บทที่ 4

การออกแบบโพรโทคอลสำหรับการแพร่ที่มีความเชื่อถือได้แบบรู้ข้อมูลตำแหน่ง สำหรับเครือข่ายไร้สายแบบแอดฮอกบนยานพาหนะ

การแพร่ที่มีความเชื่อถือได้แบบรู้ข้อมูลตำแหน่งสำหรับเครือข่ายไร้สายแบบแอดฮอกบนยานพาหนะ (Position-Aware Reliable Broadcasting Protocol on Vehicular Ad-Hoc Networks : POCA) ถูกพัฒนาขึ้นจากขั้นตอนการทำงานของการแพร่ที่มีความเชื่อถือได้แบบรู้ข้อมูลความหนาแน่นสำหรับเครือข่ายไร้สายแบบแอดฮอกบนยานพาหนะ (DECA) แต่ใช้ข้อมูลตำแหน่ง และทิศทางทดแทน นอกจากนี้ยังพัฒนาส่วนการทำงานหลักให้โพรโทคอลสามารถทำงานโดยมีประสิทธิภาพสูงขึ้น มีพื้นฐานการทำงานที่คำนึงถึงปัจจัยสำคัญ 3 ประการเช่นเดิม 1) ความเชื่อถือได้ (Reliability) 2) ค่าใช้จ่าย (Overhead) และ 3) ความเร็วในการแพร่ข้อมูล (Speed of Data Dissemination)

4.1 แนวคิดในการออกแบบ

การออกแบบสำหรับโพรโทคอลการแพร่ที่มีความเชื่อถือได้แบบรู้ข้อมูลตำแหน่งสำหรับเครือข่ายไร้สายแบบแอดฮอกบนยานพาหนะ (POCA) มีแนวคิดในการออกแบบดังนี้

1) ระยะการเชื่อมต่อสูงสุดของโหนดจะมีค่ามากกว่าความกว้างของถนน โดยเฉพาะอย่างยิ่งการทำงานในมาตรฐาน IEEE 802.11P ที่มีระยะการเชื่อมต่อสูงสุดที่ประมาณ 1 กิโลเมตร ดังนั้นความกว้างของถนนจึงไม่ใช่อุปสรรคในการแพร่ของมูล การออกแบบโพรโทคอลหากสามารถทำให้โหนดที่ส่งต่อแต่ละครั้งอยู่บริเวณขอบของวงการแพร่ จะสามารถเพิ่มความเร็วในการทำงาน และลดค่าใช้จ่ายที่เกิดขึ้นจากการแพร่ข้อมูลได้

2) ความเร็วในการแพร่ข้อมูลขึ้นอยู่กับเวลารอ (Waiting Timeout) โดยหากทราบข้อมูลของตำแหน่งจะสามารถคำนวณเวลารอที่เหมาะสมได้ จึงจะสามารถใช้ลดเวลารอโดยรวมของระบบได้ อีกทั้งหากหลีกเลี่ยงการใช้งานเวลารอให้น้อยที่สุดดังเช่นการออกแบบของ DECA จะสามารถลดความล่าช้าที่จะเกิดขึ้นในการแพร่แต่ละครั้งได้มากขึ้น ส่งผลให้การทำงานของโพรโทคอลทำงานได้เร็วขึ้น

3) เนื่องจาก POCA ต้องการเพิ่มความเร็วของโพรโทคอลให้ถึงขีดสุด จึงใช้ข้อมูลด้านตำแหน่ง และทิศทาง (ข้อมูลจากจีพีเอส) จึงมีความยืดหยุ่นในการทำงานน้อยกว่า DECA และมีค่าใช้จ่างสูงขึ้น แต่จะให้สมรรถนะที่เพิ่มขึ้น

สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดวิจัย

4.2 หลักการทำงานของโพรโทคอล

หลักการทำงานที่สำคัญของโพรโทคอลการแพร่ที่มีความเชื่อถือได้แบบรู้ข้อมูลตำแหน่งสำหรับเครือข่ายไร้สายแบบแอดฮอกบนยานพาหนะ (POCA) เหมือนดังเช่นส่วนประกอบหลักของโพรโทคอลการแพร่ที่มีความเชื่อถือได้แบบรู้ข้อมูลความหนาแน่นสำหรับเครือข่ายไร้สายแบบแอดฮอกบนยานพาหนะ (DECA) ซึ่งจะกล่าวถึงความแตกต่างภายในของการทำงานหลักในแต่ละส่วน

1) Store-and-Forward : ใช้การทำงานแบบ Store-and-Forward เพื่อรองรับการทำงานในสภาพที่มีการเชื่อมต่อเป็นช่วง ๆที่เกิดขึ้นได้บ่อยในสภาพการทำงานบนรถยนต์เช่นเดียวกับ DECA

