

# ไอเอ็มอาร์พีแอล : โพรโตคอลอาร์พีแอลแบบปรับปรุงสำหรับโหนดเคลื่อนที่บนเครือข่ายซิกโลแพน

## ImRPL: Improved RPL protocol for Mobility Node on 6LoWPAN

ภาคย์ สชนเสาวภาคย์<sup>1</sup>, ชัชชัย คุณบัว<sup>2</sup>

Pak Satanaasawapak<sup>1</sup>, Chatchai Khunboa<sup>2</sup>

Received: 7 July 2017 ; Accepted: 28 November 2017

### บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อปรับปรุงโพรโตคอลอาร์พีแอลให้สามารถสนับสนุนการทำงานในกรณีที่มีโหนดเคลื่อนที่บนเครือข่ายซิกโลแพน โดยผู้วิจัยเรียกโพรโตคอลที่ถูกปรับปรุงว่าโพรโตคอลไอเอ็มอาร์พีแอล วิธีการที่ได้เสนอในงานวิจัยนี้คือ การปรับปรุงกระบวนการเลือกโหนดแม่และการปรับปรุงกระบวนการตรวจจับโหนดเคลื่อนที่ งานวิจัยนี้ได้เปรียบเทียบประสิทธิภาพระหว่างโพรโตคอลอาร์พีแอลกับโพรโตคอลไอเอ็มอาร์พีแอลโดยใช้โปรแกรม Cooja Simulator ผลการทดลองพบว่า ในสถานการณ์แรก ที่ความเร็วของโหนด 1 เมตรต่อวินาที โพรโตคอลไอเอ็มอาร์พีแอลมีอัตราการส่งข้อมูลสำเร็จสูงถึง 88.92 เปอร์เซ็นต์ ในขณะที่โพรโตคอลอาร์พีแอลมีอัตราการส่งข้อมูลสำเร็จ 68.54 เปอร์เซ็นต์ ในสถานการณ์ที่สองโพรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันสอง โพรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งและโพรโตคอลอาร์พีแอลมีอัตราการส่งข้อมูลสำเร็จสูงเท่ากับ 70.89 เปอร์เซ็นต์, 58.84 เปอร์เซ็นต์ และ 38.98 เปอร์เซ็นต์ ตามลำดับ ยิ่งไปกว่านั้นโพรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันสองสามารถเลือกโหนดแม่ได้ถูกต้องคิดเป็น 100 เปอร์เซ็นต์ ในขณะที่โพรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งและโพรโตคอลอาร์พีแอลมีเปอร์เซ็นต์ในการเลือกโหนดแม่ที่ต้องถูกเท่ากับ 77.5 เปอร์เซ็นต์และ 47.5 เปอร์เซ็นต์ ตามลำดับ

**คำสำคัญ:** เครือข่ายที่มีโหนดเคลื่อนที่ เครือข่ายเซ็นเซอร์ไร้สาย อินเทอร์เน็ตของสรรพสิ่ง เครือข่ายซิกโลแพน โพรโตคอลค้นหาเส้นทาง

### Abstract

This research aimed to improve RPL protocol for support mobility node on 6LoWPAN. We propose an improved protocol called imRPL protocol that enhances the parent selection phase and the mobile detection phase. This research compared performance between RPL protocol and imRPL protocol using Cooja simulator. The results showed that in the first scenario where the speed of node is 1 m/s, the imRPL protocol had a packet delivery ratio at 88.92%, whereas with the RPL protocol it was 68.54%. In the second scenario, imRPLv2, imRPLv1 and RPL protocol had packet delivery ratios of 70.89%, 58.84%, and 38.98% respectively. Moreover, the imRPLv2 protocol can choose a correct new parent at 100% while imRPLv1 and RPL protocol have the correct ratio of 77.5% and 47.5% respectively.

**Keywords:** Mobility Network, Wireless Sensor Network, Internet of Things, 6LoWPAN, Routing Protocol

<sup>1</sup> นักศึกษาปริญญาเอก, <sup>2</sup>อาจารย์, สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น ถนนมิตรภาพ อำเภอเมือง จังหวัดขอนแก่น 40002 ประเทศไทย

<sup>1</sup> Doctoral degree student <sup>2</sup>Lecturer, Department of Computer Engineering, Faculty of Engineering, Khon Kaen University, Mittraphap Road, Mueang Khon Kaen District, Khon Kaen, 40002, Thailand.

## บทนำ

ซิกโลแพน (IPv6 over Low power Wireless Personal Area Networks : 6LoWPAN)<sup>1</sup> คือเครือข่ายส่วนบุคคลที่ใช้สำหรับการส่งข้อมูลไร้สายแบบประหยัดพลังงาน มีอัตราการส่งข้อมูลต่ำ ทำงานอยู่บนมาตรฐาน IEEE 802.15.4 ซึ่งจะทำให้โหนดที่อยู่ในเครือข่ายสามารถสื่อสารกันผ่านระบบเครือข่ายอินเทอร์เน็ตได้ สำหรับการใช้งานของเซ็นเซอร์ไร้สายนั้นโดยทั่วไปจะเป็นการตรวจวัดค่าจากสภาพแวดล้อมหรือสิ่งที่สนใจโดยเซ็นเซอร์โหนดที่ใช้ในการตรวจวัดจะไม่เคลื่อนที่ การประยุกต์ใช้เครือข่ายเซ็นเซอร์ไร้สายกับโหนดที่เคลื่อนที่จึงเป็นหัวข้อที่น่าสนใจในปัจจุบัน เนื่องจากจะทำให้เกิดแอปพลิเคชันที่เป็นประโยชน์และตอบโจทย์เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งได้ดี ยกตัวอย่างเช่น การเก็บข้อมูลผู้ป่วยที่เคลื่อนที่ในโรงพยาบาล การเก็บข้อมูลการจราจรจากยานพาหนะ การสื่อสารระหว่างรถกับสัญญาณไฟจราจรหรือรถต่อรถ การใช้หุ่นยนต์หรือเครื่องบินขนาดเล็กเพื่อสำรวจในพื้นที่ที่ยากจะเข้าถึง เป็นต้น แต่เนื่องจากการส่งข้อมูลขณะที่มีโหนดเคลื่อนที่จะมีปัญหาในเรื่องของการสร้างเส้นทางใหม่หรือปรับปรุงเส้นทางใหม่ ซึ่งเป็นสาเหตุที่ทำให้การสื่อสารขาดความต่อเนื่องและมีข้อมูลสูญหายมากขึ้น ดังนั้นการแก้ไขปัญหาดังกล่าวจึงต้องปรับปรุงที่โปรโตคอลค้นหาเส้นทางของมาตรฐานนี้

โปรโตคอลอาร์พีแอล (IPv6 Routing Protocol for Low-Power and Lossy Networks : RPL)<sup>2</sup> เป็นโปรโตคอลค้นหาเส้นทางที่อยู่บนเครือข่าย ซิกโลแพนซึ่งโปรโตคอลอาร์พีแอลสร้างเครือข่ายโดยใช้ Direction-Oriented Directed Acyclic Graph (DODAG) และถูกออกแบบมาเพื่อสนับสนุนการทำงานสำหรับโหนดที่ไม่เคลื่อนที่ ผู้วิจัยเล็งเห็นว่าโปรโตคอลอาร์พีแอลสามารถปรับปรุงให้สามารถรองรับการทำงานแบบโหนดที่เคลื่อนที่ได้ การปรับปรุงโปรโตคอลนั้นจะมีสองส่วนที่สำคัญ คือกระบวนการเลือกโหนดแม่ (Parent Selection) และกระบวนการตรวจจับโหนดเคลื่อนที่ (Mobile Detection) ซึ่งปัญหาหลักของการสร้างเส้นทางสำหรับโหนดที่เคลื่อนที่คือการตัดสินใจว่าจะเชื่อมต่อหรือเลือกโหนดใดเป็นโหนดแม่ในส่วนของการตรวจจับโหนดเคลื่อนที่จะเป็นการเริ่มต้นร้องขอเส้นทางใหม่เมื่อโหนดกำลังเคลื่อนที่ออกจากกระแสนำส่งข้อมูล นอกจากนี้ยังมีตัวแปรอื่น ๆ ที่ใช้ในการปรับปรุงโปรโตคอลให้รองรับการเคลื่อนที่ ยกตัวอย่างเช่น Trickle Algorithm<sup>3</sup> และกำหนดช่วงเวลาการส่งข้อความดีไอเอสเป็นคาบเวลา เป็นต้น

