

ระบบจำลองการทำงานของระบบบิทคอยน์ที่สามารถต่อขยายได้

Extensible Simulation Framework for Bitcoin

ณัฐพล ธนิตสุขการ และ ภารุจ รัตนวรพันธุ์
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
Email: nuttapon.t@ku.th, paruj.r@ku.th

I. บทนำ

ในยุคปัจจุบันเรามีการแลกเปลี่ยนสินค้า ซื้อของออนไลน์กันมากขึ้น แต่การโอนเงินข้ามประเทศหรือทำธุรกรรมข้ามประเทศนั้น มีขั้นตอนต้องยุ่งยากเยอะมาก จนในที่สุดท้ายแล้วได้เกิดระบบการแลกเปลี่ยนเงินออนไลน์แบบใหม่ขึ้นมา นั่นก็คือระบบสกุลเงินดิจิทัลที่พัฒนามาจากบล็อกเชนนั่นเอง ซึ่งบิทคอยน์หรือสกุลเงินดิจิทัลแรกของโลกนั้น ถูกสร้างขึ้นมาจากภาษาคอมพิวเตอร์ ไม่มีใครเป็นเจ้าของบิทคอยน์ ไม่มีรูปร่าง และไม่สามารถจับต้องได้ โดยระบบของบิทคอยน์ ถูกรันโดยคอมพิวเตอร์ของผู้ใช้งานทั่วโลก โดยใช้ระบบซอฟต์แวร์ในการถอดสมการคณิตศาสตร์ โดยคุณสมบัติสำคัญของ บิทคอยน์คือ โปร่งใส 100% กล่าวคือสามารถตรวจสอบรายการย้อนหลังได้ อีกทั้งยังมีความรวดเร็วและความปลอดภัยสูงมากอีกด้วย ระบบบิทคอยน์ประสบความสำเร็จอย่างมากในทางปฏิบัติ แต่ขาดทฤษฎีที่เข้มแข็งมารองรับทำให้การทำงานมีปัญหาที่จะเกิดขึ้นกับระบบทำได้ยาก ดังนั้นโครงการนี้จึงมีส่วนช่วยอย่างมากในการจำลองการทำงานภายในของระบบบิทคอยน์ สามารถเห็นถึงโครงสร้างและปรากฏการณ์ต่างๆ ทำให้ง่ายต่อการศึกษาและพยากรณ์พฤติกรรมของบิทคอยน์ในอนาคต

• วัตถุประสงค์ของการศึกษา

- เพื่อให้บัณฑิตมหาวิทยาลัยเกษตรศาสตร์ได้ทดลองใช้ระบบการแลกเปลี่ยนแบบสกุลเงินดิจิทัลที่กำลังเป็นที่นิยม ณ ปัจจุบัน
- เป็นระบบสกุลเงินดิจิทัลที่เข้าใจง่ายและใช้งานง่าย
- สามารถใช้โหนดหลายๆโหนดมาช่วยกัน mine ช่วยกันต่อบล็อกเชนได้อย่างถูกต้อง
- มีการทำงานที่รวดเร็ว เนื่องจากเขียนด้วยภาษา Golang ซึ่งเหมาะสมอย่างมากในการทำงานในระบบแบบกระจาย (distributed system)
- เป็นฐานโค้ด (codebase) ที่มีความกระชับและปรับแต่งได้ง่าย กล่าวคือสามารถนำโมดูล (module) อื่นๆ เข้ามาต่อเข้าเพื่อขยายขอบเขตการทำงานได้ง่าย

• ขอบเขตของการทำโครงการ

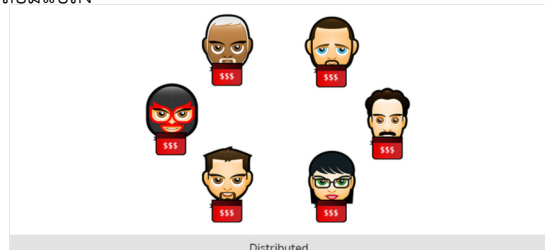
- สามารถทำการซื้อขายได้อย่างถูกต้อง เช่น การโอนเงิน
- สามารถใช้เครื่องคอมพิวเตอร์มาขุด (Mining) เพื่อช่วยระบบยืนยันการทำรายการได้อย่างถูกต้อง

