

การวิเคราะห์สมรรถนะของอัลกอริทึมต่อต้านการชนกันภายใต้วิธีการประมาณจำนวน แท็กที่เกิดการชนกันที่แม่นยำสำหรับระบบอาร์เอฟไอดี

Performance Analysis of Tag Anti-Collision Algorithms Under Accurate the Number of Colliding Tag Estimation Method for RFID Systems

นรรรัตน์ วัฒนมงคล

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา

169 ถนนลงหาดบางแสน ตำบลแสนสุข อำเภอเมือง จังหวัดชลบุรี 20131

Norrrat Wattanamongkhon

Department of Electrical Engineering, Faculty of Engineering, Burapha University

Longhad Bangsaen Road, Saensook, Mueang, ChonBuri 20131

*Corresponding author Email: norrrat@eng.buu.ac.th

(Received: September 27, 2020; Revised: December 2, 2020; Accepted: February 17, 2021)

บทคัดย่อ

ในกระบวนการของการระบุตัวตนในระบบอาร์เอฟไอดีนั้นเมื่อมีแท็กจำนวนหนึ่งต้องการส่งสัญญาณไปหาเครื่องอ่านในเวลาใกล้เคียงกันจะเกิดการชนกันขึ้น ปัญหาการชนกันของแท็กมีความสำคัญอย่างมาก เนื่องจากส่งผลถึงเวลาประวิงที่ใช้ในการระบุตัวตน การใช้พลังงานของแท็กในการติดต่อสื่อสาร หรือการใช้งานแบนด์วิดท์ของช่องสัญญาณงานวิจัยฉบับนี้นำเสนอสมการคณิตศาสตร์ใหม่เพื่อใช้ในการวิเคราะห์สมรรถนะการทำงานของอัลกอริทึมต่อต้านการชนกันสำหรับระบบอาร์เอฟไอดีซึ่งประกอบด้วย 4 อัลกอริทึมคือ สล็อตอะโลฮาแบบปรับขนาดเฟรมได้ สล็อตอะโลฮาแบบปรับขนาดเฟรมและข้ามไม่ใช้สล็อตได้ อัลกอริทึมทรีแบบปรับขนาดเฟรมได้ และอัลกอริทึมทรีแบบปรับขนาดเฟรมและข้ามไม่ใช้สล็อตได้ จากผลการทดสอบพบว่าอัลกอริทึมทรีแบบปรับขนาดเฟรมและข้ามไม่ใช้สล็อตได้จะให้ประสิทธิภาพสูงสุดเท่ากับ 0.52 เมื่อมีแท็กที่เกิดการชนกันในระบบจำนวน 100 แท็ก

คำสำคัญ: อาร์เอฟไอดี อัลกอริทึมต่อต้านการชน สล็อตอะโลฮา โพรโทคอลแบบปรับตัวได้

ABSTRACT

In the process of identification in RFID systems, a number of tags were required to send signals to nearby readers, a collision would occur. It is very important because it affects the access delay in identification, power consumption of tags, or bandwidth utilization. In this research, we present a novel mathematical models for the performance analysis of tag anti-collision algorithms for RFID systems, including AF-SA, AFS-SA, TA-AF and TA-AFSS. From the experimental results we found that the TA-AFSS algorithm provides the highest efficiency as 0.52 for the number of 100 collided tags in the system.

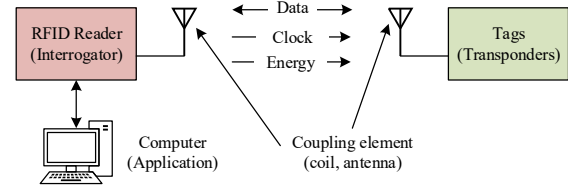
Keyword: RFID, anti-collision algorithm, slotted aloha, adaptive protocol.

1. บทนำ

อาร์เอฟไอดี [1] (Radio Frequency Identification: RFID) คือเทคโนโลยีระบุตัวตนอัตโนมัติแบบไม่สัมผัสสามารถระบุตัวตนของวัตถุเป้าหมายได้อย่างอัตโนมัติผ่านการสื่อสารข้อมูลแบบสองทาง (Full-duplex) โดยใช้สัญญาณคลื่นวิทยุแบบกระจายในทุกทิศทาง (Omni-directional) ซึ่งสามารถทำงานได้อย่างสะดวก ถูกต้อง และรวดเร็ว แทนที่จะต้องใช้การนับหรือจดบันทึกด้วยมนุษย์ที่อาจเกิดข้อผิดพลาดได้ง่ายและยังทำงานในสิ่งแวดล้อมที่อันตรายได้ ปัจจุบันได้มีการนำเอาเทคโนโลยี RFID มาใช้งานอย่างแพร่หลาย เช่น กระบวนการผลิต ระบบคลังสินค้า โลจิสติกส์ ระบบรักษาความปลอดภัย ระบบบริหารจัดการจราจร หรือระบบการสาธารณสุข ซึ่งต่างจากเทคโนโลยีระบุตัวตนแบบไร้สายชนิดอื่นๆ เช่น การอ่านบาร์โค้ด (Barcode) หรือคิวอาร์โค้ด (QR-code) ที่อาศัยคลื่นอินฟราเรดที่ต้องรับส่งสัญญาณในแนวเส้นตรง หากมีสิ่งกีดขวางมาบังจะไม่สามารถสื่อสารถึงกันได้

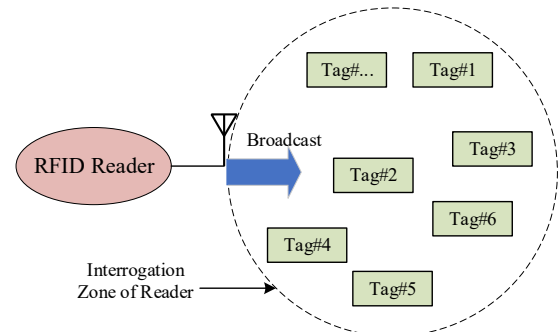
โดยทั่วไประบบ RFID ประกอบไปด้วย 3 ส่วนหลัก คือเครื่องอ่าน (Reader หรือ Interrogator) แท็กหรือทรานสปอนเดอร์ (Tag หรือ Transponder) มีลักษณะเป็นสติกเกอร์หรือชิปที่มีขนาดเล็กและระบบบริหารจัดการ (Management system) [1] ดังแสดงในรูปที่ 1 ทั้งเครื่องอ่านและแท็กจะมีอุปกรณ์สายอากาศอยู่ด้านในซึ่งถูกใช้สำหรับรับและส่งสัญญาณ โดยแท็กจะถูกติดหรือฝังลงในวัตถุที่ต้องการอ่านข้อมูล จากนั้นเครื่องอ่านจะรวบรวมและถ่ายโอนข้อมูลที่ได้รับมาจากระบบบริหารจัดการเพื่อทำการประมวลผลข้อมูลอย่างเป็นลำดับตามความต้องการที่แตกต่างกันของการนำไปใช้งาน ในระบบ RFID ทุกๆ แท็กจะมีหมายเลขประจำตัวของตนเองซึ่งได้จากการเข้ารหัสแมนเชสเตอร์ (Manchester coding) ดังนั้นเครื่องอ่านจะสามารถระบุตัวตนของแท็กที่ทราบเท่าที่มันยังได้รับหมายเลขประจำตัวของแท็กนั้นอยู่ ปัจจุบันแท็กที่ถูกนำมาใช้งานส่วนใหญ่จะเป็นแบบพาสซีฟ (Passive tag) คือแท็กที่ไม่ต้องมีแหล่งพลังงานบรรจุไว้ที่ตัวเอง แต่จะใช้การเหนี่ยวนำพลังงานจากเครื่องอ่านผ่าน

