการจองแบนด์วิดท์ ที่มีขนาดใหญ่เกินความจำเป็นเพื่อให้แน่ใจว่าสามารถมีแบนด์วิดท์ที่สามารถเรียกใช้ได้ทันทีที่ต้องการ กรรมวิธีเหล่านี้จำเป็นต้องการควบคุมส่วนกลาง (CAC) มาคอยควบคุมไม่ให้ผู้ใช้งานที่ใช้บริการในระดับ บริการที่มีความสำคัญมากๆ มีมากเกินไป [4] เนื่องจากจะส่งผลกระทบกับบริการในส่วนอื่นๆ และมีงานวิจัยที่ได้เปิดประเด็นทางด้านการควบคุมอัตราอย่างชัดแจ้งถ่วงน้ำหนักแบนด์วิดท์ของแต่ละโฟลว์ขึ้นมา [2, 8] ซึ่งปัจจัยของการถ่วงน้ำหนักในงานวิจัย [10] นี้ขึ้นอยู่กับราคาโดยที่ผู้ใช้ใดๆที่ให้ราคาที่สูงมีสิทธิในการส่งข้อมูลผ่านคอขวดด้วยอัตราการส่งที่มากกว่าผู้ใช้รายอื่นๆ ตามอัตราส่วนซึ่งขึ้นอยู่กับราคาของตัวเองกับผู้ใช้รายอื่นๆ ที่เสนอมาในขณะนั้น แต่ในงานวิจัยที่กล่าวมานี้ สนใจเฉพาะการถ่วงน้ำหนักอย่างเดียวเท่านั้นโดยไม่สนใจแก้ไขเรื่องผลกระทบของแพ็กเกตที่ถูกทิ้งที่มีสาเหตุมาจากบัฟเฟอร์ของเราเตอร์ที่คอขวดไม่สามารถรองรับแพ็กเกตที่เข้ามาได้ทำให้ต้องสูญเสียสมรรถภาพในการส่งข้อมูลไปส่วนหนึ่งเนื่องมาจากการส่งซ้ำของแพ็กเกต

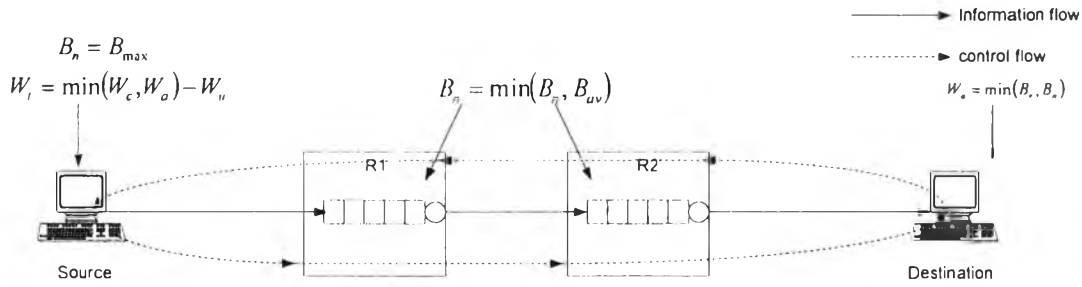
ดังนั้นจึงได้ศึกษางานวิจัยที่เกี่ยวข้องกับการควบคุมแบบป้อนกลับ (feedback control) ซึ่งมีต้นแบบมาจากการศึกษาโมเดลของพลศาสตร์ (fluid dynamic) [5] โดยสมมุติว่าที่เวลา t และ จุดต้นทาง S ส่งข้อมูลออกไปด้วยอัตราเท่ากับ $r_s(t)$ ผ่านเส้นทางต่างๆไปยังจุดปลายทาง D อัตราการการประมวลของข้อมูลที่จุดปลายทางมีค่าเท่ากับ $r_d(t)$ เมื่อ $r_s(t)$ และ $r_d(t)$ เป็นตัวแปรที่ขึ้นอยู่กับเวลา ซึ่งจะมีบัฟเฟอร์ที่จุดปลายทางเอาไว้กักเก็บข้อมูลไว้สำหรับก่อนที่จะประมวลผลกำหนดให้อัตราการไหลเข้าของข้อมูลมีค่าเท่ากับ $r_a(t)$โดยที่ $r_a(t) = r_s(t - \delta t)$ เมื่อ δt คือเวลาที่ถูกประวิงเนื่องจากเวลาที่สูญเสียไปเนื่องจากการส่งข้อมูลจากต้นทางไปยังปลายทาง ถ้ากำหนดให้ R_d คือค่าขอบเขตบนหรือค่าที่มากที่สุดที่เป็นไปได้ของ $r_d(t)$ที่จุดปลายทาง เมื่ออัตราการไหลเข้าของข้อมูล $r_a(t) \geq R_d$ หรือในกรณีที่บัฟเฟอร์ไม่ว่างจุดปลายทางจะต้องประมวลผลข้อมูลด้วยอัตราเท่ากับ R_d และจุดปลายทางจะต้องประมวลผลข้อมูลด้วยอัตราเท่ากับ