- สามารถใช้งานได้จริงในชีวิตประจำวัน
- ตัวระบบต้องมีความปลอดภัยสูง
- ประโยชน์ที่คาดว่าจะได้รับ
 - ได้ระบบสกุลเงินดิจิทัลที่ทำงานรวดเร็ว มีความเรียบง่าย และคล้ายคลึงกับระบบบิทคอยน์ที่ผู้คนทั่วโลกใช้งานอยู่ในขณะนี้
 - สามารถใช้งานได้จริงในมหาวิทยาลัยเกษตรศาสตร์ เพื่อผลิตจะได้สร้างความเคยชินกับระบบสกุลเงินเข้ารหัสที่จะเผชิญในอนาคต
 - สามารถเป็นแบบอย่างให้แก่ผู้ที่สนใจในด้านบล็อกเชนและสกุลเงินเข้ารหัสนำไปต่อยอดต่อไปได้
 - ได้นำความรู้หลากหลายสาขามาประยุกต์ใช้งานเข้าด้วยกัน ทั้งทางด้านวิทยาการเข้ารหัสลับ (Cryptography), ด้านเน็ตเวิร์กและด้านระบบการทำงานแบบกระจาย

II. ทฤษฎีที่เกี่ยวข้อง

A. บล็อกเชน (Blockchain)

บล็อกเชนคือรูปแบบการเก็บข้อมูลหรือเรียกว่าเป็นฐานข้อมูลแบบหนึ่งของระบบที่ไม่มีศูนย์กลางแต่เชื่อถือได้และโยงยาก กล่าวคือการทำให้ทุกคนถือเอกสารชุดเดียวกันคนละก๊อปปี้ด้วยวิธีแบบเพียร์ทูเพียร์ (Peer-to-Peer) ก็คือไม่ต้องมีคนตรงกลาง ทุกคนที่อยู่บนระบบเน็ตเวิร์กช่วยกันรันระบบ ซึ่งทำให้ระบบไม่มีทางล่มตราบดีที่ยังมีคนอยู่ในระบบเน็ตเวิร์กอยู่ ข้อมูลก็ไม่หายสาบสูญเพราะทุกคนช่วยกันถือไว้คนละชุด หากข้อมูลมีการเพิ่มขึ้นหรือมีบล็อกใหม่ ทุกคนในระบบเน็ตเวิร์กก็จะได้ข้อมูลใหม่ไปเพิ่มฐานข้อมูลในมือตัวเองด้วยกันทุกคน เมื่อมีการอัปเดตก็จะอัปเดตด้วยกัน โดยมั่นใจได้ว่าเอกสารเหล่านั้นเชื่อถือได้แน่นอนไม่มีการปลอมแปลง

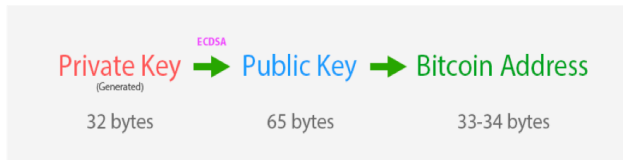


ภาพที่ 1: ทุกคนถือข้อมูลของทั้งระบบเอาไว้คนละก๊อปปี้

B. ศาสตร์การเข้ารหัสลับ (Cryptography)

เป็นสกุลเงินดิจิทัลที่ถูกพัฒนามาโดยหลักการสำคัญทาง Cryptography 2 หลักการหลักๆ คือ วิทยาการเข้ารหัสลับแบบคีย์สาธารณะ (public-key cryptography) และ แฮชฟังก์ชัน (hash function) วิทยาการเข้ารหัสลับแบบคีย์สาธารณะ ถูกนำมาใช้ทำเป็นกระเป๋าเงินและ อนุญาตรับเงิน คือ คีย์สาธารณะ (public key) และ คีย์ส่วนตัว (private key) ตามลำดับ แฮชฟังก์ชันหรือฟังก์ชันที่รับอินพุตขนาดใดๆ เข้ามาและให้อาต์พุตออกไปเป็นข้อความที่มีความยาวคงที่ เช่น 256 บิต ซึ่งมีคุณสมบัติสำคัญ 3 ประการที่จะถูกนำมาใช้ในเรื่องนี้ คือ