สายอากาศที่ฝังอยู่ภายในโดยต้องอยู่ในระยะไม่ห่างกันนัก



รูปที่ 1 ส่วนประกอบที่สำคัญของระบบ RFID

จากรูปที่ 2 ในขั้นตอนของการสื่อสารระหว่างเครื่องอ่านและแท็กของระบบ RFID หนึ่งๆ หากมีแท็กจำนวนตั้งแต่ 2 แท็กขึ้นไปในรัศมีการแพร่กระจายสัญญาณของเครื่องอ่าน (Interrogation zone of reader) หรือมีแท็กหนึ่งอยู่ในรัศมีการส่งสัญญาณของเครื่องอ่านมากกว่าหนึ่งเครื่องจะเกิดการแลกเปลี่ยนข่าวสารผ่านช่องสัญญาณความถี่เดียวกันและในเวลาเดียวกัน จึงนำไปสู่ปัญหาการชนกันของแท็กเกิดข้อมูล ส่งผลให้ไม่มีแท็กใดเลยที่สามารถส่งแท็กเกิดของตนเองไปยังเครื่องอ่านได้สำเร็จ การชนกันของแท็กเกิดเป็นปัญหาที่สำคัญมากในระบบ RFID [1,2] เพราะส่งผลให้การใช้พลังงานของแท็กมากขึ้น มีข้อจำกัดในการประมวลผลและเวลาประวิงของการส่งข่าวสารก็มากขึ้นด้วย ยิ่งไปกว่านั้น ยังส่งผลให้เกิดข้อจำกัดในการพัฒนาและเกิดผลกระทบต่อการนำไปประยุกต์ใช้งาน



รูปที่ 2 รัศมีการสื่อสารระหว่างเครื่องอ่านและแท็ก

ดังนั้นในงานวิจัยนี้มีวัตถุประสงค์นำเสนออัลกอริทึมที่แก้ปัญหการชนกันของแท็กได้อย่างมีประสิทธิภาพและสมการคณิตศาสตร์ที่สามารถวิเคราะห์สมรรถนะของอัลกอริทึมได้อย่างรวดเร็วสำหรับระบบอาร์เอฟไอดี

ภายใต้สมมติฐานที่สามารถประมาณจำนวนแท็กที่เกิดการชนกันได้อย่างแม่นยำ

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

จากการสืบค้นงานวิจัยและรวบรวมข้อมูลในอดีตจนถึงปัจจุบันที่ผ่านมามีนักวิจัยได้นำเสนอโพรโทคอลเพื่อแก้ปัญหาการชนกันในระบบ RFID แบบไร้สาย สามารถแบ่งออกได้เป็น 2 กลุ่มหลักคือ โพรโทคอลแก้ปัญหาการชนกันของแท็ก (Tag anti-collision protocols) และอีกกลุ่มคือโพรโทคอลแก้ปัญหาการชนกันของเครื่องอ่าน (Reader anti-collision protocols) [1-4] ซึ่งในงานวิจัยฉบับนี้สนใจกลุ่มที่เสนอโพรโทคอลแก้ปัญหาการชนกันของแท็ก เนื่องจากเป็นปัญหาที่มักเกิดขึ้นได้ทั่วไปเมื่อนำไปใช้งานจริง อัลกอริทึมที่นำเสนอในปัจจุบันจะอาศัยเทคโนโลยีการเข้าถึงช่องสัญญาณแบบแบ่งเวลา (Time Division Multiple Access: TDMA) เนื่องจากเป็นวิธีการเข้าถึงช่องสัญญาณที่มีความซับซ้อนน้อย โดยเราสามารถแบ่งออกได้เป็น 2 กลุ่มคือ อัลกอริทึม ALOHA-based [5]-[7] และอัลกอริทึม Tree-based [8]-[11]

แนวคิดพื้นฐานของอัลกอริทึม ALOHA-based คือการแบ่งช่วงเวลาออกเป็นไทม์สล็อตเพื่อให้แท็กแต่ละตัวสามารถสื่อสารกับเครื่องอ่านได้ในแต่ละไทม์สล็อตโดยใช้อัลกอริทึมการปรับค่าความน่าจะเป็น (Probability anti-collision algorithms) จึงช่วยลดโอกาสการชนกันของแท็กเกิดลงได้ แทนที่จะอนุญาตให้แท็กแต่ละตัวสามารถติดต่อกับเครื่องอ่านที่มีอยู่เครื่องเดียวได้อย่างอิสระซึ่งจะส่งผลให้เกิดการชนกันบ่อยมาก ด้วยอัลกอริทึมนี้สามารถพัฒนาต่อได้ง่ายและนำไปประยุกต์เพื่อระบบตัวตนของแท็กจำนวนเท่าใดก็ได้ ดังนั้นอัลกอริทึม ALOHA-based จึงถูกนำไปใช้งานอย่างแพร่หลายเพื่อแก้ปัญหาการชนกันสำหรับระบบ RFID ตัวอย่างอัลกอริทึมในกลุ่มนี้ที่ถูกนำเสนอ [5]-[7] เช่น ALOHA, Slotted ALOHA, Frame Slotted ALOHA (FSA) และ Dynamic Frame Slotted ALOHA (DFSA) อัลกอริทึมกลุ่มนี้อาจจะเกิดปัญหาที่มีบางแท็กไม่สามารถติดต่อกับ

เครื่องอ่านได้เลยซึ่งปัญหาเหล่านี้จะเพิ่มมากขึ้นตามจำนวนแท็กที่มากขึ้น ส่วนอีกกลุ่มหนึ่งคือ อัลกอริทึม Tree-based จะกำหนดค่าความน่าจะเป็นตายตัว (Deterministic anti-collision algorithms) โดยการประยุกต์ใช้หลักความเป็นคู่และความแน่นอน (Duality and certainty principle) เพื่อแก้ปัญหาการชนกันของแท็กโดยการแบ่งแยกแท็กที่เกิดการชนกันออกเป็นกลุ่มย่อยต่อเนื่องกันไปเรื่อยๆ จนกระทั่งแท็กทุกตัวได้รับการระบุตัวตนสำเร็จ ซึ่งถือเป็นข้อดีเมื่อเทียบกับอัลกอริทึม ALOHA-based ส่วนข้อด้อยคือมีการคำนวณที่ซับซ้อนและใช้เวลาประมวลผลนานกว่า โดยเฉพาะอย่างยิ่ง เมื่อมีแท็กจำนวนมากอยู่ในรัศมีการสื่อสารของเครื่องอ่าน ตัวอย่างอัลกอริทึมในกลุ่มนี้ [8]-[11] เช่น Binary Tree (BT), Modified Binary Tree (MBT), Query Tree (QT) และ Optimal Query Tracking Tree (OQTT)

2.1 Slotted ALOHA (SA)

อัลกอริทึม SA เป็นอัลกอริทึมแรกที่ได้รับการพัฒนาต่อจาก ALOHA โดยจะให้ค่าวิสัยสามารถ (Throughput) เพิ่มขึ้นจากเดิม 2 เท่า ในระบบ RFID เมื่อพิจารณาว่ามีเครื่องอ่านหนึ่งเครื่องและมีแท็กจำนวนหนึ่งอยู่ในรัศมีการส่งสัญญาณระหว่างกันโดยใช้ช่องสัญญาณสื่อสารร่วมกัน ซึ่งถูกจัดสรรเวลาเป็นไทม์สล็อต (Timeslot) ที่มีขนาดเล็กๆ จำนวนหนึ่ง โดยอนุญาตให้แท็กสามารถส่งแท็กเกิดข้อมูลได้ที่ตำแหน่งเริ่มของแต่ละสล็อตเท่านั้น (ซึ่งเดิมระบบ ALOHA จะอนุญาตให้แท็กส่งแท็กเกิดของตนได้ทันที ส่งผลให้เกิดการชนบ่อยครั้งเพราะในระหว่างส่งแท็กเกิดของแท็กแต่ละตัวหากมีการส่งซ้อนเหลื่อมกันแม้เพียงเล็กน้อย เครื่องอ่านก็จะพิจารณาว่าเกิดการชนกันทันที) ในขณะที่เครื่องอ่านจะทำหน้าที่ควบคุมจำนวนสล็อตและจัดสรรช่องทางการสื่อสารกับแท็กในช่วงเวลาที่จำกัดของแต่ละสล็อต หลังจากเครื่องอ่านส่งคำสั่งกระจายออกไป เมื่อไหร่ก็ตามที่แท็กได้รับสัญญาณนี้จะส่งหมายเลขประจำตัวของตนเองออกไปที่ตำแหน่งเริ่มต้นของสล็อต ถ้ามีเพียงแคแท็กเดียวตอบสนอง เครื่องอ่านก็จะสามารถระบุตัวตนของแท็กนั้นได้สำเร็จ (Success) แต่หาก