2) Beaconing with Adaptive Intervals : POCA ใช้ Beacon ที่ระยะเวลาในการส่งสามารถปรับเปลี่ยนได้ตามความหนาแน่นของโหนด และความหนาแน่นของเครือข่ายขณะนั้น (Adaptive Beaconing Intervals) โดยเพิ่มความซับซ้อนของฟังก์ชั่นในการคำนวณมากกว่าที่ใช้ใน DECA

3) Preferred Node Selection Algorithm : การให้โหนดต้นทางหรือโหนดก่อนหน้า (Source/Precursor Node) เป็นผู้กำหนดโหนดที่จะส่งต่อข้อความ โดยเลือกจากโหนดที่อยู่ในบริเวณที่ต้องการ (Preferred Distance) ซึ่งจะอยู่บริเวณขอบของการแพร่ เพื่อลดจำนวนครั้งการแพร่ข้อมูลในการครอบคลุมตามความยาวของถนนที่เท่ากัน

4) Waiting Timeout Calculation : POCA จะใช้เวลารอโดยคำนวณจากระยะห่างระหว่างโหนด ซึ่งจะแตกต่างกันในกรณีที่มีการรับข้อความใหม่จากโหนดอื่น และในกรณีที่พบเพื่อนบ้านไม่ได้รับข้อความ ซึ่งจะสามารถลดเวลารอได้มากกว่าแบบสุ่มใน DECA เนื่องจากมีข้อมูลในการทำงานมากกว่า

4.2.1 การเก็บข้อมูลของโพรโทคอล

รถยนต์แต่ละคัน หรือโหนดจะเก็บข้อมูลสำคัญ 3 ชุด คือ ข้อมูลของเพื่อนบ้าน คิวของข้อความที่จะถูกส่งต่อ และข้อความที่ยังไม่หมดอายุ

1) ข้อมูลของเพื่อนบ้านจะประกอบ ด้วยหมายเลขประจำตัวของโหนด ความหนาแน่นของโหนด หรือตำแหน่งของโหนดกรณีมีการใช้งานจีพีเอส ใช้ในการเลือกโหนดที่จะส่งต่อข้อความ

2) คิวของข้อความที่จะถูกส่งต่อ เป็นรายละเอียดของข้อความพร้อมทั้งเวลาที่จะส่งข้อความออกไป คิวนี้ใช้เพื่อรอเวลาที่ข้อความนั้นจะถูกส่งต่อ แต่ในกรณีที่โหนดได้ยินโหนด

เพื่อนบ้านส่งข้อความนั้นก่อน ข้อความก็จะถูกลบออกจากคิว เพื่อลดจำนวนการส่งของข้อความเดิมในบริเวณเดียวกัน ซึ่งคิวของข้อความที่จะถูกส่งต่อไม่ใช่หน่วยความจำที่ใช้ในการเก็บข้อมูล ซึ่งจะเก็บข้อความนั้นไว้จนกว่าข้อความนั้นหมดอายุ

3) ข้อความที่ยังไม่หมดอายุ จะถูกเก็บไว้ตามหลักการทำงานแบบ Store-and-Forward เพื่อใช้ในการส่งให้กับโหนดเพื่อนบ้านที่ยังไม่รับข้อมูลกรณีที่มีการเชื่อมต่อเป็นช่วง ๆ ไม่ต่อเนื่อง ข้อความจะถูกเก็บจนกระทั่งหมดอายุ

4.2.2 การแลกเปลี่ยนข้อมูลจากโหนดเพื่อนบ้าน (Beaconing)

ข้อมูลของโหนดเพื่อนบ้านจะได้จากการแลกเปลี่ยนข้อมูลผ่าน Beacon Message ซึ่งข้อมูลที่จะมีการส่งผ่านไปพร้อมกับ Beacon Message ประกอบด้วย

- หมายเลขเฉพาะตัวของโหนด

- ตำแหน่ง ทิศทาง และความเร็วของโหนด (ข้อมูลจากจีพีเอส)

- รายการของข้อความที่ได้รับ ประกอบด้วยโหนดที่แพร่ข้อความ และหมายเลขเฉพาะข้อความ ใช้เพื่อตรวจสอบว่ามีข้อความที่ยังไม่ได้รับหรือไม่ รายการของความที่ได้รับจะไม่ถูกเก็บลงในข้อมูลเพื่อนบ้าน เมื่อทำการตรวจสอบเสร็จจะถูกลบออก

เนื่องจากช่วงเวลาการทำ Beacon ส่งผลต่อประสิทธิภาพของการทำงานโพรโทคอลที่เกิดจาก ดังที่กล่าวในการออกแบบของ DECA (3.2.2 การแลกเปลี่ยนข้อมูลจากโหนดเพื่อนบ้าน) ดังนั้นการทำ Beacon ใน POCA จึงใช้ช่วงเวลาการทำ Beacon ที่เปลี่ยนแปลงตามความหนาแน่นของเครือข่าย ซึ่งความหนาแน่นจะขึ้นอยู่กับความหนาแน่นของโหนด และจำนวนของข้อความที่มีการแพร่