- 1) One-way function กล่าวคือ เราไม่สามารถทำย้อนกลับจากเอาต์พุตไปหาอินพุตได้
- 2) Hiding property กล่าวคือ เราไม่สามารถรู้ส่วนใดส่วนหนึ่งของอินพุตได้จากเอาต์พุต
- 3) Puzzle friendly กล่าวคือ เราต้องใช้การออกแรง (brute force) เท่านั้น เพื่อที่จะได้มาซึ่งอินพุตที่เข้ากับเอาต์พุตที่เราต้องการ

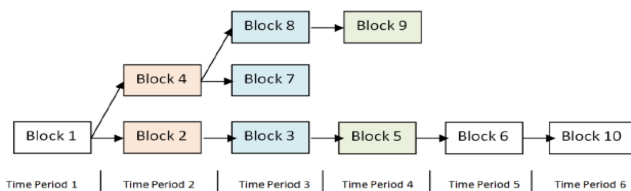


ภาพที่ 2: การใช้วิทยาการเข้ารหัสลับแบบคีย์สาธารณะในระบบสกุลเงินดิจิทัล

C. คอนเซนซัสแบบกระจาย (Distributed Consensus)

เป็นโปรโตคอลที่ใช้เพื่อการทำงานแบบกระจาย (decentralization) โดยการใช้เครือข่ายเพียร์ทูเพียร์ ให้แก่อาสาสมัครที่จะรับหน้าที่เก็บข้อมูลบัญชี (ledger) หรือบัญชีต่างๆบล็อกเชน โดยคอนเซนซัสแบบกระจาย (distributed consensus) มีวัตถุประสงค์หลักที่จะกระจายข้อมูลที่ต้องไปยังโหนดต่างๆ และทำให้แน่ใจว่าทุกโหนดบนเน็ตเวิร์คบล็อกเชน มีรายการข้อมูลที่ต้องและเหมือนกัน โดยมีหลักการการทำงาน หลักๆดังนี้

- 1) คอยสอดส่องและสะสมรายการ ใหม่ๆที่ถูกปล่อยออกมา
- 2) ตรวจสอบความถูกต้องของรายการนั้นๆ
- 3) ส่งต่อรายการให้โหนด (node) ใกล้เคียง



ภาพที่ 3: ทุกคนจะยอมรับข้อมูลที่ต้องและเหมือนกันในที่สุด

III. รายละเอียดการพัฒนา

A. การออกแบบระบบ

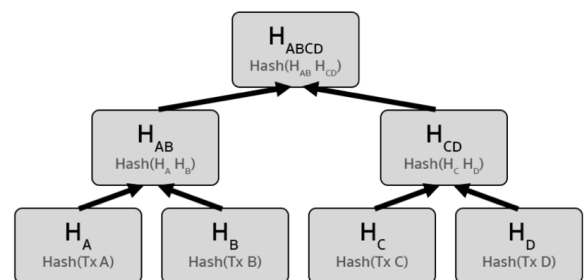
ออกแบบ header fields ของระบบ มีขนาด 80 bytes ประกอบด้วย

- 1) version 4 bytes: เป็นตัวบอกว่าบล็อกนี้ ต้องทำตามกฎการยืนยันบล็อกแบบไหน
- 2) previous block hash: ค่าแฮชของบล็อกก่อนหน้า
- 3) merkle root hash: เป็นการนำค่าแฮชของต้นไม้ merkle ที่เก็บแฮชของ transaction ทั้งหมดในบล็อกนั้นไว้
- 4) time: เป็นเวลาที่อยู่ในรูปของ unix timestamp ที่บล็อกนั้นๆ เกิดขึ้นมา
- 5) target bits: เป็นตัวกำหนดความยากของการ mine บล็อกนั้น ว่าต้องการค่าแฮชที่ต่ำกว่า target เท่าไหร่
- 6) nonce: เป็นตัวเลขใดๆ ที่ถูกสุ่มขึ้นมาเพื่อให้แฮชรวมทั้งบล็อกต่ำกว่าค่า target ที่กำหนดไว้

