การพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่

โดย

นายสมเกียรติ ดอนทองแดง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปรัชญาดุษฎีบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ แบบ 1.1 ระดับปริญญาดุษฎีบัณฑิต

ภาควิชาคอมพิวเตอร์

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2562

การพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่

โดย
นายสมเกียรติ ดอนทองแดง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปรัชญาดุษฎีบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ แบบ 1.1 ระดับปริญญาดุษฎีบัณฑิต
ภาควิชาคอมพิวเตอร์
บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร
ปีการศึกษา 2562

PROTOTYPE DEVELOPMENT OF COMPLEX NETWORK ANALYTIC SYSTEM USING BIG DATA

By

MR. Somkiat DONTONGDANG

A Thesis Submitted in Partial Fulfillment of the Requirements

for Doctor of Philosophy (INFORMATION TECHNOLOGY)

Department of COMPUTER SCIENCE

Graduate School, Silpakorn University

Academic Year 2019

หัวข้อ	การพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่
โดย	สมเกียรติ ดอนทองแดง
สาขาวิชา	เทคโนโลยีสารสนเทศ แบบ 1.1 ระดับปริญญาดุษฎีบัณฑิต
อาจารย์ที่ปรึกษาหลัก	รองศาสตราจารย์ ดร. ปานใจ ธารทัศนวงศ์

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร ได้รับพิจารณาอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปรัชญาดุษฎีบัณฑิต

.....คณบดีบัณฑิตวิทยาลัย
(รองศาสตราจารย์ ดร.จุไรรัตน์ นันทานิช)

พิจารณาเห็นชอบโดย

.....ประธานกรรมการ
(ดร.สิรักข์ แก้วจำนงค์)

.....อาจารย์ที่ปรึกษาหลัก
(รองศาสตราจารย์ ดร.ปานใจ ธารทัศนวงศ์)

.....อาจารย์ที่ปรึกษาร่วม
(ดร.เอกสิทธิ์ กิจสิพงษ์)

.....ผู้ทรงคุณวุฒิภายใน
(ดร.วัสรา รอดเหตุภัย)

.....ผู้ทรงคุณวุฒิภายนอก
(รองศาสตราจารย์ ดร.เชาวน์ดิศ อัศวกุล)

57309802 : เทคโนโลยีสารสนเทศ แบบ 1.1 ระดับปริญญาดุษฎีบัณฑิต
คำสำคัญ : เครือข่ายที่มีความซับซ้อนสูง, บิ๊กดาต้า, การตรวจจับการโจมตี, การวิเคราะห์อนุกรมเวล

นาย สมเกียรติ ดอนทองแดง: การพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ อาจารย์ที่ปรึกษาวิทยานิพนธ์ : รองศาสตราจารย์ ดร. ปานใจ ธารทัศนวงศ์