$r_d(t) = r_a(t)$ ก็ต่อเมื่ออัตราการไหลเข้าของข้อมูล $r_a(t) < R_d$ หรือในกรณีที่บัฟเฟอร์นั้นว่างอยู่ เพื่อหลีกเลี่ยงการโอเวอร์โฟลว์ (overflow) ที่บัฟเฟอร์เกิดขึ้น ขนาดของบัฟเฟอร์ในที่นี้กำหนดให้เป็นค่า C ควรจะมีขนาดใหญ่กว่าขนาดของ $q(t)$ โดยที่

$$q(t) = \int_0^t [r_s(\tau - \delta t) - r_d(\tau)] d\tau. \qquad (1.1)$$

ซึ่งสมการนี้อยู่บนสมมุติฐานที่ว่า ให้อัตราการประมวลผลที่มากที่สุดถูก R_d กำหนดให้เป็นค่าคงที่ ดังนั้นเรา จำเป็นต้องควบคุมอัตราการส่งข้อมูล $r_s(t)$ เพื่อให้ $q(t)$ มีขนาดเล็กกว่า C เพื่อป้องกันการโอเวอร์โฟลว์เกิดขึ้น ในทางคณิตศาสตร์เราสามารถที่จะหาผลเฉลยของ $r_s(t)$ โดยการแปลงลาปาซของโมเดลการไหลอย่างต่อเนื่อง (continuous fluid) และเราจะเห็นว่าผลเฉลยนั้นจะแปรผันโดยตรงกับค่าของ $q(t)$ ซึ่งแสดงให้เห็นว่าถ้าต้องการจะควบคุมอัตราการส่งข้อมูลจุดต้นทางควรที่จะทราบข้อมูลที่บ่งบอกถึงความจุของขนาดบัฟเฟอร์ที่จุดปลายทาง