ความหนาแน่นจะคำนวณได้ตามสมการเดียวกับ DECA ดังสมการที่ (1) โดยให้ d คือ ความหนาแน่นของเครือข่าย n คือ จำนวนโหนดเพื่อนบ้าน m คือ จำนวนของข้อความในระบบ w_1 และ w_2 คือ ค่าถ่วงน้ำหนักของจำนวนโหนดเพื่อนบ้าน และจำนวนของข้อความในระบบตามลำดับ

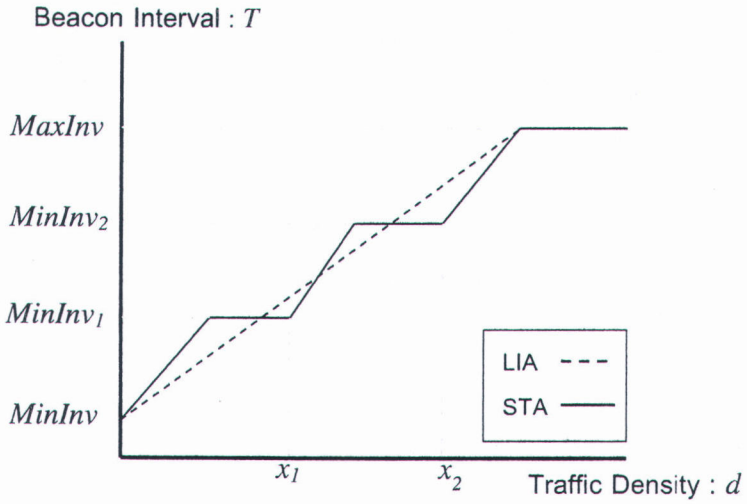
$$d = (w_1 \times n) + (w_2 \times m) \qquad (1)$$

การคำนวณช่วงเวลาสำหรับการทำ Beacon ที่มีการเปลี่ยนแปลงตามความหนาแน่นของเครือข่าย (Adaptive Beacon Interval) ใน POCA จะมีความซับซ้อนมากขึ้นโดยแบ่งการคำนวณออกเป็นช่วง ๆที่เหมาะสมสำหรับความหนาแน่นของเครือข่ายในแต่ละแบบ ลักษณะของกราฟความหนาแน่นจะมีลักษณะคล้ายขั้นบันไดจึงเรียกฟังก์ชันการคำนวณว่า การ

คำนวณช่วงเวลาปรับตัวแบบขั้น (Step Adaptive Algorithm : STA) ในวิทยานิพนธ์นี้จะแบบขั้นการคำนวณออกเป็น 3 ขั้น คือ บริเวณที่มีความหนาแน่นสูง บริเวณที่มีความหนาแน่นปานกลาง และบริเวณที่มีความหนาแน่นต่ำ เนื่องจากความหนาแน่นของเครือข่ายที่แตกต่างกันจะมีค่าความชันที่เหมาะสมของฟังก์ชั่นแตกต่างกันด้วย โดยการแบ่งออกเป็นขั้น จะสามารถหาค่าของช่วงเวลาที่เหมาะสม ลดค่าใช้จ่ายที่เกิดขึ้นและยังให้ค่าของความน่าเชื่อถือคงเดิม

การคำนวณช่วงเวลาปรับตัวแบบขั้นสามารถอธิบายได้ตามสมการที่ (3) โดย T คือ ช่วงเวลาสำหรับการทำ Beacon ครั้งถัดไป $MinInv$ คือ ช่วงเวลาสั้นสุด c_1, c_2, c_3 เป็นค่าคงที่ในการเพิ่มช่วงเวลาในแต่ละขั้น $MinInv_1$, $MinInv_2$ คือ ช่วงเวลาสั้นสุด เมื่อมีความหนาแน่นของเครือข่ายมากกว่า x_1 และ x_2 ตามลำดับ d คือ ความหนาแน่นของเครือข่าย และ $MaxInv$ คือ ช่วงเวลายาวสุด สามารถนำมาเขียนกราฟได้ตามรูปที่ 4.2 จากกราฟสามารถเปรียบเทียบความแตกต่างระหว่างการคำนวณช่วงเวลาปรับตัวแบบเชิงเส้น (LIA) และการคำนวณช่วงเวลาปรับตัวแบบขั้น (STA)

$$T = \begin{cases} d \le x_1 & min(MinInv_1+(c_1 \times s), MinInv_2). \\ x_1 \le d \le x_2 & min(MinInv_2+(c_2 \times s), MinInv_3). \\ d > x_2 & min(MinInv_3+(c_3 \times s), MaxInv). \end{cases} \quad (3)$$