สำหรับปัญหาสำคัญที่เกิดขึ้นในสถานการณ์ที่มีโหนดเคลื่อนที่คือ โหนดเคลื่อนที่ไม่สามารถติดต่อกับโหนดแม่เดิมได้เมื่อออกจากกระแสนำส่งข้อมูลและไม่สามารถเลือกโหนดแม่ตัวใหม่ได้อย่างถูกต้อง สำหรับโปรโตคอลอาร์พีแอลจะมีการ

กำหนดช่วงเวลาในการส่งข้อความควบคุม (Control Message) เพื่อนำข้อมูลมาพิจารณาปรับปรุงเส้นทางและเลือกโหนดแม่ที่ดีที่สุด โจทย์วิจัยที่น่าสนใจของปัญหานี้คือ โหนดเคลื่อนที่จะรู้ได้อย่างไรว่ากำลังจะเคลื่อนที่ออกจากกระแสนำส่งข้อมูลจากโหนดแม่ตัวนั้นและจะเลือกโหนดแม่ได้รวดเร็วและถูกต้องได้อย่างไร

ปัจจุบันมีโปรแกรมจำลองที่รองรับการทำงานของเครือข่ายเซ็นเซอร์ไร้สายอยู่จำนวนมาก<sup>4</sup> จากการศึกษาและสังเกตการทำงานของโหนดเคลื่อนที่ในโปรแกรมจำลอง ผู้วิจัยพบว่าตารางเก็บที่อยู่ของโหนดข้างเคียง (Neighbor Table) ของโหนดที่เคลื่อนที่จะมีข้อมูลโหนดข้างเคียงเดิมในแคช (Neighbor Cache)<sup>5</sup> เพิ่มขึ้นในขณะที่เคลื่อนที่ผ่านโหนดข้างเคียง ซึ่งบางครั้งในขณะที่กำลังจะเลือกโหนดแม่ที่โหนดเคลื่อนที่จะนำค่าลำดับชั้น (Rank) ของโหนดข้างเคียงที่เคยผ่านมาแล้ว (โหนดข้างเคียงที่อาจจะอยู่นอกกระแสนำส่งข้อมูล) มาคำนวณด้วยซึ่งอาจจะเกิดกรณีที่แย่มากที่สุดคือ ค่าลำดับชั้นของโหนดข้างเคียงดังกล่าวมีค่าที่ดีกว่าโหนดข้างเคียงที่อยู่ในระยะ ทำให้โหนดเคลื่อนที่เลือกโหนดข้างเคียงที่ไม่ได้อยู่ในระยะเป็นโหนดแม่ สำหรับวิธีการแก้ไขปัญหานี้ผู้วิจัยจะกล่าวต่อไปในหัวข้อ กระบวนการเลือกโหนดแม่และกระบวนการตรวจจับโหนดเคลื่อนที่

## การเคลื่อนที่ของโหนดในซิกโลแพน

การปรับปรุงโปรโตคอลอาร์พีแอลให้สนับสนุนการทำงานในกรณีที่มีโหนดเคลื่อนที่นั้นมีส่วนประกอบที่นิยมคือ กระบวนการเลือกโหนดแม่และกระบวนการตรวจจับโหนดเคลื่อนที่

สำหรับกระบวนการเลือกโหนดแม่ นั้น ความรวดเร็วและความถูกต้องจะขึ้นอยู่กับพิจารณาข้อมูลของข้อความดีไอเอส (DIO Message) ที่ได้รับจากโหนดข้างเคียง ซึ่งตามมาตรฐานของโปรโตคอลอาร์พีแอลนั้นการเลือกโหนดแม่ที่ดีที่สุดจะขึ้นอยู่กับ Objective Function (OF) ซึ่งถ้าใช้ Objective Function Zero (OF0)<sup>6</sup> แล้วโหนดข้างเคียงที่มีค่าลำดับชั้น Rank with Hysteresis Objective Function (MRHOF)<sup>7</sup> จะพิจารณาโหนดแม่จากจำนวนการส่งข้อมูลที่คาดหวัง (Expected Transmission Count : ETX) ที่ดีที่สุด ปัจจุบันมีงานวิจัยจำนวนมากที่ได้เสนอวิธีการปรับปรุงกระบวนการนี้ ยกตัวอย่างเช่น การใช้ค่าลำดับชั้นและค่าคุณภาพการเชื่อมต่อ (Link Quality Indication : LQI) จากข้อความดีไอเอสของโหนดข้างเคียงในการพิจารณาเลือกโหนดแม่ที่ดีที่สุด<sup>8</sup>, การใช้จำนวนการส่งข้อมูลที่คาดหวังในการพิจารณาโหนดแม่ตัวใหม่ กำหนดเวลาในการส่งข้อความดีไอเอสและดีไอเอสขึ้นมาใหม่หลัง

จากที่เลือกโหนดแม่ได้แล้ว<sup>9</sup>, การใช้ค่าลำดับชั้นระหว่างโหนดซึ่งจะให้ความสำคัญของโหนดที่อยู่หนึ่งกำหนดให้เป็นโหนดแม่เสมอแต่ถ้าไม่มีโหนดที่อยู่หนึ่งหรือมีโหนดที่อยู่หนึ่งทั้งคู่จะทำการเลือกโหนดแม่จากค่าลำดับชั้นที่มีค่าต่ำสุด<sup>10</sup>, การสร้างตัวเปรียบเทียบใหม่คือค่าโครนา (Corona) ซึ่งกำหนดให้แต่ละโหนดสามารถอยู่ได้เพียงหนึ่งโครนาเท่านั้น โหนดรากเป็นเพียงโหนดเดียวที่มีค่า C\_ID เป็น 0 ได้ และโหนดเคลื่อนที่จะตรวจสอบค่า C\_ID ในตารางเก็บที่อยู่เพื่อเลือกโหนดข้างเคียงที่มีค่า C\_ID น้อยที่สุดเป็นโหนดแม่และจะเพิ่มค่า C\_ID ของตัวเองเพิ่มอีกหนึ่งลำดับ<sup>11</sup>, การกำหนดโปรโตคอลอาร์พีแอลให้เป็นโหนดไม่เก็บข้อมูล (Non-storing mode) และใช้จำนวนการส่งข้อมูลที่คาดหวังในการเลือกโหนดแม่<sup>12</sup>, การกำหนดให้โหนดเคลื่อนที่ทำการเลือกโหนดแม่อีกครั้งโดยดูจากค่าความแรงของสัญญาณเฉลี่ยที่น้อยเกินกว่าที่กำหนด<sup>13</sup> และสุดท้ายเป็นการใช้ตัวประมาณค่าฟัชซี (Fuzzy Estimator) เพื่อคำนวณคุณภาพการเชื่อมต่อโดยประมวลผลจากจำนวนการส่งข้อมูลที่คาดหวัง ค่าเฉลี่ยความแรงของสัญญาณและความเปลี่ยนแปลงของอัตราความผิดพลาดในการตัดสินใจสัญลักษณ์ (Symbol Error Rate : SER) ของการเชื่อมต่อ จากนั้นจะนำค่าที่ได้จากตัวประมาณค่าฟัชซีมาพิจารณาคุณภาพการเชื่อมต่อในขณะนั้น<sup>14</sup>