ซึ่งในปัจจุบันได้มีการนำโมเดลของระบบที่ไม่ต่อเนื่อง (discrete system) มาประยุกต์ใช้กับการส่งข้อมูลของ TCP ในระบบอินเทอร์เน็ต ซึ่งในบทความ [7] ได้เสนอแนะว่าควรจะมีการป้อนกลับ (feedback) ที่บ่งบอกขนาดของบัฟเฟอร์ที่เราเตอร์ เพื่อให้ต้นทางสามารถที่จะควบคุมขนาดของหน้าต่างให้มีขนาดที่พอเหมาะเพื่อที่จะไม่เกิดการสูญหายของข้อมูลเนื่องจากการแออัดที่เกิดขึ้นที่โครงข่าย [7] แต่อย่างไรก็ตามยังมีปัญหาที่เกิดขึ้นเนื่องจากการจัดสรรแบนด์วิดท์ที่ไม่เท่าเทียมแต่สามารถแก้ไขได้โดยวิธีการจัดลำดับแบบแถวคอยต่อโฟลว์ (per-flow queuing) ที่ใช้กรรมวิธีของ round-robin scheduling ซึ่งจากงานข้างต้นทำให้เกิดกรรมวิธี Generalized Window Advertising (GWA) สำหรับใช้ควบคุมความแออัดที่เกิดขึ้นในโครงข่ายในระบบ TCP แนวความคิดพื้นฐานของ GWA แสดงให้เห็นดังรูปที่ 1.2 ที่จุดต้นทาง S แต่ละแพ็กเกตของข้อมูลจะถูกพ่วงด้วยข้อมูลที่บ่งบอกขนาดของบัฟเฟอร์ B_n ที่ถูกกำหนดให้มีค่าเริ่มต้นเท่ากับค่าสูงสุดค่าหนึ่งในที่นี้แทนด้วย B_{max} เมื่อแพ็กเกตถูกส่งไปยังเราเตอร์ $R1$ จะปรับปรุงค่า B_n ถ้าพบว่าขนาดของบัฟเฟอร์ที่มีอยู่ในที่นี้กำหนดให้มีค่าเท่ากับ B_{av} มีค่าน้อยกว่าขนาด B_n และเมื่อแพ็กเกตถูกส่งผ่านไปยังเราเตอร์ $R2$ เราเตอร์ $R2$ ก็จะทำการปรับปรุงค่า B_n ด้วยวิธีเดียวกัน เมื่อเพ็ตเกตถูกส่งไปจุดปลายทาง D ทางด้านผู้รับปลายทางจะทำการเปรียบเทียบขนาดบัฟเฟอร์ B_r ด้วยขนาดบัฟเฟอร์ B_n และจะใช้ค่าที่มีขนาดเล็กที่สุดบรรจุในหน้าต่างการประกาศ (advertised window) ทางด้านผู้รับในเฮดเดอร์ของ TCP ถึงแม้ว่าทางผู้ส่งได้รับแพ็กเกตตอบรับ (acknowledgment packet : ack) ทางด้านผู้ส่งก็คงยังใช้หน้าต่างการประกาศที่ทางด้านผู้รับประกาศมาเพื่อที่จะควบคุมขนาดของหน้าต่างของความแออัด [7] ในกรรมวิธีของ GWA ข้อมูลที่ถูกส่งไปในหน้าต่างการประกาศจะประกอบไปด้วยข้อมูลที่ระบุถึงขนาดของบัฟเฟอร์ทางด้านผู้รับและข้อมูลที่ระบุถึงขนาดบัฟเฟอร์ของเน็ตเวิร์ค



รูปที่ 1.2 แนวคิดพื้นฐานของ GWA

จากที่กล่าวมาข้างต้นจะเห็นว่าแบนด์วิดท์ มีความสำคัญอย่างยิ่งในการควบคุมความแออัดในโครงข่าย (เนื่องจากว่าในโครงข่ายที่มีค่าของค่าประวิงเวลาการแผ่กระจาย (propagation delay) นานๆ ควรจะมีขนาดบัฟเฟอร์ที่มากขึ้นตามไปด้วยอย่างน้อยควรมีค่าเท่ากับ ผลคูณของค่าประวิงเวลาจริงในการเข้าถึงข้อมูล (round trip time delay) และขนาดของแบนด์วิดท์เพื่อให้มี utilization ของแบนด์วิดท์ที่สูง [6]) และเมื่อพิจารณาร่วมกับการให้บริการที่แตกต่างแล้ว จึงเสนอกรรมวิธีในการควบคุมความแออัดในโครงข่ายในรูปแบบของข้อมูลที่มีหลากหลายประเภทและควบคุมการแบ่งแบนด์วิดท์โดยอาศัยการควบคุมอัตราอย่างชัดแจ้ง ซึ่งพิจารณาตามความต้องการของผู้ใช้บริการอย่างเท่าเทียม โดยอาศัยเฮดเดอร์ส่วนขยาย (extension header) ที่ปัจจุบันยังไม่ได้กำหนดการใช้งานอย่างชัดเจนของอินเทอร์เน็ตโพรโทคอลรุ่นที่ 6 ส่วนการกำหนดขนาดของแบนด์วิดท์จะเป็นหน้าที่ของเราเตอร์ซึ่งขนาดของแบนด์วิดท์ของแต่ละผู้ใช้ โดยจะทำการปรับเปลี่ยนขนาดของแบนด์วิดท์ทุกครั้งที่มีการหยุดหรือเริ่มต้นการส่งข้อมูลระหว่างผู้ใช้ปลายทางและต้นทาง หรือในกรณีที่มีผู้ใช้รายใหม่เพิ่มเข้ามาในโครงข่าย โดยจะทำการจำลองเสมือนจริงบนโทโพโลยี (topology) ที่แตกต่างกัน 3 รูปแบบเพื่อวิเคราะห์และเปรียบเทียบถึงประสิทธิภาพที่เกิดขึ้น