รูปที่ 4.1 กราฟแสดงการคำนวณช่วงเวลาปรับตัวแบบขั้น

4.2.3 การเลือกโหนดส่งต่อข้อความ (Preferred Node Selection Algorithm)

เมื่อมีโหนดต้องการแพร่ข้อมูลให้กับโหนดอื่นๆในเครือข่าย โหนดต้นทาง (Source) จะเหลือโหนดที่อยู่ในระยะที่เหมาะสมกับระยะเชื่อมต่อของตน โดยมีค่าประมาณ 80% ของระยะเชื่อมต่อสูงสุด เพื่อให้โหนดถัดไปที่จะส่งต่อข้อความมีระยะที่ไกลจากโหนดต้นทางเพื่อลด

จำนวนครั้งที่จะต้องแพร่ข้อความเพื่อครอบคลุมความยาวของเส้นทางที่เท่ากัน นอกจากนี้การไม่เลือกโหนดที่บริเวณขอบสุด เนื่องจากบริเวณขอบของการแพร่อาจจะเกิดความผิดพลาดในการส่งข้อมูล และลดความผิดพลาดจากการเลือกโหนดที่มีการเปลี่ยนแปลงความเร็วสูงอย่างรถยนต์

ขั้นตอนการเลือกโหนด ก่อนการเลือกโหนด โหนดต้นทาง (Source) จะปรับปรุงตำแหน่งของโหนดเพื่อนบ้านโดยการใช้ข้อมูลล่าสุดจาก Beacon Message ที่ได้รับจากเพื่อนบ้าน ซึ่งมีทั้งตำแหน่ง ความเร็ว ทิศทางและเวลาที่ได้รับ Beacon Message จากนั้นโหนดต้นทางจะเลือกโหนดประกอบด้วย 2 โหนดจากทางด้านหน้าและด้านหลัง แล้วแนบหมายเลขของโหนดนั้นพร้อมกับส่งข้อความออกไป โดยโหนดที่ได้รับข้อความนั้นแล้วพบว่าตัวเองเป็นโหนดที่ถูกเลือก ก็จะทำการเลือกโหนดที่อยู่ระยะที่เหมาะสมสูงสุดจากรายชื่อของตัวเอง แล้วแนบหมายเลขของโหนดนั้นลงไปกับข้อความก่อนส่งข้อความออกไป โดยจะเลือกโหนดถัดไปเพียงโหนดเดียวตามเส้นทางการเคลื่อนที่ของข้อความ ซึ่งกระบวนการนี้จะเกิดขึ้นจนกว่ารถยนต์ในบริเวณได้รับข้อความนั้นทั้งหมด หรือข้อความนั้นหมดอายุ

การเลือกโหนดที่จะส่งข้อความต่อนั้น จะเลือกโหนดที่อยู่ในระยะเชื่อมต่อ โดยที่โหนดนั้นจะไม่ใช่โหนดที่ส่งข้อความก่อนหน้า (Precursor Node) และโหนดที่ถูกเลือกจะทำงานก็ต่อเมื่อโหนดนั้นไม่เคยได้รับข้อความมาก่อน เพื่อป้องกันการเลือกโหนดวนซ้ำเดิม ซึ่งอาจจะเกิดได้กรณีการเปลี่ยนแปลงตำแหน่ง แต่ก็เป็นกรณีที่สามารถเกิดขึ้นได้ยาก เมื่อความหนาแน่นของโหนดมีมากพอจนไม่เกิดการเชื่อมต่อเป็นช่วง ความเร็วในการเดินทางของข้อความจะมากกว่าความเร็วการเดินทางของรถยนต์มาก ดังนั้นการเลือกโหนดย้อนกลับจึงเกิดขึ้นได้ยาก

การซ่อมแซมการแพร่ข้อความเป็นกระบวนการที่เกิดขึ้นเมื่อโหนดที่ถูกเลือกไม่ทำงานตามที่กำหนด เกิดขึ้นได้เมื่อโหนดที่ถูกเลือกได้รับข้อความนั้นแล้วเช่นกรณีข้างต้น หรือเมื่อเกิดการชนทำให้โหนดที่ถูกเลือกไม่ได้รับข้อความนั้น หรือเกิดจากโหนดที่เห็นแก่ตัว ซึ่งได้รับข้อความแล้วไม่ทำการส่งข้อความนั้นต่อ กระบวนการซ่อมแซมนั้นจะเกิดขึ้นทันทีหลังจากที่โหนดที่ไม่ใช่โหนดที่ถูกเลือกได้รับข้อความใหม่ โหนดจะตั้งเวลาตามระยะทาง(รายละเอียดการคำนวณอยู่ในส่วนถัดไป) เพื่อรอการส่งข้อความนั้นซ้ำอีกครั้ง ซึ่งหากมีการส่งข้อความนั้นซ้ำ ข้อความที่ถูกตั้งเวลาไว้จะถูกลบออกจากคิว แต่ในกรณีที่โหนดนั้นไม่ได้ยินการส่งข้อความซ้ำอีกครั้งจนกระทั่งเวลาที่ตั้งไว้หมด โหนดจะทำการส่งข้อความนั้นซ้ำอีกครั้ง พร้อมทั้งเลือกโหนดที่จะส่งต่อข้อความใหม่จากข้อมูลเพื่อนบ้านที่มีอยู่ ซึ่งเมื่อโหนดหนึ่งได้ทำการซ่อมแซมโดยส่งต่อข้อความนั้นแล้ว โหนดอื่นๆที่อยู่ในบริเวณเดียวกันก็จะทำการลบข้อความนั้นออกจากคิว

