รายงานการวิจัย

การหาเส้นทางเพื่อรองรับทราฟฟิคซึ่งการันตีคุณภาพการให้บริการในเครือข่ายเคลื่อนที่แบบแอดฮอค โดยใช้เทคนิครีอินฟอร์สเม้นท์เลิร์นนิ่ง

(Quality-of-Service Routing in Mobile Ad Hoc Networks using Reinforcement Learning Techniques)

ผู้วิจัย

วิภาวี อุสาหะ

สาขาวิชาวิศวกรรมโทรคมนาคม

สำนักวิชาวิศวกรรมศาสตร์

ได้รับทุนอุดหนุนการวิจัยจากมหาวิทยาลัยเทคโนโลยีสุรนารี

ผลงานวิจัยเป็นความรับผิดชอบของหัวหน้าโครงการวิจัยแต่เพียงผู้เดียว

ธันวาคม 2547

บทคัดย่อ

วัตถุประสงค์ของรายงานฉบับนี้คือการพัฒนากระบวนการตัดสินใจแบบออนไลน์เพื่อการค้นเส้นทางที่รองรับคุณภาพการบริการในเครือข่ายเคลื่อนที่แบบแอดฮอค โดยสามารถลดโอเวอร์เฮดในกระบวนการสื่อสาร เพิ่มประสิทธิภาพของระบบในระยะยาว และสามารถปฏิบัติการการได้อย่างดีเยี่ยมภายใต้ข้อมูลที่ไม่ชัดเจนสำหรับเครือข่ายที่มีรูปร่างเครือข่ายแบบพลวัติ องค์ความรู้ที่ได้จากรายงานฉบับนี้คือการประยุกต์ใช้กระบวนการเรียนรู้แบบรีอินฟอร์สเม้นและกลยุทธ์พาธ แคชชิ่ง สามารถลดขนาดของเมสเสจโอเวอร์เฮดในการค้นเส้นทางที่รองรับคุณภาพการบริการในเครือข่ายเคลื่อนที่แบบแอดฮอค

รายงานฉบับนี้ทำการกำหนดปัญหาการควบคุมเมสเสจโอเวอร์เฮดสำหรับการค้นเส้นทางที่รองรับคุณภาพการบริการในเครือข่ายเคลื่อนที่แบบแอดฮอคให้เป็นกระบวนการการตัดสินใจแบบมาร์คอฟภายใต้สภาวการณ์ที่สังเกตได้บางส่วน (partially observable Markov decision process หรือ POMDP) ด้วยการนำวิธีการตรวจสอบด้วยตั๋ว (Ticket-based probing หรือ TBP) แบบเดิมมาผนวกกับกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์สำหรับ POMDP ที่เรียกว่า วิธีออนโพลิซี มอนติ คาร์โล (on-policy Monte Carlo หรือ ONMC) และกลยุทธ์พาธ แคชชิ่ง (path caching) เพื่อใช้หานโยบายที่เหมาะสมในการค้นเส้นทางที่รองรับคุณภาพการบริการสำหรับเครือข่ายเคลื่อนที่แบบแอดฮอค จากผลการทดลองพบว่าภายใต้กระบวนการ POMDP สามารถเลือกนโยบายที่ดีสำหรับการจำหน่ายตั๋ว ซึ่งแสดงให้เห็นในรูปของผลตอบแทนสะสมต่อเอพพิโซดเมื่อเปรียบเทียบกับวิธี TBPแบบเดิมและวิธี ONMCที่ไม่มีพาธ แคชชิ่ง นอกจากนี้กระบวนการที่นำเสนอสามารถเพิ่มประสิทธิภาพในการควบคุมเมสเสจค้นหาโดยสามารถลดขนาดของเมสเสจโอเวอร์ได้ดีกว่าวิธีดังเดิม ซึ่งแสดงในรูปของอัตราความสำเร็จและมูลค่าเฉลี่ยตลอดเส้นทาง

Abstract

The underlying aim of this report is to develop on-line decision-making algorithm for QoS routing in mobile ad hoc networks (MANETs) which would minimize communication overhead, maximize the overall long-term performance criterion and can perform well under the presence of uncertainty for dynamic topology networks. The contributions in this report is the experimental evidence that, RL techniques equipped with suitable path caching strategies can be employed to reduce the amount of message overhead in QoS routing in MANETs.

A novel partially observable Markov decision process (POMDP) formulation of a message overhead control problem for QoS routing in MANETs is introduced. The proposed scheme integrates the original the Ticket-Based Probing (TBP) scheme with a reinforcement learning method for POMDPs, called the on-policy first-visit Monte Carlo method with path caching (ONMCP) scheme, is applied to support QoS routing at the network level in a MANET. Results obtained from various scenarios of mobility and imprecise information, and stringent QoS requirements show that the POMDP framework can achieve good ticket-issuing policies, in terms of the accumulated reward per episode when compared to the original heuristic TBP scheme and the ONMC scheme without path caching. Furthermore, our approach can lead to more efficient control of search messages, i.e., a reduction of message overhead with marginal difference in the success ratio and average path cost.

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญรูป

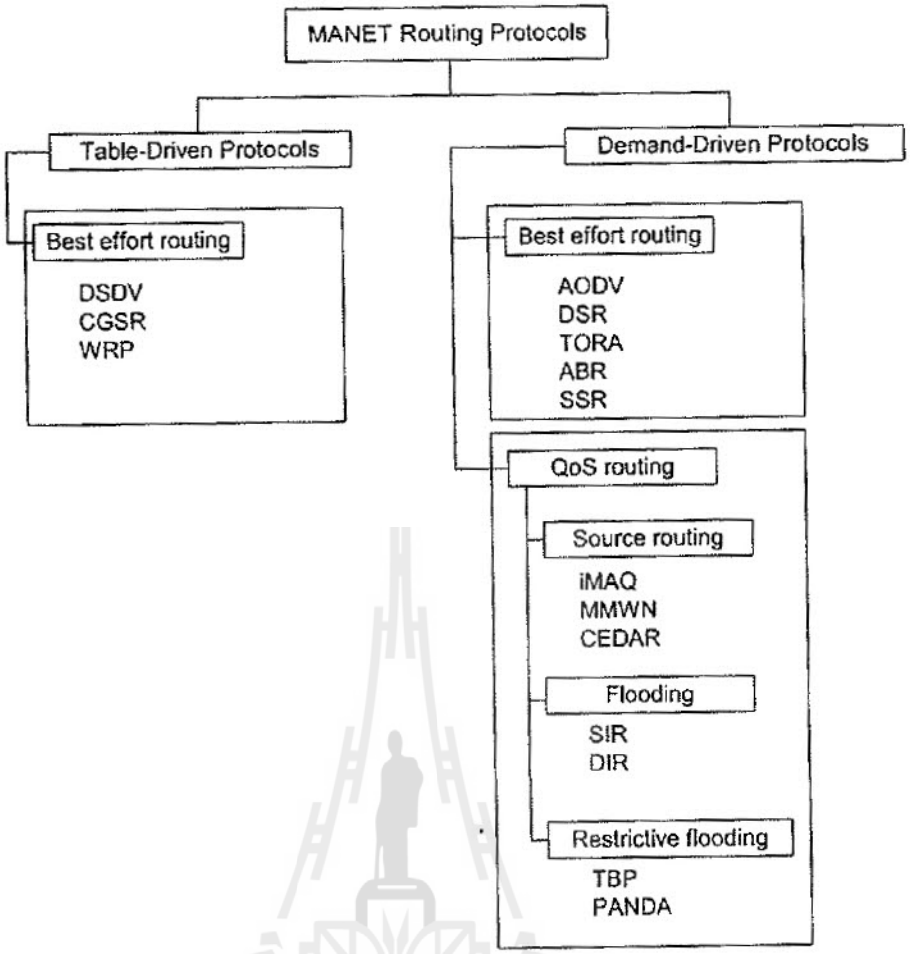
บทที่1
ที่มาและความสำคัญ

บทนี้กล่าวถึงพื้นฐานเครือข่ายเคลื่อนที่แบบแอดฮอค มุ่งเน้นถึงความสำคัญของปัญหาการหาเส้นทางที่รองรับกับคุณภาพการบริการในเครือข่ายเคลื่อนที่แบบแอดฮอค อีกทั้งยังกล่าวถึงแรงจูงใจในการนำกระบวนการเรียนรู้แบบรีอินฟอสเม้นท์มาใช้เพื่อปรับปรุงโปรโตคอลค้นหาเส้นทางที่มีอยู่เดิมให้เกิดประสิทธิภาพดียิ่งขึ้น ซึ่งเป็นเป้าหมายหลักของงานวิจัยนี้

1.1 ความสำคัญของปัญหา

เครือข่ายเคลื่อนที่แบบแอดฮอค (Mobile ad hoc network หรือ MANET) เป็นเครือข่ายเพื่อการติดต่อสื่อสาร เมื่อทุกโหนดภายในเครือข่ายต่างร่วมมือกันเพื่อเชื่อมต่อเส้นทางในเครือข่ายโดยปราศจากควบคุมจากศูนย์กลาง คุณลักษณะทั่วไปของเครือข่ายเคลื่อนที่แบบแอดฮอค คือ ทุกโหนดภายในเครือข่ายสามารถเคลื่อนที่ได้อย่างอิสระภายใต้ขอบเขตของแบนวิธ, ความจุของเส้นทางเชื่อมต่อ และรูปแบบของเครือข่ายที่ไม่สามารถทำนายได้ แต่ละโหนดจะมีระยะการส่งข้อมูลที่จำกัด โหนดต้นทางจะติดต่อสื่อสารกับโหนดปลายทางที่อยู่นอกเหนือรัศมีการส่งด้วยการใช้โหนดข้างเคียง ดังนั้นทุกๆโหนดในเครือข่ายจึงมีความสามารถเช่นเดียวกับตัวค้นหาเส้นทางเคลื่อนที่ที่สามารถส่งต่อข้อมูลและเป็นเสมือนโฮสได้ในเวลาเดียวกัน จากคุณลักษณะของเครือข่ายเคลื่อนที่แบบแอดฮอคจะเห็นได้ว่า เมื่อแต่ละโหนดในเครือข่ายสามารถเคลื่อนที่ได้อย่างเป็นอิสระต่อกันส่งผลให้การค้นหาเส้นทางจากต้นทางไปยังปลายทางนั้นซับซ้อนมากกว่าระบบเครือข่ายไร้สายทั่วไป เนื่องจากข้อมูลเส้นทางการสื่อสารนั้นจะมีการเปลี่ยนแปลงอยู่ตลอดเวลาตามสภาพของเส้นทางและรูปร่างเครือข่าย

ในเครือข่าย MANETs งานวิจัยส่วนใหญ่มุ่งเน้นไปที่การพัฒนาโปรโตคอลเลือกเส้นทาง (Routing protocol) เพื่อค้นหา เลือก และบำรุงรักษาเส้นทางการส่งข้อมูลที่สั้นที่สุด เพื่อให้เกิดการถ่ายโอนข้อมูลที่ดีที่สุด โปรโตคอลเลือกเส้นทางสามารถจำแนกได้เป็น 2 ประเภทคือ โปรโตคอลเลือกเส้นทางด้วยตารางเส้นทาง (Table-driven routing protocol) และโปรโตคอลเลือกเส้นทางตามอุปสงค์ (demand-driven routing protocol) ซึ่งแสดงในรูปที่ 1.1



รูปที่ 1.1 โปรโตคอลค้นหาเส้นทางในเครือข่ายเคลื่อนที่แบบแอดฮอค

โดยโปรโตคอลเลือกเส้นทางที่ขับเคลื่อนด้วยตารางเส้นทางจะจัดเตรียมตารางเส้นทางเพื่อเก็บข้อมูลเกี่ยวกับเส้นทางจากโหนดหนึ่งไปยังทุกๆโหนดในเครือข่ายไว้ล่วงหน้า ทำให้แต่ละโหนดไม่เสียเวลาในการประมวลผลเลือกเส้นทาง อย่างไรก็ตามด้วยการกระทำเช่นนี้ทำให้สิ้นเปลืองพื้นที่จัดเก็บข้อมูลตารางเส้นทาง ตัวอย่างโปรโตคอลประเภทนี้ได้แก่ โปรโตคอลเลือกเส้นทางจากลำดับของระยะทางจากเวกเตอร์ปลายทาง (Destination-sequenced distance-vector protocol หรือ DSDV) [1] โปรโตคอลเลือกเส้นทางด้วยการสับเปลี่ยนหัวหน้ากลุ่มเกตเวย์ (Clusterhead gateway switch routing หรือ CGSR) [2] โปรโตคอลเลือกเส้นทางแบบไร้สาย (Wireless routing protocol หรือ WRP) [3]

สำหรับโปรโตคอลเลือกเส้นทางตามฟังก์ชันอุปสงค์นั้น เส้นทางจะถูกค้นหาและถูกบำรุงรักษาตามการร้องขอ โหนดต้นทางจะเริ่มกระบวนการค้นหาเส้นทางก็ต่อเมื่อต้องการเส้นทางเพื่อส่งข้อมูลไปยังโหนดปลายทางเท่านั้น กระบวนการนี้จึงสามารถหลีกเลี่ยงโอเวอร์เฮด (overhead) ขนาด

ใหญ่ในการเก็บรักษาตารางเส้นทางจากโปรโตคอลเลือกเส้นทางที่ขับเคลื่อนด้วยตารางเส้นทางลงได้ ตัวอย่างโปรโตคอลประเภทนี้ได้แก่ โปรโตคอลเลือกเส้นทางด้วยการใช้เวกเตอร์ระยะทางตามฟังก์ชั่นอุปสงค์ภายในเครือข่ายแอดฮอค (Ad hoc on-demand distance vector routing หรือ AODV) [4] โปรโตคอลเลือกเส้นทางจากต้นทางแบบพลวัติ (Dynamic source routing หรือ DSR) [5] โปรโตคอลเลือกเส้นทางโดยใช้ลำดับเวลา (Temporally ordered routing algorithm หรือ TORA) [6] โปรโตคอลเลือกเส้นทางด้วยการเปลี่ยนหมู่ (Associativity-based routing protocol หรือ ABR) [7] และโปรโตคอลเลือกเส้นทางด้วยการปรับเสถียรภาพของสัญญาณ (Signal stability-based routing protocol หรือ SSR) [8]

จากโปรโตคอลเลือกเส้นทางที่กล่าวมาข้างต้นไม่ได้มีการรองรับคุณภาพการบริการ (QoS) ของเส้นทาง เช่น แบนวิธและเวลาหน่วงตลอดเส้นทาง (end-to-end bandwidth and delay) และเงื่อนไขของเวลาหน่วง อย่างไรก็ตามการรับรอง QoS ของเส้นทางในเครือข่าย MANETs ที่มีรูปร่างเครือข่ายแบบพลวัติเป็นเรื่องยากเนื่องมาจาก ประการแรก QoSของเส้นทางต้องการการจองทรัพยากรตลอดเส้นทางการส่งข้อมูลระหว่างคู่โหนดต้นทางไปยังโหนดปลายทาง การจองทรัพยากรดังกล่าวขึ้นอยู่กับอัลกอริธึมค้นหาเส้นทางซึ่งต้องอาศัยข้อมูลสถานะของพลังงานและรูปร่างเครือข่ายที่แม่นยำ อีกทั้งข้อมูลดังกล่าวยังเป็นข้อมูลที่ไม่สามารถระบุได้อย่างชัดเจนสำหรับเครือข่ายMANETs ดังนั้นการตัดสินใจเลือกเส้นทางด้วยข้อมูลที่ไม่แน่ชัดหรือข้อมูลที่ไม่มีการอัพเดทอาจทำให้ได้เส้นทางที่ไม่เหมาะสม ประการที่สองทุกโหนดในเครือข่ายสามารถเคลื่อนที่ได้ ดังนั้นเส้นทางที่เชื่อมต่ออยู่อาจขาดหายได้ตลอดเวลา ซึ่งอาจทำให้เกิดปัญหาการรักษาเส้นทางเชื่อมต่อตามมา ดังนั้นการประกันคุณภาพของเส้นทางในเครือข่ายประเภทนี้จึงแทบเป็นไปไม่ได้เลยถ้าโหนดในเครือข่ายมีการเคลื่อนที่มากเกินไป ทำให้งานวิจัยส่วนใหญ่มักพิจารณาเครือข่ายMANETsที่มีการเปลี่ยนแปลงรูปร่างเครือข่ายที่ไม่รวดเร็วมากนัก