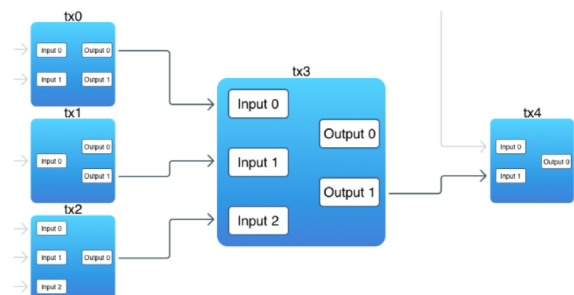
โดยแต่ละ block จะบรรจุ transaction หลายๆอันอยู่ภายใน ซึ่ง transaction แต่ละอันจะต้องระบุ input และ output เพื่อที่จะบอกว่าอ้างอิงเงินจาก transaction อะไร แล้วจะส่งเงินไปที่ transaction ไหนต่อไป

Block Header	version	4 bytes
	prev_block_hash	32 bytes
	merkle_root_hash	32 bytes
	time	4 bytes
	bits	4 bytes
	nonce	4 bytes
Transactions	Coinbase Transaction	
	Transaction 1	
	Transaction 2	
	...	
	...	
block_hash		32 bytes

ภาพที่ 4: โครงสร้างของบล็อก

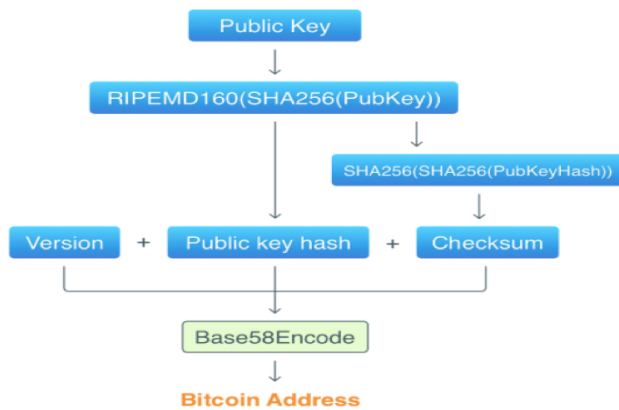


ภาพที่ 5: โครงสร้างของต้นไม้ merkle root



ภาพที่ 6: การอ้างอิง transaction input จาก transaction output ก่อนหน้า

โดยใช้หลักการของศาสตร์การเข้ารหัสลับที่ชื่อว่า Elliptic Curve Cryptography (ECDSA) ซึ่งเป็นหนึ่งในศาสตร์การเข้ารหัสโดยใช้คีย์สาธารณะ (Public-key Cryptography) เป็นตัวรองรับความปลอดภัยของระบบ ECDSA จะสร้าง public key และ private key ที่เข้าคู่กันขนาด 512 bytes โดยมี public key เป็น Bitcoin address ซึ่งสามารถให้ผู้ใช้อื่น โอนเงินมาให้เราผ่านทางที่อยู่นี้ได้ และเก็บ private key ไว้เป็นตัวแทนความเป็นเจ้าของสมุดเล่มนี้ เพื่อใช้ในการถอนเงินหรือทำธุรกรรมต่างๆได้แต่เพียงผู้เดียว



ภาพที่ 7: ขั้นตอนการสร้าง Bitcoin address จาก public key

การที่ผู้ใช้งานสามารถส่งต่อข้อมูลไปให้ผู้ใช้อื่นในระบบได้ผ่านการทำงานแบบเพียร์ทูเพียร์ (Peer-to-Peer) ได้มีการทำ cluster node membership, status dissemination และ failure detection เพื่อทำให้ระบบเพียร์ทูเพียร์เราเสถียรและป้องกันทันท่วงทีต่อความผิดพลาดได้ นอกจากนี้ยังทำ Atomic broadcast ด้วย TCP แทนที่จะใช้การ Broadcast/Multicast แบบปกติที่เป็น UDP ที่มีข้อเสียด้าน UDP Buffer Size ที่จะไม่รับรองการส่งข้อมูล เมื่อส่งข้อมูลขนาดใหญ่กว่า 512 bytes