เมื่อพิจารณาการสูญหายของข้อมูลในการสื่อสารผ่านโครงข่ายไร้สายจะเห็นว่าการสูญหายนั้นเกิดขึ้นในอัตราที่สูงอันเป็นผลเนื่องมาจากคุณลักษณะบางประการของโครงข่าย ซึ่งจะไปกระตุ้นกรรมวิธีการควบคุมความแออัดในโครงข่ายให้ลดขนาดของหน้าต่างความแออัดโดยไม่จำเป็นซึ่งส่งผลให้ประสิทธิภาพลดลง ซึ่ง TCP ไม่มีกรรมวิธีที่จะแยกแยะการสูญหายแบบสุ่ม (random loss) ที่เกิดจากคุณสมบัติช่องสัญญาณไร้สาย ที่มีลักษณะสุ่มที่ไม่สามารถคาดเดาได้ออกจากการสูญหายที่เกิดจากความแออัดในโครงข่าย (congestion loss) ด้วยเทคนิค TCP แบบดั้งเดิมนั้นจะมองว่าการสูญหายที่เกิดขึ้นนั้นเป็นการสูญหายแบบเดียวกันหมดและแก้ไขการสูญหายเหล่านั้นด้วยกรรมวิธีการควบคุมความแออัดทั้งหมด ดังนั้นจึงได้มีการคิดค้นกรรมวิธีสำหรับบรรเทาการสูญหายแบบสุ่มของโครงข่ายไร้สายเกิดขึ้นหลากหลายวิธี เช่น เทคนิคข้อมูลซ้ำใหม่

หอสมุดกลาง สำนักงานวิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

จาก link layer (link layer retransmission) ที่มีชื่อเรียกว่า Snoop Protocol [13] หรือ เทคนิคการส่งข้อมูลแบบแยกช่องสัญญาณ (split TCP connection) ที่เรียกว่า I-TCP [12] หรือ เทคนิคในกลุ่มของการส่งข้อมูลจากปลายถึงปลาย (end to end mechanism) เช่น Explicit Loss Notification, Explicit Bad State Notification and Explicit Congestion Notification [11] ถ้าแบ่งตามวิธีการปรับปรุงการกู้ข้อมูลแบบ TCP ตามลักษณะของการสูญหายสามารถแบ่งได้ 2 วิธี คือ หนึ่ง คือการปิดบัง (blind) ข้อมูลที่สูญหายแบบสุ่มจากผู้ใช้ต้นทาง (sender) ไม่ให้ผู้ใช้ต้นทางเห็นเพื่อที่จะไม่ต้องแก้ไขการสูญหายโดยการลดขนาดหน้าต่างความแออัด และ สอง ให้ผู้ใช้ต้นทางคอยตรวจสอบหรือระวังการสูญหายที่ไม่ได้เกิดจากความแออัดในโครงข่าย แนวความคิดหลักๆของวิธีแรกคือเมื่อการสูญหายเกิดขึ้นในท้องถิ่น (local) ควรที่จะแก้ไขโดยท้องถิ่น โดยที่ transport layer ไม่ต้องมากังวลหรือระมัดระวังเรื่องคุณสมบัติของโครงข่ายไร้สาย ดังนั้นผู้ใช้ต้นทางจะเห็นเฉพาะการสูญหายที่เกิดขึ้นเนื่องจากความแออัดเท่านั้นและสามารถแก้ไขข้อมูลที่สูญหายด้วยกรรมวิธีการควบคุมความแออัดในโครงข่าย TCP/IP แบบดั้งเดิม ส่วนในวิธีที่สองนั้นเนื่องจากว่าทางผู้ใช้ต้นทางจะเป็นส่วนที่คอยตรวจสอบการสูญหายที่เกิดจากคุณสมบัติของช่องสัญญาณไร้สายออกจากการสูญหายที่เกิดจากความแออัด ดังนั้นจึงมีส่วนของฟังก์ชันที่ไว้แยกแยะว่าการสูญหายที่เกิดขึ้นเกิดจากจากลักษณะของช่องสัญญาณไร้สายหรือเกิดจากความแออัดในโครงข่ายและทำการกู้ข้อมูลหรือแก้ไขตามลักษณะของการสูญหายที่เกิดขึ้น