ในปัจจุบันนี้การโจมตีด้วย DDoS นั้นสามารถสร้างความเสียหายให้กับระบบเครือข่ายอย่างมหาศาลเนื่องจากระบบเครือข่ายที่มีขนาดใหญ่อาทิเช่น องค์กรต่างๆ มีปริมาณของผู้ใช้จำนวนมากทำให้ปริมาณข้อมูลที่ถูกใช้งานมีจำนวนมากขึ้น ซึ่งจะส่งผลกระทบอย่างมากต่อระบบเครือข่ายขององค์กรที่มีความซับซ้อนสูง เนื่องด้วยขนาดของระบบเครือข่ายที่มีขนาดใหญ่และจำนวนของผู้ใช้ที่มีปริมาณมาก การศึกษานี้มุ่งเน้นไปที่การวิเคราะห์ระบบเครือข่ายของ UniNet ที่มีความซับซ้อนสูง งานวิจัยนี้ช่วยให้นำข้อมูลที่มีอย่างมหาศาลมาใช้ประโยชน์ ได้อย่างมีประสิทธิภาพ ข้อมูลที่มีขนาดใหญ่มากจะทำให้ระบบเครือข่ายจัดการได้ยาก จึงได้มีการพัฒนาต้นแบบขึ้นโดยใช้หลักการการประมวลผลแบบกระจายทำงานร่วมกับ Big Data Technic เพื่อใช้ในการแก้ไขปัญหาเหล่านี้ได้ตรงจุดโดยต้นแบบที่พัฒนาขึ้นจะทำการประมวลผลกระจายในแต่ละชั้นของระบบเครือข่าย ในงานวิจัยนี้ใช้ Hadoop platform เป็นตัวจัดการกับ Big Data โดยจัดเก็บข้อมูลขนาดใหญ่ด้วย HDFS และการคัดกรองข้อมูลด้วย MapReduce การทดลองจะเกิดขึ้นด้วยการนำ NetFlow log file ในระบบเครือข่าย UniNet ที่มีการโจมตีด้วย DDoS นำมาประเมินผ่านอัลกอริทึม ของ Vishal Masheshwari ในเพื่อหาค่าความถูกต้อง (accuracy) และค่าความล่าช้า (Delay) โดยได้ผล accuracy เท่ากับร้อยละ 71 และค่าความล่าช้า (Delay) ตั้งแต่เริ่มกระบวนการจัดเก็บ log file จนถึงการประมวลผลใช้เวลา 7 นาที

57309802 : Major (INFORMATION TECHNOLOGY)
Keyword : Complex Network, Big Data, Detect DDoS, Time series analysis

MR. SOMKIAT DONTONGDANG : PROTOTYPE DEVELOPMENT OF COMPLEX NETWORK ANALYTIC SYSTEM USING BIG DATA THESIS ADVISOR : ASSOCIATE PROFESSOR PANJAI TANTATSANAWONG, Ph.D.

Nowadays, DDoS attacks cause much damage in the network system. The organization's network system has a large number of users in which the heavy usage will be a great impact on the complex network system. Concerning an enormous size of the network system and an aggregation of users, this study aims to regularly analyze the complication of the UniNet network (complex network). This issue attempts to avoid an overload of data consumption and traffic congestion in the network. Hereafter, the super-sized data will be too exhausted to manipulate. The distribution processing principle becomes a simulation of Big Data model in order to treat problems properly. The principle is a processing of each network system level. Also, Hadoop platform could be in use as Big data storage (HDFS) and the data filtration (MapReduce). This experiment is done by using Netflow log file in the UniNet system and evaluating itself whether there are some attacks through Vishal Maheshwari's algorithm. As a result, accuracy is 71%, and a delay value is obviously 7 minute form beginning at storing to evaluating log file.

กิตติกรรมประกาศ

งานวิจัยนี้สำเร็จลุล่วงได้ด้วยความกรุณาช่วยเหลือ แนะนำ ให้คำปรึกษา ตรวจสอบแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่อย่างดียิ่งจาก รศ.ดร.ปานใจ ธารทัศนวงศ์ อาจารย์ที่ปรึกษาหลัก ผู้เขียนกราบขอบพระคุณเป็นอย่างสูงขอขอบพระคุณทุกท่านที่ให้ความอนุเคราะห์ เกี่ยวกับอุปกรณ์และข้อมูลที่ใช้ในการวิจัยครั้งนี้ ขอขอบคุณบิดา มารดา ญาติพี่น้อง และผู้ที่คอยช่วยเหลือสนับสนุนด้านกำลังใจด้วยดีตลอดมา นอกจากนี้ยังมีผู้ที่ให้ความร่วมมือช่วยเหลืออีกหลายท่าน ซึ่งผู้เขียนไม่สามารถกล่าวนามในที่นี้ได้หมด จึงขอขอบคุณทุกท่านเหล่านั้นไว้ ณ โอกาสนี้ด้วย