เพื่อบรรเทาการพิจารณาการเลือกเส้นทางจากข้อมูลที่ไม่แน่ชัด งานวิจัยส่วนใหญ่จึงนำเสนอโปรโตคอลเลือกเส้นทางที่รองรับQos ในเครือข่ายเคลื่อนที่แบบแอดฮอคซึ่งอาศัยการฟลัดดิ้ง (flooding) เพื่อหาเส้นทาง อาทิ อัลกอริธึมเลือกเส้นทางด้วยโหนดต้นทาง (Source-initiated routing algorithm หรือ SIR) [9] อัลกอริธึมเลือกเส้นทางด้วยโหนดปลายทาง (Destination-initiated routing algorithm หรือ DIR) [9] อย่างไรก็ตามการฟลัดดิ้งมีขีดจำกัดเมื่อเครือข่ายมีขนาดใหญ่ขึ้น

ในทางตรงกันข้ามการฟลัดดิ้งแบบวงจำกัด (restrictive flooding) ถูกนำเสนอขึ้นซึ่งเป็นวิธีที่กึ่งกลางระหว่างการฟลัดดิ้ง (แบบไม่จำกัด) และการเลือกเส้นทางโดยโหนดต้นทาง (source

routing) โดยการฟลัดดิ้ง แบบวงจำกัดยังคงมีการหาเส้นทางด้วยวิธีการเดิมอยู่ แต่เมสเสจฟลัดดิ้งจะถูกควบคุมด้วยหน่วยวัดบางประการที่โหนดต้นทางด้วยการใช้ข้อมูลที่ครอบคลุม (global information) เช่น วิธีการตรวจสอบด้วยตั๋ว (Ticket-based probing หรือ TBP) [10] ซึ่งวิธีนี้สามารถควบคุมการฟลัดดิ้งที่ถี่เกินไปได้ด้วยการจำกัดจำนวนตั๋วเชิงตรรกะ (logical ticket) ที่โหนดต้นทาง อย่างไรก็ตามวิธีการนี้ยังมีประเด็นเปิดที่ควรพิจารณาอยู่ นั่นคือ การคำนวณจำนวนตั๋วเชิงตรรกะที่เหมาะสมสำหรับการแจกจ่ายที่โหนดต้นทาง นอกจากนี้วิธีนี้ยังอาศัยกฎฮิวริสติก (heuristic rule) สำหรับการคำนวณตั๋วอีกด้วย งานวิจัย [11] ได้นำวิธี TBP แบบเดิมมาผนวกกับกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ (reinforcement learning หรือ RL) ซึ่งเป็นวิธีที่สามารถเรียนรู้ นโยบายที่ดีสำหรับการจำหน่ายตั๋วด้วยการเลือกการกระทำโต้ตอบกับสิ่งแวดล้อมโดยตรงโดยใช้กระบวนการตัดสินใจอย่างมีเหตุผลในรูปแบบออนไลน์ ซึ่งสามารถหลีกเลี่ยงการคำนวณตั๋วที่ผูกกับกฎฮิวริสติกจากวิธี TBPเดิมได้อีกด้วย อย่างไรก็ตามวิธีการนี้ยังคงมีโอเวอร์เฮดขนาดใหญ่อันเนื่องมาจากความถี่ในการถูกร้องขอให้ค้นหาเส้นทางใหม่ทุกครั้งที่มีการร้องขอเส้นทาง

รายงานวิจัยฉบับนี้ทำการพัฒนาโปรโตคอลค้นหาเส้นทางที่รองรับคุณภาพการบริการในเครือข่ายเคลื่อนที่แบบแอดฮอค โดยนำวิธีค้นหาเส้นทางแบบ TBP ที่มีอยู่เดิมมาผนวกกับกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ภายใต้กระบวนการตัดสินใจแบบมาร์คอฟภายใต้สภาวการณ์ที่สังเกตได้บางส่วน (partially observable Markov decision process หรือ POMDP) ที่เรียกว่า วิธีออนโพลิซีมอนติ คาร์โล (on-policy Monte Carlo หรือ ONMC) [11] และกลยุทธ์พาธ แคชชิ่ง (path caching) เพื่อใช้หานโยบายที่เหมาะสมสำหรับการค้นหาเส้นทางที่รับรองคุณภาพการบริการ (QoS routing) อีกทั้งยังสามารถลดโอเวอร์เฮดในการค้นหาเส้นทาง (routing overhead) สำหรับเครือข่ายเคลื่อนที่แบบแอดฮอคที่มีรูปร่างเครือข่ายแบบพลวัติ

1.2 วัตถุประสงค์

วัตถุประสงค์ของรายงานฉบับนี้คือ

1.2.1 เพื่อพัฒนากระบวนการตัดสินใจสำหรับการหาเส้นทางในเครือข่ายเคลื่อนที่แบบแอดฮอคที่สามารถลดโอเวอร์เฮดในการค้นหาเส้นทาง และเพิ่มประสิทธิภาพของเครือข่ายในระยะยาว

1.2.2 เพื่อประยุกต์ใช้กระบวนการเรียนรู้แบบรีอินฟอร์สเม้นท์กำหนดปัญหาการค้นหาเส้นทางที่รอบรับคุณภาพการบริการสำหรับเครือข่ายเคลื่อนที่แบบแอดฮอคที่มีรูปแบบเครือข่ายแบบพลวัติ

1.2.3 เพื่อประเมินประสิทธิภาพของการผนวกวิธี POMDP RL เข้ากับวิธี TBP ด้วยการเปรียบเทียบผลกับโปรโตคอลหาเส้นทางที่มีอยู่ โดยพิจารณาจากผลตอบแทนสะสมต่อเอพพิโซด และจำนวนเมสเสจค้นหาโดยเฉลี่ย

1.2.4 เพื่อเปรียบเทียบข้อแลกเปลี่ยนในการพบเส้นทางที่รอบรับคุณภาพการบริการสำหรับกระบวนการเรียนรู้แบบรีอินฟอร์สเม้นท์

1.3 ส่วนประกอบของรายงานวิจัย

ส่วนที่เหลือของรายงานวิจัยฉบับนี้ประกอบด้วย บทที่2 กล่าวถึงทฤษฎีพื้นฐานของกระบวนการเรียนรู้แบบรีอินฟอร์สเม้นท์ซึ่งเป็นส่วนจำเป็นที่ทำให้เกิดองค์ความรู้ในรายงานวิจัยฉบับนี้ ประการแรกอธิบายแนวคิดของกระบวนการตัดสินใจแบบมาร์คอฟ (Markov Decision Process หรือ MDP) และแนะนำกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์เพื่อหาผลเฉลยของกระบวนการการตัดสินใจแบบมาร์คอฟที่กำหนดขึ้น โดยใช้กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ที่เรียกว่าวิธี ออนโพลิซี มอนติ คาร์โล (On-policy Monte Carlo หรือ ONMC) ซึ่งเรียนรู้จากประสบการณ์ที่เกิดจากผลของการกระทำที่ส่งผลต่อสิ่งแวดล้อมในแต่ละเอพพิโซด (episode)

บทที่3 กล่าวถึงการศึกษาโปรโตคอลเลือกเส้นทางที่รองรับคุณภาพการบริการสำหรับเครือข่ายเคลื่อนที่แบบแอดฮอค การประยุกต์วิธี TBP ผนวกกับกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์แบบ POMDP ที่เรียกว่า วิธีออนโพลิซี มอนติ คาร์โล (ONMC) ด้วยการใช้กลยุทธ์พาธ แคชชิ่ง ซึ่งเกิดสมดุลแลกเปลี่ยนระหว่างการเพิ่มความสำเร็จในการค้นพบเส้นทางและการใช้เมสเสจค้นหาในปริมาณต่ำ พร้อมทั้งเปรียบเทียบประสิทธิภาพของเส้นทางในแง่ของ ผลตอบแทนสะสมต่อเอพพิโซดและจำนวนเมสเสจค้นหาโดยเฉลี่ย ซึ่งวิธี ONMC ด้วยการใช้กลยุทธ์พาธ แคชชิ่ง จะถูกเปรียบเทียบประสิทธิภาพกับโปรโตคอลเลือกเส้นทางที่ใช้อยู่สำหรับเครือข่ายเคลื่อนที่แบบแอดฮอค

บทที่4 กล่าวถึงการสรุปผล และแนวทางการพัฒนาในอนาคต

บทที่ 2

ปริทัศน์วรรณกรรม งานวิจัยที่เกี่ยวข้อง

2.1 กล่าวนำ

เนื้อหาในบทนี้กล่าวถึงวิธีหาเส้นทางการเชื่อมต่อที่รองรับคุณภาพการบริการสำหรับเส้นทาง (QoS routing) ในเครือข่ายเคลื่อนที่แบบแอดฮอค (mobile ad hoc network หรือ MANET) ที่มีรูปร่างเครือข่ายแบบพลวัติ รายงานวิจัยฉบับนี้นำกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ (reinforcement Learning หรือ RL) ประยุกต์ใช้กับเครือข่ายเคลื่อนที่แบบแอดฮอคที่มีรูปร่างเครือข่ายพลวัติ กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ไม่ต้องการรูปแบบการคำนวณทางคณิตศาสตร์ที่ชัดเจน วิธีนี้มีกระบวนการตัดสินใจที่ดีซึ่งได้จากการเรียนรู้แบบลองผิดลองถูก (trial and error) ซึ่งผู้เรียนจะเรียนรู้ผลได้รับจากการกระทำซึ่งได้รับจากสิ่งแวดล้อม แล้วนำไปปรับปรุงนโยบายจนบรรลุเป้าหมายที่ต้องการ

กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์จะกำหนดปัญหาให้เป็นกระบวนการตัดสินใจแบบมาร์คอฟ (Markov decision process หรือ MDP) ด้วยการระบุปัญหาว่า ระบบที่มีสิ่งแวดล้อมแบบพลวัติจะสามารถเรียนรู้นโยบายในการเลือกการกระทำที่ดีที่สุดเพื่อให้บรรลุเป้าหมายได้อย่างไร ในกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์นั้นจะมองปัญหาในรูปแบบของผู้เรียนสามารถเรียนรู้พฤติกรรมที่ดีจากการลองผิดลองถูกซึ่งเป็นการเลือก การกระทำ และสังเกตผลจากการกระทำที่ส่งผลต่อสิ่งแวดล้อมแบบพลวัติ รายงานวิจัยฉบับนี้จะประยุกต์ใช้กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ที่แบ่งการเรียนรู้ออกเป็นเอพพิโซด ด้วยวิธีการที่เรียกว่า ออนโพลิซี มอนติ คาร์โล (On-policy Monte Carlo หรือ ONMC) [11] โดยฟังก์ชั่นค่าการกระทำจะถูกประมาณค่า และนโยบายจะถูกปรับปรุงหลังจากการเรียนรู้ในแต่ละเอพพิโซด ภายใต้สมมุติฐานที่กล่าวมา วิธีออนโพลิซี มอนติ คาร์โล จะสามารถหาผลเฉลยที่ลู่เข้าสู่

นโยบายที่ดีที่สุดได้และได้ค่าฟังก์ชั่นที่ดีที่สุดจากการเรียนรู้ในแต่ละเอพพิโซดโดยไม่จำเป็นต้องรู้ข้อมูลของสิ่งแวดล้อมแบบพลวัติที่ชัดเจน

ดังนั้นรายงานฉบับนี้จะนำเสนอวิธีONMC ประยุกต์ใช้ในปัญหาการค้นพบเส้นทางในเครือข่ายเคลื่อนที่แบบแอดฮอค หัวข้อถัดไปจะนำเสนอทฤษฎีพื้นฐานเกี่ยวกับกระบวนการตัดสินใจแบบมาร์คอฟ และกล่าวแนะนำกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ในหัวข้อที่ 2.3 ในหัวข้อ2.4 จะกล่าวแนะนำวิธี ONMC และกล่าวสรุปเนื้อหาในบทนี้ในหัวข้อสุดท้าย

2.2 พื้นฐานทฤษฎีการตัดสินใจแบบมาร์คอฟ

2.2.1 คุณสมบัติมาร์คอฟ

คุณสมบัติมาร์คอฟกล่าวว่าทุกสิ่งที่เกิดขึ้นในระยะยาวเป็นผลสืบเนื่องมาจากสถานะปัจจุบัน ดังนั้นความน่าจะเป็นของสถานะถัดไป ณ เวลา $k+1$ สามารถนิยามได้โดยใช้เงื่อนไขอย่างง่ายภายใต้ข้อมูลที่ทราบจากสถานะปัจจุบัน ณ เวลา k ดังนี้

$$\Pr\{s_{k+1} = s' | s_k = s\} = \Pr\{s_{k+1} = s' | s_k = s, s_{k-1} = s, ..., s_0 = s\}. \quad (2.1)$$

โดยรายงานฉบับนี้จะประยุกต์ใช้กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ด้วยการกำหนดปัญหาให้มีคุณสมบัติสอดคล้องกับคุณสมบัติมาร์คอฟ สถานะของผู้เรียนอธิบายถึงข้อมูลของสิ่งแวดล้อมซึ่งเป็นประโยชน์ต่อการตัดสินใจ ถ้าสถานะของผู้เรียนมีคุณสมบัติมาร์คอฟจะทำให้การตอบสนองของสิ่งแวดล้อมที่เวลา $k+1$ ขึ้นอยู่กับผลที่เกิดจากสถานะปัจจุบันที่เวลา k ซึ่งเรียกสถานะเช่นนี้ว่า สถานะมาร์คอฟ (Markov state)

2.2.2 กระบวนการตัดสินใจแบบมาร์คอฟ

กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ที่มีสิ่งแวดล้อมที่สอดคล้องกับคุณสมบัติมาร์คอฟ ถูกเรียกว่า กระบวนการตัดสินใจแบบมาร์คอฟ (Markov decision process หรือ MDP) สมมุติให้เวลาปัจจุบันคือ ช่วงเวลา k ซึ่งมีสถานะจากสิ่งแวดล้อม s ผู้เรียนจะเลือกการกระทำ a ผลที่ได้รับจากการเลือกการกระทำ a ณ สถานะ s คือการตอบสนองจากสิ่งแวดล้อมทำให้ได้สถานะใหม่เป็น s' ดังนั้นความน่าจะเป็นในการเกิดสถานะใหม่ที่เป็นไปได้ คือ

$$P_{ss'}^{a} = \Pr\{s_{k+1} = s' | s_k = s, a_k = a\}. \quad (2.2)$$

สมการนี้ถูกเรียกว่า ความน่าจะเป็นในการส่งสถานะ(transition probabilities) สมมุติให้สถานะและการกระทำ ณ เวลาปัจจุบันคือ s_k และ a_k และสถานะใหม่ที่เกิดขึ้นคือ s_{k+1} ส่งผลให้ได้รับ