มีแท็กตั้งแต่ 2 ตัวขึ้นไปส่งสัญญาณออกมาพร้อมกันจะเกิดการชนกันขึ้น (Collision) ส่งผลให้เครื่องอ่านไม่สามารถระบุตัวตนของแท็กใดได้เลย ดังนั้นแท็กที่เกิดการชนกันทั้งหมดจะต้องเริ่มต้นติดต่อกับเครื่องอ่านซ้ำไปเรื่อยๆ ซึ่งจะต้องใช้พลังงานและเวลาประวิงในการประมวลผลมากขึ้น ปัญหานี้จะรุนแรงมากขึ้น เมื่อมีจำนวนแท็กมากขึ้น นอกจากนั้นแล้ว ในกรณีที่ไม่มีแท็กใดเลยส่งสัญญาณกลับมาที่เครื่องอ่านจะเกิดสล็อตว่าง (Idle) ซึ่งจะเป็นการเสียโอกาสในการส่งข้อมูล

2.2 Frame Slotted ALOHA (FSA)

ข้อบกพร่องหลักของอัลกอริทึม SA คือประสิทธิภาพการทำงานจะลดลงมาก เมื่อจำนวนแท็กในระบบเพิ่มขึ้น โอกาสในการชนกันก็จะมากขึ้นตามไปด้วย อัลกอริทึม Frame Slotted ALOHA (FSA) จึงถูกนำเสนอเพื่อเอาชนะปัญหานี้ โดยกำหนดให้แท็กแต่ละตัวจะได้รับอนุญาตให้ส่งข้อมูลได้เพียงครั้งเดียวเท่านั้นในแต่ละเฟรม [6] (ในหนึ่งเฟรมประกอบด้วยสล็อตจำนวนหนึ่งซึ่งมีจำนวนคงที่) เพื่อลดโอกาสในการชนลง จึงสามารถเพิ่มค่าวิสัยสามารถ (Throughput) ของระบบได้

จากรูปที่ 3 แสดงขั้นตอนการทำงานของอัลกอริทึม FSA ในหนึ่งเฟรมมี 3 สล็อต ในขณะที่มีแท็กที่เกิดการชนกันเริ่มต้นคือ A, B, ..., E เมื่อพิจารณาเฟรมแรกจะมีเพียงแท็ก C เท่านั้นที่ส่งสำเร็จ แท็กที่เหลือทั้งหมดจะสุ่มเลือกสล็อตอีกครั้งในเฟรมถัดไป ซึ่งมีแท็กที่ส่งแพ็กเก็ตสำเร็จคือ E ในขณะที่แท็ก A, B และ D ยังคงชนกันอยู่ จากนั้นแท็กที่เหลืออยู่จะดำเนินการตามขั้นตอนเช่นนี้วนซ้ำไปเรื่อยๆ จนกระทั่งถึงเฟรมที่ 4 แท็กทุกตัวสามารถส่งแพ็กเก็ตได้สำเร็จ จากตัวอย่างนี้จะเห็นว่าในขั้นตอนการแก้ปัญหาการชนกันของแท็กทั้งหมดนี้ระบบจะต้องใช้เวลาไปทั้งหมดเท่ากับ 12 สล็อต ในจำนวนนี้จะเกิดสล็อตว่างสะสมรวมเท่ากับ 3 สล็อต

Frame#	1			2			3			4		
Slot#	1	2	3	1	2	3	1	2	3	1	2	3
Tag#	A,D	B,E	C	A,B,D	E		A	B,D			D	B
A,B,C,D,E			A,B,D,E			A,B,D			B,D			สิ้นสุด

รูปที่ 3 ตัวอย่างการทำงานของอัลกอริทึม FSA

2.3 Dynamic Frame Slotted ALOHA (DFSA)

ข้อเสียเปรียบหลักของอัลกอริทึม FSA คือโดยทั่วไปเครื่องอ่านจะไม่ทราบจำนวนแท็ก เนื่องจากทุกเฟรมจะมีความยาวเท่ากันหรือมีจำนวนสล็อตในแต่ละเฟรมคงที่ ดังนั้นจึงเกิดปัญหาที่สำคัญ 2 เหตุการณ์คือ 1. เมื่อจำนวนสล็อตในแต่ละเฟรมน้อยกว่าจำนวนแท็กที่มีอยู่มาก ดังนั้นอาจเกิดการชนกันในหลายสล็อตของเฟรม ส่งผลให้เครื่องอ่านไม่สามารถระบุตัวตนได้เป็นระยะเวลานาน และ 2. เมื่อจำนวนสล็อตของเฟรมมีค่ามากกว่าจำนวนแท็กที่มีอยู่จะเกิดสล็อตว่างจำนวนมาก ด้วยเหตุนี้จึงได้มีนักวิจัยนำเสนออัลกอริทึม Dynamic Frame Slotted ALOHA (DFSA) เพื่อแก้ปัญหาดังกล่าว โดยประเด็นหลักที่นำเสนอคือความยาวของเฟรมจะเปลี่ยนแปลงไปตามจำนวนแท็ก ร่วมกับใช้วิธีการประมาณจำนวนแท็กซึ่งมีความสำคัญอย่างมาก แต่วิธีการประมาณจำนวนแท็กไม่มีอัลกอริทึมที่กำหนดเป็นมาตรฐาน

จากรูปที่ 4 เมื่อใช้อัลกอริทึม DFSA ในการแก้ปัญหาการชน เมื่อมีแท็ก A, B, ..., E ในเฟรมแรกจะใช้ขนาด 5 สล็อต ซึ่งเท่ากับจำนวนแท็กที่เกิดการชนกัน โดยเหลืออีก 2 แท็กที่ยังส่งไม่สำเร็จ จากนั้นในเฟรมที่ 2 จะใช้เฟรมขนาด 2 สล็อต แต่ไม่ยังคงส่งไม่สำเร็จเช่นเดิม ดังนั้นในเฟรมที่ 3 จะใช้เฟรมขนาด 2 สล็อต เช่นเดิมและส่งได้สำเร็จในที่สุด โดยใช้เวลาไปทั้งหมดเท่ากับ 9 สล็อต และในจำนวนนี้จะเกิดสล็อตว่างสะสมเท่ากับ 2 สล็อต ซึ่งใช้จำนวนสล็อตในการแก้ปัญหการชนน้อยกว่าตัวอย่างในรูปที่ 3 เมื่อพิจารณาจำนวนแท็กที่เกิดการชนเริ่มต้นเท่านั้น

Frame#	1					2		3	
Slot#	1	2	3	4	5	1	2	1	2
Tag#		E	C	B,D	A	B,D		B	D
A,B,C,D,E					B,D		B,D		สิ้นสุด

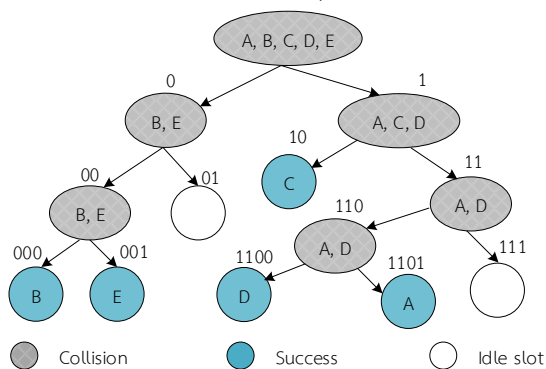
รูปที่ 4 ตัวอย่างการทำงานของอัลกอริทึม DFSA

2.4 Binary tree algorithm (BTA)

ตามหลักการพื้นฐานของอัลกอริทึมต้นไม้ หลังจากที่เกิดการชนกันของแท็กขึ้นจะเข้าสู่กระบวนการแก้ปัญหาการชน แท็กอื่นๆ ที่ไม่เกี่ยวข้องกับการชนจะต้องรอ

จนกระทั่งปัญหาการชนกันได้รับการแก้ไข เริ่มต้นจะแบ่งแท็กที่เกิดการชนทั้งหมดออกเป็น 2 กลุ่มคือ 0 และ 1 ซึ่งจะพิจารณาจากหมายเลขประจำตัวแท็กทีละบิต โดยกลุ่มบิต 0 จะได้รับการพิจารณาก่อนเสมอ ในขณะที่แท็กในกลุ่มบิต 1 จะต้องรอจนกระทั่งแท็กทุกตัวในกลุ่มบิต 0 สำเร็จ (มีเพียงแท็กเดียวที่ส่งข้อมูลในสล็อตนั้น) ทุกๆ ครั้งที่เกิดการชนกันจะแบ่งแยกแท็กออกเป็น 2 กลุ่ม เช่นเดิมพร้อมกับพิจารณาบิตในลำดับถัดไป กระบวนการเช่นนี้จะวนซ้ำไปเรื่อยๆ จนกระทั่งแท็กทุกตัวได้รับการระบุตัวตนสำเร็จ