สมเกียรติ ดอนทองแดง

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ญ

สารบัญรูปภาพ ..... ฎ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 3

1.3. ขอบเขตการวิจัย 3

1.4. ประโยชน์ที่คาดว่าจะได้รับ 3

บทที่ 2 แนวคิดและงานวิจัยที่เกี่ยวข้อง 4

2.1 ระบบเครือข่ายที่มีความซับซ้อนสูง (Complex Network) 4

2.2 ปัญหาของระบบเครือข่าย 8

2.3 การวิเคราะห์ระบบเครือข่าย (Network Analysis) 11

2.4 การจัดการข้อมูลขนาดใหญ่ 18

บทที่ 3 วิธีดำเนินการวิจัย 29

3.1 เครื่องมือและอุปกรณ์ที่ใช้ในการทำวิจัย 29

3.2 ขั้นตอนการดำเนินการวิจัย 30

3.2.1 ศึกษาและวิเคราะห์ปัญหาที่เกิดขึ้นกับระบบเครือข่ายที่มีความซับซ้อนสูง 30

3.2.2 ศึกษาและออกแบบต้นแบบเพื่อแก้ปัญหาที่เกิดขึ้น 36

3.2.3 พัฒนาต้นแบบตามที่ได้ออกแบบไว้ข้างต้น 38
3.2.4 ทดสอบระบบ 41
3.2.5 การประเมินผล 45
3.2.6 สรุปผล 46
บทที่ 4 ผลการดำเนินการวิจัย 47
4.1 ผลการดำเนินงานที่ได้จากการศึกษาและวิเคราะห์ปัญหา 47
4.2 ผลการดำเนินการในการศึกษาเพื่อออกแบบต้นแบบ 48
4.2.2 การคัดกรองข้อมูล (Filtering Data) 50
4.2.3 ทดสอบอัลกอริทึมในการทดสอบระบบ 52
4.2.4 วิเคราะห์และประเมินผลการทดสอบ อัลกอริทึม 58
4.3 ผลการดำเนินงานในการทดสอบระบบที่พัฒนาขึ้น 59
4.3.1 จัดเก็บข้อมูล log file จาก ระบบเครือข่าย 59
4.3.2 การประเมินผล 71
4.4 สรุปผล 73
บทที่ 5 สรุปผลการดำเนินวิจัย 74
5.1 แก้ปัญหาที่พบในการวิเคราะห์และศึกษาระบบ 74
5.1.1 แก้ปัญหาข้อมูลที่มีจำนวนมากเกินไปที่ระบบจะทำการจัดเก็บได้ 74
5.1.2 แก้ปัญหาการจัดเก็บข้อมูลที่อยู่แบบกระจาย 74
5.1.3 ความล่าช้าในในการประมวลผล 75
5.2 สรุปผลการออกแบบต้นแบบที่จะทำการพัฒนา 76
5.3 สรุปผลการทดลองต้นแบบที่พัฒนาขึ้น 76
5.4 สรุปผลการวิจัย 78
5.5 แนวทางการวิจัยในอนาคต 79
รายการอ้างอิง 80

ประวัติผู้เขียน 83

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงสมาชิกเครือข่ายของ UniNet [2] 7

ตารางที่ 2.2 แสดงงานวิจัยที่เกี่ยวข้องโดยแบ่งเป็นการเพิ่มประสิทธิภาพในด้านต่าง ๆ 28