เมื่อผู้ใช้เยอะขึ้นมากๆ การส่งต่อ transaction, การส่งต่อบล็อกย่อมก่อให้เกิดความไม่สอดคล้องกันได้ง่ายขึ้น เช่นบางกรณีนั้น ได้เกิดการแยกกันของบล็อกเป็นสองสาย ซึ่งสุดท้ายจะถูกมองเป็นสายเดียวโดยยึดนโยบาย longest chain ตามบิทคอยน์โดย consensus Proof-of-Work ที่เราเลือกจะเป็นตัวการหลักในการจัดการปัญหานี้

B. รายละเอียดของระบบ

- 1) Core: ตัวส่วนหลักของระบบซึ่งมีพื้นฐานอยู่บนระบบเดียวกันกับ Bitcoin ซึ่งภายในจะมี public ledger เป็น blocks ต่อการเป็นสาย และในแต่ละ block จะบรรจุไปด้วย transaction ที่เก็บธุรกรรมของทุกคนในระบบที่ผ่านการยืนยันมาแล้วเอาไว้
- 2) Proof-of-Work: ระบบมีการประยุกต์โปรโตคอลคอนเซนซัสหลากหลายรูปแบบ เพื่อทดสอบหาระบบที่ดีที่สุดที่จะนำมาใช้ในระบบจริงของมหาวิทยาลัยเกษตรศาสตร์ได้ เช่น Proof-of-Stake แต่เนื่องด้วยยังไม่มีจัดการปัญหา Nothing-at-Stake และ LongRange-Attack ได้ จึงยังไม่ควรนำมาใช้จริงในระบบ สุดท้ายจึงได้เลือกใช้ตัว Proof-of-Work แบบเดียวกันกับ Bitcoin ซึ่ง

อาจจะมมีปัญหาการกินทรัพยากรบ้าง แต่ก็เป็น consensus ที่เสถียรและนิยมกันมากที่สุดในขณะนี้

- 3) Peer-to-Peer: มีการใช้งานระบบเพียร์ทูเพียร์ในรูปแบบของโปรโตคอลข่าวลือ (Gossip Protocol) ซึ่งเป็นวิธีการที่มีประสิทธิภาพมากที่สุดวิธีหนึ่งที่มักจะถูกนำมาใช้ในระบบแบบกระจาย โดยในระบบได้มีการนำมาใช้เพื่อเป็นการสื่อสารกันระหว่างโหนด ให้โหนดได้คุยกัน เช่น ทำการกระจายหรือแลกเปลี่ยน transaction และ block กับโหนดที่อยู่ติดต่อกัน นอกจากโหนดต่างๆ จะต้องมีการทำส่งข่าวลือต่อกันแล้ว โหนดจำเป็นต้องมีการแพร่กระจายของสถานะโหนดข้างเคียงและแจ้งให้โหนดอื่นๆรู้ด้วย เราเรียกเหตุการณ์เช่นนี้ว่า status dissemination และโหนดต่างๆเหล่านี้มีโอกาสที่จะดับไปหรือมีโอกาสที่จะเป็นโหนดชั่วร้ายได้อีกด้วย เราจึงจำเป็นต้องมีวิธีการจัดการในด้านนี้หรือที่เรียกว่า failure detection ในโปรโตคอลอีกด้วย