จากรูปที่ 5 แสดงตัวอย่างการทำงานของอัลกอริทึม BTA เมื่อมีแท็กที่เกิดการชนกันเริ่มต้นคือ A, B, ..., E โดยจะเริ่มแก้ปัญหาการชนในกลุ่มบิต 0 ทางซ้ายมือก่อนเสมอในแนวลึกลับซ้ายกับรากของต้นไม้ จนกระทั่งแท็กเหล่านั้นส่งแพ็กเก็ตได้สำเร็จ จากนั้นจึงแก้ปัญหาการชนต่อไปในกลุ่มบิต 1 จากรูปนี้จะใช้เวลาในการแก้ปัญหาการชนกันทั้งหมด 12 สล็อต โดยมีสล็อตว่างจำนวน 2 สล็อต สำหรับแท็กที่เกิดการชนกันจำนวน 5 แท็ก ซึ่งแสดงให้เห็นว่าจากหลักการแก้ปัญหาการชนกันของแท็กด้วยอัลกอริทึม Tree-based จะมีประสิทธิภาพสูงกว่าอัลกอริทึม ALOHA-based เนื่องจากวิธีการแบ่งแยกกลุ่มจะทำให้โอกาสในการชนกันของแท็กลดลงอย่างรวดเร็ว ในขณะที่อัลกอริทึม ALOHA-based จะนำแท็กที่เกิดการชนกันทั้งหมดไม่ว่าเกิดขึ้นที่ตำแหน่งในสล็อตใดของเฟรมก็ตามก็ต้องทำการสุ่มเลือกสล็อตเพื่อส่งแพ็กเก็ตในเฟรมถัดไปเสมอ จึงมีโอกาสเป็นไปได้สูงที่แท็กบางตัวอาจใช้เวลานานกว่าที่จะได้รับการระบุตัวตนสำเร็จ

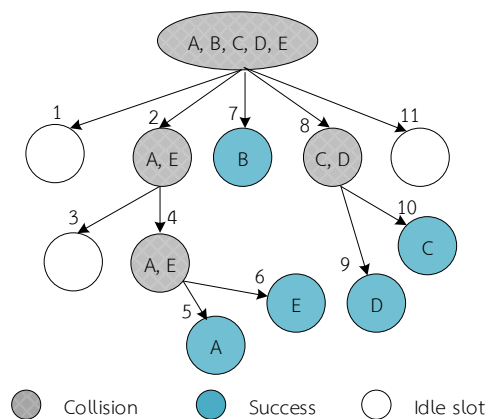


รูปที่ 5 ตัวอย่างการทำงานของอัลกอริทึม BTA

2.5 Adaptive Tree algorithm (ATA)

อัลกอริทึม Adaptive Tree algorithm (ATA) จะเป็นอัลกอริทึมที่ถูกพัฒนาให้มีประสิทธิภาพสูงขึ้นจาก Tree-based โดยทั่วไป โดยอาศัยแนวความคิดการปรับขนาดเฟรมเช่นในอัลกอริทึม ALOHA-based ดังนั้นจึงถือเป็นการเอาข้อดีของทั้งเทคนิคการแบ่งแยกกลุ่มและเทคนิคการปรับขนาดของเฟรมได้เอง ส่งผลให้ใช้เวลาในการระบุตัวตนน้อยลงในขณะที่จำนวนสล็อตก็มีความเหมาะสมกับจำนวนแท็กที่เหลืออยู่ในขณะนั้นด้วย ซึ่งขั้นตอนการแก้ปัญหาการชนกันโดยใช้เทคนิคเช่นนี้จะต้องอาศัยอัลกอริทึมที่สามารถประมาณจำนวนแท็กได้อย่างถูกต้องและรวดเร็ว

จากรูปที่ 6 แสดงตัวอย่างการทำงานของอัลกอริทึม ATA เมื่อมีแท็กที่เกิดการชนกันเริ่มต้นจำนวน 5 แท็ก คือ A, B, ..., E และได้แสดงหมายเลขลำดับสล็อตที่ถูกใช้แก้ปัญหาการชนกันของแท็กเพื่อให้สามารถเปรียบเทียบและเข้าใจกลไกการทำงานได้ง่ายยิ่งขึ้น เริ่มต้นจากการแบ่งแยกออกเป็น 5 สล็อต ให้เท่ากับจำนวนแท็กที่มีอยู่ในรอบแรกนี้แท็ก B สามารถสุ่มเลือกสล็อตเพื่อส่งแพ็กเก็ตได้สำเร็จ ในขณะที่แท็ก A และ E ยังคงสุ่มเลือกได้สล็อตเดียวกัน จึงเกิดการชนกันขึ้นอีก ดังนั้นจึงต้องแบ่งแยกกลุ่มเพื่อแก้ปัญหาการชนต่อเนื่องไปเรื่อยๆ จนกระทั่งสำเร็จ ในขณะที่แท็ก C และ D เมื่อเกิดการชนกันก็ต้องรอเวลาและดำเนินการเช่นเดียวกัน จากตัวอย่างนี้จะใช้เวลาในการแก้ปัญหาการชนกันของแท็กทั้งหมด 11 สล็อต และมีสล็อตว่าง 3 สล็อต ซึ่งดีกว่าอัลกอริทึม BTA



รูปที่ 6 ตัวอย่างการทำงานของอัลกอริทึม ATA

3. วิธีการดำเนินงานวิจัย

จากงานวิจัยในอดีต [5-11] มักนำเอาเทคนิคการปรับขนาดเฟรมเองได้ (Adaptive frame size) และเทคนิคการข้ามไม่ใช้งานสล็อต (Skipping slotted) มาใช้เพื่อปรับปรุงประสิทธิภาพการทำงานของอัลกอริทึมให้สูงขึ้นทั้งในกลุ่ม ALOHA-based และ Tree-based ด้วยเหตุนี้เราจึงได้นำเสนอสมการคณิตศาสตร์เพื่อใช้วิเคราะห์สมรรถนะของอัลกอริทึมต่อต้านการชนกันของแท็กในระบบ RFID จำนวน 4 อัลกอริทึม ซึ่งประกอบด้วย 1. Adaptive Frame Slotted Aloha (AF-SA) 2. Adaptive Frame and Skipping Slotted Aloha (AFS-SA) 3. Tree algorithm with Adaptive Frame (TA-AF) และ 4. Tree algorithm with Adaptive Frame and Skipping Slotted (TA-AFSS) ภายใต้สมมติฐานว่าเครื่องอ่านสามารถประมาณจำนวนแท็กที่มีอยู่ในปัจจุบันได้ในระดับสล็อต เนื่องจากมีงานวิจัยจำนวนมากที่นำเสนอวิธีการประมาณจำนวนแท็กที่มีความแม่นยำ [12-16] โดยอาศัยการประมวลผลภายหลังจากแท็กพยายามส่งแพ็กเก็ตไปยังเครื่องอ่านเพื่อค้นหาจำนวนแท็กที่เหลืออยู่ในแต่ละเฟรมหรือในแต่ละสล็อต ดังนั้นอัลกอริทึมที่นำเสนอในงานวิจัยนี้จึงสามารถลดการใช้งานสล็อตลงได้อย่างชัดเจน และส่งผลให้แท็กได้รับการระบุตัวตนได้รวดเร็วยิ่งขึ้น

ในงานวิจัยนี้จะแสดงสมการคณิตศาสตร์สำหรับวิเคราะห์สมรรถนะของอัลกอริทึมที่นำเสนอในรูปของจำนวนสล็อตโดยเฉลี่ยที่ต้องการใช้ระบุตัวตนของแท็กทั้งหมด N แท็ก (Average number of required slots, $\bar{L}_N$) ซึ่งค่านี้สามารถนำไปประเมินเวลาประวิงของระบบการใช้พลังงานในการสื่อสาร หรือปริมาณการใช้แบนด์วิดท์ของช่องสัญญาณ สำหรับแท็กที่เกิดการชนกันทั้งหมดจำนวน N แท็ก (โดยที่ $N \geq 2$) และในหนึ่งเฟรมจะมีจำนวน M สล็อต เนื่องจากเรานำเสนออัลกอริทึมที่ปรับขนาดเฟรมของได้มาใช้เป็นเทคนิคพื้นฐาน ดังนั้นจึงต้องกำหนดให้จำนวนสล็อตในแต่ละเฟรมเท่ากับจำนวนแท็ก ($M = N$) ที่เกิดการชนกันทั้งหมด โดยมีรายละเอียดดังนี้