ตารางที่ 4.1 ขนาดของ log file ในเวลา 1 ชั่วโมง/Interface ใน Router 47

ตารางที่ 4.2 แสดง Scenarios ทั้งหมดของ CTU-13 Dataset 49

ตารางที่ 4.3 แสดงการจัดการแยก log file ออกเป็นคอลัมน์ 49

ตารางที่ 4.4 Field ทั้งหมดของ CTU-13 Dataset 50

ตารางที่ 4.5 แสดงรูปแบบการแบ่งข้อมูลตาม Timestamp 54

ตารางที่ 4.6 แสดงการคำนวณหาค่า Unique, Total Packets และ γi 55

ตารางที่ 4.7 แสดงการคำนวณหาค่า ai βi และ Li. 55

ตารางที่ 4.8 การตรวจจับพบ DDoS ด้วยอัลกอริทึมตามเงื่อนไข ai >1 56

ตารางที่ 4.9 การตรวจจับพบ DDoS ด้วยอัลกอริทึมตามเงื่อนไข $\beta i \leq 1$ 57

ตารางที่ 4.10 การตรวจจับพบ DDoS ด้วยอัลกอริทึมตามเงื่อนไข Li >0 57

ตารางที่ 4.11 ผลลัพธ์การประมวลผล DDoS 69

ตารางที่ 4.12 ผลลัพธ์การประมวลผลที่เป็นปกติ 70

ตารางที่ 4.13 แสดงผลค่าความถูกต้องในการทดลอง 73

สารบัญรูปภาพ

หน้า

ภาพที่ 2.1 แสดงรูปแบบการเชื่อมต่อของเครือข่าย UniNet [2] 5

ภาพที่ 2.2 แสดงโครงข่าย UniNet [2] 6

ภาพที่ 2.3 แสดงการโจมตีแบบ DDoS 10

ภาพที่ 2.4 สถาปัตกรรม Netflow 14

ภาพที่ 2.5 ภาพแสดง NetFlow Cache [8] 14

ภาพที่ 2.6 แสดงส่วนประกอบของ Flow record [7] 17

ภาพที่ 2.7 Hadoop Architecture [13] 21

ภาพที่ 2.8 แสดงคุณสมบัติที่เพิ่มขึ้นจาก Hadoop V.1 [16] 22

ภาพที่ 2.9 แสดง Hadoop Ecosystem [13] 23

ภาพที่ 3.1 แสดงขั้นตอนในการดำเนินการวิจัย 30

ภาพที่ 3.2 แสดงจุดการเชื่อมต่ออุปกรณ์หาเส้นทางในระบบ UniNet [2] 32

ภาพที่ 3.3 IP Address ในระบบ UniNet มีการปิดกั้นเนื่องจากเป็นภัยคุกคาม [2] 33

ภาพที่ 3.4 แจ้งประชาสัมพันธ์เหตุขัดข้องที่ไม่สามารถจะระบุสาเหตุได้ [2] 34

ภาพที่ 3.5 แสดงเครื่องแม่ข่ายที่ติดตั้งตามภูมิภาคที่ติดตั้ง NetFlow Application [2] 35