C. ขั้นตอนการพัฒนา

- 1) ศึกษาการทำงานของระบบเหรียญดิจิทัล โดยศึกษาพื้นฐานต่างๆจาก e-book ของ Prince-ton University คือ Bitcoin and cryptocurrency technology ซึ่งมีหัวข้อสำคัญต่างๆ ดังนี้
 - a) เหตุผลที่เกิดเหรียญดิจิทัลขึ้นมาเพราะต้องการแก้ปัญหาการโอนเงินข้ามประเทศที่มีความล่าช้าเพราะขั้นตอนที่ยุ่งยากหรือมีการเก็บค่าธรรมเนียมที่มากเกินไป
 - b) การทำให้ทั่วโลกใช้เงินในสกุลเดียวกันเพื่อที่ในอนาคตจะเป็นโลกที่ไร้พรมแดนที่แท้จริง
 - c) หลักการเข้าข้อมูลและการนำมาใช้ในการสร้างเหรียญดิจิทัล เพื่อให้มั่นใจว่าเราสามารถเชื่อถือในตัวเหรียญได้ว่ามีความปลอดภัยและมีความมั่นคงสูง
 - d) เหรียญดิจิทัลตัวแรกของโลกซึ่งก็คือบิทคอยน์ ว่ามีโครงสร้างอย่างไรและทำอย่างไรถึงสามารถโอนเงินไปมาข้ามโลกได้ โดยที่ไม่จำเป็นต้องมีศูนย์กลางของระบบ
 - e) แมคคานิคซึมหลักของบิทคอยน์ว่ามีการทำงานอย่างไร ทั้งการทำ transaction และการต่อบล็อก
- 2) ศึกษาโปรโตคอลคอนเซนซัสที่ใช้ในการทำงานด้านระบบแบบกระจายหลายๆตัว เพื่อที่จะสามารถนำมาพิจารณาเลือกใช้ เช่น Proof of Work, Proof of Stake, Proof of Importance, Proof of Authority, Proof of Elapsed Time, Proof of Burn, Proof of Capacity, Practical Byzantine Fault Tolerance, Tendermint Core, Loop Fault Tolerance, Delegated Byzantine Fault Tolerance, Federated Byzantine Agreement, Paxos Consensus และ Raft Consensus
- 3) ศึกษาจุดประสงค์และจุดเด่นของเหรียญดิจิทัลหลายๆสกุล เพื่อที่จะสามารถนำมาประยุกต์ใช้กับโครงการ โดยมุ่งเน้นไปที่ความเร็วในการโอน, ความเป็นส่วนตัวของผู้ใช้, จำนวนบล็อกที่สามารถต่อได้ นอกจากนี้แต่ละเหรียญยังมีจุดประสงค์ที่แตกต่างกัน เช่น Rip-

ple ร่วมมือกับธนาคารหลายแห่งทั่วโลก เพื่อที่จะต้องการเป็น Financial coin หรือ ZCash coin ที่เน้นที่ความเป็นส่วนตัวของผู้ใช้ หรือ ไม่สามารถตรวจสอบได้ และยังมีเหรียญที่เป็น Assets ที่ไป Contact กับเหรียญอื่น เช่น Ethereum , Waves, NEO เป็นต้น