ในกระบวนการตรวจจับโหนดเคลื่อนที่นั้น จะเป็นการกระบวนการที่ระบุว่าโหนดใดเคลื่อนที่อยู่ หากตรวจพบการเคลื่อนที่ที่จะเริ่มกระบวนการเลือกโหนดแม่ สำหรับตัวแปรที่นิยมใช้ในการตัดสินใจ คือ ค่าความแรงของสัญญาณ (Received Signal Strength Indicator : RSSI) ซึ่งถ้าเกินกว่าค่าที่กำหนดจะถือว่าโหนดนั้นกำลังเคลื่อนที่ออกจากโหนดแม่ปัจจุบัน นอกจากนี้ยังมีงานวิจัยจำนวนมากได้เสนอวิธีการอื่นเพื่อปรับปรุงกระบวนการนี้ เช่น การกำหนดให้มีโหนดพี่น้อง (Sibling Node) เพื่อรับข้อมูลแทนโหนดแม่ที่ไม่สามารถส่งข้อมูลไปถึง<sup>8</sup>, การกำหนดให้โหนดที่เคลื่อนที่ไม่สามารถเลือกโหนดที่เคยเป็นโหนดลูกเป็นโหนดแม่ได้ เพื่อป้องกันการเกิดวนซ้ำบนเส้นทาง ซึ่งใช้วิธีการตรวจสอบค่าไอดีของโหนดแม่ (Parent ID) ในข้อความดีไอโอว่ามีค่าตรงกับไอดีของโหนดมันหรือไม่<sup>9</sup>, การปรับเปลี่ยนช่วงเวลาการส่งข้อความดีไอเอสจากความถี่ในการเลือกโหนดแม่ถ้ามีการเปลี่ยนแปลงบ่อยในช่วงเวลาที่กำหนดจะทำให้อัตราการส่งข้อความดีไอเอสจะลดลง ในทางกลับกันถ้าใช้โหนดแม่ตัวเดิมบ่อยจะทำให้อัตราการส่งข้อความดีไอเอสเพิ่มมากขึ้น<sup>10</sup>, การกำหนดให้เราเตอร์เคลื่อนที่ (Mobile Router) ทำการกระจายข้อความดีไอเอสไปยังโหนดข้างเคียงจนกว่าจะได้รับค่าข้อความดีไอโอหรือจนกระทั่งหมดเวลาในการส่งข้อความดีไอเอส<sup>11</sup>, การกำหนดให้

โหนดรับรู้ว่ามีโหนดเคลื่อนที่จากการติดป้ายการเคลื่อนที่ (Mobility Flag) ในข้อความดีไอโอ (DAO Message) ที่ได้รับ ซึ่งจะเปลี่ยนเวลาในการส่งข้อความดีไอโอให้ใช้ Trickle Algorithm แบบย้อนกลับเพื่อให้เหมาะสมกับสถานการณ์ที่มีการเคลื่อนที่ สำหรับโหนดที่ไม่ใช่โหนดเคลื่อนที่จะใช้ Trickle Algorithm ตามมาตรฐานของโปรโตคอลอาร์พีแอล<sup>12</sup>, การกำหนดให้โหนดเคลื่อนที่ส่งข้อความดีไอเอสตามช่วงเวลาที่กำหนดไปยังโหนดข้างเคียงซึ่งโหนดข้างเคียงจะอ่านค่าความแรงของสัญญาณและนับจำนวนของแพ็คเก็ตดีไอเอสเพื่อใช้ในการคำนวณเวลาตอบกลับและจะส่งข้อความดีไอโอพร้อมกับค่าเฉลี่ยความแรงของสัญญาณไปยังโหนดเคลื่อนที่<sup>13</sup>, การกำหนดให้เมื่อตรวจสอบพบเส้นทางเสียหายและมีความต้องการที่จะซ่อมแซมเส้นทางแล้ว จะส่งข้อความออกไปยังโหนดข้างเคียงและเมื่อโหนดข้างเคียงได้รับข้อความนี้ส่งข้อความที่ประกอบไปด้วยจำนวนการส่งข้อมูลที่คาดหวัง ค่าเฉลี่ยความแรงของสัญญาณ และความเปลี่ยนแปลงของอัตราความผิดพลาดในการตัดสินใจสัญลักษณ์ไปยังโหนดเคลื่อนที่เพื่อใช้ในการพิจารณาการเลือกโหนดแม่<sup>14</sup>

## วัตถุประสงค์

- (1) เพื่อพัฒนาโปรโตคอลให้สามารถรับรู้ถึงการเคลื่อนที่ของโหนด
- (2) เพื่อพัฒนาโปรโตคอลให้สามารถเลือกโหนดแม่ที่ดีที่สุดได้อย่างถูกต้องและรวดเร็วมากขึ้น
- (3) เพื่อเพิ่มอัตราความสำเร็จในการส่งข้อมูลในขณะที่โหนดมีการเคลื่อนที่โดยเปรียบเทียบกับโปรโตคอลอาร์พีแอล

## วิธีดำเนินการวิจัย

### การปรับปรุงโปรโตคอล

ผู้วิจัยได้ทำการปรับปรุงโปรโตคอลอาร์พีแอลและได้เรียกโปรโตคอลที่ถูกปรับปรุงใหม่ว่าโปรโตคอลไอเอ็มอาร์พีแอล (Improve mobility for RPL: imRPL) โดยในสถานการณ์ที่ทดลองจะมีโหนด 3 ประเภท คือ โหนดราก (Root Node), โหนดเราเตอร์ (Router Node) และโหนดเคลื่อนที่ (Mobile Node) ซึ่งโหนดรากนั้นทำหน้าที่รับและส่งข้อมูลตอบรับ (Acknowledgement: ACK) กลับไปยังต้นทาง โหนดเราเตอร์ทำหน้าที่ค้นหาเส้นทางและส่งต่อข้อมูลจากโหนดเคลื่อนที่ไปยังโหนดรากและสุดท้ายโหนดเคลื่อนที่จะทำหน้าที่ส่งข้อมูลไปยังโหนดราก

โปรโตคอลไอเอ็มอาร์พีแอลจะมีกระบวนการเลือกโหนดแม่และกระบวนการตรวจจับโหนดเคลื่อนที่ สำหรับรายละเอียดของแต่ละกระบวนการผู้วิจัยได้อธิบายในหัวข้อต่อไป

**กระบวนการเลือกโหนดแม่**

การเลือกโหนดแม่ของโปรโตคอลอาร์พีแอลนั้นจะพิจารณาจากค่าลำดับชั้น (Rank) จากข้อความดีไอโอที่ได้รับ จากนั้นจะเปรียบเทียบกับค่าลำดับชั้นของโหนดข้างเคียงที่มีในแคชของตารางเก็บที่อยู่ของโหนดข้างเคียงเพื่อพิจารณาว่าจะใช้โหนดแม่ตัวเดิมหรือเปลี่ยนโหนดแม่ตัวใหม่ โดยจะเลือกโหนดข้างเคียงที่มีค่าลำดับชั้นที่ดีที่สุดเป็นโหนดแม่ เมื่อได้โหนดแม่แล้วโหนดจะเปลี่ยนเส้นทางจากโหนดแม่เดิมไปยังโหนดแม่ตัวใหม่ จากวิธีการดังกล่าวเมื่อโหนดได้เคลื่อนที่ไปในเครือข่ายในระยะหนึ่งแล้วผู้วิจัยพบว่า รายการโหนดข้างเคียงที่อยู่ในแคชของโหนดเคลื่อนที่นั้นเพิ่มมากขึ้น ซึ่งโหนด