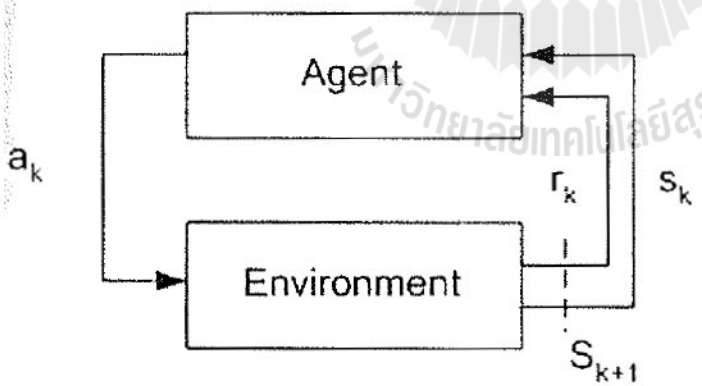
ค่าตอบแทน g_k ซึ่งเป็นค่าที่ได้รับจากการกระทำของผู้เรียน ค่าคาดหวัง (the expected value) ของผลตอบแทนถัดไปคือ

$$G^a_{ss'} = E\{g_k | s_k = s, a_k = a, s_{k+1} = s'\} \quad (2.3)$$

เมื่อผู้เรียนได้รับผลตอบแทนจากสมการนี้แล้ว ผู้เรียนจะสามารถเรียนรู้เพื่อหาการกระทำที่ดีและปรับปรุงกระบวนการตัดสินใจเพื่อให้ได้รับผลตอบแทนสูงสุดยะระยาว

2.3 กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์

กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์เป็นวิธีการคำนวณด้วยการกำหนดให้ระบบที่มีสิ่งแวดล้อมแบบพลวัติสามารถเรียนรู้เพื่อเลือกการกระทำที่ส่งผลให้ผู้เรียนบรรลุเป้าหมายที่วางไว้ [12] ผู้เรียน (The learner) จะมีกลไกการเรียนรู้โดยจะไม่สามารถระบุอย่างชัดเจนว่าการกระทำไหนควรเลือก แต่จะค้นหาการกระทำที่เหมาะสมจากการกระทำที่ให้ผลตอบแทนมากที่สุดซึ่งได้มาจากการลองผิดลองถูกภายใต้ขอบเขตของสิ่งแวดล้อมที่ศึกษา ผู้เรียนจำเป็นต้องใช้ประโยชน์จากประสบการณ์ที่ได้จากการทดลองเลือกการะกระทำและผลที่ได้รับจากการกระทำนั้น การเปลี่ยนแปลงสถานะไปจนถึงผลตอบแทนที่ได้รับเพื่อใช้ในการปรับปรุงการเลือกการกระทำที่ดีที่สุดให้กับตัวเอง นอกจากนี้กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์เป็นการเรียนรู้แบบออนไลน์



รูปที่ 2.1 แผนผังการกระทำโต้ตอบระหว่างผู้เรียนและสิ่งแวดล้อมใน
กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์

รูปที่ 2.1 แสดงให้เห็นถึงแนวทางการเรียนรู้ในกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์เพื่อหาผลเฉลยจากปัญหาที่ซับซ้อนด้วยการเรียนรู้จากการกระทำโต้ตอบระหว่างผู้เรียนและสิ่งแวดล้อม

ซ้ำไปซ้ำมา เมื่อเอเจนท์หมายถึงผู้เรียน (learner) หรือผู้ตัดสินใจ (decision maker) ทุกสิ่งนอกเหนือเอเจนท์ถูกกำหนดให้เป็นสิ่งแวดล้อม โดยทั่วไป *การกระทำ*ใช้นิยามถึงการตัดสินใจของผู้เรียน ในขณะที่สถานะหมายถึงข้อมูลที่ทราบจากสิ่งแวดล้อมซึ่งใช้ประกอบการตัดสินใจของผู้เรียน

เป้าหมายหลักของผู้เรียนคือการหานโยบาย π ซึ่งเป็นการจับคู่ระหว่างสถานะและการกระทำที่ให้ผลตอบแทนระยะยาวสูงสุด รูปแบบมาตรฐานของกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์คือ ผู้เรียนจะพยายามเลือก *การกระทำ* จากชุดของการกระทำที่เป็นไปได้ทั้งหมด ณ สภาพแวดล้อมปัจจุบัน จากนั้น การกระทำที่ถูกเลือก จะส่งผลให้สภาพแวดล้อมมีการเปลี่ยนแปลงและผู้เรียนก็จะได้*ผลตอบแทน* ซึ่งขึ้นอยู่กับว่าการกระทำดังกล่าวส่งผลให้สภาพแวดล้อมเปลี่ยนไปในทิศทางใด กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ประกอบด้วยปัจจัยพื้นฐานสามอย่างคือ สิ่งแวดล้อม, ฟังก์ชั่นรีอินฟอร์สเมนท์ และฟังก์ชั่นมูลค่า (value function)

1) สิ่งแวดล้อม

ในระบบการเรียนรู้แบบรีอินฟอร์สเมนท์จะเรียนรู้การจับคู่จากสถานะไปยังการกระทำด้วยการสุ่มทดลองการกระทำโต้ตอบ (interactions) กับสิ่งแวดล้อมแบบพลวัติสิ่งแวดล้อมเหล่านี้จะถูกเฝ้าสังเกตจากกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ด้วยการสังเกตจากค่าที่อ่านได้จากอุปกรณ์เซนเซอร์, สัญลักษณ์ หรือ จากสถานการณ์ที่ผิดปกติ ถ้ากระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์สามารถสังเกตรายละเอียดต่างๆของสิ่งแวดล้อมที่สนใจได้ดีเยี่ยมจะส่งผลให้กระบวนการนี้สามารถเลือกการกระทำที่เหมาะสมกับสถานะจริงที่เกิดขึ้นได้ แนวคิดนี้จึงเป็นแนวคิดพื้นฐานที่ดีที่สุดของกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ อย่างไรก็ตามกระบวนการนี้ยังต้องอาศัยปัจจัยที่จำเป็นต่อการเรียนรู้เพื่อการจับคู่ที่เหมาะสมตามทฤษฎีดังกล่าว

2) ฟังก์ชั่นรีอินฟอร์สเมนท์

จากที่กล่าวมาข้างต้น ระบบที่มีการเรียนรู้แบบรีอินฟอร์สเมนท์ซึ่งทำการจับคู่จากสถานะไปยังการกระทำด้วยการทดลองสุ่มการกระทำโต้ตอบกับสิ่งแวดล้อม โดยเป้าหมายของกระบวนการเรียนรู้นี้คือ การใช้แนวคิดของฟังก์ชั่นรีอินฟอร์สเมนท์ซึ่งเป็นฟังก์ชั่นจริงของการเสริมกำลังในอนาคต (future reinforcements) ของผู้เรียนเพื่อค้นหาการกระทำที่ให้ผลตอบแทนระยะยาวสูงสุด โดยหลังจากการเลือก *การกระทำ* ที่ *สถานะปัจจุบัน* แล้วผู้เรียนจะได้รับ ผลตอบแทน (reward) ในรูปแบบของปริมาณเชิงตัวเลข ผู้เรียนจะเรียนรู้การกระทำที่ให้ผลตอบแทนระยะยาวสูงสุด

3) ฟังก์ชั่นมูลค่า

ฟังก์ชั่นมูลค่าคือการจับคู่จาก *สถานะ* (state) ไปยัง *มูลค่าของสถานะ* (state values) สมมุติให้นโยบาย π ใช้ในการกำหนด การกระทำที่ควรเลือกในแต่ละสถานะ มูลค่าของสถานะ $V^{\pi}(s)$ ถูกนิยามด้วยผลรวมของค่าคาดหวังที่ผู้เรียนจะได้รับเมื่ออยู่ในสถานะ s

$$V^{\pi}(s) = E_{\pi}\left\{\sum_{n=1}^{\infty} g_{t+n} | s_t = s\right\} \quad (2.4)$$

ดังนั้นนโยบายที่ดีที่สุด V^{*} จะถูกจับคู่จาก สถานะ ไปยัง การกระทำ ที่ให้ผลตอบแทนสูงสุดซึ่งเริ่มต้นจากสถานะแรกและทำการเลือกการกระทำจนกระทั่งสิ้นสุดสถานะสุดท้ายจึงได้ว่า

$$V^{*} = \max_{\pi}\left\{V^{\pi}(s)\right\} \quad (2.5)$$

โดยทั่วไปการเลือกการกระทำในแต่ละช่วงเวลามักถูกคาดหวังให้เป็นการกระทำที่ให้ผลตอบแทนสูงที่สุดในระยะยาว

2.3.1 วิธีมอนติ คาร์โล

วิธีมอนติ คาร์โลเป็นวิธีที่ใช้หาคำตอบของปัญหาในกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์บนพื้นฐานของการเรียนรู้จากมูลค่าเฉลี่ยสะท้อนกลับ วิธีมอนติ คาร์โลต้องการเพียงประสบการณ์ในการเรียนรู้เพื่อหาคำตอบเท่านั้น เช่น ลำดับของสถานะตัวอย่าง, การกระทำ และผลตอบแทนที่ได้รับจากกระบวนการเรียนรู้แบบออนไลน์ กระบวนการเรียนรู้จากประสบการณ์ตรงแบบออนไลน์กำลังเป็นที่สนใจเนื่องจากวิธีนี้ไม่จำเป็นต้องใช้ข้อมูลจากสิ่งแวดล้อมแบบพลวัติแต่ยังสามารถตอบโจทย์ของปัญหาโดยการได้มาซึ่งพฤติกรรมที่ดีที่สุดที่ควรปฏิบัติ รายงานฉบับนี้ประยุกต์วิธีมอนติ คาร์โลกับปัญหาโดยแบ่งการเรียนรู้เป็นเอพพิโซด โดยการสมมุติให้ประสบการณ์ในการเรียนรู้ถูกแบ่งออกเป็นเอพพิโซด แต่ละเอพพิโซดจะจบลงเมื่อมีมีการเปลี่ยนแปลงของมูลค่าที่ประมาณและนโยบายการเลือกพฤติกรรมเท่านั้น ดังนั้นวิธีมอนติ คาร์โลจึงเป็นการเรียนรู้แบบเอพพิโซดต่อเอพพิโซด

พิจารณาวิธีมอนติ คาร์โลสำหรับการเรียนรู้ *ฟังก์ชั่นมูลค่าสถานะ*(the state-value function) สมมุติให้นโยบาย $\pi : S \rightarrow A$ โดยมี *มูลค่าของสถานะ* เป็นค่าคาดคะเนผลตอบแทนย้อนกลับ (the expected return) หรืออาจกล่าวได้อีกนัยว่า เป็นค่าคาดคะเนผลตอบแทนในอนาคตแบบลดทอน (the expected cumulative future discounted reward) ของสถานะนั้น [12] วิธีการดั้งเดิมที่ใช้ประมาณค่าฟังก์ชั่นมูลค่าสถานะคือค่าผลเฉลี่ยย้อนกลับ (สมการที่ 2.4) ซึ่งได้รับหลังการพบสถานะนั้น โดยที่ค่าตอบแทนผลเฉลี่ยนี้ควรลู่เข้าสู่ค่า *ค่าคาดคะเน* (the expected value) ปัญหาการ

ประเมินนโยบายสำหรับเลือกพฤติกรรมคือ การประมาณค่า $Q^{\pi}(s,a)$ ซึ่งเป็น *ค่าคาดคะเนย้อนกลับ* หลังจากการเลือกการกระทำ a ที่สถานะ s แล้วได้นโยบาย π

$$Q^{\pi}(s,a) = E_{\pi}\left\{\sum_{n=1}^{\infty} g_{t+n} \middle| s_t = s, a_t = a\right\}. \tag{2.6}$$

ค่าการพบสถานะแรกของวิธีมอนติคาร์โล (the first-visit Monte Carlo method) ได้จากการเฉลี่ย *ค่าย้อนกลับ* จากการพบสถานะแรกที่เป็นผลจากเกิดจากการการกระทำที่ถูกเลือกดังนี้

$$Q^{\pi}(s,a) = \frac{c(s,a,1)}{1}. \tag{2.7}$$

เมื่อ *c(s, a, 1)* คือ ค่าย้อนกลับ หลังจากพบคู่สถานะ-การกระทำแรก *(s,a)*

ดังนั้นค่าการพบสถานะถัดไปที่เหลือทั้งหมดของวิธีมอนติคาร์โล (the every-visit Monte Carlo method) จึงได้จากการประมาณค่าจากคู่สถานะ-การกระทำนั้น ซึ่งเป็นการเฉลี่ยค่าย้อนกลับจากการพบสถานะที่เป็นผลจากเกิดจากการการกระทำที่ถูกเลือกดังนี้

$$Q^{\pi}(s,a) = \frac{\sum_{k=1}^{n} c(s,a,k)}{n(s,a)}. \tag{2.8}$$

เมื่อ *c(s, a, k)* คือ ค่าย้อนกลับ หลังจากพบคู่สถานะ-การกระทำ *(s, a)* และ *n(s, a)* คือจำนวนครั้งในการพบคู่สถานะ-การกระทำ *(s, a)*

วิธีคำนวณค่าผลตอบแทนทั้งสองวิธีที่กล่าวมาจะทำให้ค่าย้อนกลับลู่เข้าสู่ค่าคาดคะเนจริงได้ถ้าจำนวนครั้งของการพบคู่สถานะ-การกระทำแต่ละคู่เป็นอนันต์กระบวนการนี้ถูกเรียกว่า การประเมินนโยบายภายใต้นโยบายคงที่ π ในแต่ละเอพพิโซด ค่าผลตอบแทน จะถูกสังเกตเพื่อนำไปประเมินและปรับปรุงนโยบายเมื่อทุกสถานะจะต้องถูกพบครบทุกสถานะในการเรียนรู้แต่ละเอพพิโซด การปรับปรุงนโยบายเป็นกระบวนการที่ประกอบด้วย นโยบายใหม่ ซึ่งถูกปรับปรุงมาจาก นโยบายเดิมด้วยการใช้ค่ากรีดี (หรือ ε-greedy) นโยบายกรีดี (the greedy policy) จะเลือกการกระทำที่ดีที่สุดจากการคาดการณ์ค่าประมาณมูลค่าการกระทำปัจจุบัน (the current action-value estimates) สำหรับนโยบายอีกรีดี (the ε-greedy policy) ผู้เรียนจะประพฤติตัวอย่างละโมบด้วยการการกระทำที่ดีที่สุดจากการคาดการณ์ค่าประมาณมูลค่าการกระทำปัจจุบันเป็นส่วนใหญ่ แต่จะมีช่วงเวลาขณะหนึ่งด้วย

ค่าความน่าจะเป็นน้อยๆที่นโยบายอีกรีดีจะเลือกการกระทำจากการสุ่มค่าประมาณมูลค่าการกระทำเหตุที่กระทำเช่นนี้เนื่องมาจากการเลือกการกระทำการการประมาณมูลค่าเพื่อให้ได้การกระทำที่ดีที่สุดอาจยังไม่เพียงพอถ้ายังไม่มีการเข้าพบทุกสถานะที่เป็นไปได้ทั้งหมด ดังนั้นแล้วทุกคู่สถานะ-การกระทำที่ยังไม่เคยถูกพบก็จะไม่มีมูลค่าผลตอบแทน ซึ่งคู่สถานะ-การกระทำที่ไม่เคยถูกสำรวจนี้อาจมีมูลค่าย้อนกลับที่ดีกว่าคู่สถานะ-การกระทำอื่นก็ได้ แต่ด้วยการใช้นโยบายอีกรีดีทำให้คู่สถานะ-การกระทำที่ยังไม่เคยถูกพบได้มีโอกาสถูกสำรวจมากขึ้นดังนั้นผู้เรียนจึงจำเป็นอย่างยิ่งในการประมานมูลค่าของการกระทำที่เป็นไปได้ทั้งหมดในแต่ละสถานะเพื่อที่จะได้ข้อมูลประกอบการตัดสินใจที่ครอบคลุมซึ่งจะทำให้ได้พฤติกรรมที่ถูกต้อง ดังนั้นด้วยการใช้นโยบายอีกรีดีจะช่วยให้ผู้เรียนสามารถสำรวจการกระทำที่เป็นไปได้ทั้งหมดในแต่ละสถานะ