3.1 Adaptive Frame Slotted Aloha (AF-SA)

อัลกอริทึม AF-SA จะใช้หลักการเช่นเดียวกับงานวิจัยอื่นๆ ที่ได้นำเสนอแล้วคือการปรับขนาดเฟรมตามจำนวนแท็กที่เกิดการชนกัน โดยเครื่องอ่านจะสามารถทราบจำนวนแท็กที่เหลืออยู่หลังจากประมวลผลเมื่อสิ้นสุดในแต่ละเฟรมโดยไม่สนใจว่าจะเกิดการชนกันที่สล็อตใด ดังนั้นก่อนเริ่มต้นการแก้ปัญหาการชนกันของแท็กในแต่ละเฟรมเครื่องอ่านจะกำหนดจำนวนสล็อตในแต่ละเฟรมให้มีค่าเท่ากับจำนวนแท็กที่เกิดการชนกันคงเหลืออยู่ในปัจจุบันจนกระทั่งแท็กทุกตัวได้รับการระบุตัวตนสำเร็จ โดยขั้นตอนแก้ปัญหาการชนกันของอัลกอริทึม AF-SA จะเหมือนกับอัลกอริทึม DFSA โดยสามารถพิจารณาได้จากรูปที่ 4 ดังนั้นจะสามารถแสดงสมการคณิตศาสตร์เพื่อคำนวณหาจำนวนสล็อตโดยเฉลี่ยที่ต้องการใช้ระบุตัวตนของแท็กทั้งหมด N แท็กได้เป็น

$$\bar{L}_N = N + p_c \bar{L}_c \quad (1)$$

สำหรับกรณีที่ไม่เกิดการชนกันของแท็กเลยจะได้ว่า $\bar{L}_0 = 0$ และ p_c คือความน่าจะเป็นของเหตุการณ์ที่แท็กเกิดการชนกันขึ้นทั้งหมดในแต่ละเฟรม โดยที่

$$p_c = \frac{(N-1)!}{N^{N-1}} \sum_{R_c}^N \left(\binom{N}{k_1, k_2, \dots, k_N} \frac{1}{\prod_{j=0}^N \left(\sum_{i=1}^N I(k_i = j) \right)!} \right) \quad (2)$$

R_c คือรูปแบบของการสุ่มเลือกในแต่ละสล็อตเฉพาะกรณีที่เกิดการชนกันที่เป็นไปได้ทั้งหมด สามารถเขียนเงื่อนไขแทนด้วยสมการ (3)

$$R_c = \{k_1, k_2, \dots, k_N \mid \sum_{i=1, k_i > 1}^N k_i = c\} \quad (3)$$

จากสมการ (2) เทอมของ $\sum_{k_1, k_2, \dots, k_N}^N (\cdot)$ คือผลรวมของการเลือกสล็อตของแท็กที่เป็นไปได้ทั้งหมดของ $k_1, k_2, \dots, k_N$

ซึ่งเป็นจำนวนแท็กที่สุ่มเลือกหมายเลขสล็อต 1,2,...,N ตามลำดับ เช่น ถ้าในหนึ่งเฟรมมีจำนวนสล็อตเท่ากับ 3 จะมีรูปแบบการเลือกสล็อตของแท็กที่แตกต่างกันทั้งหมด 3 รูปแบบคือ 1) $k_1=1, k_2=1, k_3=1$ 2) $k_1=2, k_2=1, k_3=0$ และ 3) $k_1=3, k_2=0, k_3=0$ โดยที่ $\sum_{i=1}^N k_i = N$ เมื่อ $N \geq 2$ และ $k_i \geq 0$ และค่าสัมประสิทธิ์ทวินาม (Multinomial coefficient) ในสมการ (2) มีค่าเท่ากับ

$$\binom{N}{k_1, k_2, \dots, k_N} = \frac{N!}{k_1! k_2! \dots k_N!} \quad (4)$$

ฟังก์ชันตัวบ่งชี้ (Indicator function) ในสมการ (2) คือ $I(k_i = j)$ จะให้ค่าเป็น 1 เมื่อเงื่อนไขเป็นจริง และจะให้ค่าเป็น 0 เมื่อเงื่อนไขเป็นเท็จ

3.2 Adaptive Frame and Skipping Slotted

Aloha (AFS-SA)

อัลกอริทึม AFS-SA ได้รับการพัฒนาต่อจากอัลกอริทึม AF-SA โดยการเพิ่มเทคนิคการข้ามสล็อต (Skipping slot) ร่วมกับเทคนิคการปรับขนาดเฟรมตัวเอง ซึ่งสามารถเกิดเหตุการณ์ข้ามไม่ใช้งานสล็อตได้ 2 รูปแบบดังแสดงในรูปที่ 7 โดยรูปที่ 7(ก) กรณีที่มีจำนวนแท็กตั้งแต่ 2 แท็กขึ้นยังไม่ได้สุ่มเลือกสล็อตใดเลย จนกระทั่งเหลือสล็อตสุดท้ายซึ่งจะเกิดการชนกันอย่างแน่นอน และรูปที่ 7(ข) แท็กทั้งหมดได้สุ่มเลือกสล็อตไปในช่วงแรกแล้ว และยังเหลือสล็อตอีกจำนวนหนึ่งที่ไม่ถูกใช้งานแน่นอน จึงสามารถลดการใช้สล็อตเหล่านี้ได้

เราจะสามารถแสดงสมการคณิตศาสตร์เพื่อคำนวณหาจำนวนสล็อตโดยเฉลี่ยที่ต้องการใช้ระบุตัวตนของแท็กทั้งหมด N แท็กได้เป็น

$$\bar{L}_N = N + p_c \bar{L}_c - p_{skip} \bar{L}_{skip} \quad (5)$$

เมื่อ $\bar{L}_c$ คือจำนวนสล็อตโดยเฉลี่ยที่ต้องการใช้ระบุตัวตนของแท็กในกรณีที่แท็กเกิดการชนกัน $\bar{L}_{skip}$ คือจำนวนสล็อตโดยเฉลี่ยที่ต้องการใช้ระบุตัวตนของแท็กใน

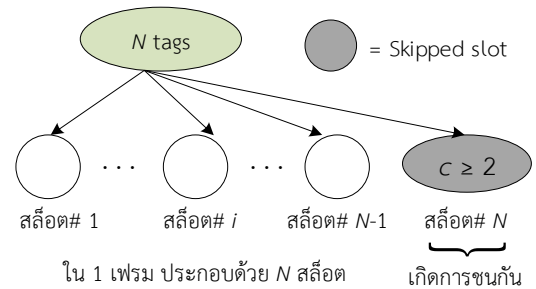
เหตุการณ์กรณีที่เกิดเหตุการณ์ข้ามไม่ใช้งานสล็อต และ p_{skip} คือความน่าจะเป็นของเหตุการณ์ที่สามารถข้ามไม่ใช้งานสล็อตได้ซึ่งมี 2 รูปแบบดังแสดงในรูปที่ 7 โดยที่

$$p_{skip} = \frac{(N-1)!}{N^{N-1}} \sum_{R_{skip}}^N \left[\binom{N}{k_1, k_2, \dots, k_N} \frac{1}{\prod_{j=0}^N \left(\sum_{i=1}^N I(k_i = j) \right)!} \right] \quad (6)$$

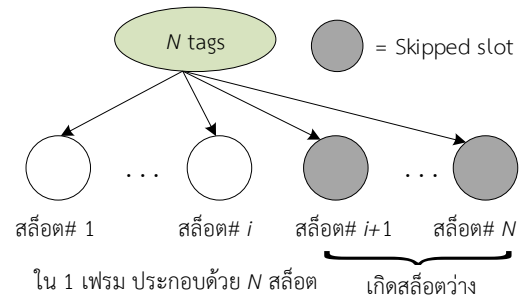
R_{skip} คือรูปแบบที่เป็นไปได้ทั้งหมดของการสุ่มเลือกในแต่ละสล็อตที่สามารถข้ามสล็อตไม่ใช้งานสล็อตได้ สามารถเขียนแทนด้วยสมการ (7)

$$R_{skip} = \{k_1, \dots, k_i, \dots, k_N \mid \begin{cases} k_N \geq 0 \text{ or} \\ k_{i+1} = k_{i+2} = \dots = k_N = 0 \end{cases} \quad (7)$$