ข้างเคียงบางโหนดอาจจะไม่อยู่ในระยะการส่งข้อมูลอีกแล้ว ยิ่งไปกว่านั้นในบางกรณีค่าลำดับชั้นของโหนดข้างเคียงที่ไม่ได้อยู่ในระยะการส่งข้อมูลนั้นมีค่าลำดับชั้นที่ดีกว่าโหนดข้างเคียงที่สามารถติดต่อได้ จะส่งผลให้โหนดเคลื่อนที่เลือกโหนดข้างเคียงดังกล่าวเป็นโหนดแม่และส่งผลให้เกิดการสูญเสียข้อมูลจำนวนมาก จาก (Figure 1) แสดงการเปรียบเทียบข้อมูลในแคชของโหนดเคลื่อนที่ระหว่างโปรโตคอลไอเอ็มอาร์พีแอลและอาร์พีแอลในระหว่างที่โหนดมีการเคลื่อนที่ ซึ่งจะเห็นว่าอาร์พีแอลจะบันทึกข้อมูลโหนดข้างเคียงของโหนดเดิมที่เคยผ่านไว้

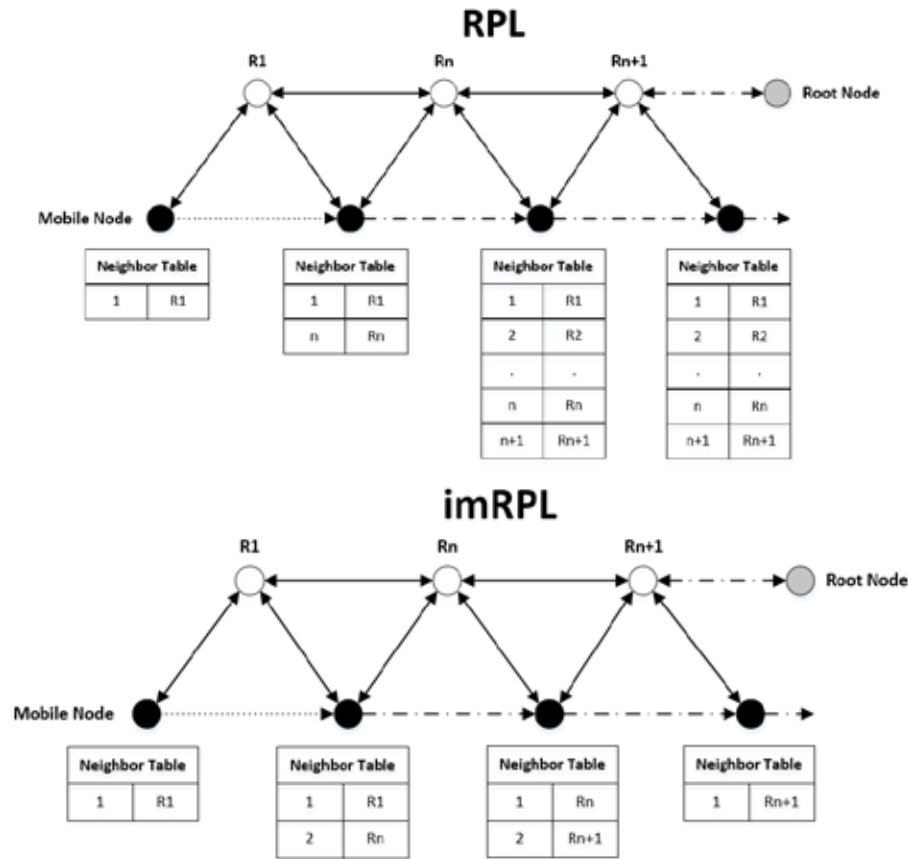


Figure 1 Neighbor table of RPL and imRPL

ในตารางเก็บที่อยู่ของโหนดข้างเคียงซึ่งในส่วนนี้เป็นสาเหตุที่ทำให้เกิดข้อผิดพลาดในการคำนวณหาโหนดแม่ที่ดีที่สุดในช่วงที่มีการเคลื่อนที่

ดังนั้นผู้วิจัยจึงได้ทำการปรับปรุงกระบวนการนี้ โดยกำหนดให้โหนดเคลื่อนที่ทำการขจัดแคชทุกครั้งเมื่อเริ่มกระบวนการเลือกโหนดแม่ เพื่อป้องกันการคำนวณที่ผิดพลาดและลดการใช้หน่วยความจำ นอกจากนี้ยังมีการตรวจสอบข้อความดีไอโอที่ได้รับว่ามีค่าความแรงของสัญญาณ (RS-

$SI_{DIO}$ ) น้อยกว่าค่าความแรงของสัญญาณที่กำหนดไว้ ( $RS-SI_{Threshold}$ ) หรือไม่ ถ้าน้อยกว่าค่าที่ตั้งไว้จะทำการทิ้งข้อความดีไอโออันนั้นและไม่นำมาพิจารณา ผู้วิจัยได้กำหนดให้ค่า  $RS-SI_{Threshold}$  เท่ากับ  $-85$  dBm ซึ่งเป็นค่าจากการทดลองแล้วให้ผลลัพธ์ที่ดีที่สุด ในส่วนนี้จะมีการนับจำนวนการทิ้งข้อความดีไอโอซึ่งถ้ามากกว่าค่าที่ตั้งไว้จะยอมให้ใช้ข้อความดีไอโอที่มีค่าความแรงของสัญญาณต่ำได้ หลังจากนั้นจะนำข้อความดีไอโอที่ได้รับจากโหนดข้างเคียงมาพิจารณาโหนดแม่ที่ดีที่สุด

ผู้วิจัยได้กำหนดช่วงเวลา timer<sub>DIS Threshold</sub> เพื่อกำหนดช่วงเวลาในการรอข้อความดีไอโอหลังจากที่ส่งข้อความดีไอเอสออกไป ผู้วิจัยได้กำหนดค่า timer<sub>DIS Threshold</sub> เท่ากับ 5 วินาที โดยได้มาจากการทดลองวัดเวลาเฉลี่ยตั้งแต่ส่งข้อความดีไอเอสจนกระทั่งได้รับข้อความดีไอโอ ซึ่งได้เวลาเฉลี่ยเท่ากับ 3.3446 วินาทีและเวลาที่มากที่สุดเท่ากับ 4.235 วินาที

**Algorithm 1. Parent Selection**

```

Begin
  Remove Neighbor Cache;
  Broadcast DIS Message;
  timerDIS = 0;
  While timerDIS < timerDIS threshold
    If RSSIDIO >= RSSIThreshold or DropDIO > 2
      Calculate Best Parent;
      DropDIO = 0;
    Else If RSSIDIO < RSSIThreshold
      Drop DIO Message;
      DropDIO++;
      Parent Selection;
    End
  End
End
    
```

**Figure 2** Algorithm 1: Parent Selection

สำหรับกระบวนการดังกล่าวที่แสดงใน (Figure 2) ผู้วิจัยได้ปรับปรุงโปรโตคอลเป็นสองเวอร์ชัน คือ ไอเอ็มอาร์พีแอลเวอร์ชันหนึ่ง (imRPLv1) ที่เป็นโปรโตคอลที่ไม่มีการจัดแคชและไอเอ็มอาร์พีแอลเวอร์ชันสอง (imRPLv2) ที่เป็นโปรโตคอลที่มีการจัดแคชในกระบวนการเลือกโหนดแม่เพื่อเปรียบเทียบผลกระทบของการจัดแคชและไม่จัดแคช