ภาพที่ 3.6 แสดงต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ 37

ภาพที่ 3.7 แสดง NetFlow log file ของ CTU-13 Dataset [23] 38

ภาพที่ 3.8 แสดงขั้นตอนการจัดเก็บ log file ลงใน HDFS 39

ภาพที่ 3.9 แสดงการคัดกรองข้อมูล 40

ภาพที่ 3.10 แสดงการส่งข้อความแจ้งเตือนและผลลัพธ์ไปยังศูนย์กลาง 41

ภาพที่ 3.11 การจัดเก็บ log file DDoS Attack 42

ภาพที่ 3.12 แสดงการทำงานMap Function เพื่อทำการคัดกรองข้อมูล 43

ภาพที่ 3.13 แสดงการทำงานของอัลกอริทึมในการตรวจจับ DDoS 44
ภาพที่ 4.1 แสดงการสร้างกราฟเพื่อแสดงค่าคงที่ (Stationary) ของข้อมูล 48
ภาพที่ 4.2 แสดงผังงานในการคัดกรองข้อมูล 51
ภาพที่ 4.3 แสดงผลลัพธ์ที่ได้จากการคัดกรองข้อมูลเพื่อลดขนาดลง 52
ภาพที่ 4.4 แสดงจำนวนข้อมูลที่ลดลงหลังจากการคัดกรองข้อมูล 52
ภาพที่ 4.5 แสดงการทำงานอัลกอริทึมการตรวจจับ DDoS 53
ภาพที่ 4.6 แสดงการหาค่าความถูกต้องแม่นยำด้วยค่า Accuracy 58
ภาพที่ 4.7 อุปกรณ์ที่ใช้ในการทดสอบระบบ 59
ภาพที่ 4.8 แสดงผังการจัดอุปกรณ์ที่ใช้ในการทดลอง 60
ภาพที่ 4.9 ตั้งค่า Netflow ให้ส่ง log file มาที่เครื่อง Collector 60
ภาพที่ 4.10 ไฟล์ nfcap 61
ภาพที่ 4.11 คำสั่งในการใช้ Slowloris DDoS Tool 62
ภาพที่ 4.12 กราฟแสดงปริมาณแพ็คเกตที่มีการโจมตี DDoS 62
ภาพที่ 4.13 กราฟแสดงปริมาณแพ็คเกตที่มีการใช้งานปกติ 63
ภาพที่ 4.14 log file ที่อยู่ใน Router 64
ภาพที่ 4.15 แสดงผังงานการนำ Log file ไปจัดเก็บใน HDFS 65
ภาพที่ 4.16แสดงการทำงานของการคัดกรองข้อมูล 66
ภาพที่ 4.17 แสดงการทำงานของระบบ 67
ภาพที่ 4.18 แสดงกล่องข้อความแจ้งเตือนไปยังผู้ดูแลระบบเมื่อพบ DDoS 71

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันระบบเครือข่ายคอมพิวเตอร์มีบทบาทสำคัญต่อการดำเนินกิจการต่าง ๆ เป็นอย่างมาก ไม่ว่าจะเป็นด้านการติดต่อสื่อสาร ธุรกิจ ความบันเทิง รวมไปถึงระบบการศึกษาและวิจัย อย่างไรก็ตามความสะดวกสบาย และช่องทางในการสื่อสารที่เพิ่มมากขึ้นเท่าใด ก็ส่งผลให้ผู้บุกรุกมีช่องทางในการเข้าถึงระบบได้มากขึ้นเท่านั้น เช่น การโจมตีเพื่อหยุดการให้บริการ (DoS: Denial of Service Attacks) และการโจมตีแบบกระจายเพื่อหยุดการให้บริการ (DDoS: Distributed Denial of Service Attacks) [1] เป็นต้น ซึ่งจะส่งผลกระทบที่รุนแรงต่อการใช้งานระบบอินเตอรเน็ตโดยรวมภายในองค์กร และอาจก่อความเสียหายในด้านความมั่นคงขององค์กรได้ ผู้บริหารเครือข่ายต้องคำนึงถึงประสิทธิภาพและความปลอดภัยในระบบเครือข่าย ที่อาจเกิดความเสี่ยงต่อการบุกรุกและเกิดภัยอันตรายได้ตลอดเวลา เนื่องจากระบบวิเคราะห์เครือข่ายที่ใช้งานอยู่ยังไม่มีประสิทธิภาพเพียงพอที่จะช่วยให้ผู้ดูแลระบบรู้ถึงสถานการณ์ของระบบเครือข่ายได้ทันเหตุการณ์

ในปัจจุบันการนิยมใช้การเฝ้าสังเกตระบบ (Monitoring) โดยใช้ MRTG (Multi Router Traffic Grapher) ซึ่งเป็นเครื่องมือในการดักจับแพ็คเกตผ่านตัวประสาน (Interface) ที่กำหนดและเก็บค่ามาสร้างเป็นกราฟ แต่ยังมีจุดด้อยคือ ต้องใช้คนในการอ่านค่ากราฟ และวิเคราะห์ระบบเครือข่าย ซึ่งหากเกิดผิดปกติกับระบบเครือข่ายอาจจะไม่สามารถป้องกันหรือแก้ไขได้ทันเวลา ทำให้ส่งผลกระทบและสร้างความเสียหายต่อหน่วยงานได้ ดังนั้นระบบเครือข่ายที่ดีต้องสามารถทำการวิเคราะห์สิ่งผิดปกติได้ก่อนที่จะก่อให้เกิดความเสียหายต่อระบบ และสามารถที่จะป้องกันหรือแก้ไขได้อย่างทันท่วงทีโดยไม่สิ้นเปลืองเสียค่าใช้จ่ายในการใช้แรงงานคนในการเฝ้าระวังระบบเครือข่าย