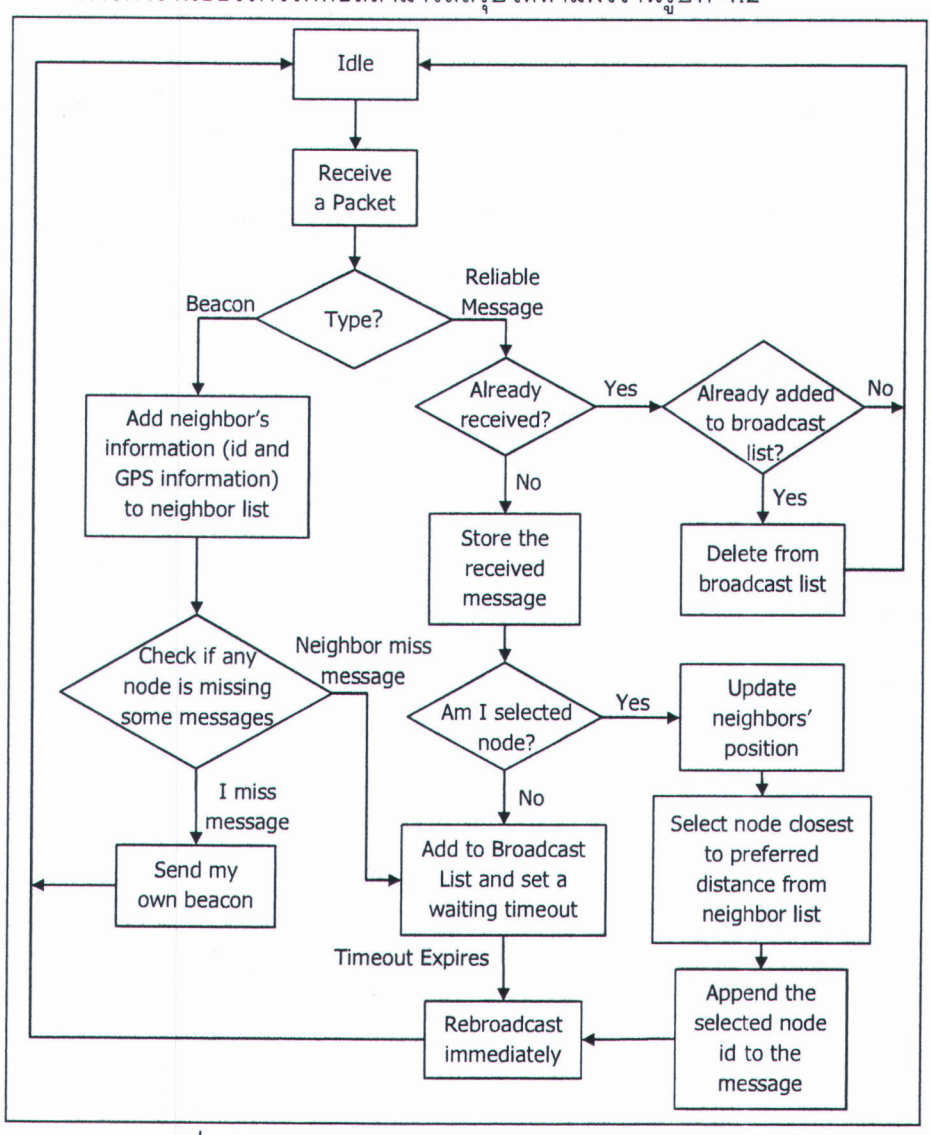
เนื่องจากลักษณะเฉพาะของรถยนต์ที่ทำให้เกิดการเชื่อมต่อเป็นช่วงๆได้ ดังนั้นการใช้ Beacon Message จึงสามารถตรวจสอบได้เมื่อโหนดได้รับข้อความไม่ครบ โดยที่โหนดอื่นที่

ได้รับ Beacon Message สามารถตรวจสอบได้ว่าข้อความแพร่ใดที่โหนดเพื่อนบ้านยังไม่ได้รับ รวมทั้งข้อความใดที่ตนเองยังไม่ได้รับเช่นกัน