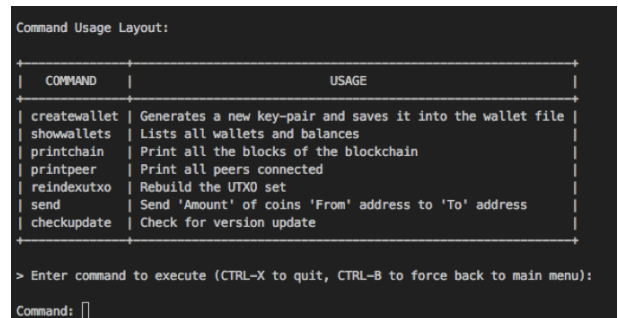
- 4) ศึกษาข้อดี-ข้อเสียของภาษาต่างๆ และดูความเหมาะสมของภาษาที่จะเอามาใช้งานจริง สุดท้ายจึงได้เลือกใช้ภาษา Go เพราะมีข้อดีที่เหมาะสมกับการทำ blockchain programming ในหลายๆ ด้าน เช่น
 - a) โค้ดที่เขียนในภาษา Go สามารถเข้าใจได้ง่ายและมีความสะอาด
 - b) เร็วและมีประสิทธิภาพสูง ไม่เหมือน Python ที่เป็น interpreted language แต่ Go เป็น compiled language เหมือนกับ C ซึ่งมีจุดเด่นเรื่องความเร็วเพราะมี over-head ตอนจัดการข้อผิดพลาดประเภท on-the-fly ได้ดี
 - c) ถูกออกแบบมาเพื่อทำงานด้านระบบแบบกระจายได้ดี โดยสามารถเห็นได้จากซอฟต์แวร์ต่างๆ หลายตัว เช่น Docker และ MongoDB ซึ่งถูกเขียนขึ้นมาจากภาษา Go แม้กระทั่ง Codebase ของ Ethereum และ Hyperledger ก็ถูกเขียนด้วยภาษา Go เช่นกัน
 - d) Goroutines ที่สามารถทำงานด้านการเห็นพ้อง (concurrency) ได้อย่างมีประสิทธิภาพสูงสุด เนื่องจากการทำเทร็ด (thread) ด้วยภาษาโปรแกรมมิ่งอื่นๆ 1 เทร็ด อาจใช้แรมมากที่สุดถึง 1024 กิโลไบต์ แต่การใช้ Goroutines จะใช้แรมมากที่สุดไม่เกิน 4 กิโลไบต์
- 5) ทำความเข้าใจเรื่องการโปรแกรมมิ่งบิทคอยน์จาก 3 assignments ที่มาจาก Bitcoin and Cryptocurrency Technology ของ Princeton University คือ
 - a) Scrooge Coin assignment จะให้สร้างการจัดการธุรกรรม (transaction handler) เพื่อให้สามารถยืนยันความถูกต้องของการทำธุรกรรม, หรือตรวจจับการทำธุรกรรมที่ไม่ถูกต้อง เช่น double spending attack
 - b) Consensus from Trust assignment โดยงานนี้เราจะได้รับกราฟความน่าเชื่อถือของโหนดแต่ละโหนดในระบบเน็ตเวิร์กมา โดยเรามีหน้าที่ต้องตรวจจับโหนดชั่วร้ายที่มีอยู่ในระบบ
 - c) Block Chain assignment เราจะต้องจัดการกับธุรกรรม โดยการนำไปเข้าบล็อก แล้วต่อไปอยู่บนบล็อกเชนจำลองได้
- 6) ศึกษาตัวโค้ดหลักจาก <https://github.com/bitcoin/bitcoin> ซึ่งเป็นโค้ดจากระบบ บิทคอยน์หลัก เขียนด้วยภาษา C++ และบล็อกโพสของ Jeiwon เป็นแนวทางในการเขียนและออกแบบในรูปแบบของภาษา Go
- 7) ทำการทดสอบระบบทั้งหมด
- 8) จัดทำเอกสารโครงการ

IV. ผลการพัฒนาโครงการ

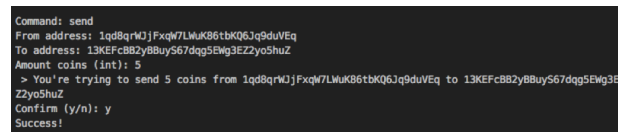
A. ผลการออกแบบและทดสอบระบบ

- 1) ในส่วนของ Core จะได้หน้าต่างผู้ใช้งานที่สามารถทำหน้าที่ได้ครบถ้วนและเพียงพอต่อการใช้งาน

- createwallet ใช้สร้าง address ของกระเป๋าเงินใหม่
- showwallets ใช้สำหรับแสดงกระเป๋าเงินทั้งหมด
- printchain แสดงสถานะบล็อกเชนตั้งแต่ต้นจนถึงปัจจุบัน
- printpeer แสดง neighbor nodes
- reindexutxo จัดอินเด็กซ์การเก็บ unspent transaction output ใหม่
- send ทำการโอนเงินไปให้ address อื่นๆ
- checkupdate เป็นการ manual update สถานะของบล็อก



ภาพที่ 8: หน้า User Interface



ภาพที่ 9: ทดสอบการโอนเงิน

- 2) ในส่วนของ Web Visualization สามารถแสดงผลสถานะบล็อกปัจจุบันและรายละเอียดของแต่ละบล็อกได้เป็น tree ที่สวยงาม