จากการเปรียบเทียบการทำงานของเทคนิคต่างๆ ในการเพิ่มประสิทธิภาพของการส่งข้อมูลและกู้คืนข้อมูลบนช่องสัญญาณไร้สายใน [14] ได้แสดงให้เห็นว่ากรรมวิธีการส่งข้อมูลและการกู้คืนข้อมูลด้วยเทคนิค Link layer retransmission ที่เรียกว่าโพรโทคอลสนูป (Snoop Protocol) มีประสิทธิภาพที่ดีกว่าเทคนิคอื่นๆ เนื่องมาจากความสามารถในการกู้คืนข้อมูลจากท้องถิ่นทั้งนี้เพราะว่าค่าประวิงเวลาการแผ่กระจายของช่องสัญญาณไร้สายระหว่างผู้รับปลายทางและโหนดสุดท้าย เมื่อเทียบกับค่าประวิงเวลาการแผ่กระจายระหว่างต้นทางถึงปลายทางของเทคนิคอื่นๆ ถือว่ามีค่าน้อยมาก และโพรโทคอลสนูปให้ประสิทธิภาพทางด้านประมาณงาน (throughput) มี utilization ที่สูงและมีการตอบสนองต่อการสูญหายแบบสุ่มของช่องสัญญาณไร้สายได้รวดเร็วกว่าวิธีอื่นๆ เมื่ออัตราการสูญหายของแพ็กเกตในช่องสัญญาณไร้สายมีค่าเพิ่มมากขึ้น

แต่อย่างไรก็ตามโพรโทคอลสนูปนั้นยังคงมีข้อเสีย เพราะว่าตามลำพังเพียงตัวมันเองสามารถกู้คืนแพ็กเกตได้เพียง 1 แพ็กเกตต่อ 1 รอบ RTT เท่านั้น (เพราะข้อจำกัดในการในการใช้แพ็กเกตตอบรับที่ใช้ลำดับหมายเลขซ้ำในการกู้คืนข้อมูล) ซึ่งการแก้ไขปัญหาในจุดนี้สามารถทำได้โดยนำมาทำงานร่วมกับการส่งข้อมูลด้วย SACK TCP แต่ในบทความ [15] ได้เปรียบเทียบการทำงานของวิธีการส่งข้อมูลด้วย TCP รูปแบบต่างๆ ทำงานร่วมกับโพรโทคอลสนูปและการส่งข้อมูล

ของ TCP รูปแบบต่างๆ ที่ไม่มีโพรโทคอลสนูปได้แสดงให้เห็นว่า SACK TCP เมื่อทำงานร่วมกับโพรโทคอลสนูปให้ผลที่ดีกว่าเมื่อเปรียบเทียบกับการส่งข้อมูลร่วมกับโพรโทคอลสนูปของ TCP รูปแบบอื่นๆ เพราะความสามารถในการกู้คืนแพ็กเกตที่สูญหายมากกว่า 1 แพ็กเกตใน 1 ช่วงของค่าประวิงเวลาจริงในการเข้าถึงข้อมูล ถึงแม้ว่าจะมีผลจากการส่งซ้ำแพ็กเกตโดยไม่จำเป็นและยังมีปัญหาในส่วนของพฤติกรรมของโพรโทคอลสนูปที่ไม่ส่งผ่านแพ็กเกตตอบรับไปยังผู้ใช้ต้นทางจนกว่าจะมีการตอบรับของแพ็กเกตที่เพิ่งได้ทำการกู้คืนจากผู้ใช้ปลายทางทำให้แพ็กเกตตอบรับไปถึงมือผู้ใช้ต้นทางนั้นขาดช่วงส่งผลกระทบต่อสมรรถนะของโครงข่ายได้ เพราะโดยพื้นฐานการทำงานของการส่งข้อมูลด้วย TCP นั้นเป็นการส่งส่งข้อมูลแบบใช้แพ็กเกตตอบรับเป็นการขับดันให้ผู้ใช้ต้นทางส่งแพ็กเกตไปยังผู้ใช้