นอกจากนี้ ข้อมูลการจราจร (Log file) ต่างๆในปัจจุบันส่วนหนึ่งมาจากการใช้งานระบบการสื่อสารและเครือข่ายอินเทอร์เน็ตกันมากขึ้น จนมีปริมาณข้อมูลเกิดขึ้นจำนวนมหาศาล หรือข้อมูลขนาดใหญ่เรียกว่า บิ๊กดาต้า (Big Data) ปัญหาเกี่ยวกับการจัดการข้อมูลขนาดใหญ่เหล่านี้กำลังได้รับความสนใจมาก เนื่องจากข้อมูลต่างๆที่จัดเก็บมีความสำคัญเพื่อใช้ในการสืบค้น วิเคราะห์ ประมวลผลข้อมูล อีกทั้งข้อมูลที่จัดเก็บยังมีขนาดใหญ่ขึ้นเรื่อยๆทำให้การจัดการกับข้อมูลมีความลำบากและยุ่งยากมากขึ้น จึงจำเป็นต้องมีเครื่องมือเข้ามาช่วยจัดการเพื่ออำนวยความสะดวกในการจัดเก็บ การ

ค้นหา และการวิเคราะห์ข้อมูล ซึ่งมีหลายหน่วยงาน หลายเครือข่ายสนใจ และจำเป็นต้องใช้เทคโนโลยีเหล่านี้

ในประเทศไทยมีเครือข่ายเพื่อการศึกษาวิจัยไทย (Thai-REN) เป็นเครือข่ายที่มีความซับซ้อนสูง (Complex Network) เป็นลักษณะของการเชื่อมต่อชนิดหนึ่งซึ่งตัวเครือข่ายเองนั้นมีขนาดใหญ่ มีความซับซ้อนและเป็นเครือข่ายที่มีการเชื่อมต่อหลายลำดับชั้น (Layer) เครือข่ายที่มีความซับซ้อนสูงดังกล่าวนี้มีการดำเนินการสร้างเครือข่ายความเร็วสูงเชื่อมโยงมหาวิทยาลัย/สถาบันระดับอุดมศึกษาในประเทศไทย เชื่อมโยงไปยังเครือข่ายเพื่อการศึกษาวิจัยในต่างประเทศ เพื่อให้สถาบันการศึกษาสามารถจัดการเรียนการสอนและการวิจัยร่วมกันได้โดยสะดวกและรวดเร็วผ่านเครือข่ายอินเทอร์เน็ตความเร็วสูง นอกจากนี้ Thai-REN พัฒนาแหล่งความรู้โดยพัฒนาความร่วมมือระหว่างห้องสมุดสถาบันอุดมศึกษาในประเทศไทย ภายใต้โครงการ ThaiLIS และพัฒนากลุ่มวิจัยด้านต่างๆ เพื่อสนับสนุนพัฒนาและปรับปรุงเครือข่ายให้เหมาะสมสำหรับใช้งานเพื่อการศึกษาวิจัยโดยเฉพาะ ซึ่งมีสมาชิกที่เป็นมหาวิทยาลัยจำนวน 293 แห่ง เชื่อมต่อกันที่ความเร็ว 1-10 Gbps สถาบันอาชีวะจำนวน 415 แห่ง เชื่อมต่อกันด้วยความเร็ว 100-1,000 Mbps และ โรงเรียนจำนวน 9,000 แห่ง เชื่อมต่อกันที่ความเร็ว 10-100 Mbps [2]