- กรณีที่โหนดเพื่อนบ้านได้รับข้อความไม่ครบ โหนดจะนำข้อความที่เก็บไว้ใส่ในคิว เพื่อรอการส่งต่อ และตั้งเวลา โดยการตั้งเวลาจะกล่าวถึงอย่างละเอียดในส่วนถัดไป โหนดที่มีเวลารอน้อยที่สุดเท่านั้นจะทำการส่งข้อความให้โหนดเพื่อนบ้าน โหนดอื่นๆที่ได้ยินจะลบข้อความออกจากคิว ในการส่งข้อความในกรณีนี้ โหนดที่ส่งข้อความจะไม่เลือกโหนดที่จะส่งต่อข้อมูล เนื่องจากโหนดเพื่อนบ้านน่าจะมีข้อมูลของโหนดเพื่อนบ้านในบริเวณนั้นที่ดีกว่า

- กรณีที่โหนดพบว่าตนเองมีข้อความที่ขาดไป จะส่ง Beacon Message ทันที เพื่อขอรับข้อความจากโหนดที่มีข้อความที่ตนเองไม่มีอยู่ ก่อนที่โหนดนั้นจะหายไปจากบริเวณนั้น

การทำงานของโพรโทคอลสามารถสรุปได้ตามผังงานรูปที่ 4.2



รูปที่ 4.2 ผังงานแสดงการทำงานของโพรโทคอล POCA

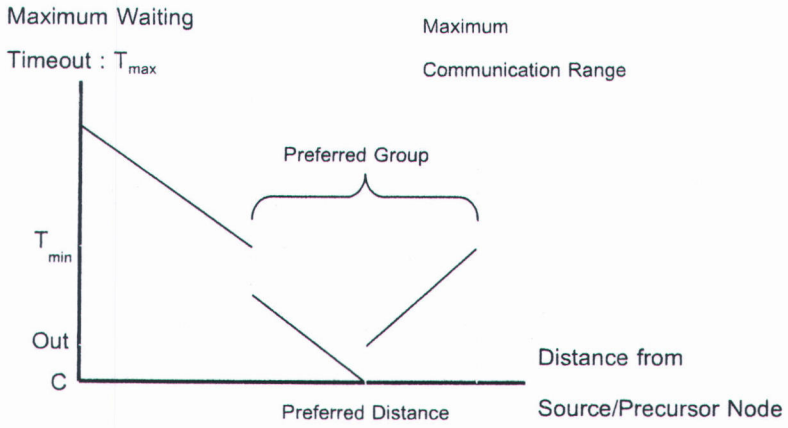
4.2.4 การคำนวณเวลารอ (Waiting Timeout Calculation)

การคำนวณเวลารอเป็นขั้นตอนที่สำคัญในการลดเวลาในการทำงานโพรโทคอล และส่งผลต่อประสิทธิภาพของโพรโทคอล ดังที่กล่าวในการออกแบบการทำงานของ DECA แต่เนื่องจาก POCA มีข้อมูลของตำแหน่งของโหนดในการทำงาน ดังนั้นการคำนวณเวลารอจึงมีความซับซ้อนมากขึ้นเพื่อนให้โพรโทคอลมีสมรรถนะที่ดีขึ้น โดยแยกกรณีการคำนวณออกเป็น 2 กรณี คือ เมื่อโหนดที่ถูกเลือกไม่ทำงาน และกรณีที่โหนดเพื่อนบ้านมีข้อความที่ยังไม่ได้รับ ซึ่งทั้ง 2 กรณีมีวิธีการคำนวณเวลารอที่แตกต่างกันดังนี้