ระหว่างการเรียนรู้การประเมินนโยบายจากหลายๆเอพพิโซดด้วยวิธีการประมาณค่าฟังก์ชั่นมูลค่าการกระทำ(the action-value function)และฟังก์ชั่นมูลค่าคาดคะเน (the expected value function) สมมุติให้ผู้เรียนมีการเฝ้าสังเกตการเปลี่ยนแปลงในเอพพิโซดแบบอนันต์และแต่ละเอพพิโซดจะเริ่มต้นจากการสำรวจ สุดท้ายสมมุติให้ทุกๆคู่สถานะ-การกระทำในแต่ละเอพพิโซดมีความน่าจะเป็นที่ไม่เป็นศูนย์ ภายใต้สมมุติฐานเหล่านี้จะทำให้วิธีมอนติ คาร์โลสามารถคำนวณค่า Q^{π_k} ได้โดยเริ่มจากการสุ่มเลือกนโยบาย π_k หลังจากจบแต่ละเอพพิโซดผู้เรียนจะต้องสังเกตมูลค่าย้อนกลับที่ถูกใช้ไปในการประเมินนโยบาย และนโยบายจะถูกปรับปรุงที่ทุกๆสถานะที่ถูกสำรวจของแต่ละเอพพิโซด การปรับปรุงนโยบายทำได้ด้วยการใช้นโยบายกรีดีคาดประมาณฟังก์ชั่นมูลค่าการกระทำปัจจุบัน ฟังก์ชั่นมูลค่าการกระทำ $Q^{\pi}(s,a)$ ใดๆภายใต้นโยบาย π ซึ่งสัมพันธ์นโยบายกรีดีกล่าวคือ แต่ละสถานะ s ในชุดของสถานะ $(s \in S)$ จะเลือกการกระทำที่คาดการณ์ได้ด้วยฟังก์ชันมูลค่าการกระทำสูงสุด (ซึ่งอ้างถึง มูลค่าQ หรือ Q-value)

$$\pi(s) = \arg\max_{a} \{Q(s,a)\} \tag{2.9}$$

การปรับปรุงนโยบายจะถูกสร้างขึ้นจากนโยบายใหม่ π_{k+1} ที่ได้รับในแต่ละครั้งด้วยนโยบายกรีดีที่พิจารณาจาก Q^{π_k} นโยบายจะถูกปรับปรุงจาก π_k และ π_{k+1} สำหรับทุกชุดของสถานะ $(s \in S)$.

$$\begin{aligned} Q^{\pi_k}\left(s,\pi_{k+1}(s)\right) &= Q^{\pi_k}\left(s,\arg\max_{a}\left\{Q^{\pi_k}(s,a)\right\}\right) \\ &= \max_{a}\left\{Q^{\pi_k}(s,a)\right\} \\ &\geq Q^{\pi_k}\left(s,\pi_k(s)\right). \end{aligned} \tag{2.10}$$

ด้วยความสัมพันธ์ข้างต้นทำให้มั่นใจได้ว่า แต่ละนโยบาย π_{k+1} จะมีค่าที่ดีกว่า π_k หรืออย่างน้อยก็เท่ากัน ในกรณีที่มีนโยบายที่ดีที่สุดสองทาง นอกจากนี้วิธีการดังกล่าวทำให้เชื่อมั่นได้อีกว่าจะสามารถลู่เข้าสู่นโยบายที่ดีที่สุดได้ด้วยมูลค่าฟังก์ชั่นสูงที่สุด

2.4 วิธีออนโพลิซีมอนติ คาร์โล

รายงานฉบับนี้นำเสนอกระบวนการเลือกเส้นทางที่ใช้พลังงานอย่างมีประสิทธิภาพสำหรับเครือข่ายเคลื่อนที่แบบแอดฮอค (MANET) ด้วยการประยุกต์ใช้กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ที่แบ่งการเรียนรู้เป็นเอพพิโซดด้วยวิธีการที่เรียกว่า ออนโพลิซี มอนติ คาร์โล (the on-policy Monte Carlo หรือ ONMC) วิธีการนี้เน้นการเรียนรู้แบบเป็นเอพพิโซดสำหรับประเมินว่าสถานะหรือการกระทำใดที่เหมาะสมในการดำเนินงานระยะยาว ฟังก์ชั่นดังกล่าวเรียกว่า ฟังก์ชั่นมูลค่าการกระทำ ซึ่งเป็นฟังก์ชั่นคู่สถานะ-การกระทำที่ใช้กำหนดปริมาณเฉลี่ยของผลตอบแทนที่ผู้เรียนใช้คาดคะเนเพื่อให้ได้ผลตอบแทนสูงสุดในระยะยาวจากผลตอบแทนเฉลี่ยย้อนกลับที่ได้รับจากคู่สถานะ-การกระทำ

วิธีออนโพลิซีจะพยายามประเมินหรือปรับปรุงนโยบายที่เกิดขึ้นในปัจจุบันเพื่อใช้ในการตัดสินใจเลือกการกระทำ วิธีทั่วไปพยายามทำให้มั่นใจว่าทุกการกระทำได้ถูกเลือกอย่างต่อเนื่องจนเป็นอนันต์แล้ว กำหนดให้ S เป็นชุดของสถานะที่เป็นไปได้ทั้งหมด และ A เป็นชุดของการกระทำที่เป็นไปได้ทั้งหมด สมมุติให้การกระทำที่ถูกเลือกในเอพพิโซด t ถูกควบคุมโดยนโยบาย π_t เมื่อ $\pi_t : S \to A$ กำหนดให้ฟังก์ชั่นมูลค่าการกระทำที่ (s,a) โดย $Q^{\pi_t}(s,a)$ คือผลตอบแทนที่ถูกคาดหวังว่าจะได้รับจาก (s,a) และนโยบาย π_t จะถูกปฏิบัติตามอีกครั้งในภายหลัง กำหนดให้นโยบายเริ่มต้นเป็น π_0 และ $Q^{\pi_t}(s,a)$ เริ่มต้นที่จุดเริ่มต้นในแต่ละเอพพิโซด สำหรับเอพพิโซด t ใดๆ จะเลือกการกระทำที่เป็นไปได้จากสถานะนั้นตามนโยบาย π_t เมื่อเอพพิโซด t จบลงค่า $Q^{\pi_t}(s,a)$ จะถูกอัพเดทตามสมการดังนี้

$$Q^{\pi_t}(s,a) = Q^{\pi_{t-1}}(s,a) + \frac{1}{t}\left[\sum_{n=\tau_t(s,a)}^{N_t} g(s_n,a_n) - Q^{\pi_{t-1}}(s_n,a_n)\right]. \qquad (2.11)$$

เมื่อ N_t คือช่วงเวลาหรือจำนวนครั้งของขั้นเวลา (time step) ในเอพพิโซด t และ $\tau_t(s,a)$ เมื่อ $0 \le \tau_t(s,a) \le N_t$ คือ ขั้นเวลาเมื่อเกิดการสำรวจคู่สถานะ-การกระทำ (s,a) ในเอพพิโซด t และ $g(s,a)$ ผลตอบแทนที่ได้รับจากการเลือก การกระทำ a ที่สถานะ s โดยที่เทอมของผลรวมคือค่าผลตอบแทนสะสมที่เกิดจากการสำรวจคู่สถานะ-การกระทำ (s,a) แรก

นโยบายใหม่สำหรับเอพพิโซดถัดไป π_{t+1} ถูกปรับปรุงมาจากนโยบายเดิม π_t ซึ่งใช้นโยบาย อีกรีดี (the ε-greedy policy) ในกระบวนการปรับปรุงดังนี้

$$\pi_{t+1}(s)=\begin{cases} a^* & \text{with probability } 1-\varepsilon+\dfrac{\varepsilon}{|\mathrm{A}|} \\ a\in A-\{a^*\} & \text{with probability } \dfrac{\varepsilon}{|\mathrm{A}|}, \end{cases} \tag{2.12}$$

เมื่อ a^* คือ นโยบายกรีดีที่ถูกพบโดย $a^*=\arg\max_{a\in A}\{Q^{\pi_t}(s,a)\}$, $\varepsilon\in[0,1]$ และ $|A|$ คือขนาดของชุดการกระทำ ภายใต้เงื่อนไขดังกล่าวจะทำให้นโยบาย ε-greedy ซึ่งได้จาก Q^{π} ถูกการันตีว่าเหมาะสมกว่าหรือเทียบเท่า π

2.5 สรุป

เนื้อหาบทนี้กล่าวถึงแนวคิดของกระบวนการตัดสินใจแบบมาร์คอฟ การแนะนำแนวคิดของกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์เพื่อหาผลเฉลยของปัญหาที่มีการตัดสินใจแบบมาร์คอฟ กรอบงานของกระบวนการตัดสินใจแบบมาร์คอฟถูกนำมาใช้ในการกำหนดปัญหาการเลือกเส้นทางในเครือข่ายเคลื่อนที่แบบแอดฮอค สำหรับปัญหาการเลือกเส้นทางในรายงานเล่มนี้ได้แบ่งลักษณะงานออกเป็นเอพพิโซดเมื่อเอพพิโซดหนึ่งๆเริ่มต้นจากการที่โหนดต้นทางค้นหาเส้นทางไปยังโหนดปลายทาง เอพพิโซดสิ้นสุดเมื่ออย่างน้อยมีหนึ่งเส้นทางที่ถูกพบ ด้วยเหตุนี้กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ที่แบ่งการเรียนรู้ออกเป็นเอพพิโซดด้วยวิธีการที่เรียกว่า วิธีออนโพลิซี มอนติ คาร์โลจึงถูกอธิบายในบทนี้ สำหรับบทถัดไปจะเสนอถึงการกำหนดปัญหาการเลือกเส้นทางที่รองรับคุณภาพการบริการในเครือข่ายเคลื่อนที่แบบแอดฮอค ด้วยวิธีออนโพลิซี มอนติ คาร์ พร้อมทั้งเปรียบเทียบประสิทธิภาพของวิธีออนโพลิซี มอนติ คาร์กับวิธีเลือกเส้นทางที่มีอยู่เดิมเช่น [10], [11]

บทที่ 3

กระบวนการเรียนรู้แบบรีอินฟอสเมนท์สำหรับการค้นพบเส้นทางในเครือข่ายเคลื่อนที่แบบแอดฮอคด้วยกลยุทธ์พาธ แคชชิ่ง

3.1 กล่าวนำ

เนื้อหาบทนี้กล่าวถึงวิธีการเลือกเส้นทางที่รองรับคุณภาพการบริการในเครือข่ายเคลื่อนที่แบบแอดฮอค โดยมีวัตถุประสงค์เพื่อให้เกิดข้อตกลงสำหรับการเกิดสมดุลแลกเปลี่ยนระหว่างการเพิ่มความสำเร็จในการค้นหาเส้นทางและการใช้เมสเสจค้นหาในปริมาณต่ำ โดยการประยุกต์ใช้กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ด้วยวิธีออนโพลิซี มอนติ คาร์โล (ONMC) ด้วยกลยุทธ์พาธ แคชชิ่ง ซึ่งวิธีการดังกล่าวเหมาะสำหรับการตัดสินใจที่มีการแบ่งเรียนรู้เป็นเอพพิโซด

เนื้อหาสำคัญที่จะกล่าวถึงในบทนี้ ได้แก่

1. กล่าวแนะนำการจัดการคุณภาพการบริการสำหรับเส้นทางในเครือข่ายเคลื่อนที่แบบแอดฮอค ด้วยกระบวนการตัดสินใจแบบมาร์คอฟภายใต้สภาวการณ์ที่สังเกตได้บางส่วน
2. การหาเส้นทางด้วยวิธี TBP และกลยุทธ์พาธ แคชชิ่ง
3. การเปรียบเทียบประสิทธิภาพการเลือกเส้นทางด้วยวิธีออนโพลิซี มอนติ คาร์โล ด้วยกลยุทธ์พาธ แคชชิ่ง (ONMCP) เทียบกับวิธีการเลือกเส้นทางที่มีอยู่เดิมอีก 2 วิธี ได้แก่ วิธี TBP และวิธี ONMC

3.2 คุณภาพการบริการสำหรับเส้นทางในเครือข่ายเคลื่อนที่แบบแอดฮอค

องค์ประกอบสำคัญของการจัดการคุณภาพการบริการของเส้นทาง (QoS routing) ในเครือข่ายเคลื่อนที่แบบแอดฮอค (mobile ad hoc network หรือ MANET) คือ ข้อมูลเกี่ยวกับแหล่งพลังงานที่เหลืออยู่ในโหนดเคลื่อนที่ ข้อมูลดังกล่าวขึ้นอยู่กับการอัพเดทข้อมูลระหว่างโหนดเคลื่อนที่

ดังนั้นการแลกเปลี่ยนข้อมูลระหว่างโหนดเคลื่อนที่จึงเป็นสิ่งจำเป็นอย่างยิ่ง การแลกเปลี่ยนข้อมูลนั้นจะกระทำเป็นช่วงเวลา หรือ เมื่อพบว่ารูปร่างเครือข่ายมีการเปลี่ยนแปลง อย่างไรก็ตามยังมีข้อมูลที่คลุมเครือสืบเนื่องมาจากเมสเสจบอกการอัพเดทถูกส่งมาช้าหรือมีการสูญหาย ซึ่งอาจทำให้ข้อมูลถูกยับยั้ง

ข้อมูลของเครือข่ายที่ถูกต้องนั้นยากต่อการสังเกต เนื่องมาจากโหนดเคลื่อนที่แต่ละตัวจะถูกพบจากการเฝ้าสังเกต (observation) จากสิ่งแวดล้อมของตนเองเท่านั้น ซึ่งอาจทำให้ข้อมูลที่รวบรวมมาไม่สมบูรณ์และอาจเกิดความผิดพลาดได้ ข้อมูลที่ได้จากการเฝ้าสังเกตเครือข่ายของโหนดเคลื่อนที่แต่ละตัวจะถูกนำไปใช้ในการตัดสินใจในเรื่องต่างๆ เช่น ต้องใช้เมสเสจควบคุมจำนวนเท่าไรเพื่อใช้ในการค้นหาเส้นทางการส่งข้อมูลที่เป็นไปได้ เป็นต้น

ภายใต้การพิจารณาสมมติฐานเกี่ยวกับการเคลื่อนที่และข้อมูลด้านแหล่งจ่ายพลังงาน มีความเป็นไปได้ที่จะจำลองโมเดลการส่งสถานะ (state transitions) ให้เป็นกระบวนการมาร์คอฟ [11] และเนื่องจาก สถานะ (state) ที่ถูกต้องของเครือข่ายไม่สามารถระบุได้ ดังนั้นจึงมีการใช้แบบจำลองการตัดสินใจที่เรียกว่า กระบวนการตัดสินใจแบบมาร์คอฟภายใต้สภาวการณ์ที่สังเกตได้บางส่วน (partially observable Markov decision process หรือ POMDP) ภายในเครือข่ายเคลื่อนที่แบบแอดฮอค โดยมีวัตถุประสงค์เพื่อค้นหานโยบายที่ให้ประสิทธิภาพสูงสุดภายใต้ขอบเขตจำกัด โดยที่ปัญหาขอบเขตจำกัดจะถูกพิจารณาในที่นี้เนื่องมาจากการแลกเปลี่ยนข้อมูลระหว่างโหนดเคลื่อนที่จะกระทำเป็นแบบเอพพิโซด (episode) โดยเอพพิโซดจะเริ่มต้นหลังจากมีการแลกเปลี่ยนเมสเสจเกิดขึ้นและจะสิ้นสุดเมื่อมีการแลกเปลี่ยนเมสเสจถัดมา