สำหรับ p_c และ R_c ยังคงสามารถคำนวณได้จากสมการ (2) และ (3) ตามลำดับเช่นเดิม



(ก) รูปแบบที่ 1



(ข) รูปแบบที่ 2

รูปที่ 7 รูปแบบเหตุการณ์การข้ามไม่ใช้งานสล็อต

3.3 Tree algorithm with Adaptive Frame (TA-AF)

เหตุผลที่สำคัญที่ทำให้สมรรถนะของอัลกอริทึมต้นไม้ต่ำคือการกำหนดให้ขนาดของเฟรมมีค่าคงที่ (Fixed frame size, Q) และอยู่ในรูปกำลังสอง (ตามการเข้ารหัสแมนเชสเตอร์ที่เป็นเลขไบนารี) เช่น $Q = 2, 4, 8, \dots$ ถึงแม้จะมีนักวิจัยนำเสนอว่าควรกำหนด Q มีค่าคงที่เป็นเท่าใดเพื่อให้ได้สมรรถนะที่ต่ำสุดตลอดเวลา แต่เนื่องจากจำนวนแท็กที่เกิดการชนกันเปลี่ยนแปลงอยู่เสมอทุกครั้งที่ผ่านการแก้ปัญหาการชนจนกระทั่งแท็กทุกตัวได้รับการระบุตัวตนสำเร็จจึงเกิดแนวความคิดการปรับขนาดเฟรมได้ตามจำนวนแท็กที่มีอยู่ในปัจจุบันโดยอาศัยอัลกอริทึมที่สามารถประมาณจำนวนแท็กได้อย่างแม่นยำในทุกๆ สล็อต เราจึงได้นำเสนออัลกอริทึม TA-AF ซึ่งมีความแตกต่างจากอัลกอริทึมอื่นที่เคยเสนอมามากตรงที่การกำหนดขนาดของเฟรมที่ไม่อยู่ในรูปของเลขจำนวนเต็มยกกำลังสอง เพราะหากแท็กสุ่มเลือกสล็อตได้สำเร็จแล้วจึงอ่านหมายเลขประจำตัวแท็กต่อเนื่องกันไป การทำเช่นนี้จะช่วยลดโอกาสการเกิดสล็อตว่างได้เพิ่มขึ้น ในขณะที่หลักการแก้ปัญหาการชนในแนวลึกของกลุ่มทางซ้ายมือก่อน จากนั้นค่อยแก้ปัญหาการชนในกลุ่มขวามือต่อเนื่องกันไปเรื่อยๆ ซึ่งจะระจนกระทั่งแท็กในกลุ่มซ้ายมือได้รับการระบุตัวตนสำเร็จทั้งหมดก่อน

เราจะสามารถแสดงสมการคณิตศาสตร์เพื่อคำนวณหาจำนวนสล็อตโดยเฉลี่ยที่ต้องการใช้ระบุตัวตนของแท็กทั้งหมดได้เป็น

$$\bar{L}_N = \frac{1}{(1 - N^{1-N})} \left[N + \sum_{c=2}^{N-1} \binom{N}{c} \frac{(N-1)^{N-c}}{N^{N-1}} \bar{L}_c \right] \quad (8)$$

จากสมการ (8) จะอยู่ในรูปของสมการรีเคอร์ซีฟ (Recursive equation) เนื่องจากต้องมีการคำนวณแบบวนซ้ำเพื่อหาค่า $\bar{L}_c$ ซึ่งเราไม่สามารถหาค่า $\bar{L}_N$ ใดๆ ได้โดยตรง ดังนั้นนอกจากนั้นเรายังสังเกตได้ว่ามีความแตกต่างกันกับสมการ (1) ซึ่งแท็กทั้งหมดที่เกิดการชนกันไม่ว่าตำแหน่งสล็อตใดจะนำไปแก้ปัญหาในเฟรมถัดๆ ไปเสมอ แต่ใน

สมการ (8) จะพิจารณาจำนวนแท็กที่เกิดการชนกันในระดับสล็อตเพื่อแก้ปัญหาการชนจนกระทั่งสำเร็จ

3.4 Tree algorithm with Adaptive Frame and Skipping Slotted (TA-AFSS)

ในทำนองเดียวกัน อัลกอริทึม TA-AFSS ก็ได้รับการพัฒนาต่อจากอัลกอริทึม TA-AF โดยการเพิ่มเทคนิคการข้ามไม่ใช้งานสล็อตเข้าไปพร้อมกับเทคนิคการปรับขนาดเฟรมได้เอง โดยการเกิดเหตุการณ์ข้ามไม่ใช้งานสล็อตสามารถพิจารณาได้จากอัลกอริทึม AFS-SA ในหัวข้อ 3.2 และรูปที่ 7 ดังนั้นเราจะสามารถแสดงสมการคณิตศาสตร์เพื่อคำนวณหาจำนวนสล็อตโดยเฉลี่ยที่ต้องการใช้ระบุตัวตนของแท็กทั้งหมดได้เป็น

$$\bar{L}_N = \frac{1}{(1 - N^{1-N})} \left[N + \sum_{c=2}^{N-1} \binom{N}{c} \frac{(N-1)^{N-c}}{N^{N-1}} \bar{L}_c - p_{skip} \bar{L}_{skip} \right] \quad (9)$$

เมื่อ $\bar{L}_{skip}$ คือจำนวนสล็อตโดยเฉลี่ยที่ต้องการใช้ระบุตัวตนของแท็กในเหตุการณ์กรณีที่เกิดเหตุการณ์ข้ามไม่ใช้งานสล็อต และ p_{skip} คือความน่าจะเป็นของเหตุการณ์ที่สามารถข้ามไม่ใช้งานสล็อตได้ซึ่งสามารถคำนวณได้จากสมการ (6)

3.5 การวัดประสิทธิภาพการทำงานของอัลกอริทึม

นอกจากการประเมินความสามารถในการทำงานของอัลกอริทึมที่นำเสนอจากจำนวนสล็อตโดยเฉลี่ยที่ต้องการใช้ระบุตัวตนของแท็กที่สามารถลดจำนวนการใช้สล็อตในการแก้ปัญหาการชนกันของแท็กเมื่อพิจารณาที่จำนวนแท็กค่าหนึ่งแล้ว เรายังสามารถประเมินประสิทธิภาพ (Efficiency) การทำงานของอัลกอริทึมได้ในอีกมุมมองหนึ่ง ด้วยการนอร์มอลไลซ์ค่าจำนวนแท็กด้วยนวนการใช้สล็อตในการแก้ปัญหาการชนกันของแท็กซึ่งจะเห็นความสามารถของอัลกอริทึมในภาพรวมทั้งหมด โดยสามารถเขียนอยู่ในรูปสมการคณิตศาสตร์ได้เป็น

$$Efficiency = \frac{N}{\bar{L}_N} \quad (10)$$

4. ผลการทดลอง

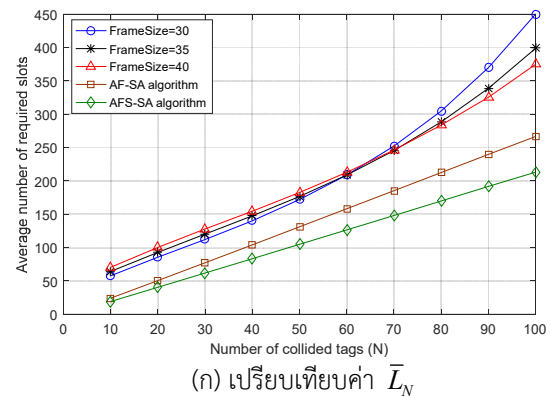
ในหัวข้อนี้จะแสดงผลการทดสอบสมรรถนะของอัลกอริทึมที่นำเสนอและเปรียบเทียบกับอัลกอริทึมบางส่วนที่มีผู้ได้นำเสนอมาแล้ว โดยการเปรียบเทียบจำนวนการใช้สล็อตในการแก้ปัญหาการชนกันของแท็กและค่าประสิทธิภาพของอัลกอริทึม ในการทดลองเราจะกำหนดแกนนอนเป็นค่าจำนวนแท็กโดยเปลี่ยนค่าเป็น 10, 20, 30,..., 100 แท็ก ตามลำดับ