1) การคำนวณเวลารอเมื่อโหนดที่ถูกเลือกไม่ทำงาน

การตั้งเวลารอในกรณีนี้จะกระทำทันทีเมื่อโหนดได้รับข้อความใหม่ เพื่อให้การทำงานของโพรโทคอลสามารถทำได้ต่อเนื่องเมื่อมีโหนดที่ถูกเลือกไม่ทำงาน ดังนั้นการตั้งเวลารอจำเป็นจะต้องให้มีระยะเวลาสั้นที่สุดเพื่อลดความล่าช้าที่จะเกิดขึ้นกันโพรโทคอล โดยโหนดที่จะทำหน้าที่แทนควรจะอยู่ในตำแหน่งที่ใกล้กับระยะของโหนดที่ถูกเลือกมากที่สุด โดยสามารถคำนวณระยะทางได้จากตำแหน่งของโหนดที่แนบมากับข้อความ ดังนั้นตำแหน่งของโหนดจึงนำมาใช้ในการคำนวณค่าเวลารอ โดยสามารถสรุปได้ดังกราฟในรูปที่ 4.3 ซึ่ง โหนดที่มีค่าห่างจากบริเวณที่โหนดต้องการ (Preferred Distance) ในช่วงหนึ่งจะถือว่าอยู่ในกลุ่มที่ต้องการมากกว่า(Preferred Group) จะถูกแบ่งออกเป็นกลุ่มภายที่มีระยะห่างไม่เกินค่าระยะที่ต้องโหนดต้องการ ค่าของเวลารอจะถูกสุ่มให้มีค่าระหว่าง $(C,T_{max}]$ โดยที่ C คือ ค่าคงที่เป็นสองเท่าของเวลาที่ใช้ในการส่ง (Propagation Delay) และ T_{max} ได้จากการคำนวณ ค่าของโหนดที่อยู่ในกลุ่มแต่มีระยะห่างมากกว่าที่โหนดต้องการจะมีค่าเวลารอถูกสุ่มระหว่าง $(C+Out,T_{max}]$ การที่โหนดในกลุ่มนี้ถูกเพิ่มค่าขั้นต่ำ เนื่องจากโหนดใกล้บริเวณขอบของระยะเชื่อมต่อสูงสุดมีโอกาสที่จะได้รับข้อความที่ไม่สมบูรณ์ ส่วนค่าของโหนดที่อยู่นอกกลุ่มจะมีเวลารอโดยสุ่มจาก $(T_{min},T_{max}]$ เพื่อให้โหนดที่อยู่ใกล้บริเวณของโหนดที่ถูกเลือกมีโอกาสทำงานแทนมากกว่า ตัวอย่างช่วงเวลาที่โหนดกลุ่มสุ่มได้ดังนี้ ให้ระยะการเชื่อมต่อสูงสุดมีค่า 250 เมตร ระยะที่ต้องการ (Preferred Distance) คือ 200 เมตร

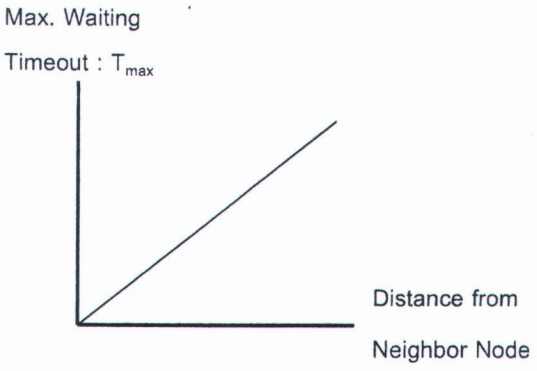
- โหนดในช่วงระยะที่โหนดก่อนหน้าต้องการ (Preferred Group) ระยะ 150-250 เมตร
  - โหนดที่มีระยะห่างน้อยกว่าระยะที่ต้องการ (150-200 เมตร) : $(C,T_{max}]$
  - โหนดที่มีระยะห่างน้อยกว่าระยะที่ต้องการ (200-250 เมตร) : $(C+Out,T_{max}]$
- โหนดนอกช่วงระยะที่โหนดก่อนหน้าต้องการ
  - โหนดที่มีระยะห่างระหว่าง 0-150 เมตร : $(T_{min},T_{max}]$



รูปที่ 4.3 กราฟแสดงการคำนวณเวลารอสูงสุดกรณีโหนดที่ถูกเลือกไม่ทำงาน

2) การคำนวณเวลารอเมื่อโหนดเพื่อนบ้านมีข้อความที่ยังไม่ได้รับ

การคำนวณเวลารอในกรณีนี้เพื่อป้องกันโหนดที่จะพยายามส่งข้อความให้กับเพื่อนบ้านส่งพร้อมๆกัน แต่เพื่อให้โหนดเพื่อนบ้านนั้นสามารถได้รับข้อความจากการส่งเพียงครั้งเดียว ลดความผิดพลาดที่อาจจะเกิดขึ้นจากการเปลี่ยนแปลงตำแหน่ง และความผิดพลาดจากการส่งข้อความ ดังนั้นโหนดที่อยู่ใกล้โหนดเพื่อนบ้านมากที่สุดจะเป็นโหนดที่ส่งข้อมูลให้ โดยการคำนวณเวลารอจะแปรผันตรงกันระยะทางระหว่างโหนดนั้นๆกับโหนดเพื่อนบ้าน ดังกราฟในรูปที่ 4.4 ค่าของเวลารอจะถูกสุ่มในช่วง (0,Tmax]

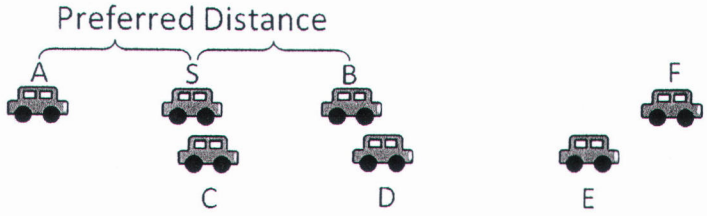


รูปที่ 4.4 กราฟแสดงการคำนวณเวลารอสูงสุดกรณีโหนดที่เพื่อนบ้านมีข้อความที่ยังไม่ได้รับ