ดังนั้นจึงได้เสนอวิธีสำหรับเพิ่มประสิทธิภาพการส่งข้อมูลและการกู้คืนข้อมูลที่สูญหายบนช่องสัญญาณไร้สายให้เป็นหน้าที่ของโพรโทคอลสนูปและทำการปิดบังการสูญหายที่เกิดขึ้นในส่วนของช่องสัญญาณไร้สายนี้ไม่ให้ทางผู้ใช้ต้นทางรู้และทำการกู้คืนข้อมูลที่เกิดขึ้น โดยไม่ส่งผลกระทบต่อประสิทธิภาพในการส่งข้อมูลเพราะการลบแพ็กเกตตอบรับของโพรโทคอลสนูปเพื่อทำการเปรียบเทียบและวิเคราะห์ถึงประสิทธิภาพที่ได้จากการทดลองบนแบบจำลองเสมือนจริงที่มีช่องสัญญาณไร้สายอยู่ระหว่างผู้ใช้ปลายทางและโหนดสุดท้าย ด้วยโทโพโลยีที่แตกต่างกันตามที่ได้กล่าวไว้ข้างต้น

1.2 วัตถุประสงค์

เพื่อพัฒนาประสิทธิภาพของของการควบคุมความแออัดในโครงข่ายในปัจจุบัน โดยอาศัยการควบคุมอัตราอย่างชัดแจ้งเพื่อร้องขออัตราการส่งจากผู้ใช้ต้นทาง เพื่อกำหนดให้ผู้ใช้แต่ละรายแบ่งแบนด์วิดท์อย่างเท่าเทียมอย่างเป็นสัดส่วนบนโครงข่าย TCP/IP และเพิ่มประสิทธิภาพในการส่งข้อมูลและกู้คืนข้อมูลบนช่องสัญญาณไร้สายโดยใช้โพรโทคอลสนูปรูปแบบใหม่ตามที่ได้นำเสนอมาจำลองแบบการทำงานบนแบบจำลองโดยใช้ network simulator (ns-2)

1.3 เป้าหมายและขอบเขตของวิทยานิพนธ์

1. เสนอกรรมวิธีการควบคุมความแออัดในโครงข่าย TCP/IP และการแบ่งแบนด์วิดท์อย่างเท่าเทียม อาศัยการควบคุมอัตราอย่างชัดแจ้งเพื่อกำหนดการอัตราการส่งที่เหมาะสมจากผู้ใช้ต้นทางบนโครงข่าย TCP/IP

2. ศึกษาผลกระทบที่มีผลต่อการควบคุมความแออัดและสมรรถนะของการจัดสรรแบนด์วิดท์ ตามรูปแบบของโทโพโลยีต่างๆ คือ

2.1 โทโพโลยี แบบ dumbell

2.2 โทโพโลยี แบบ parking lot

2.3 โทโพโลยี แบบ upstream

3. ศึกษาถึงข้อดีและข้อเสียในการควบคุมอัตราอย่างชัดแจ้งจากผู้ใช้ต้นทางเพื่อกำหนดอัตราการส่งที่มีความเท่าเทียมและเพิ่มอัตราการใช้ประโยชน์ของลิงก์บริเวณคอขวด และความซับซ้อนในการควบคุมความแออัดและการจัดสรรแบนด์วิดท์

4. เสนอกรรมวิธีในการเพิ่มประสิทธิภาพในการส่งข้อมูลและกู้คืนข้อมูลที่เหมาะสมกับสภาพบนช่องสัญญาณไร้สาย

5. ศึกษาถึงผลกระทบของประสิทธิภาพที่เพิ่มขึ้นที่มีต่อทรัพยากรของโครงข่ายบางประเภท อาทิเช่น ค่าประวิงเวลาในการแผ่กระจาย ขนาดของแบนด์วิดท์ของลิงก์

1.4 ขั้นตอนและวิธีการดำเนินงาน

1. ศึกษาบทความทางวิชาการที่เกี่ยวข้องกับวิทยานิพนธ์

1.1 ศึกษาถึงที่มาของปัญหาและแนวทางการแก้ปัญหาความแออัดบนโครงข่าย TCP/IP

1.2 ศึกษากรรมวิธีการควบคุมความแออัดในรูปแบบการให้บริการที่แตกต่างกันบนโครงข่าย TCP/IP