3.3 วิธีการค้นหาเส้นทางแบบ TBP และพาธ แคชชิ่ง

วิธีออนโพลิซี เฟิร์ส-วิสิท มอนติ คาร์โล (on-policy first-visit Monte Carlo หรือ ONMC) [7] ถูกนำมาใช้เพื่อหานโยบายภายใต้การเฝ้าสังเกตสิ่งแวดล้อมในกระบวนการตัดสินใจแบบมาร์คอฟภายใต้สภาวการณ์ที่สังเกตได้บางส่วน (partially observable Markov decision process หรือ POMDP)

วิธี ONMC จากกระบวนการ POMDP ถูกรวมเข้าไปในวิธีการค้นหาเส้นทางที่เรียกว่า Ticket-Based Probing (TBP) โดยวิธี TBP เป็นอัลกอริธึมค้นหาเส้นทางจากหลายเส้นทางแบบกระจาย (multipath distributed routing algorithm) สำหรับระบบที่มีเวลาหน่วงตลอดเส้นทาง (end-to-end delay) หรือการร้องขอแบนวิธด์สำหรับข้อมูลสถานะที่มีความคลุมเครือระดับสูง [10] จุดประสงค์ของ

อัลกอริธึมนี้คือการเลือกเส้นทางการส่งข้อมูลที่เหมาะสมด้วยความน่าจะเป็นในการส่งสำเร็จสูงสุดสำหรับเครือข่ายที่มีรูปร่างเครือข่ายแบบพลวัติด้วยข้อมูลที่ไม่แม่นยำ แนวคิดพื้นฐานของอัลกอริธึมนี้คือ เมื่อโหนดต้นทาง s ต้องการเส้นทางการส่งที่เหมาะสมกับเวลาหน่วง (หรือ แบนวิดธ์) ที่ถูกร้องขอต่อโหนดปลายทาง d จำนวนของโพรบ (เมสเสจค้นหา) ถูกส่งจากโหนด s ไปยังโหนด d จำนวนโพรบทั้งหมดที่ใช้สำหรับค้นหาเส้นทางถูกควบคุมโดย จำนวนตั๋วเชิงตรรกะ (logical tickets) หรือ M_0 โดยพารามิเตอร์ M_0 จะถูกคำนวณที่โหนดต้นทาง s เมื่อโหนดข้างเคียง j ได้รับโพรบจากโหนด s โหนด j จะทำสำเนาโพรบนั้นไว้และทำการคำนวณจำนวนของตั๋วเพื่อดำเนินการทำสำเนาโพรบอีกครั้ง การคำนวณตั๋วที่โหนด j จะกระทำภายใต้ข้อมูลตลอดเส้นทางที่หามาได้ (นั่นคือ จากโหนด j ไปยังโหนด d) และจะไม่เกินกว่าจำนวนของตั๋วในโพรบที่โหนด j ได้รับมา แต่ละโพรบจะถือตั๋วไว้อย่างน้อยหนึ่งใบ และตั๋วทั้งหมดที่อยู่ในเครือข่ายจะถูกควบคุมโดยความแปรปรวนของพารามิเตอร์ M_0

3.3.1 การคำนวณตั๋วตั้งต้น: ภาพรวมของวิธี TBPดั้งเดิม

งานวิจัยนี้ศึกษาปัญหาการเลือกเส้นทางที่มีมูลค่าเวลาหน่วงน้อยที่สุด (delay-constrained least-cost routing) พิจารณาการร้องขอการเชื่อมต่อของโหนดต้นทาง โหนดปลายทาง และความต้องการของค่าเวลาหน่วงเฉลี่ยตลอดเส้นทาง (mean end-to-end delay) คือ s, d และ D_{req} ตามลำดับ กำหนดให้ D_{ij} คือค่าเวลาหน่วงเฉลี่ยของการเชื่อมต่อ (mean link delay) ระหว่างโหนด i และโหนด j โดยที่เวลาหน่วงเฉลี่ยตลอดเส้นทางของเส้นทางที่มีเวลาหน่วงน้อยที่สุดคือ r^* และ $D_n(d) = \sum_{(i,j)\in r^*} D_{ij}$ โดยพารามิเตอร์ $\Delta D_n(d)$ คือค่าความแปรปรวนของเวลาหน่วงเฉลี่ยตลอดเส้นทาง หาได้จาก

$$\Delta D_n^{new}(d) = \rho \Delta D_n^{old}(d) + (1-\rho)\beta \left| D_n^{new}(d) - D_n^{old}(d) \right| \qquad (3.1)$$

พารามิเตอร์ ρ คือตัวแปรเพิกเฉย ซึ่งใช้กำหนดความเร็วในการเพิกเฉยค่า $\Delta D_n^{old}(d)$ และพารามิเตอร์ $1-\rho$ ใช้กำหนดความรวดเร็วของการทำให้ค่า $\Delta D_n^{new}(d)$ ลู่เข้าสู่ $|D_n^{new}(d) - D_n^{old}(d)|$ และ β คือพารามิเตอร์สำหรับรองรับค่า $\Delta D_n^{new}(d)$ ในงานวิจัย [1] จำนวนของตั๋ว (M_0) หาได้จาก $M_0 = Y_0 + G_0$ เมื่อ Y_0 และ G_0 คือตั๋วสีเหลืองและตั๋วสีเขียวตามลำดับ โดยตั๋วสีเหลือง

สำหรับโอกาสสูงสุดในการค้นพบเส้นทางที่เหมาะสม ในขณะที่ตั๋วสีเขียวหมายถึงโอกาสสูงสุดในการพบเส้นทางที่มีมูลค่าต่ำ

พารามิเตอร์ Y_0 ถูกกำหนดจากกฎฮิวริสติก (heuristic rules) [10]

$$Y_0 = \begin{cases} 1 & , D_{reg} > D_{hi} \\ \dfrac{D_s(d) + \Delta D_s(d) - D_{req}}{2 \times \Delta D_s(d)} \times \theta_Y & , D_{lo} \le D_{req} \le D_{hi} \\ 0 & , D_{req} < D_{lo} \end{cases} \quad (3.2)$$

เมื่อ $D_{hi} = D_s(d) + \Delta D_s(d)$, $D_{lo} = D_s(d) - \Delta D_s(d)$ และ θ_Y คือพารามิเตอร์สำหรับกำหนดจำนวนสูงสุดของตั๋วสีเหลือง

สำหรับพารามิเตอร์ G_0 จะแตกต่างจากกฎฮิวริสติกเล็กน้อย

$$G_0 = \begin{cases} 1 & , D_{reg} > \Theta D_{hi} \\ \dfrac{\Theta(D_s(d) + \Delta D_s(d)) - D_{req}}{\Theta(D_s(d) + \Delta D_s(t)) - D_s(d)} \times \theta_G & , D_s(d) \le D_{req} \le \Theta D_{hi} \\ \dfrac{D_{req} - D_s(d) + \Delta D_s(d)}{\Delta D_s(d)} \times \theta_G & , D_{lo} \le D_{req} \le D_s(d) \\ 0 & , D_{req} < D_{lo} \end{cases} \quad (3.3)$$

เมื่อ θ_G คือพารามิเตอร์สำหรับกำหนดจำนวนสูงสุดของตั๋วสีเขียว ค่า $\Theta > 1$ใช้กำหนดระดับตั้งต้น (threshold) ที่เกินกว่า D_{req} ซึ่งใช้สำหรับค้นหาเส้นทางที่มีเวลาหน่วงสูงมาก

3.3.2 การคำนวณตั๋วตั้งต้น: วิธี TBP ภายใต้กระบวนการ ONMC

กระบวนการ ONMC สำหรับ POMDPs ถูกนำมาใช้ทั้งในระบบจริงหรือระบบจำลอง เพื่อหานโยบายการแจกจ่ายตั๋วที่ดีในแง่ของสมดุลแลกเปลี่ยนระหว่างจำนวนของตั๋วที่ถูกแจกจ่ายไปกับความน่าจะเป็นในการค้นพบเส้นทางที่เหมาะสม นอกจากนี้วิธีนี้ยังมีการแทนที่การคำนวณค่า M_0 จากกฎฮิวริสติกในสมการที่ (3.2) และ (3.3) ด้วยการเลือก M_0 จากเซตจำกัดในลำดับของกระบวนการตัดสินใจแบบมาร์คอฟ

พิจารณาเซตของโหนดเคลื่อนที่ N ในเครือข่ายเคลื่อนที่แบบแอดฮอค แต่ละโหนดจะเก็บข้อมูลของเวลาหน่วงตลอดเส้นทางของทุกโหนดปลายทางไว้ สำหรับคู่ของโหนดต้นทาง-ปลายทาง (s,d) จะมีเซตของการเฝ้าสังเกตดังนี้

$$O_{sd} = \{[q_D(m), q_{\Delta D}(l)] : 1 \le m \le n,\ 1 \le l \le n_\Delta\}$$

เมื่อ n (n_Δ) คือช่วงของเวลาหน่วงตลอดเส้นทางแบบเต็มหน่วย และ $q_D(m)$ $(q_{\Delta D}(l))$ คือช่วงลำดับที่ m (l) บนช่วงจำนวน $[0,\infty)$

พิจารณา $o_k \in O_{sd}$ ที่เวลา k โหนด s จะเลือกการกระทำ $a_k \in A = \{0,...,M_{\max}\}$ เมื่อ $M_{\max}$ คือจำนวนตั๋วสูงสุดที่ยอมรับได้ โดยที่ตั๋วสีเขียวจะไม่ถูกพิจารณา $(G_0 = 0)$ และจะมีเพียงตั๋วสีเหลืองเท่านั้นที่ถูกพิจารณา จึงทำให้ $M_{\max} = \theta_\gamma$ เพื่อเป็นการเน้นความสำคัญไปที่การหาเส้นทางที่เหมาะสมมากกว่าการหาเส้นทางที่มีมูลค่าต่ำ ถ้ามีการค้นพบเส้นทางที่เหมาะสมอย่างน้อย 1 เส้นทาง จะได้รับผลตอบแทนเป็น $g(o_k, a_k)$ ถ้าไม่เช่นนั้นแล้ว การกระทำดังกล่าวจะถูกลงโทษ ซึ่งผลตอบแทนนิยามโดย

$$g(\cdot, a_k) = \begin{cases} \varsigma_j - \log a_k & , a_k > 0, X = \aleph \\ -(\varsigma_j - \log a_k) & , a_k > 0, X = 0 \\ -\log a_k & , a_k > 0, X > \aleph \\ 0 & , a_k = 0 \end{cases} \qquad (3.4)$$

เมื่อ $\varsigma_j \in R^+$ คือผลตอบแทนที่ได้รับสำหรับการบริการโหนด j X คือจำนวนเส้นทางการส่งที่ถูกค้นพบ และ $\aleph$ คือ จำนวนเส้นทางที่ต้องการสูงสุดจากเส้นทางที่ค้นพบ

ถ้าเส้นทางที่เหมาะสมที่ถูกพบมีหลายเส้นทาง โหนดปลายทาง d จะเลือกเส้นทางที่มีมูลค่าน้อยที่สุด จากนั้นจะส่งเมสเสจตอบกลับ (acknowledge message) ซึ่งประกอบด้วยค่าใหม่ของค่าเวลาหน่วงเฉลี่ยตลอดเส้นทาง ไปยังโหนดต้นทาง s ด้วยเส้นทางที่ถูกเลือกไว้ หลังจากได้รับเมสเสจตอบกลับแล้ว โหนด s จะอัพเดทข้อมูลเครือข่ายของตน นั่นคือการอัพเดท $D_s^{new}(d)$ และ $\Delta D_s^{new}(d)$ ที่คำนวณได้จากสมการ (3.1) กระบวนการดังกล่าวจะถูกกระทำซ้ำทุกเส้นทางการเชื่อมต่อที่ถูกร้องขอที่โหนดต้นทาง s จนกระทั่งเกิดการแลกเปลี่ยนของเวกเตอร์ระยะทางที่โหนด s ดังนั้นวัตถุประสงค์ของการใช้ OPMC เพื่อใช้กำหนดนโยบายที่ใกล้เคียงนโยบายที่ดีที่สุด $\pi : O_{sd} \to A$ ภายใต้กระบวนการเฝ้าสังเกต

3.3.3 พาธ แคชชิ่ง

วิธีพาธ แคชชิ่ง (path caching) ภายใต้กระบวนการ ONMC [11] ถูกนำมาใช้ในการหาเส้นทางสำหรับทุกการเชื่อมต่อที่ถูกร้องขอ เพื่อหลีกเลี่ยงการร้องขอเส้นทางที่ถี่เกินไป แต่ละโหนดจึงมีการเก็บรักษาเส้นทางไว้ [13], [14] ดังนั้นกลยุทธ์พาธ แคชชิ่งจึงช่วยลดโอเวอร์เฮด (overhead) ในเครือข่าย MANETs ได้ ดังนั้นในงานวิจัยฉบับนี้จึงมีการใช้กลยุทธ์พาธ แคชชิ่ง ซึ่งถูกสนับสนุนด้วยวิธี TBP โดยพาธ แคช (path cache) เป็นเซตของเส้นทางซ้ำซ้อนที่ถูกค้นพบด้วยวิธี TBP ขนาดของพาธ แคชขึ้นอยู่กับระดับความซ้ำซ้อนของเส้นทางที่ต้องการ

3.4 การทดสอบและวิเคราะห์ผล

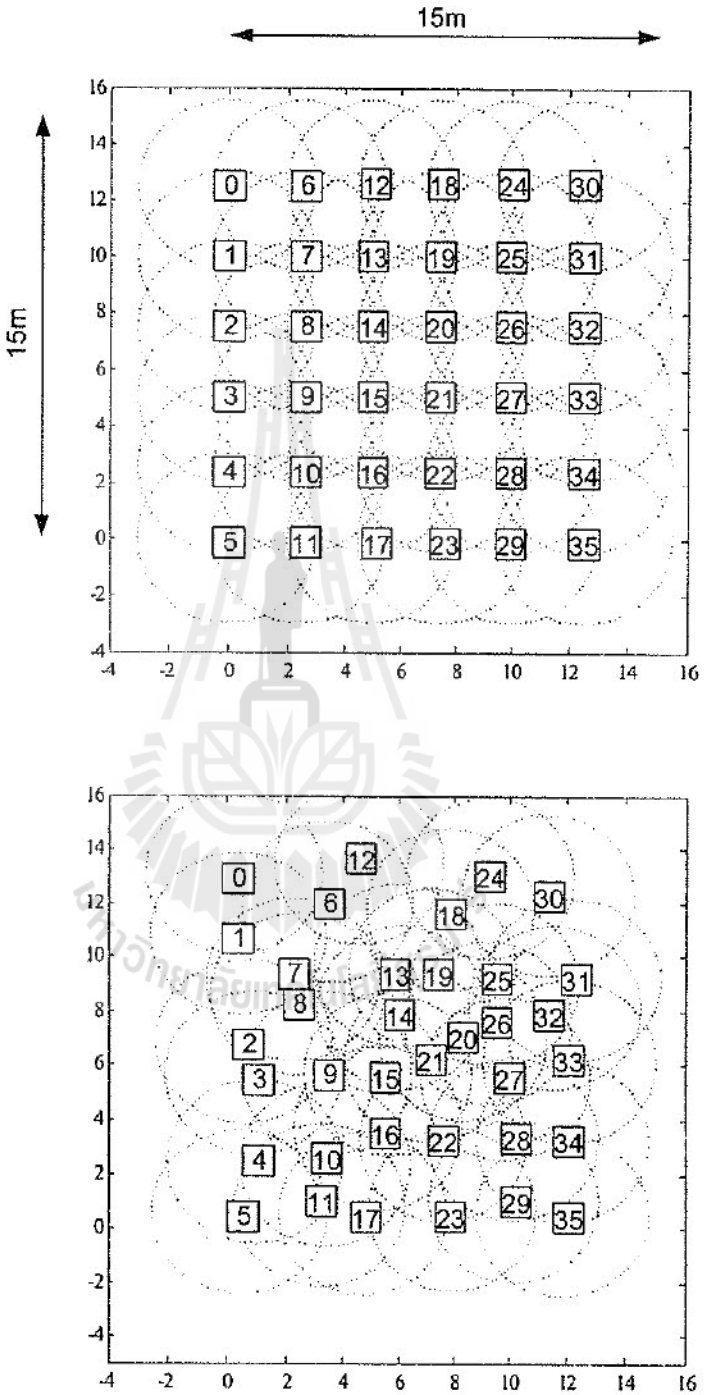
ประสิทธิภาพสำหรับการประยุกต์วิธี TBP ภายใต้กระบวนการ ONMC ในเครือข่ายเคลื่อนที่แบบแอดฮอคจะถูกประเมินด้วยแบบจำลองระบบในคอมพิวเตอร์ งานวิจัยฉบับนี้ทำการทดสอบประสิทธิภาพของระบบในหลายด้านด้วยกัน กล่าวคือ