**กระบวนการตรวจจับโหนดเคลื่อนที่**

กระบวนการนี้จะกำหนดเวลาเพื่อรอรับค่าข้อมูลตอบกลับจากโหนดรากซึ่งถ้าหากไม่ได้ข้อมูลตอบกลับมาตามระยะเวลาที่กำหนด (Timer<sub>Threshold</sub>) โหนดจะเข้าสู่กระบวนการเลือกโหนดแม่อีกครั้ง โดยจะคำนวณค่า Threshold จากสูตร

$$Threshold = 0.616h^2 - 1.4183h + 0.9292$$

ซึ่งได้มาจากการทดลองวัดค่าความล่าช้า (Delay) ที่สัมพันธ์กับฮอป (Hop) จำนวน 1 ถึง 5 ฮอปจะได้กราฟที่มีลักษณะคล้ายสมการพหุนามกำลังสองแล้วคำนวณเป็นสูตรตั้งสมการข้างต้น จากสูตรกำหนดให้ *h* คือ จำนวนฮอปของเส้นทางที่ใช้ในการสื่อสารปัจจุบัน การคำนวณค่าฮอปนั้นสามารถคำนวณได้จากสูตร

$$h = \left(\frac{Rank}{256}\right) - 1$$

สำหรับค่าลำดับชั้นจะอยู่ในข้อความดีไอโอซึ่งสามารถถอดแพ็คเก็ตแล้วนำมาใช้ประโยชน์ได้และ Threshold คือค่าเวลาความล่าช้าของแพ็คเก็ตโดยมีความสัมพันธ์กับจำนวนฮอปซึ่งค่า Timer<sub>Threshold</sub> ของแต่ละฮอปจะขึ้นอยู่กับค่า Threshold ที่คำนวณได้โดยพิเศษขึ้นเป็นจำนวนเต็มดัง Table 1

**Table 1** Relation between hop and Timer<sub>Threshold</sub>

Hop	Average Delay	Threshold	Timer <sub>Threshold</sub>
1	0.2173	0.1269	1
2	0.478	0.5566	1
3	1.9109	2.2183	3
4	5.5995	5.112	6
5	9.0441	9.2377	10

เมื่อโหนดตรวจสอบค่าความแรงของสัญญาณของข้อความที่ตอบกลับ (RSSI<sub>ACK</sub>) แล้วพบว่ามีความเกินกว่าค่าความแรงของสัญญาณที่กำหนด (RSSI<sub>Threshold</sub>) หรือไม่ ถ้าน้อยกว่าค่าที่กำหนดจะเริ่มกระบวนการเลือกโหนดแม่และรีเซ็ตเวลา Timer<sub>ACK</sub> ใหม่ จากนั้นจะรอรับข้อความดีไอโอเพื่อนำมาคำนวณหาโหนดแม่ที่ดีที่สุดในการกระบวนการเลือกโหนดแม่ต่อไป กระบวนการดังกล่าวที่แสดงใน Figure 3

**Algorithm 2. Mobile Detection**

```

Begin
  If TimerACK < TimerThreshold then
    If RSSIACK < RSSIThreshold then
      Parent Selection;
      Reset TimerACK;
    Else
      Send Data;
    End
  Else
    Parent Selection;
    Reset TimerACK;
  End
End
    
```

**Figure 3** Algorithm 2: Mobile Detection

**เกณฑ์ในการประเมินประสิทธิภาพ**

งานวิจัยนี้ได้กำหนดเกณฑ์ในการประเมินประสิทธิภาพของโปรโตคอล คือ 1) ผลการวัดจำนวนข้อความควบคุม

2) ผลการวัดอัตราความสำเร็จในการส่งข้อมูล 3) ผลการวัดความเร็วและความถูกต้องในกระบวนการเลือกโหนดแม่ สำหรับการประเมินประสิทธิภาพผู้วิจัยจะทดสอบบนโปรแกรมจำลองโดยเปรียบเทียบประสิทธิภาพของโปรโตคอลที่พัฒนาขึ้นกับโปรโตคอลอาร์พีแอล

**วิธีการจำลองและสถานการณ์ในการจำลอง**

ผู้วิจัยได้ใช้โปรแกรม Cooja Simulator<sup>15</sup> บนระบบปฏิบัติการ Contiki 3.0<sup>16</sup> ในการทดสอบประสิทธิภาพและได้ใช้ปลั๊กอินการเคลื่อนที่<sup>17</sup> ในการทำให้โหนดสามารถเคลื่อนที่ได้ การทดลองแบ่งออกเป็นสองสถานการณ์ โดยสถานการณ์ที่หนึ่ง กำหนดให้โหนดเราเตอร์และโหนดรากวางเรียงกันเป็นเส้นตรงและกำหนดให้โหนดเคลื่อนที่วนรอบโหนดดังกล่าวเป็นจำนวน 10 รอบ ในการทดลองแต่ละครั้งจะเพิ่มโหนดเราเตอร์ขึ้นทีละ 1 โหนดจนกระทั่งถึง 5 โหนดโดยที่โหนดเราเตอร์แต่ละตัวนั้นวางห่างกัน 40 เมตร (ระยะห่างที่มากที่สุดที่โหนดไม่สามารถสื่อสารกันได้คือ 50 เมตร) สำหรับตำแหน่งการวางโหนดในสถานการณ์ที่หนึ่งและสองดังแสดงใน Figure 4 และ Figure 5 ตามลำดับ

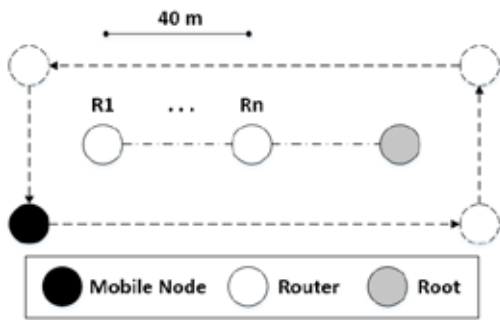


Figure 4 Scenario one

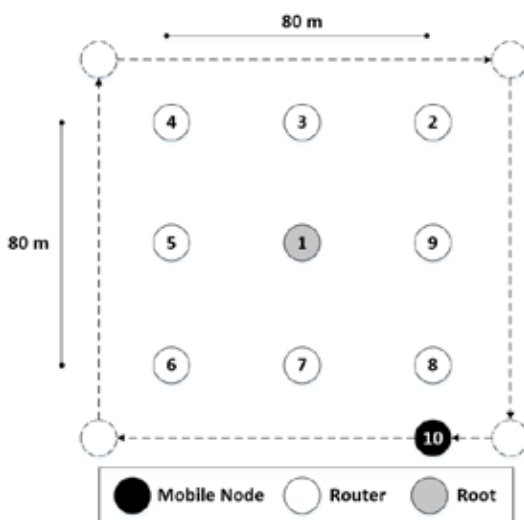


Figure 5 Scenario two

**การกำหนดตัวแปรบนโปรแกรมจำลอง**

ผู้วิจัยได้กำหนดให้ในสถานการณ์ที่หนึ่งและสถานการณ์ที่สองมีตัวแปรดัง Table 2 และ Table 3 ตามลำดับ โดยเป็นการเปรียบเทียบระหว่างโปรโตคอลอาร์พีแอลกับโปรโตคอลไอเอ็มอาร์พีแอลซึ่งเป็นการวัดจำนวนของข้อความควบคุมและวัดอัตราความสำเร็จในการส่งข้อมูล โหนดเคลื่อนที่จะเดินทางด้วยความเร็วคงที่ซึ่งกำหนดให้มีความเร็ว 1 เมตรต่อวินาทีถึง 5 เมตรต่อวินาที ในขณะที่เดินทางโหนดเคลื่อนที่จะส่งข้อมูลไปยังโหนดรากด้วยอัตรา 1 แพ็คเก็ตต่อวินาทีสำหรับในสถานการณ์ที่สองผู้วิจัยได้ทำการวัดเวลาและความถูกต้องในการเลือกโหนดแม่ ซึ่งจะเป็นการเปรียบเทียบระหว่างโปรโตคอลอาร์พีแอล ไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งและไอเอ็มอาร์พีแอลเวอร์ชันสอง

Table 2 Parameter of scenario one

Parameters	Detail
Network Simulator	Cooja Simulator (Contiki OS 3.0)
Protocol	RPL, imRPLv2
Performance Measure	Number of Control Message, Packet Delivery Ratio
Number of Node	1 Root Node, 1-5 Router Node, 1 Mobile Node
Number of Round	10 Laps
Emulated Nodes	Zolertia (Z1)
Packet Rate	1 Packet / Second
Speed of Mobile Node	1 m/s, 2 m/s, 3 m/s, 4 m/s, 5m/s