1.3 ศึกษากรรมวิธีการการส่งข้อมูลและการกู้คืนข้อมูลแบบต่างๆบนช่องสัญญาณไร้สาย

2. สร้างแบบจำลองการทำงานของระบบเครือข่ายอินเทอร์เน็ต เพื่อใช้ในการทดสอบอัลกอริทึมของการควบคุมความแออัดของโครงข่ายโดยใช้การควบคุมอัตราอย่างชัดแจ้งเพื่อร้องขออัตราการส่งจากผู้ใช้ต้นทางผ่านทางแพ็กเกตข้อมูล และทดสอบกรรมวิธีการส่งข้อมูลและกู้คืนข้อมูลบนช่องสัญญาณไร้สายตามที่ได้เสนอ

3. ศึกษาถึงปัจจัยที่เป็นไปได้ที่มีผลกระทบต่อการสร้างแบบจำลองและการความคุมความแออัดตามกรรมวิธีที่ได้เสนอไว้

4. วิเคราะห์และประเมินผลการทำงานของวิธีการควบคุมความแออัดรูปแบบใหม่ที่ใช้การควบคุมอัตราอย่างชัดแจ้งเพื่ออาศัยการร้องขออัตราการส่งจากผู้ใช้ต้นทาง ผ่านทางแพ็กเกตข้อมูลเพื่อที่จัดสรรอัตราการส่งและความเท่าเทียมแก่ผู้ใช้แต่ละราย

5. วิเคราะห์และประเมินผลการทำงานของกรรมวิธีที่ใช้ในการส่งข้อมูลและกู้คืนข้อมูลของช่องสัญญาณไร้สายเมื่อเทียบกับกรรมวิธีการส่งข้อมูลและการกู้คืนข้อมูลบนช่องสัญญาณไร้สายรูปแบบอื่นๆที่อยู่ในกลุ่มเดียวกัน

6. รวบรวมและสรุปผลการวิจัย เพื่อเขียนวิทยานิพนธ์

1.5 ประโยชน์ที่ได้รับ

1. แนวทางการแก้ปัญหาการควบคุมความแออัดในโครงข่ายในรูปแบบของการให้บริการที่แตกต่างกันในโครงข่าย TCP/IP โดยคำนึงถึงการจัดสรรแบนด์วิดท์อย่างเท่าเทียมตามความต้องการของผู้ใช้บริการ

2. สามารถเพิ่มประสิทธิภาพในการติดต่อสื่อสารข้อมูลบนโครงข่าย TCP/IP

3. สามารถเพิ่มความสามารถในการกู้คืนข้อมูลที่สูญหายในช่องสัญญาณไร้สาย

1.6 เค้าโครงวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งรายละเอียดออกเป็น 5 บท ดังนี้

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของวิทยานิพนธ์ เป้าหมายและขอบเขตของวิทยานิพนธ์ ขั้นตอนและวิธีการดำเนินงาน ประโยชน์ที่ได้รับ และเค้าโครงวิทยานิพนธ์

บทที่ 2 ความรู้พื้นฐาน ประกอบไปด้วยหลักการทำงานของ TCP รูปแบบต่างๆที่ใช้ในปัจจุบัน เทคนิคการส่งข้อมูลบนช่องสัญญาณไร้สายบนโครงข่าย TCP แบบที่มีการส่งซ้ำบนระดับชั้นลิงก์ (link layer retransmission) และการพิจารณาความเท่าเทียมของผู้ใช้แต่ละราย

บทที่ 3 ระบบที่นำเสนอ มีเนื้อหาที่ประกอบไปด้วย หลักการ แบบจำลองของระบบ และวิธีของการส่งข้อมูลและการกู้คืนข้อมูลบนช่องสัญญาณไร้สายที่ได้นำเสนอ

บทที่ 4 ผลการทดสอบ กล่าวถึงผลการจำลองระบบและการวิเคราะห์ผลการจำลองระบบเพื่อเปรียบเทียบประสิทธิภาพของการส่งข้อมูลของ TCP รูปแบบต่างๆ และวิธีที่ได้นำเสนอในโทโพโลยีที่ได้กล่าวไว้ข้างต้น ในโครงข่ายแบบสายและโครงข่ายที่มีช่องสัญญาณระหว่างโหนดตัวสุดท้ายและผู้ใช้ปลายทางเป็นแบบไร้สาย

บทที่ 5 บทสรุป กล่าวถึงบทสรุป และข้อเสนอแนะ