1) ผลตอบแทนสะสม ซึ่งเป็นค่าที่ได้จากผลตอบแทนสะสมจากทุกเอพพิโซดหารด้วยจำนวนเอพพิโซดทั้งหมด
2) อัตราความสำเร็จ ซึ่งเป็นค่าที่ได้จากจำนวนการเชื่อมต่อที่ตอบรับทั้งหมดหารด้วยจำนวนการร้องขอการเชื่อมต่อทั้งหมด
3) มูลค่าเส้นทางโดยเฉลี่ย ซึ่งเป็นค่าที่ได้จากมูลค่าทั้งหมดจากการสร้างเส้นทางเชื่อมต่อหารด้วยจำนวนการเชื่อมต่อที่ถูกสร้างขึ้น
4) จำนวนเมสเสจค้นหาโดยเฉลี่ย ซึ่งเป็นค่าที่ได้จากจำนวนของเมสเสจค้นหาทั้งหมดที่ถูกส่งไปหารด้วยจำนวนการร้องขอการเชื่อมต่อทั้งหมด

พิจารณาโหนดเคลื่อนที่ 36 โหนดในเครือข่ายเคลื่อนที่แบบแอดฮอคบนพื้นที่15x15 เมตร^2 รูปร่างเครือข่ายจะถูกสุ่มอย่างสม่ำเสมอด้วยแบบจำลองการเคลื่อนที่ ความเร็วของโหนดเคลื่อนที่มีค่าอยู่ระหว่าง 0.3-0.7 เมตร/วินาที แต่ละโหนดมีรัศมีการส่ง 3 เมตร ลิงค์เชื่อมต่อ (link) จะถูกสร้างขึ้นระหว่าง 2 โหนดเคลื่อนที่ใดๆที่อยู่ภายใต้รัศมีการส่งที่กำหนด

การร้องขอการเชื่อมต่อจะถูกสร้างขึ้นที่โหนดต้นทางด้วยอัตราการเชื่อมต่อ 0.2 วินาที มูลค่าของแต่ละลิงค์จะมีค่าสม่ำเสมออยู่ในช่วง [0,1] แต่ละลิงค์เชื่อมต่อระหว่างโหนด i และโหนด j จะมีเวลาหน่วง 2 แบบเกิดขึ้น คือ ดีเวลาหน่วงเฉลี่ยที่เกิดขึ้นจริง (D_{ij}) และเวลาหน่วงเฉลี่ยสำหรับการ

ประกาศ ($\tilde{D}_{ij}$) เวลาหน่วงในแบบหลังจะถูกประกาศไปที่เครือข่ายและนำมาใช้เพื่อคำนวณค่าเวลาหน่วงเฉลี่ยตลอดเส้นทาง $D_j(d)$ สำหรับทุกโหนด j และ d ในเครือข่ายเคลื่อนที่แบบแอดฮอค เวลาหน่วงเฉลี่ยที่เกิดขึ้นจริงในแต่ละลิงค์จะถูกสร้างขึ้นอย่างสม่ำเสมอในช่วง [0,50] มิลลิวินาที

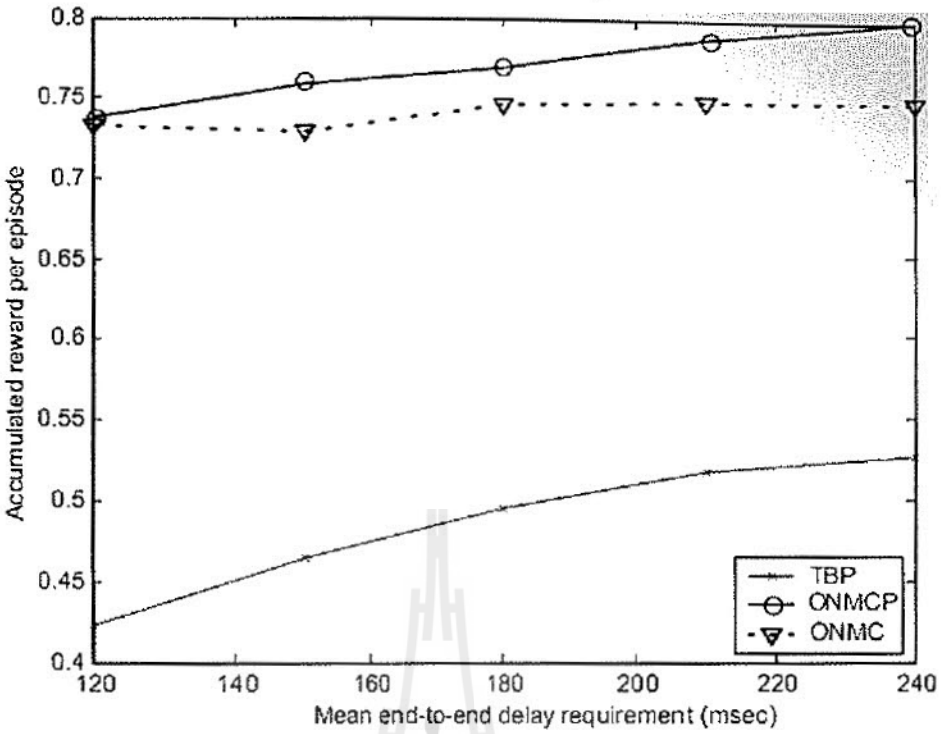


รูปที่ 3.1 แบบจำลองเครือข่ายโหนดเคลื่อนที่ 36 โหนดในเครือข่ายเคลื่อนที่แบบแอดฮอคบนพื้นที่ 15x15 เมตร^2 แต่ละโหนดมีรัศมีการส่ง 3 เมตรแสดงโดยวงกลมเส้นประ รูปบนแสดงความสัมพันธ์เริ่มต้นของโหนดเคลื่อนที่ รูปล่างแสดงถึงความสัมพันธ์โหนดเคลื่อนที่ที่เปลี่ยนไปหลังจากเริ่มการทดสอบ

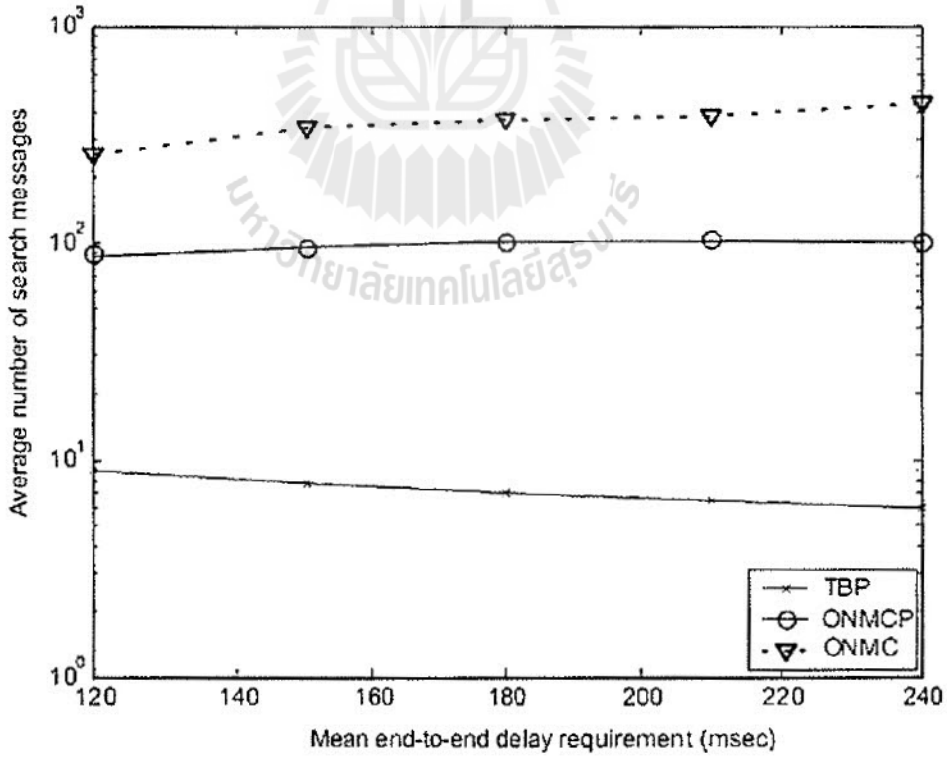
กระบวนการเลือกเส้นทาง 3 วิธีถูกนำมาทดสอบประสิทธิภาพ คือ วิธีTBP วิธี TBP ภายใต้กระบวนการ ONMC และวิธี TBP ภายใต้กระบวนการ ONMC ด้วยพาธ แคชชิง (ONMCP) ทั้ง 3 กระบวนการนี้จะพิจารณาเพียงตั๋วสีเหลืองโดยมี $M_{\max} = \theta_Y = 100$ ใบ เพื่อเน้นความสำคัญในการหาเส้นทางที่เป็นไปได้มากว่าการหาเส้นทางที่มีมูลค่าต่ำ นอกจากนี้การร้องขอการเชื่อมต่อจะถูกปฏิเสธทันทีถ้าเวลาหน่วงเฉลี่ยตลอดเส้นทางที่ร้องขอมีค่ามากกว่าเวลาหน่วงตลอดเส้นทางสูงสุดที่เป็นไปได้

กำหนดให้เซตของการกระทำ (เมื่อการร้องขอการเชื่อมต่อไม่ถูกปฏิเสธ) สำหรับทุกอัลกอริธึมกำหนดโดย $M_0 \in A = \{1,10,20,...,100\}$ เวลาหน่วงเฉลี่ยตลอดเส้นทางและการเปลี่ยนแปลงของเวลาหน่วง (ในหน่วย มิลลิวินาที) $q_D(m) \in \{[0,10), [10,20), ... ,[250,\infty)\}$ และ $q_{\Delta D}(l) \in \{[0,10),[10,\infty)\}$ เมื่อ m = 1,...,26 l =1,2 และ $q_D(m)$ คือช่วงการแบ่งที่ m ของเวลาหน่วงเฉลี่ยตลอดเส้นทางระหว่างโหนด s และโหนด d และ $q_{\Delta D}(l)$ คือช่วงการแบ่งที่ l ของการเปลี่ยนแปลงของเวลาหน่วงเฉลี่ยตลอดเส้นทางระหว่างสองโหนดเคลื่อนที่ วิธี ONMC และ ONMCP จะถูกฝึกทั้งหมด 4×10^6 ครั้งของการร้องขอการเชื่อมต่อ หลังจากนั้นจึงจะนำมาทดสอบประสิทธิภาพและเปรียบเทียบกับวิธี TBP โดยทุกอัลกอริธึมจะถูกทดสอบด้วยการรันทั้งหมด 1×10^6 ครั้งของการร้องขอการเชื่อมต่อ

รูปที่ 3.2 แสดงให้เห็นว่าผลตอบแทนเฉลี่ยสะสมต่อเอพพิโซดจะเพิ่มขึ้นเมื่อความต้องการของเวลาหน่วงเฉลี่ยตลอดเส้นทางเพิ่มขึ้น ที่เป็นเช่นนี้เพราะ เมื่อความต้องการของเวลาหน่วงเฉลี่ยตลอดเส้นทางเพิ่มขึ้น ทำให้ง่ายขึ้น ดังนั้นผลตอบแทนเฉลี่ยสะสมจึงมีค่าเพิ่มขึ้น นอกจากนี้เมื่อความต้องการของเวลาหน่วงเฉลี่ยตลอดเส้นทางเพิ่มขึ้น จะเห็นว่าวิธี ONMCP มีประสิทธิภาพดีกว่าวิธี ONMC เนื่องจากมีตั๋วจำนวนน้อยมากถูกแจกจ่ายไปเมื่อพาท แคชชิ่งถูกนำมาใช้ รูปที่ 3.3 ชี้ให้เห็นว่าวิธี TBP ให้จำนวนเมสเสจค้นหาเฉลี่ยน้อยที่สุดแม้แต่บริเวณที่มีค่าผลตอบแทนสะสมต่ำสุดต่อเอพพิโซด วิธี ONMCP มีจำนวนเมสเสจค้นหาเฉลี่ยน้อยกว่าวิธี ONMC เนื่องมาจากการใช้งานพาท แคชชิ่ง