Table 3 Parameter of scenario two

Parameters	Detail
Network Simulator	Cooja Simulator (Contiki OS 3.0)
Protocol	RPL, imRPLv1, imRPLv2
Performance Measure	Number of Control Message, Packet Delivery Ratio, Parent Selection Time, Correct Parent Selection
Number of Node	1 Root Node, 8 Router Node, 1 Mobile Node
Number of Round	5 Laps
Emulated Nodes	Zolertia (Z1)
Packet Rate	1 Packet / Seconds
Speed of Mobile Node	1 m/s

### ผลการวิจัยและอภิปรายผล

#### ผลการวัดจำนวนของข้อความควบคุม

DODAG จะใช้ข้อความควบคุมเพื่อสร้างและปรับปรุงเส้นทางของเครือข่าย ซึ่งถ้าหากเป็นสถานการณ์ที่มีโหนดเคลื่อนที่จะทำให้เกิดข้อความควบคุมเพิ่มขึ้นเนื่องจากการสร้างและปรับปรุงเส้นทางใหม่อยู่เสมอ ดังนั้นถ้าสามารถควบคุมการส่งข้อความควบคุมให้เหมาะสมจะทำให้จำนวนของข้อความควบคุมในเครือข่ายมีจำนวนน้อยลง ผู้วิจัยได้เปรียบเทียบจำนวนข้อความควบคุมของข้อความดีไอเอส ดีไอโอและดีไอโอระหว่างโปรโตคอลอาร์พีแอลและไอเอ็มอาร์พีแอล ผลการทดลองเป็นดังนี้

จาก Figure 6 แสดงผลการเปรียบเทียบจำนวนข้อความดีไอเอสของโปรโตคอลอาร์พีแอลและไอเอ็มอาร์พีแอลเวอร์ชันสอง สำหรับสถานการณ์ที่หนึ่งผู้วิจัยพบว่าโปรโตคอลอาร์พีแอลมีจำนวนของข้อความดีไอเอสน้อยกว่าโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันสอง เนื่องจากโปรโตคอลอาร์พีแอลใช้ Trickle Algorithm ในการควบคุมการส่งข้อความควบคุมซึ่งอัลกอริทึมนี้ไม่เหมาะกับสถานการณ์ที่มีโหนดเคลื่อนที่ เนื่องจากช่วงเวลาในการสร้างหรือปรับปรุงเส้นทางใหม่อาจจะไม่อยู่ในช่วงเวลาของ Trickle Algorithm ดังนั้นจึงมีโอกาสที่แพ็คเก็ตจะส่งไปไม่ถึงปลายทาง ในส่วนของโปรโตคอลไอเอ็มอาร์พีแอล

เวอร์ชันสองมีจำนวนของข้อความดีไอเอสสูงตามความเร็วของโหนดเคลื่อนที่เนื่องจากโหนดที่เคลื่อนที่ด้วยความเร็วจะมีโอกาสในการเปลี่ยนโหนดแม่บ่อยมากขึ้นในสถานการณ์ที่สองแสดงให้เห็นถึงจำนวนของข้อความดีไอเอสที่มีมากขึ้นในแต่ละรอบสำหรับโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งและไอเอ็มอาร์พีแอลเวอร์ชันสอง ซึ่งในส่วนของโปรโตคอลอาร์พีแอลจะมีจำนวนข้อความดังกล่าวเพิ่มขึ้นเล็กน้อย จำนวนของข้อความดีไอเอสที่เพิ่มขึ้นเป็นผลจากการร้องขอข้อมูลเพื่อเลือกโหนดแม่ เมื่อเข้าสู่กระบวนการตรวจจับโหนดเคลื่อนที่จาก Figure 7 จำนวนของข้อความดีไอโอที่มีมากสำหรับโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งและโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันสองเป็นผลมาจากการร้องขอเส้นทางใหม่จากข้อความดีไอเอส

ซึ่งถ้าหากกระบวนการตรวจจับโหนดเคลื่อนที่เกิดขึ้นบ่อยจะส่งผลให้จำนวนของข้อความดีไอโอส่งขึ้นตามไปด้วยจาก Figure 8 เป็นผลจากการปรับปรุงเส้นทางใหม่ ซึ่งโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันหนึ่ง และไอเอ็มอาร์พีแอลเวอร์ชันสองจะมีจำนวนของข้อความดีไอโอส่งกว่าโปรโตคอลอาร์พีแอลเนื่องจากโปรโตคอลไอเอ็มอาร์พีแอลมีการปรับปรุงเส้นทางมากกว่าโปรโตคอลอาร์พีแอล

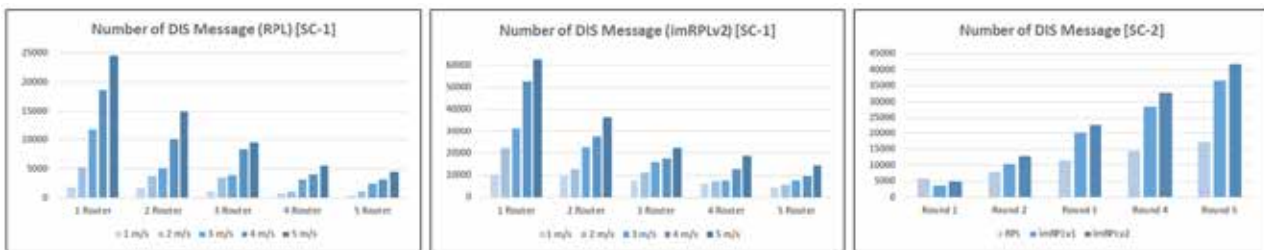


Figure 6 Scenario one and two: number of DIS message (RPL and imRPL)



Figure 7 Scenario one and two: number of DIO message (RPL and imRPL)

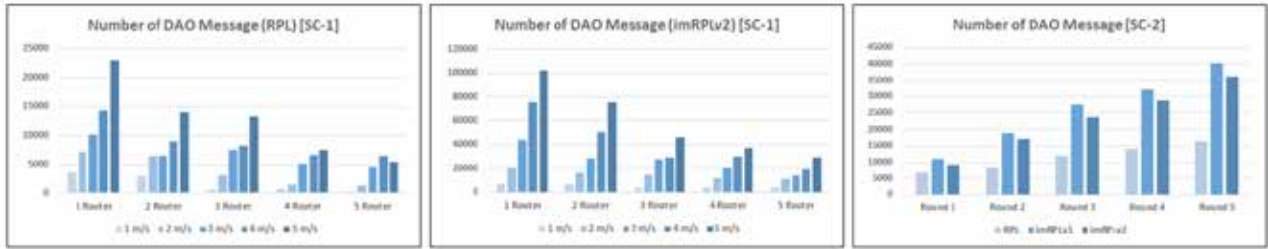


Figure 8 Scenario one and two: number of DAO message (RPL and imRPL)



Figure 9 Scenario one and two: packet delivery ratio (RPL and imRPL)

**ผลการวัดอัตราความสำเร็จในการส่งข้อมูล**

ความต่อเนื่องในการเชื่อมต่อเครือข่ายในขณะที่เคลื่อนที่มีผลกระทบต่ออัตราความสำเร็จในการส่งข้อมูลเป็นอย่างมาก ซึ่งถ้าหากไม่สามารถเลือกโหนดแม่ได้จะส่งผลให้ข้อมูลไม่สามารถส่งไปถึงปลายทาง ยิ่งไปกว่านั้นความเร็วในการเลือกโหนดแม่ ยังส่งผลกระทบต่อความต่อเนื่องในการส่งข้อมูลด้วยเช่นกัน ดังนั้นการเลือกโหนดแม่ที่รวดเร็วและถูกต้องที่สุดจึงเป็นสิ่งสำคัญต่อการปรับปรุงโปรโตคอล ผู้วิจัยได้เปรียบเทียบอัตราความสำเร็จในการส่งข้อมูลของโปรโตคอลอาร์พีแอลและโปรโตคอลไอเอ็มอาร์พีแอลได้ผลดังนี้