ในส่วนของรูปแบบการเชื่อมต่อของเครือข่ายนั้นจะเชื่อมต่อเป็น 2 ลำดับชั้นคือ Backbone ลิงค์เชื่อมต่อกันที่ความเร็ว 50 Gbps และ Distribution ลิงค์เชื่อมต่อกันที่ความเร็ว 10 Gbps โดยรูปแบบการเชื่อมต่อจะเชื่อมต่อกันด้วยอุปกรณ์เลือกเส้นทาง (Router) 120 ตัว จากปริมาณของข้อมูลการจราจร (Log file) ในอุปกรณ์เลือกเส้นทาง (Router) ที่สร้างขึ้นจาก NetFlow application ในเครือข่ายซึ่งมีจำนวน Router ทั้งหมด 120 ตัว ที่มีการเก็บข้อมูลปริมาณของ Log file ใน Router โดยในแต่ละตัวจะสร้างข้อมูล Log file ออกมาประมาณ 3.5 GB ในเวลา 1 ชั่วโมง แต่ถ้ารวมกันทั้งหมดใน 1 วัน จะมีปริมาณข้อมูล 9.8 TB และในระยะเวลา 1 ปี จะมีปริมาณข้อมูลการจราจรเข้า-ออกของเครือข่ายเท่ากับ 3.4 PB ถือว่าเป็นปริมาณข้อมูลที่มีจำนวนมาก หากต้องการที่จะนำข้อมูลทั้งหมด มาใช้เพื่อทำการประมวลผลก่อนนำไปใช้ในการวิเคราะห์ระบบเครือข่าย อาจจะทำเกิดปัญหาด้านพื้นที่ในการจัดเก็บข้อมูลที่มีอยู่อย่างจำกัด และความเร็วในการประมวลผลเพื่อทำการวิเคราะห์ระบบเครือข่าย

จากปัญหาดังกล่าวข้างต้นผู้วิจัยจึงสนใจที่จะศึกษารูปแบบการวิเคราะห์ประสิทธิภาพและปัญหาบนระบบเครือข่าย เพื่อนำข้อมูลเหล่านี้มาพัฒนาระบบเครือข่ายให้มีประสิทธิภาพโดยการนำข้อมูลขนาดใหญ่มาใช้ในการวิเคราะห์ระบบเครือข่ายที่มีความซับซ้อนสูงให้มีประสิทธิภาพ มีความ

รวดเร็ว และทำให้ระบบเครือข่ายมีความปลอดภัยสามารถแจ้งเตือนเมื่อระบบเกิดความผิดปกติ ช่วยให้ผู้ควบคุมระบบรู้ถึงสถานการณ์ของเครือข่ายได้อย่างทันเหตุการณ์

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาและการวิเคราะห์ปัญหาที่เกิดขึ้นในระบบเครือข่ายที่มีความซับซ้อนสูง (Complex Network)

1.2.2 เพื่อพัฒนารูปแบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่

1.2.3 เพื่อการพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่

1.3. ขอบเขตการวิจัย

1.3.1 ศึกษาและวิเคราะห์ปัญหาเกี่ยวกับการโจมตีแบบ DDoS ที่เกิดขึ้นในระบบเครือข่ายที่มีความซับซ้อนสูง (Complex Network)

1.3.2 ออกแบบพัฒนาระบบการวิเคราะห์เครือข่ายให้มีประสิทธิภาพสูงสุดเพื่อความมีเสถียรภาพของระบบเครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ที่รองรับได้ทั้ง IPv4 และ IPv6 ข้อมูลที่ใช้เป็นกรณีศึกษาจะใช้ข้อมูลซึ่งสนับสนุนโดยสมาคมเครือข่ายเพื่อการศึกษาวิจัยไทย (Thai-REN: Thailand Research Education Network Association) โดยให้ผู้เกี่ยวข้องมีส่วนร่วม

1.3.3 ทดสอบการทำงานและสรุปผล

1.4. ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ทราบถึงปัญหาที่เกิดขึ้นในระบบเครือข่ายที่มีความซับซ้อนสูง

1.4.2 ได้แนวทางในการแก้ปัญหาการจัดการข้อมูลขนาดใหญ่บนระบบเครือข่ายที่มีความซับซ้อนสูง

1.4.3 ได้ต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ที่มีประสิทธิภาพ และสามารถแจ้งเตือนเมื่อระบบเกิดความผิดปกติได้