4.4 ตัวอย่างการทำงานของโพรโทคอล

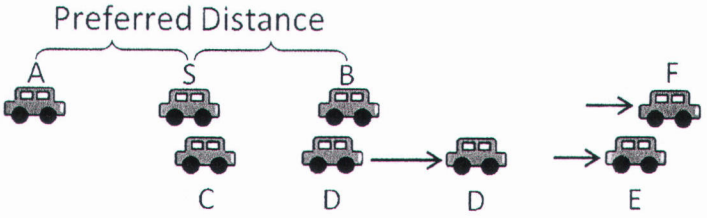
การทำงานของ POCA มีข้อแตกต่าง และส่วนเพิ่มเติมเพียงจาก DECA ดังตัวอย่างการทำงานต่อไปนี้ ในรูปที่ 4.4 เป็นการทำงานในลักษณะปกติ เมื่อโหนด S ต้องการเริ่มต้นแพร่ข้อมูลออกไป โหนด S มีโหนด A, B, C และ D เป็นโหนดเพื่อนบ้าน ในกรณีนี้ให้ A และ B เป็นโหนดที่อยู่ใกล้ระยะที่ S ต้องการมากที่สุด (Preferred Distance) ดังนั้นในการแพร่ข้อความ S จึงเลือก A และ B แนบไปกับข้อความ หลังจากการแพร่ข้อความของ S โหนด A, B, C และ D ได้รับข้อความพร้อมกัน เมื่อ A และ B ได้รับข้อความ A และ B ซึ่งทราบว่าตัวเองเป็นโหนดที่ถูกเลือกจะทำการส่งข้อมูลพร้อมทั้งเลือกโหนดส่งต่อข้อความต่อไป ในกรณีของ B โหนดเพื่อนบ้านที่จะถูกเลือกอาจจะเป็น E หรือ F

C และ D เมื่อได้รับข้อความแล้วพบว่าตนเองไม่ใช่โหนดที่ถูกเลือกจะนำข้อความใส่ลงในคิวและตั้งเวลา เพื่อรอการส่งต่อข้อความของ B ซึ่งในกรณีที่ B ไม่ได้ส่งต่อข้อความตามที่ S กำหนด C และ D จะทำการซ่อมแซม โดยโหนดที่มีเวลาในการรอสั้นที่สุดจะเป็นผู้ส่งต่อข้อความ ในกรณีนี้จะเป็น D เนื่องจาก D มีระยะทางใกล้ระยะที่ S ต้องการมากกว่า C จึงมีเวลารอสั้นกว่า C ดังนั้น B จะทำการเลือกเพื่อนบ้านที่ใกล้ระยะที่ต้องการมากที่สุด และส่งข้อความต่อ C ที่ได้ยินการส่งต่อของ B จะลบข้อความนั้นออกจากคิว



รูปที่ 4.5 ลักษณะของรถในการเชื่อมต่อแบบปกติ

ในรูปที่ 4.5 แสดงการทำงานในกรณีที่มีการเชื่อมต่อเป็นช่วง ๆ (Intermittent Connectivity) ในกรณีนี้สมมติให้ E และ F อยู่ห่างออกไปจากระยะสัญญาณสื่อสารของ B เมื่อ S เป็นโหนดเริ่มต้นการแพร่ข้อความ และ B ถูกเลือก จากนั้น B จะเลือกไม่โหนดส่งต่อถัดไป เนื่องจากไม่พบโหนดที่อยู่ด้านหน้าของตนในทิศทางการส่งข้อความ แต่เมื่อเวลาผ่านไป D แซง B ทำให้พบกับรถยนต์บริเวณด้านหน้าซึ่งคือ E และ F หลังจากได้รับ Beacon Message D จะทราบว่าทั้ง 2 โหนดได้รับข้อความไม่ครบถ้วน D จะส่งต่อข้อความแต่ไม่ระบุโหนดส่งต่อ เนื่องจากว่าในพื้นที่ใหม่นั้น E และ F น่าจะมีข้อมูลของโหนดเพื่อนบ้านที่ดีกว่า D ดังนั้นหลังจากที่ E และ F ได้รับข้อความก็จะตั้งเวลา F ที่มีระยะทางใกล้ระยะที่ D ต้องการมากกว่า E โหนด F จึงมีเวลารอสั้นที่สุดและเลือกโหนดที่จะส่งต่อถัดไปแทนที่การเลือกของ B



รูปที่ 4.6 ลักษณะของรถในการเชื่อมต่อเป็นช่วงๆ