จาก Figure 9 เป็นการเปรียบเทียบอัตราความสำเร็จในการส่งข้อมูลของโปรโตคอลอาร์พีแอลและโปรโตคอลไอเอ็มอาร์พีแอลจากสถานการณ์ที่หนึ่ง แสดงให้เห็นถึงจำนวนโหนดเราเตอร์และความเร็วของการเคลื่อนที่ของโหนดที่มีมากขึ้นส่งผลให้อัตราความสำเร็จในการส่งข้อมูลลดลง ในสถานการณ์ที่สองแสดงให้เห็นว่าจำนวนรอบที่มากขึ้นทำให้ค่าอัตราความสำเร็จในการส่งข้อมูลลดลงเล็กน้อยซึ่งจากการเปรียบเทียบประสิทธิภาพระหว่างโปรโตคอลอาร์พีแอล, ไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งและไอเอ็มอาร์พีแอลเวอร์ชันสองในทั้งสองการทดลองพบว่าโปรโตคอลไอเอ็มอาร์พีแอลมีอัตราความสำเร็จในการส่งข้อมูลมากกว่าโปรโตคอลอาร์พีแอลโดยโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันสองมีอัตราความสำเร็จในการส่งข้อมูลมากที่สุด

**ผลการวัดความเร็วและความถูกต้องในกระบวนการเลือกโหนดแม่**

ความเร็วในการเลือกโหนดแม่นั้นเกิดจากโหนดคำนวณค่าลำดับชั้นของฟังก์ชันจุดประสงค์ที่อยู่ในโปรโตคอล

อาร์พีแอลซึ่งเป็นการเปรียบเทียบค่าลำดับชั้น ในตารางเก็บที่อยู่ของโหนดข้างเคียงสำหรับโหนดที่เคลื่อนที่นั้นเมื่อมีการเดินทางผ่านโหนดข้างเคียงไปแล้วจะตารางเก็บที่อยู่ของโหนดข้างเคียงยังคงเก็บรายการโหนดข้างเคียงเดิมที่เคยผ่านมาและนำค่าดังกล่าวมาใช้ในการพิจารณาเลือกโหนดแม่ที่ดีที่สุด ซึ่งการขจัดแคชทุกครั้งก่อนที่จะมีการเลือกโหนดแม่ตัวใหม่จะทำให้โหนดแม่ที่ต้องพิจารณามีน้อยลงอีกทั้งยังเป็นโหนดข้างเคียงล่าสุดที่โหนดเคลื่อนที่สามารถติดต่อได้ จึงส่งผลให้การหาค่าโหนดแม่ที่ดีที่สุดทำได้อย่างรวดเร็วและถูกต้อง ผู้วิจัยได้ทดลองเปรียบเทียบความเร็วและความถูกต้องในการเลือกโหนดแม่ระหว่างโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งกับ ไอเอ็มอาร์พีแอลเวอร์ชันสองโดยกำหนดให้วิธีการเลือกโหนดแม่และการตรวจจับโหนดเคลื่อนที่ยังคงเดิม

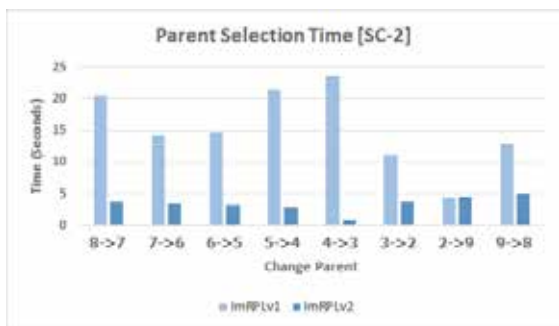
ผลการทดลองพบว่าความเร็วในการเลือกโหนดแม่หลังจากกระบวนการตรวจจับโหนดเคลื่อนที่ ในสถานการณ์ที่สองนั้นโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันสองใช้เวลาในกระบวนการเลือกโหนดแม่เฉลี่ย 3.46 วินาที โดยใช้เวลาน้อยที่สุดเท่ากับ 0.79 วินาที และได้ใช้เวลาที่มากที่สุดเท่ากับ 4.99 วินาที ในขณะที่โปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งใช้เวลาเฉลี่ย 15.39 วินาที โดยใช้เวลาน้อยที่สุดเท่ากับ 4.36 วินาที และใช้เวลามากที่สุดเท่ากับ 23.72 วินาที ผลการทดลองแสดงดัง Figure 10

นอกจากนี้ผู้วิจัยยังได้ทดสอบความถูกต้องในการเลือกโหนดแม่ที่โหนดเคลื่อนที่ได้เดินทางผ่านโดยในการทดลองนี้ได้นำโปรโตคอลอาร์พีแอลมาเปรียบเทียบกับซึ่งการทดลองจะทดลองในสถานการณ์ที่สองผลการทดลองพบว่าโปรโตคอลอาร์พีแอล, ไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งและไอเอ็มอาร์

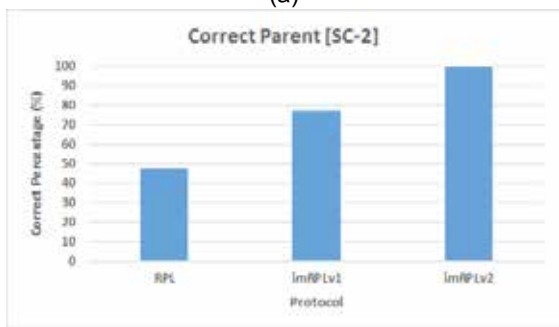


พีแอลเวอร์ชันสอง สามารถเลือกโหนดแม่ได้ถูกต้องเท่ากับ 47.5 เปอร์เซ็นต์, 77.5 เปอร์เซ็นต์และ 100 เปอร์เซ็นต์ ตามลำดับ

จากผลการทดลองทั้งสองการทดลองแสดงให้เห็นว่าการขจัดแคชทำให้โหนดมีความเร็วในการเลือกโหนดแม่มากขึ้น เนื่องจากการลดจำนวนโหนดที่ต้องนำมาพิจารณาและโหนดที่พิจารณานั้นเป็นโหนดที่สามารถเชื่อมต่อได้ ยิ่งไปกว่านั้นยังส่งผลให้สามารถเชื่อมต่อโหนดแม่ได้ทุกครั้งจึงทำให้โอกาสที่ส่งข้อมูลได้ต่อเนื่องมีมากขึ้น นอกจากนี้ยังส่งผลให้อัตราความสำเร็จในการส่งข้อมูลของโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันสองมีค่าสูงกว่าโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งและโปรโตคอลอาร์พีแอล



(a)



(b)

**Figure 10** Scenario two: (a) Time of Parent Selection (b) Correct Parent Selection

**สรุปผลการทดลอง**

งานวิจัยนี้ได้เสนอโปรโตคอลไอเอ็มอาร์พีแอลที่เป็นการปรับปรุงโปรโตคอลอาร์พีแอลของเครือข่ายแบบซิกโลแพนเพื่อสนับสนุนการทำงานในกรณีที่มีโหนดเคลื่อนที่ในเครือข่าย งานวิจัยนี้ได้เสนอการปรับปรุงสองกระบวนการคือ กระบวนการเลือกโหนดแม่และกระบวนการตรวจจับโหนดเคลื่อนที่ ซึ่งในกระบวนการเลือกโหนดแม่นั้นจะทำให้โหนดเคลื่อนที่สามารถเลือกโหนดแม่ที่ดีที่สุดได้อย่างรวดเร็วและถูกต้อง สำหรับกระบวนการตรวจจับโหนดเคลื่อนที่นั้นจะทำให้โหนดสามารถรับรู้ได้ว่าจะต้องส่งข้อความร้องขอไปยังโหนดแม่ตัวใหม่เมื่อใดและจะรับรู้