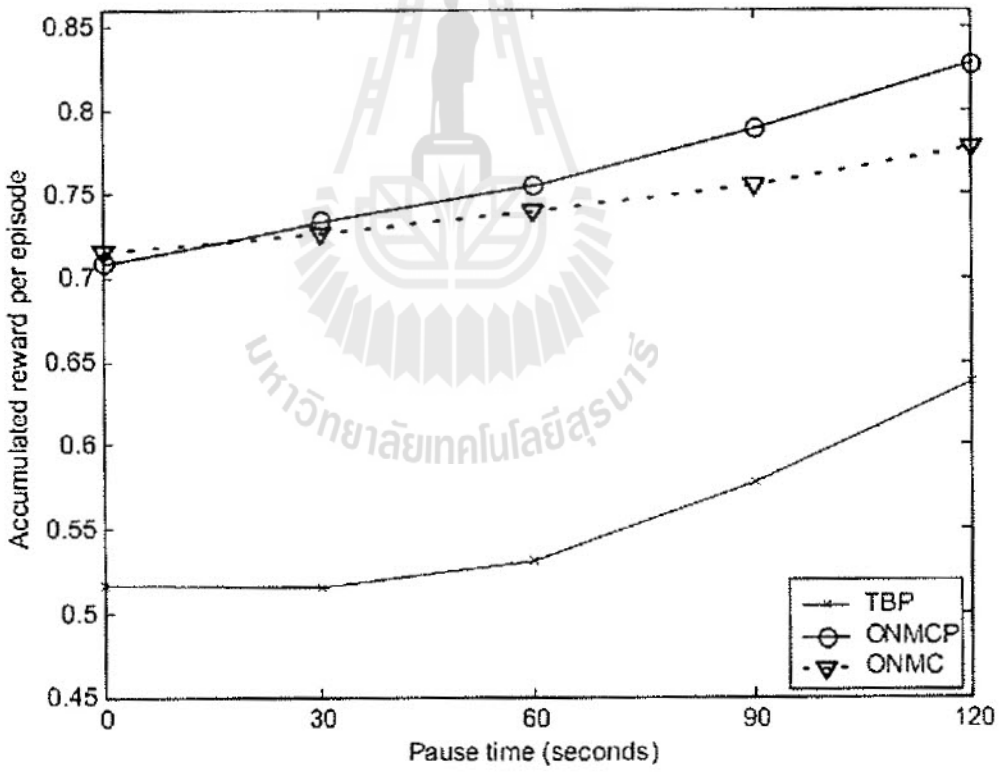


รูปที่ 3.2 ผลตอบแทนเฉลี่ยสะสมต่อเอพพิโซดที่อัตราความคลุมเครือ 0.5

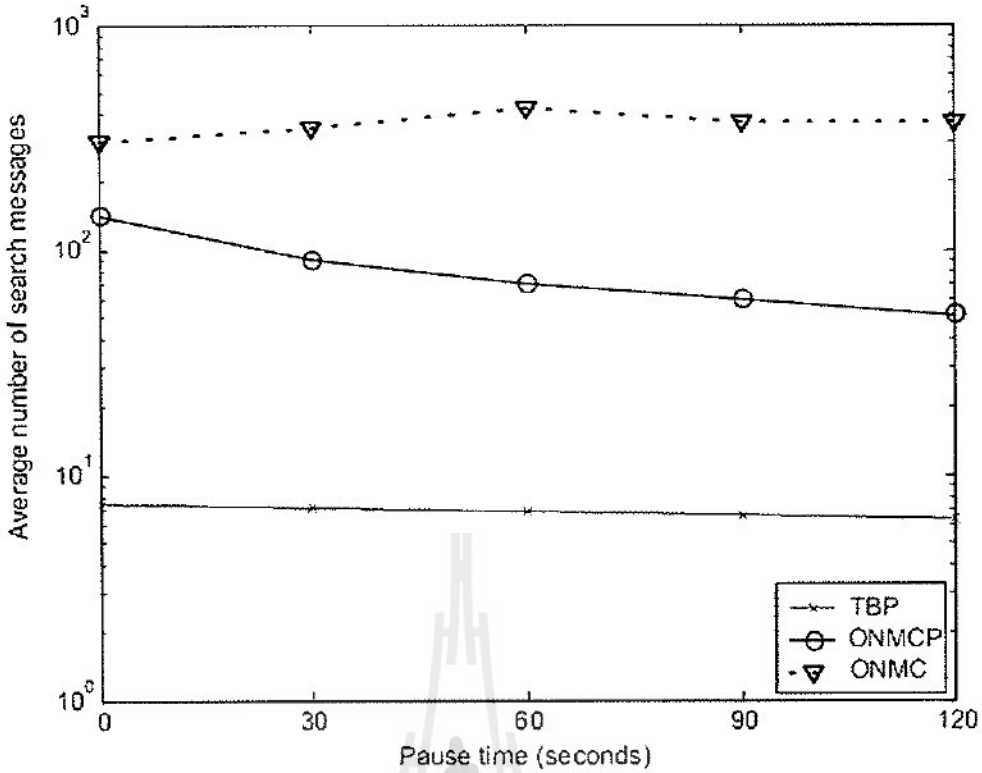


รูปที่ 3.3 จำนวนของเมสเสจค้นหาเฉลี่ยที่อัตราความคลุมเครือ 0.5

การทดลองสุดท้ายแสดงถึงผลกระทบจากการเคลื่อนที่ของทุกอัลกอริธึม (โดยการเพิ่มเวลาเพื่อให้โหนดอยู่ในสภาวะคงที่ซึ่งเรียกว่า เวลาหยุดพักชั่วขณะ หรือ pause time) ภายใต้ความต้องการของเวลาหน่วงเฉลี่ยตลอดเส้นทางคงที่ รูปที่ 3.4 แสดงให้เห็นว่าวิธี ONMCP และวิธี ONMC มีความแตกต่างของผลตอบแทนเฉลี่ยสะสมต่อเอพพิโซดเพียงเล็กน้อยเท่านั้น ส่วนวิธี TBP ให้ผลตอบแทนเฉลี่ยสะสมต่อเอพพิโซดน้อยที่สุด ทั้งนี้เนื่องมาจากเมื่อโหนดอยู่ในสภาวะคงที่นานขึ้นทำให้ง่ายต่อการค้นหาเส้นทาง ดังนั้นผลตอบแทนเฉลี่ยสะสมต่อเอพพิโซดของทุกอัลกอริธึมจึงเพิ่มขึ้นนั่นเอง รูปที่ 3.5 ชี้ให้เห็นว่าวิธี ONMCP และ ONMC มีจำนวนเมสเสจค้นหาเฉลี่ยลดลงเนื่องจากเส้นทางการส่งที่เป็นไปได้ถูกค้นพบง่ายขึ้นอันเนื่องมาจากโหนดอยู่ในสภาวะคงที่นานขึ้น ดังนั้นทั้งสองวิธีจึงเรียนรู้การแจกจ่ายตัวให้น้อยลงเพื่อให้มีการใช้จำนวนของเมสเสจค้นหาน้อยที่สุด อย่างไรก็ตามจะเห็นว่าวิธี ONMCP มีการสร้างจำนวนเมสเสจค้นหาเฉลี่ยน้อยที่สุดเนื่องจากการใช้พาท แคชซึ่งสามารถหลีกเลี่ยงความถี่ในการร้องขอการค้นหาเส้นทางได้



รูปที่ 3.4 ผลตอบแทนเฉลี่ยสะสมต่อเอพพิโซดที่ช่วงเวลาหยุดพักชั่วขณะที่แตกต่างกัน



รูปที่ 3.5 จำนวนของเมสเสจค้นหาเฉลี่ยที่ช่วงเวลาหยุดพักชั่วขณะที่แตกต่างกัน

3.5 สรุป

งานวิจัยฉบับนี้นำเสนอวิธี TBP ภายใต้กระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์และพาธ แคชซิ่ง ซึ่งเรียกกระบวนการนี้ว่า กระบวนการ ONMCP โดยนำไปประยุกต์ใช้ในการค้นหาเส้นทางที่มีการรับรองคุณภาพการบริการสำหรับเครือข่ายเคลื่อนที่แบบแอดฮอค ผลการจำลองระบบแสดงให้เห็นว่ากระบวนการที่ได้นำเสนอสามารถได้มาซึ่งนโยบายในการแจกจ่ายตั๋วที่ดีในด้านของผลตอบแทนสะสมต่อเอพพิโซด เมื่อเทียบกับวิธี TBP ดั้งเดิมและวิธี ONMC นอกจากนี้วิธี ONMCP ยังสามารถลดโอเวอร์เฮดจากการค้นหาเส้นทางอันเนื่องมาจากการใช้กลยุทธ์พาธ แคชซิ่งอีกด้วย

บทที่ 4
บทสรุป

4.1 บทสรุป

ในรายงานวิจัยฉบับนี้นำเสนอกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์ภายใต้การเฝ้าสังเกตสิ่งแวดล้อมในกระบวนการตัดสินใจแบบมาร์คอฟภายใต้สภาวการณ์ที่สังเกตได้บางส่วน (partially observable Markov decision process หรือ POMDP) ด้วยวิธีการที่เรียกว่า ออนโพลิซี มอนติ คาร์โล (On-policy Monte Carlo หรือ ONMC) เพื่อปรับปรุงวิธีหาเส้นทางที่รองรับคุณภาพในการบริการสำหรับเครือข่ายเคลื่อนที่แบบแอดฮอคที่มีอยู่เดิมนั่นคือ วิธีตรวจสอบด้วยตั๋ว (Ticket-based probing หรือ TBP) โดยมีวัตถุประสงค์เพื่อหาสมดุลแลกเปลี่ยนระหว่างการใช้จำนวนเมสเสจค้นหาและความสำเร็จในการค้นพบเส้นทาง โดยองค์ความรู้ที่ได้รับในรายงานวิจัยนี้ได้แก่

4.1.1 การกำหนดปัญหา

กำหนดปัญหาการค้นหาเส้นทางให้เป็นกระบวนการตัดสินใจแบบมาร์คอฟภายใต้สภาวการณ์ที่สังเกตได้บางส่วน (partially observable Markov Decision Process หรือ MDP) เพื่อหาสมดุลแลกเปลี่ยนระหว่างการใช้จำนวนเมสเสจค้นหาและความสำเร็จในการค้นพบเส้นทางที่รองรับคุณภาพการบริการ (QoS routing) ในเครือข่ายเคลื่อนที่แบบแอดฮอค

4.1.2 การค้นหาเส้นทางที่รองรับคุณภาพการบริการในเครือข่ายเคลื่อนที่แบบแอดฮอคด้วยกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์

ในบทที่ 3 นั้นได้ประยุกต์ใช้กระบวนการ POMDP RL เพื่อค้นหาเส้นทางที่รองรับคุณภาพการบริการ (QoS routing) ในเครือข่ายเคลื่อนที่แบบแอดฮอคด้วยการใช้วิธีการ ONMC ผนวกเข้ากับวิธี TBP และกลยุทธ์พาธ แคชชิ่งเพื่อเรียนรู้นโยบายที่ดีในการแจกจ่ายตั๋วที่โหนดต้นทาง และสามารถลดจำนวนโอเวอร์เฮดในการค้นหาเส้นทาง

ผลการทดลองที่ได้จากการจำลองระบบบนคอมพิวเตอร์แสดงให้เห็นว่าวิธี TBP ภายใต้กระบวนการ ONMC ด้วยกลยุทธ์พาธ แคชชิ่ง (ONMCP) สามารถเรียนรู้นโยบายการจำหน่ายตัวที่ดี ในแง่ของการให้ผลตอบแทนสะสมต่อเอพพิโซดสูงสุดเมื่อเปรียบเทียบกับวิธี TBP ดั้งเดิมและวิธี TBP ภายใต้กระบวนการ ONMC ในขณะที่มีการใช้เมสเสจค้นหาน้อยกว่า

วิธี TBP ภายใต้กระบวนการ ONMC มีความต้องการในการคำนวณเพื่อให้ได้การตัดสินใจในหนึ่งครั้งต้องใช้การโต้ตอบ $O(|S||A|)$ ครั้ง เมื่อ $|S|$ และ $|A|$ แทนขนาดของเซตของสถานะและการทำงานที่เป็นไปได้ทั้งหมดตามลำดับ โดยเมื่อขนาดของเครือข่ายใหญ่ขึ้นก็ไม่ได้ส่งผลต่อความต้องการในการคำนวณนี้ อย่างไรก็ตามความต้องการด้วยหน่วยความจำจะแปรผันตรงกับจำนวนโหนดที่อยู่ในเครือข่าย ในทางปฏิบัติ การใช้โครงสร้างลำดับชั้นถูกนำมาใช้แก้ไขปัญหาความต้องการด้านหน่วยความจำที่เพิ่มขึ้นเพื่อนำไปปฏิบัติจริงกับเครือข่ายเคลื่อนที่แบบแอดฮอคขนาดใหญ่ได้

4.2 งานวิจัยในอนาคต

4.2.1 การรักษาเส้นทาง

รายงานวิจัยฉบับนี้กล่าวครอบคลุมเนื้อหาของการค้นหาเส้นทางเมื่อมีการร้องขอการเชื่อมต่อเท่านั้น อย่างไรก็ตามกระบวนการเรียนรู้แบบรีอินฟอร์สเมนท์สามารถขยายองค์ความรู้เพื่อการพัฒนาเป็นอัลกอริธึมรักษาเส้นทางเพื่อคงเส้นทางสื่อสารไว้ ในเครือข่าย MANETs เมื่อมีความเสียหายของเส้นทาง เส้นทางใหม่จะถูกสร้างขึ้นมาอีกครั้ง [5] หรือ มีการซ่อมบำรุงเส้นทางเก่า [7], [9] โดยที่การสร้างเส้นทางใหม่จะไปเพิ่มโอเวอร์เฮดในการสร้างเส้นทางได้ แต่ก็ยังมีการใช้ต้นทุนต่ำกว่าการซ่อมบำรุงเส้นทาง ดังนั้นกระบวนการตัดสินใจแบบ MDP สามารถนำไปใช้แก้ปัญหาการรักษาเส้นทางเพื่อหาสมดุลแลกเปลี่ยนระหว่างประสิทธิภาพของเส้นทางและโอเวอร์เฮดที่เพิ่มขึ้น

4.2.2 การพิจารณาพลังงานจากแบตเตอร์รี่

เครือข่าย MANETs ที่ถูกพิจารณาในรายงานวิจัยฉบับนี้สมมุติว่าแหล่งพลังงานจากแบตเตอร์รี่ของแต่ละโหนดเคลื่อนที่มีระดับพลังงานคงที่ตลอดการทดลอง ดังนั้นการเชื่อมต่อระหว่างโหนดจะไม่เกิดขึ้นเมื่อโหนดอยู่ห่างกันมากเกินไปเท่านั้น ดังนั้นประเด็นที่น่าสนใจคือการขยายกรอบงานไปสู่การพิจารณาให้รัศมีการส่งข้อมูลของโหนดแปรผันตามระดับพลังงานจากแบตเตอร์รี่

4.2.3 การประสานงานข้ามชั้น

รายงานวิจัยฉบับนี้มีการพิจารณาระดับชั้นโปรโตคอลในเครือข่ายเคลื่อนที่แบบแอดฮอค ในชั้นกายภาพ (physical layer) การเข้าใช้ช่องสัญญาณในชั้นเชื่อมต่อข้อมูล (media-access control layer) และชั้นเครือข่าย (network layer) ซึ่งถูกพิจารณาแบบแยกส่วนกัน อย่างไรก็ตามการปรับปรุงประสิทธิภาพของเครือข่ายและและการลดการใช้พลังงานในเครือข่ายจำเป็นต้องทำการพิจารณาร่วมกันทั้งสามชั้น ดังนั้นสิ่งที่น่าสนใจคือการขยายกรอบงานไปสู่การพิจารณาการประสานงานข้ามชั้นกัน

บรรณานุกรม

[1] Perkins, C., Bhangwat, P. 1994. Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers. Proceedings of ACM SIGCOMM'94, Vol.24, No.4, pp.234-244.

[2] Chiang, C.C. 1997. Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel. Proceedings of IEEE SICON'97, pp.197-211.

[3] Murphy, S., Garcia-Luna-Aceves, J.J. 1996. An Efficient Routing Protocol for Wireless Networks. Proceedings of ACM PETRA'09, pp.183-197.

[4] Perkins, C.E., Royer, E.M. 1999. Ad Hoc On-Demand Distance Vector Routing. Proceedings of IEEE WMCSA'99, pp.90-100.

[5] Johnson, D.B., Maltz, D.A. 1996. Dynamic Source Routing in Ad Hoc Wireless Networks. Mobile Computing, Vol.353, pp153-181.

[6] Park, V.D., Corson, M.S. 1997. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. Proceedings of IEEE INFOCOM'97, pp.1405-1413.

[7] Toh, C.K. 1997. Associativity-based Routing for Ad Hoc Mobile Networks. ACM Journal on Wireless Personal Communication, Vol.4, No.2, pp.103-139.

[8] Dube, R., Wang, K., Rais, C.D., Tripathi, S.K. 1997. Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks. IEEE Personal Communication, Vol.4, No.1, pp.36-45.

[9] Chen, S. 1999. Routing Support for Providing Guarantedd End-to-End Quality-of-Service. PhD Thesis, University of Illinois at Urbana-Champaign, IL.

[10] Chen, S., Nahrstedt, K. 1999. Distributed Quality-of-Service Routing in Ad Hoc Networks. IEEE Journal on Selected Areas in Communications, Vol.17, No.8, pp.1488-1505.

บรรณานุกรม (ต่อ)

[11] Usaha, W. 2004. Resource Allocation in Networks with Dynamic Topology. PhD Thesis, University of London, London, U.K.