ภาพที่ 10: หน้า Web Visualization

B. วิธีวัดผลและประเมินผล

การวัดและประเมินผล จะทดลองโดยการทำธุรกรรมจำนวนมหาศาล และมีโหนดที่ทำกรยืนยัน transaction อยู่เยอะมากๆ เพื่อทดสอบดูว่าระบบจะเข้าสู่ consensus ได้ในที่สุด

V. ผลการดำเนินโครงการและวิจารณ์

A. สรุปผลการพัฒนาโครงการ

ทั้งในส่วน Core และ Web Visualization สามารถทำงานและแสดงผลออกมาได้อย่างถูกต้อง

สามารถทำการซื้อขายได้อย่างถูกต้อง เช่น การโอนเงิน	[ผ่าน]
ระบบจะเข้าสู่คอนเซนซัสในที่สุด	[ผ่าน]
สามารถมีคนยืนยัน transaction ได้หลายคน	[ผ่าน]
ระบบมีความปลอดภัยจากการเกิด double spending	[ผ่าน]
ผู้ใช้ใหม่สามารถเข้าร่วมได้ตลอดเวลา	[ผ่าน]
มีเว็บไซต์สถานะบล็อกเชนของแต่ละผู้ใช้งาน	[ผ่าน]
โค้ดมีความกระชับ, สามารถเข้าใจและต่อยอดได้ง่าย	[ผ่าน]
โครงการสามารถนำไปใช้ได้จริง	[ผ่าน]

B. ปัญหาและอุปสรรคในการดำเนินงาน

- 1) ไม่สามารถทำงานให้เสร็จได้ตามแผนที่วางไว้ เนื่องจากงานในบางส่วนกินเวลามากกว่าที่คาดไว้
- 2) มีไลบรารีให้เลือกใช้น้อยมาก เนื่องจากเป็นฟิลด์ที่ยังใหม่และเลือกใช้ภาษาที่มีเครื่องมือสำเร็จรูปยังไม่มากพอ
- 3) ในส่วนของโค้ดหลัก ยังขาดการทำคู่มือการใช้งานเพื่อเพิ่มความง่ายให้นำไปใช้ได้

C. ข้อเสนอแนะในการพัฒนาต่อไป

- 1) สามารถนำไปต่อยอดเป็นระบบจำลองการทำงานเพื่อตรวจจับอัตราการเกิดบล็อกกำพไรในระบบบิทคอยน์ ซึ่งจะมีส่วนช่วยอย่างมากในการทำนายพฤติกรรมบางอย่างของระบบ
- 2) ใช้ระบบการจำลองนี้เป็นโค้ดพื้นฐาน ในการใช้ต่อยอดบนระบบสกุลเงินดิจิทัลอื่นๆ เช่น Ethereum เพื่อทำการจำลองและทำนายพฤติกรรมได้เช่นเดียวกัน

เอกสารอ้างอิง

- [1] Bitcoin and Cryptocurrency technologies, https://d28rh4a8wq0iu5.cloudfront.net/bitcointech/readings/princeton_bitcoin_book.pdf, [สืบค้นเมื่อ มกราคม 2561]
- [2] Understanding Blockchain Consensus Models, <https://www.persistent.com/wp-content/uploads/2017/04/WP-Understanding-Blockchain-Consensus-Models.pdf?pdf=Understanding-Blockchain-Consensus-Models>, [สืบค้นเมื่อ มกราคม 2561]
- [3] Blockchain DIY with Python, <https://clumdee.github.io/blockchain-DIY-with-python>, [สืบค้นเมื่อ มกราคม 2561]
- [4] Blockchain for Geek, https://nuuneoi.com/blog/blog.php?read_id=900, [สืบค้นเมื่อ กุมภาพันธ์ 2561]
- [5] Go by Example, <https://gobyexample.com>, [สืบค้นเมื่อ กุมภาพันธ์ 2561]
- [6] Blockchain programming with Go, <https://jeiwan.cc>, [สืบค้นเมื่อ กุมภาพันธ์ 2561]
- [7] Consensus protocol for Blockchain, https://nuuneoi.com/blog/blog.php?read_id=933, [สืบค้นเมื่อ มีนาคม 2561]