ว่าควรจะต้องเลือกข้อมูลจากโหนดข้างเคียงใดที่ใช้ในการพิจารณาโหนดแม่ตัวใหม่

ผู้วิจัยได้ทดสอบประสิทธิภาพของโปรโตคอลโดยการใช้โปรแกรม Cooja Simulator โดยทดสอบกับสองสถานการณ์ที่จำลองขึ้นมา ผลการทดลองพบว่าในสถานการณ์ที่หนึ่งนั้นที่ความเร็วการเคลื่อนที่ของโหนด 1 เมตรต่อวินาทีและจำนวนเราเตอร์โหนดตั้งแต่ 1 ถึง 5 โหนดนั้นโปรโตคอลไอเอ็มอาร์พีแอลมีอัตราความสำเร็จในการส่งข้อมูลเท่ากับ 88.92 เปอร์เซ็นต์, 82.30 เปอร์เซ็นต์ 75.52 เปอร์เซ็นต์, 66.20 เปอร์เซ็นต์และ 60.69 เปอร์เซ็นต์ ตามลำดับ ส่วนโปรโตคอลอาร์พีแอลมีอัตราความสำเร็จในการส่งข้อมูลเท่ากับ 68.54 เปอร์เซ็นต์, 52.54 เปอร์เซ็นต์, 46.78 เปอร์เซ็นต์, 42.27 เปอร์เซ็นต์และ 25.51 เปอร์เซ็นต์ตามลำดับ ซึ่งในสถานการณ์ที่มีโหนดเคลื่อนที่ทำให้โปรโตคอลไอเอ็มอาร์พีแอลมีจำนวนของข้อความควบคุมมากกว่าโปรโตคอลอาร์พีแอลซึ่งการทดลองในสถานการณ์ที่สองนั้นที่ความเร็วการเคลื่อนที่ของโหนด 1 เมตรต่อวินาทีโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันสองมีอัตราความสำเร็จในการส่งข้อมูลเท่ากับ 70.89 เปอร์เซ็นต์ โปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งมีอัตราความสำเร็จในการส่งข้อมูลเท่ากับ 58.84 เปอร์เซ็นต์ และอาร์พีแอลมีอัตราความสำเร็จในการส่งข้อมูลเท่ากับ 38.98 เปอร์เซ็นต์ ส่วนจำนวนของข้อความควบคุมนั้นโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันสองและไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งยังมีมากกว่าโปรโตคอลอาร์พีแอล

นอกจากนี้ผู้วิจัยยังได้ทดสอบเวลาในกระบวนการเลือกโหนดแม่ซึ่งพบว่าความเร็วในการเลือกโหนดแม่ตัวใหม่ระหว่างโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันสองกับไอเอ็มอาร์พีแอลเวอร์ชันหนึ่ง เฉลี่ยเท่ากับ 3.46 วินาที และ 15.39 วินาทีตามลำดับ ยิ่งไปกว่านั้นโปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันสองยังมีเปอร์เซ็นต์การเลือกโหนดแม่ที่ถูกต้องถึง 100 เปอร์เซ็นต์ ในขณะที่โปรโตคอลไอเอ็มอาร์พีแอลเวอร์ชันหนึ่งและโปรโตคอลอาร์พีแอลมีเปอร์เซ็นต์เท่ากับ 77.5 เปอร์เซ็นต์ และ 47.5 เปอร์เซ็นต์ ตามลำดับ ซึ่งการเลือกโหนดแม่ที่ถูกต้องและมีความรวดเร็วส่งผลให้อัตราความสำเร็จในการส่งข้อมูลมีจำนวนมากขึ้น

**เอกสารอ้างอิง**

1. G. Mulligan. The 6loWPAN architecture. Proceedings of the 4<sup>th</sup> Workshop on Embedded Networked Sensors (EmNets); 2007 June 25-26; Cork, Ireland: ACM; 2007. p. 78-82.
2. T. Winter, P. Thubert, A. Brandt, J. Hui, and R. Kelsey. RPL: IPv6 Routing Protocol for Low Power and

- Lossy Networks; IETF Request for Comments 6550; March 2012.
3. P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko. The Trickle Algorithm; IETF Request for Comments 6206; March 2011.
  4. Leila Ben Saad, Cedric Chauvenet, Bernard Tourancheau. Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: two cases studies. International Conference on Sensor Technologies and Applications (SENSORCOMM); 2011, Sep 2011, Nice, France. IARIA, 2011.
  5. Z. Shelby, S. Chakrabarti, E. Nordmark. Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs); IETF Request for Comments 6775; November 2012.
  6. P. Thubert. RPL Objective Function 0; IETF Request for Comments 6552; March 2012.
  7. O. Gnawali and P. Levis. The Minimum Rank with Hysteresis Objective Function; IETF Request for Comment 6719; September 2012.
  8. Ki-Sup Hong and Lynn Choi. DAG-based multipath routing for mobile sensor networks. *International Conference on ICT Convergence (ICTC)*; 2011, Sep 28-30; Seoul, Korea: IEEE; 2011. p. 261-6.
  9. K. Lee, R. Sudhaakar, L. Dai, S. Addepalli, M. Gerla. RPL under mobility. IEEE Consumer Communications and Networking Conference (CCNC); 2012 Jan 14-17; Las Vegas, NV, USA; IEEE; 2012. p. 300-4.
  10. In`es El Korbi, Mohamed Ben Brahim, Cedric Adjihy and Leila Azouz Saidane. Mobility Enhanced RPL for Wireless Sensor Networks. Proceedings of the 3<sup>rd</sup> International Conference on the Network of the Future (NOF); 2012 Nov 21-23; Tunis, Tunisia; IEEE; 2012. p. 1-8.
  11. Olfa Gaddour, Anis Koubaa, Raghuraman Rangarajan, Omar Cheikhrouhou, Eduardo Tovar, Mohamed Abid. Co-RPL: RPL Routing for Mobile Low Power Wireless Sensor Networks using Corona Mechanism. Proceedings of the 9<sup>th</sup> IEEE International Symposium on Industrial Embedded Systems (SIES), 2014 June 18-24; Pisa, Italy; IEEE; 2014. P. 200-9.
  12. C. Cobarzan, J. Montavont, T. Noel. Analysis and performance evaluation of rpl under mobility. IEEE Symposium on Computers and Communication (ISCC), 2014 June 23-26; Funchal, Madeira, Portugal: IEEE; 2014. p. 1-6.
  13. Hossein Fotouhi, Daniel Moreira, Mário Alves. mRPL: Boosting mobility in the Internet of Things. *Journal on Ad Hoc Networks* 2015; 26:17-35.
  14. JeongGil Ko and Marcus Chang. MoMoRo: Providing Mobility Support for Low-Power Wireless Applications. *IEEE Systems Journals* 2015; 9(2):585-94.
  15. anrg.usc.edu [Internet]. Cooja Simulator. [cited 16 March 2016]. Available from: [http://anrg.usc.edu/contiki/index.php/Cooja\\_Simulator](http://anrg.usc.edu/contiki/index.php/Cooja_Simulator)
  16. contiki-os.org [Internet]. Get Started with Contiki [cited 14 March 2016]. Available from: <http://www.contikios.org>
  17. anrg.usc.edu [Internet]. Mobility of Nodes in Cooja. [cited 23 April 2016]. Available from: [http://anrg.usc.edu/contiki/index.php/Mobility\\_of\\_Nodes\\_in\\_Cooja](http://anrg.usc.edu/contiki/index.php/Mobility_of_Nodes_in_Cooja)