4.1 การทดสอบสมรรถนะของอัลกอริทึม AF-SA และ AFS-SA

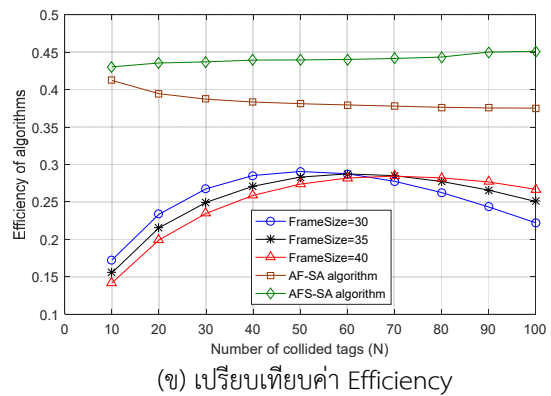
จากรูปที่ 8 จะแสดงผลการเปรียบเทียบค่า $\bar{L}_N$ ของอัลกอริทึมที่เรานำเสนอคือ AF-SA และ AFS-SA รวมถึงการเปรียบเทียบกับอัลกอริทึม FSA ที่มีผู้เสนอมาแล้ว ซึ่งประกอบด้วยกรณีที่กำหนดจำนวนสล็อตต่อเฟรมให้คงที่เป็น 30, 35 และ 40 จากรูป 8(ก) เช่นที่ค่า $N=100$ แท็ก เมื่อ Framesize เท่ากับ 30, 35 และ 40 สล็อตจะต้องใช้สล็อตเป็นจำนวนเฉลี่ยเท่ากับ 450, 400 และ 375 สล็อต เมื่อพิจารณาจะเห็นว่าอัลกอริทึม FSA จะใช้จำนวนสล็อตในการแก้ปัญหาการชนมากกว่าอัลกอริทึมที่เราได้นำเสนอซึ่งมีค่าเท่ากับ 266 และ 213 ตามลำดับ เนื่องจากอัลกอริทึมสามารถปรับขนาดเฟรมได้ตามจำนวนแท็กที่มีอยู่ในเวลานั้น เพราะจำนวนสล็อตต่อเฟรมสำหรับแก้ปัญหาการชนกันจะมีค่าเหมาะสมกับจำนวนแท็กเพียงค่าหนึ่งเท่านั้น ในขณะที่อัลกอริทึมที่นำเสนอคือ AFS-SA ซึ่งมีเทคนิคการข้ามไม่ใช้งานสล็อตจะดีกว่า AF-SA อย่างชัดเจน โดยเฉพาะเมื่อจำนวนแท็กในระบบมีจำนวนมากและจะต้องใช้จำนวนสล็อตในการแก้ปัญหามากขึ้น จึงเกิดเหตุการณ์ที่สามารถข้ามไม่ใช้งานสล็อตมากขึ้นด้วย

จากรูป 8(ข) จะแสดงผลการเปรียบเทียบค่าประสิทธิภาพ (Efficiency) ซึ่งจะเห็นว่าอัลกอริทึม FSA จะให้ค่าประสิทธิภาพสูงขึ้นในช่วงแรกจนมีค่าสูงที่สุดและมีค่าลดลงตามจำนวนแท็กที่เพิ่มมากขึ้นตามลำดับ โดยค่าสูงที่สุดนี้อธิบายได้ว่าจำนวนสล็อตต่อเฟรมสำหรับแก้ปัญหาการชนกันจะมีค่าเหมาะสมกับจำนวนแท็กเพียงค่าหนึ่งเท่านั้น เช่นที่ Framesize=30 สล็อต จะให้ค่า

ประสิทธิภาพสูงสุดเป็น 0.29 เมื่อมีจำนวน $N=50$ แท็ก ส่วนอัลกอริทึมที่นำเสนอคือ AF-SA จะมีประสิทธิภาพลดลงเพียงเล็กน้อยต่อเนื่องไปตลอดตามจำนวนการเพิ่มขึ้นของแท็ก เหตุผลที่ประสิทธิภาพลดลงเนื่องจากการใช้จำนวนสล็อตเพื่อแก้ปัญหาการชนกันมากขึ้น ส่งผลให้เกิดสล็อตที่ไม่ได้งานจำนวนมากขึ้นตามไปด้วย ดังนั้นอัลกอริทึม AFS-SA จึงมีส่วนช่วยลดการใช้งานสล็อตได้จำนวนมาก ส่งผลให้ประสิทธิภาพสูงขึ้นตามจำนวนแท็กที่เพิ่มขึ้นดังรูปนั่นเอง



(ก) เปรียบเทียบค่า $\bar{L}_N$



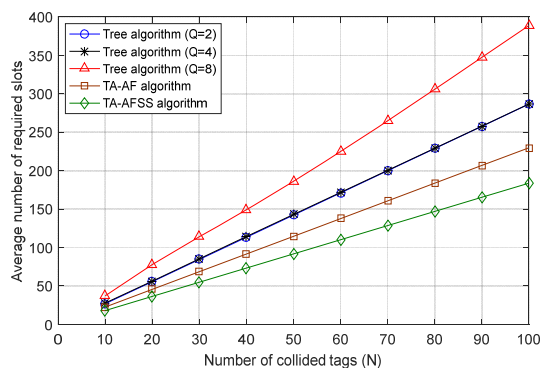
(ข) เปรียบเทียบค่า Efficiency

รูปที่ 8 สมรรถนะของอัลกอริทึม AF-SA และ AFS-SA

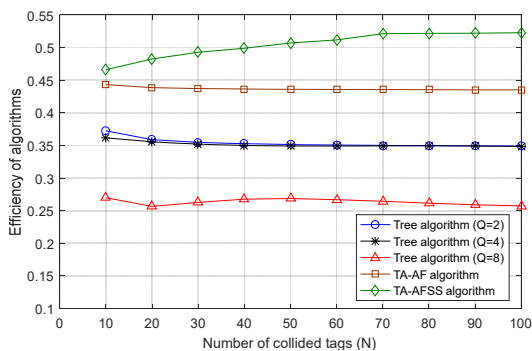
4.2 การทดสอบสมรรถนะของอัลกอริทึม TA-AF และ TA-AFSS

จากรูปที่ 9 จะแสดงผลการเปรียบเทียบค่า $\bar{L}_N$ ของอัลกอริทึมที่เรานำเสนอคือ TA-AF และ TA-AFSS รวมถึงการเปรียบเทียบกับอัลกอริทึม Tree algorithm ที่มีผู้เสนอมาแล้วซึ่งประกอบด้วยกรณีที่ Q-ary = 2, 4 และ

8 สล็อต ตามลำดับ เมื่อพิจารณาจากรูปที่ 9(ก) จะเห็นว่า กรณี Q -ary = 8 สล็อต จะใช้สล็อตเพื่อแก้ปัญหาการชนกันมากที่สุดเท่ากับ 389 สล็อต ที่ค่า $N=100$ แท้ก็ ในขณะที่ Q -ary = 2 และ 4 สล็อต มีค่าใกล้เคียงกันมากต้องการใช้สล็อตเป็นจำนวนโดยเฉลี่ยเท่ากับ 286 สล็อต โดยอัลกอริทึมที่นำเสนอคือ TA-AF และ TA-AFSS จะใช้จำนวนสล็อตน้อยลงเป็น 230 และ 184 สล็อต ตามลำดับ เนื่องจากเทคนิคการปรับขนาดเฟรมและเทคนิคการข้ามไม่ใช้งานสล็อต



(ก) การเปรียบเทียบค่า $\bar{L}_N$



(ข) การเปรียบเทียบค่าประสิทธิภาพ

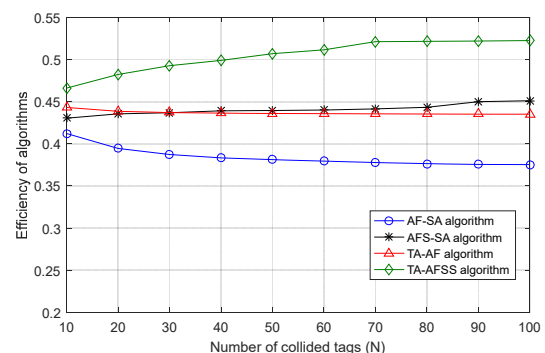
รูปที่ 9 สมรรถนะของอัลกอริทึม TA-AF และ TA-AFSS

จากรูปที่ 9(ข) จะแสดงผลการเปรียบเทียบค่าประสิทธิภาพ โดยภาพรวมจะเห็นว่าอัลกอริทึม TA-AF และ TA-AFSS จะให้ค่าประสิทธิภาพค่อนข้างคงที่ถึงแม้ว่าจำนวนแท็กจะเพิ่มมากขึ้น (ซึ่งแตกต่างจาก AFS-SA และ AFS-SA อย่างชัดเจน) และมีความสอดคล้องกับจำนวนการใช้สล็อตในรูปที่ 9(ก) โดยเรียงลำดับ