[12] Chang, J.H., Tassiulas, L. 2004. Maximum Lifetime Routing in Wireless Sensor Networks. ACM Transactions, Vol.12, No.4, pp.609-619.

[13] Hu, Y., Johnson, D.B. 2000. Caching Strategies in On-demand Routing Protocols for Wireless Ad Hoc Networks. Proceedings of ACM MobiCom'00, pp.231-242.

[14] Papadimitratos, P., Haas, Z.J., Sirer, E.G. 2002. Path Set Selections in Mobile Ad Hoc Networks. Proceedings of ACM MobiHOC'02, pp1-11.

ภาคผนวก

A Reinforcement Learning Approach for Path Discovery in MANETs with Path Caching Strategy

Wipawee Usaha
School of Telecommunication Engineering
Suranaree University of Technology, Nakhon Ratchasima, Thailand 30000
Email: wipawee@ccs.sut.ac.th

Abstract— **In this paper, we enhance an existing path discovery scheme called the Ticket-Based Probing (TBP) which supports QoS routing in mobile ad hoc networks (MANETs) to increase its accumulated reward. The scenario of QoS routing in MANETs with the presence of network information uncertainty is considered and modelled as a partially observable Markov decision process (POMDP). The proposed scheme integrates the original TBP scheme with a reinforcement learning method for POMDPs, called the on-policy first-visit Monte Carlo (ONMC) method, and a suitable path caching strategy. Simulation results shows that the inclusion of patch caching with the ONMC method can indeed achieve message overhead reduction with marginal difference in the path search ability and additional computational and storage requirements.**

I. Introduction

Routing in a mobile ad hoc network (MANET) is a challenging task due to node mobility. Difficulties arise even further in the development of routing schemes which support QoS connections. One key to support QoS routing is feasible route search [1], [2], [5]. Feasible route search can be done by distributed routing whereby other nodes apart from the source node are involved in the feasible path(s) search by identifying their neighboring nodes as the next hop router. It can also be performed by source routing where a feasible path(s) is computed solely at the source node.

Alternatively, certain methods like the Ticket-Based Probing (TBP) scheme [1] combine the features of distributed and source routing. More specifically, flooding is still invoked but the amount of flooding is controlled by issuing a limited number of logical tickets at the source node. Although the TBP scheme enjoys several advantages such as high tolerance to imprecise state information, some challenging issues still remain—one of which relates to the restricted flooding method: the computation of a suitable number of logical tickets issued at the source node. More specifically, the original TBP scheme relies on an heuristic rule of ticket computation. In [7], the original TBP scheme is enhanced by integrating it with a reinforcement learning (RL) technique. Results in [7] show that the RL-based TBP scheme is able to learn a "good" rule for issuing tickets by interacting directly with the environment or by simulation—at the expense of reasonable storage and computational requirements of on-line decision parameters.

In this paper, we study the effect of the inclusion of path caching to the RL-based TBP scheme. Our motivation is that by maintaining a path cache at each mobile node, we can avoid frequently invoking the path discovery scheme and therefore reduce the amount of routing overhead in the MANET. The contribution in this paper is the experimental evidence that, RL techniques equipped with suitable path caching strategies can be employed to reduce the amount of message overhead in QoS routing in MANETs [7].

The paper is organized as follows. In the next section, we present an introduction to the partially observable Markov decision process (POMDP) model which the QoS routing problem in MANETs is based on. Section III describes the TBP path discovery schemes and path caching to support QoS routing in MANETs. In this section, the original TBP scheme and the enhanced TBP scheme are presented. The following section shows the numerical study results and Section V provides the conclusion.

II. QoS routing in MANET as a partially observable Markov decision process

A vital component for QoS routing in MANETs is the residual resource information in the network. Such information depends on up-to-date information between mobile nodes. Message exchanges between mobile nodes are therefore required. These information exchanges are done periodically or when a topology change is detected. But even so, imprecise information can still arise due to delayed-arrival or lost update messages and restricted transmission of updating messages.

Because accurate network information is difficult to obtain, each mobile node is faced with only an "observation" of its environment which is most likely incomplete and inaccurate. With only the current network observation at hand, each mobile node must make certain decisions, e.g., how many control messages are needed to find a feasible path for some new connection arrival, when and how to perform path maintenance if an existing path is about to break, etc.

Under certain assumptions regarding the movement and resource information, it is possible to (approximately) model the state transitions as a Markov process [7]. Furthermore, since the accurate state of the network is hidden from each mobile node, we can (approximately) model the decision-making problem in MANETs as a partially observable Markov decision process (POMDP). The goal is to find a policy which optimizes some performance criterion in finite horizon. The finite horizon problem is considered here due to the *episodic*

nature of message exchanges between the mobile nodes—an episode starts immediately after a message exchange and terminates at the subsequent message exchange.

III. TBP PATH DISCOVERY SCHEME AND PATH CACHING

The on-policy first-visit Monte Carlo (ONMC) method for POMDPs [7] is employed here to find an observation-based policy in *partially observable* MDPs. The method is extended from *completely observable* Markov decision processes (MDPs) in [6].

The ONMC method for POMDPs is integrated into a path discovery scheme called the *Ticket-Based Probing* (TBP) scheme. The TBP scheme is a multipath distributed routing algorithm for supporting end-to-end delay or bandwidth requirements proposed to tolerate high degrees of imprecise state information [1]. The design objective of this algorithm is to maximize the probability of success in finding a feasible route in dynamic networks in the presence of inaccurate information. The basic idea of the algorithm is outlined as follows. When a source node s needs to find a route that satisfies a delay (or bandwidth) requirement to a destination node d, a number of probes (search messages) are sent from s towards d. The total number of probes used in the path discovery is controlled by the initial number of *logical* tickets, M_0. The parameter M_0 is computed at the source node s depending on the contention level of network resources and the inaccuracy of available information. When a neighboring node j receives a probe from node s, it makes copies of that probe and recomputes the number of tickets to be carried on the copied probes. The computation of the tickets at node j is based on the available end-to-end information (i.e., from node j to d) and cannot exceed the number of tickets in the probe that node j has received. The end-to-end information, which is obtained through probing on an on-demand basis, is used to guide the distribution of the tickets and the probes along the directions of *most probable* feasible paths towards the destination d. Each probe carries at least one ticket. Since no additional tickets are issued along the intermediate nodes and each probe searches one path, the number of paths found are also bounded by the number of tickets M_0 issued at the source node. Consequently, the amount of probes that enter the network is simply controlled by varying M_0.

A. Initial ticket calculation: Overview of the original TBP scheme

In this paper, we study a *delay-constrained least-cost routing* problem. Consider a connection request whose source, destination nodes and mean end-to-end delay requirement are s, d and D_{req}, respectively. Let D_{ij} be the mean link delay between node i and j. The mean end-to-end delay of the lowest delay route r^*, $D_n(d)$, is found by $D_n(d) = \sum_{(i,j)\in r^*} D_{ij}$. The parameter $\triangle D_n(d)$ is the variation of the mean end-to-end delay which is computed from

$$\triangle D_n^{new}(d) = \rho \triangle D_n^{old}(d) + (1-\rho)\,\beta \left| D_n^{new}(d) - D_n^{old}(d) \right|. \quad (1)$$

The parameter ρ is the forgetting factor which determines how fast $\triangle D_n^{old}(d)$ is forgotten, $(1-\rho)$ determines how fast $\triangle D_n^{new}(d)$ converges to $\left|D_n^{new}(d) - D_n^{old}(d)\right|$, and β is a parameter chosen to ensure a large value of $\triangle D_n^{new}(d)$. Note that by increasing β, we increase $\triangle D_n^{new}(d)$ and consequently, the certainty that the actual delay falls in the imprecise range. In [1], the number of tickets (M_0) is found from $M_0 = Y_0 + G_0$ where Y_0 and G_0 are the number of yellow and green tickets, respectively. The yellow tickets are for maximizing the chances of finding feasible paths while the green tickets are for maximizing the chances low cost paths.

The parameter Y_0 is determined according to these heuristic rules [1]:

$$Y_0 = \begin{cases} 1 & , D_{req} > D_{hi} \\ \left\lceil \frac{D_s(d)+\triangle D_s(d)-D_{req}}{2\times \triangle D_s(d)} \times \theta_Y \right\rceil & , D_{lo} \le D_{req} \le D_{hi} \\ 0 & , D_{req} < D_{lo} \end{cases} \quad (2)$$

where $D_{hi} = D_s(d) + \triangle D_s(d)$, $D_{lo} = D_s(d) - \triangle D_s(d)$, θ_Y is a system parameter specifying the maximum allowable number of yellow tickets.

The other parameter, G_0, follows a slightly different set of rules:

$$G_0 = \begin{cases} 1 & , D_{req} > \Theta D_{hi} \\ \left\lceil \frac{\Theta(D_s(d)+\triangle D_s(d))-D_{req}}{\Theta(D_s(d)+\triangle D_s(t))-D_s(d)} \times \theta_G \right\rceil & , D_s(d) \le D_{req} < \Theta D_{hi} \\ \left\lceil \frac{D_{req}-D_s(d)+\triangle D_s(d)}{\triangle D_s(d)} \times \theta_G \right\rceil & , D_{lo} \le D_{req} < D_s(d) \\ 0 & , D_{req} < D_{lo} \end{cases} \quad (3)$$

where θ_G specifies the maximum allowable number of green tickets, $\Theta > 1$ specifies the threshold beyond D_{req} which we allow to search for large-delay paths.

The intuitive reasoning behind the above rules is simple. If D_{req} is very large, then a single yellow ticket suffices. If D_{req} is within the estimated range, then more yellow tickets are assigned for more stringent D_{req}. In the case where D_{req} is less than the best estimated end-to-end delay, no tickets are issued since such a tight requirement is unlikely to be satisfied. The connection request is rejected or some negotiation for a less stringent requirement is made. The green tickets undergo a similar strategy. The selection of the system parameters (θ_Y, θ_G and Θ) is a practical design issue which can depend on level of overhead control imposed on the network [1].

B. Initial ticket calculation: TBP scheme based on the ONMC method

The ONMC method for POMDPs can be applied to the actual system or simulator to obtain a good ticket issuing policy—one that balances the trade-off in the number of issued tickets and the probability of discovering feasible paths. More specifically, instead of calculating M_0 from an heuristic rule like in (2) and (3), M_0 is selected from some finite set in a sequential decision-making process in the presence of state uncertainty with the objective of maximizing some performance criterion.

Consider a $\mathcal{N}$-node MANET. Each mobile node maintains end-to-end delay information to all the destination nodes in the network. For each source node s, a policy is determined separately for each destination node d in the network. Hence, for each source-destination node pair (s, d), the observation set is defined as

$$\mathcal{O}_{sd} = \{[q_D(m), q_{\Delta D}(l)] : 1 \le m \le n, 1 \le l \le n_\Delta\}$$

where n (n_Δ) is the number of discrete end-to-end delay (end-to-end delay variation) intervals and $q_D(m)$ ($q_{\Delta D}(l)$) is the m^{th} (l^{th}) interval on $[0, \infty)$. The variable $q_{\Delta D}(l)$ is included to reduce the uncertainty of the actual end-to-end delay.

Based on $o_k \in \mathcal{O}_{sd}$ at time k, node s takes an action $a_k \in A = \{0, \ldots, M_{\max}\}$ by selecting some $M_0 \in A$ tickets, where $M_{\max}$ is the maximum allowable number of tickets. To maximize the probability of discovering a feasible path, note that high-cost (e.g. longer hops) paths can be tolerated as long as a feasible path can be discovered. The green tickets are omitted ($G_0 = 0$) and only the yellow tickets are considered so that $M_{\max} = \theta_Y$ in order to put more emphasis on finding feasible paths rather than low-cost paths. If the selected action is $a_k = M_0 > 0$, the tickets are distributed in the manner as the original TBP scheme. If at least one feasible path is found once the path discovery is completed, a reward $g(o_k, a_k)$ is generated. Otherwise, the action is penalized. Such reward scheme is defined as

$$g(\cdot, a_k) = \begin{cases} \zeta_j - \log a_k & , a_k > 0,\ X = \varkappa \\ -(\zeta_j - \log a_k) & , a_k > 0,\ X = 0 \\ -\log a_k & , a_k > 0,\ X > \varkappa \\ 0 & , a_k = 0 \end{cases} \quad (4)$$

where $\zeta_j \in \mathcal{R}^+$ is the immediate reward parameter for service type-j, X is the number of discovered feasible paths, $\varkappa$ is the desired maximum number of discovered feasible paths. Note that this scheme favors issuing tickets which can find up to $\varkappa$ paths only—issuing too few or too many tickets than necessary is penalized.

If multiple feasible paths are discovered, the destination node d selects the least-cost path. It then returns an acknowledge message which includes the new mean end-to-end delay, $D_s^{new}(d)$, to node s by backtracking the selected path. Upon receiving the acknowledge message, node s updates its network information with the new entries, i.e., $D_s^{new}(d)$ and $\Delta D_s^{new}(d)$, the latter having been computed from (1). Note that all other entries to other destination nodes remain the same. If no feasible route is found, no acknowledgment is returned and the global information at node s remains unchanged.

The process is repeated for every connection request at node s until an exchange of distance vectors occurs at node s. Such exchange occurs periodically or whenever a topology change is detected, causing an update to the entries of the global information at node s—independent of the previous actions taken (i.e., the number of M_0 selected). Therefore, using the on-policy first-visit Monte Carlo method in this scenario, we want to determine a near-optimal observation-based deterministic policy $\pi : \mathcal{O}_{sd} \to A$.

C. Path Caching

In [7], the TBP scheme based on the ONMC method is invoked to discover new paths for every connection request. To avoid frequently invoking the path discovery algorithm for every connection request, a path cache can be maintained at each mobile node [3], [4]. Path caching strategies are thus likely to help reduce overhead in MANETs.

In this paper, we use a simple path caching strategy which is readily supported by the TBP scheme. The entries of the path cache are the set of redundant paths discovered from the TBP scheme. The size of the path cache depends on the desired degree of path redundancy. Since paths can be broken at any time, the entries in the path cache can become out-of-date. To deal with such dynamic nature, each path entry in the path cache is validated by a timeout procedure. That is, each path entry requires a *refreshing* message periodically in order to remain in the path cache. The refreshing message is periodically initiated from the destination node, and propagates to intermediate nodes along the path. Once the refreshing message is received at a node, the timer for that path entry is reset and the refreshing message is propagated upstream towards the source node. If no refreshing message is received within a time period, the path entry is deleted.

IV. Numerical Study

The performance of the modified TBP schemes based on the ONMC method are evaluated on MANETs through simulations. To assess their performance, the following four metrics are considered: i) *Accumulated reward* which is equal to the accumulated reward over all episodes divided by the total number of episodes, ii) *Success ratio* which is equal to the total number of accepted connections divided by the total number of connection requests, iii) *Average path cost* which is equal to the total cost of all established connections divided by the total number of established connections and iv) *Average number of search messages* which is equal to the total number of search messages sent divided by the total number of connection requests,. Note that one search message is counted each time a probe is sent over a link. Therefore, a probe which has traversed l hops in the network has created l search messages.

We consider a MANET of 36 nodes placed in a 15×15 square meter area. The topology of the MANET is randomly generated by a random way point mobility model. The velocity is uniformly chosen between 0.3 to 0.7 meters per second. Each node has a circular transmission range with a radius of 3 meters. A link is formed between any two mobile nodes located within this transmission range.

Connection requests are generated at a source node at rate 0.2 connections per second. The cost of each link is uniformly distributed in $[0, 1]$. Each link connecting nodes i and j has two types of link delays associated to it, namely, the actual (D_{ij}) and announced mean link delay $\left(\tilde{D}_{ij}\right)$. The latter type