ประสิทธิภาพของอัลกอริทึมจากสูงสุดจนถึงต่ำสุดตามลำดับคือ TA-AFSS, TA-AF, $Q = 2, 4$ และ 8 โดยเมื่อพิจารณาที่ค่า $N=100$ แท้ก็ จะมีค่าประสิทธิภาพเท่ากับ 0.52, 0.43, 0.35, 0.35 และ 0.26 ตามลำดับ

4.3 ประสิทธิภาพของอัลกอริทึมทั้งหมดที่นำเสนอ

จากรูปที่ 10 จะแสดงการเปรียบเทียบประสิทธิภาพของอัลกอริทึมทั้งหมดที่นำเสนอในงานวิจัย เมื่อพิจารณาจากรูปจะพบว่าในภาพรวมจะมีประสิทธิภาพเปลี่ยนแปลงเล็กน้อยถึงแม้ว่าจะมีจำนวนแท็กในระบบเพิ่มมากขึ้น ซึ่งแสดงให้เห็นถึงข้อดีว่าอัลกอริทึมที่นำเสนอทั้งหมดนี้มีประสิทธิภาพในการแก้ปัญหาการชนได้ดีในทุกๆ สถานการณ์ โดยเฉพาะอย่างยิ่ง อัลกอริทึม TA-AFSS ที่มีเทคนิคการข้ามไม่ใช้งานสล็อตจะช่วยลดจำนวนการใช้สล็อตได้จำนวนมากและมีผลต่อการเพิ่มประสิทธิภาพของระบบอย่างชัดเจน นอกจากนั้นผลลัพธ์นี้ยังแสดงให้เห็นว่าเทคนิคการข้ามไม่ใช้งานสล็อตสามารถช่วยอัลกอริทึม AFS-SA ให้มีประสิทธิภาพสูงขึ้นเทียบเท่ากับ TA-AF ได้ ในกรณีที่แท็กในระบบมีจำนวนมากๆ ทั้งนี้เมื่อพิจารณาที่ค่า $N=100$ แท้ก็ เมื่อเรียงลำดับประสิทธิภาพของอัลกอริทึมทั้งหมดจากสูงสุดจนถึงต่ำสุดตามลำดับคือ TA-AFSS, AFS-SA, TA-AF และ AF-SA จะมีค่าประสิทธิภาพเท่ากับ 0.52, 0.45, 0.43 และ 0.37 ตามลำดับ



รูปที่ 10 ประสิทธิภาพของอัลกอริทึมที่นำเสนอทั้งหมด

5. สรุป

งานวิจัยฉบับนี้นำเสนอการวิเคราะห์สมรรถนะของอัลกอริทึมต่อต้านการชนกันสำหรับระบบอาร์เอฟไอดีซึ่ง

ประกอบด้วยอัลกอริทึม AF-SA, AFS-SA, TA-AF และ TA-AFSS โดยอาศัยเทคนิคการปรับขนาดเฟรมได้เองและเทคนิคการข้ามไม่ใช้งานสล็อต เราได้นำเสนอสมการคณิตศาสตร์เพื่อประเมินประสิทธิภาพการทำงานของอัลกอริทึมโดยพิจารณาจากจำนวนสล็อตโดยเฉลี่ยที่ต้องการใช้ระบุตัวตนของแท็กทั้งหมดและค่าประสิทธิภาพการทำงานของอัลกอริทึม

จากผลการทดสอบพบว่าอัลกอริทึมทั้งหมดที่นำเสนอภาพรวมจะมีประสิทธิภาพเปลี่ยนแปลงเล็กน้อยถึงแม้ว่าจะมีจำนวนแท็กในระบบเพิ่มมากขึ้น ซึ่งแสดงให้เห็นถึงข้อดีว่าอัลกอริทึมที่นำเสนอทั้งหมดนี้มีเสถียรภาพในการแก้ปัญหาการชนได้ดีในทุกๆ สถานการณ์ โดยอัลกอริทึม TA-AFSS มีประสิทธิภาพสูงสุด ในขณะที่ประสิทธิภาพของอัลกอริทึม TA-AF มากกว่า AFS-SA เล็กน้อยเมื่อมีแท็กที่เกิดการชนในระบบจำนวนมากๆ ส่วนอัลกอริทึม AF-SA มีประสิทธิภาพต่ำที่สุด นอกจากนั้นยังสรุปได้ว่าเทคนิคการข้ามไม่ใช้งานสล็อตส่งผลให้ประสิทธิภาพของระบบสูงขึ้นมากกว่าเทคนิคปรับขนาดเฟรมได้เอง

6. กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณคณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา ที่ได้สนับสนุนทุนอุดหนุนการวิจัยและพัฒนา สัญญาเลขที่ วจพ. 6/2560

7. เอกสารอ้างอิง

[1] K. Finkenzeller, *RFID Handbook: Fundamental and applications in contactless smart cards and radio frequency identification, near-field communication*. The 3rd Edition, John Wiley and Sons, 2010.

[2] D. K. Klair, K. W. Chin, R. Raad, "A survey and tutorial of rfid anti-collision protocols," *IEEE Commu. Surveys & Tutorial*, vol. 12, no. 3, pp. 400-421, 2010.

[3] D. H. Shih, P. L. Sun, D. C. Yen, and S. M. Huang, "Taxonomy and survey of rfid anti-

collision protocols: short survey," *Computer Commu.*, vol. 29, no. 11, 2006, pp. 2150–2166.

- [4] Z. Tang and Y. He, "Research of multi-access and anti-collision protocols in rfid systems," in *IEEE International Workshop on Anti-counterfeiting, Security, Identification*. Xiamen, China, 2007, pp. 377–380.
- [5] D. K. Klair, K.-W. Chin, and R. Raad, "An investigation into the energy efficiency of pure and slotted Aloha based rfid anti-collision protocols," in *IEEE WoWMoM*, Helsinki, Finland, 2007.
- [6] H. Wu and Y. Zeng, "Efficient framed slotted aloha protocol for rfid tag anticollision," *IEEE Trans. Automation Science and Eng.*, vol. 8, no. 3, pp. 581-588, 2011.
- [7] C. F. Lin and F. Y. S. Lin, "Efficient estimation and collision-group-based anticollision algorithms for dynamic frame-slotted aloha in rfid networks," *IEEE Trans. Automation Sci. and Eng.*, vol. 7, no. 4, pp. 840–48, Oct. 2010.
- [8] H. Wu, Y. Zeng, J. Feng, and Y. Gu, "Binary tree slotted aloha for passive rfid tag anti-collision," *IEEE Trans. Parallel Distrib. Syst.*, vol. 1, pp. 19–31, 2013.
- [9] J. Shin, B. Jeon, and D. Yang, "Multiple rfid tags identification with M-ary query tree scheme," *IEEE Comm. Letters*, vol. 17, no. 3, 2013.
- [10] X. Jia, Q. Feng, and C. Ma, "An efficient anti-collision protocol for rfid tag identification," *IEEE Communication Letter*, vol. 14, no. 11, pp. 1014–1016, 2010.

- [11] J. H. Choi, D. Lee, and H. Lee, “Query tree-based reservation for efficient rfid tag anti-collision,” *IEEE Communication Letter*, vol. 11, no. 1, pp. 85–87, 2007.
- [12] J. B. Eom and T. J. Lee, “Accurate tag estimation for dynamic framed-slotted aloha in rfid systems,” *IEEE Communication Letter*, vol. 14, no. 1, pp. 60–62, 2010.
- [13] W.T. Chen, “An accurate tag estimate method for improving the performance of a rfid anti-collision algorithm based on dynamic frame length aloha,” *IEEE Trans. Automation Science and Eng.*, vol. 6, no. 1, pp. 9-15, 2009.
- [14] H. Wu and Y. Zeng, “Bayesian tag estimate and optimal frame length for anti-collision aloha rfid system,” *IEEE Trans. Automation Science and Eng.*, vol. 7, no. 4, pp. 963-969, 2010.
- [15] T. F. La Porta, G. Maselli, and C. Petrioli, “Anti-collision protocols for single-reader rfid systems: temporal analysis and optimization,” *IEEE Trans. Mobile Computing*, vol. 10, no. 2, pp. 267-279, 2011.
- [16] Y. C. Lai, L. Y. Hsiao, and B. S. Lin., “Optimal slot assignment for binary tracking tree protocol in rfid tag identification,” *IEEE/ACM Trans. on Networking*, vol. 23, no. 1, pp. 255–268, 2015.