บทที่ 2
แนวคิดและงานวิจัยที่เกี่ยวข้อง

ในการศึกษาเพื่อดำเนินการจัดทำวิทยานิพนธ์ฉบับนี้ ผู้วิจัยได้ทำการศึกษาแนวคิดและค้นคว้าทบทวนเอกสารงานวิจัยที่เกี่ยวข้องกับปัญหาของระบบเครือข่ายที่มีความซับซ้อนสูง (Complex Network) การวิเคราะห์ระบบเครือข่าย ประสิทธิภาพการวิเคราะห์ระบบเครือข่าย และการจัดการข้อมูลขนาดใหญ่ (Big Data) ดังมีรายละเอียดดังต่อไปนี้

2.1 ระบบเครือข่ายที่มีความซับซ้อนสูง (Complex Network)

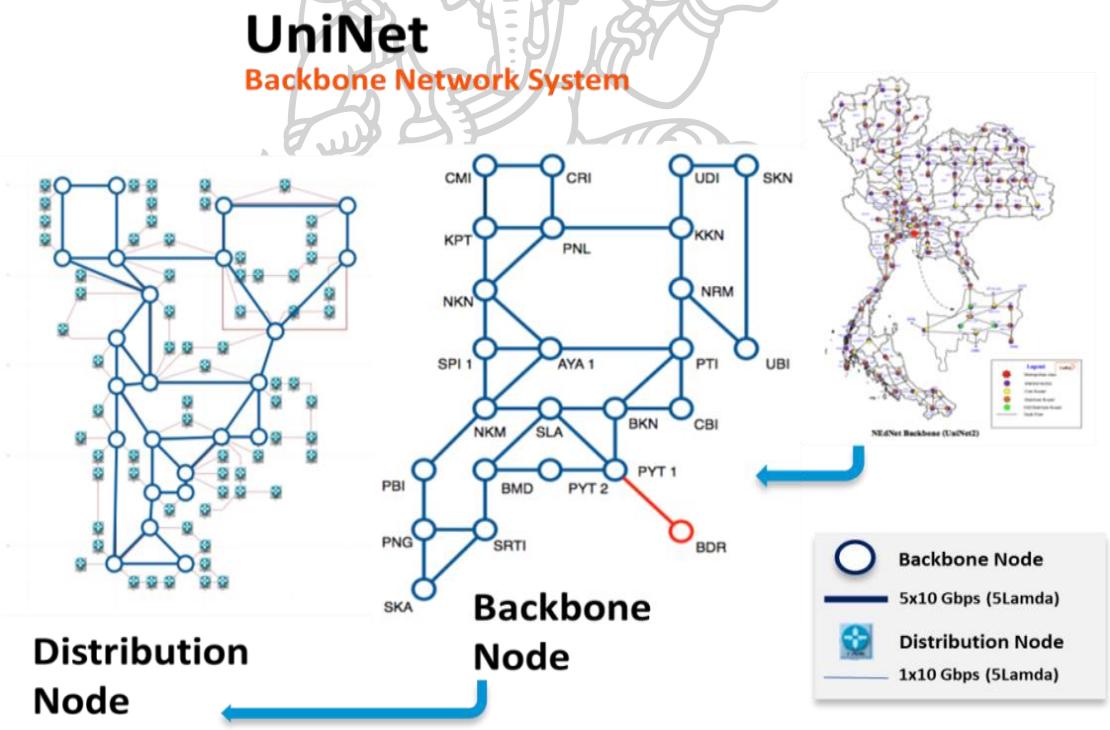
เครือข่ายที่มีความซับซ้อน (complex network) มีอยู่มากมายทุกหนทุกแห่ง ทั้งในธรรมชาติและมนุษย์สร้างขึ้น มีโครงสร้างประกอบด้วยจุด nodes (vertices) จำนวนมากที่เชื่อมโยงกันด้วยเส้น links [3] (edges) ตัวอย่างเช่น เครือข่ายอินเทอร์เน็ตซึ่งประกอบด้วย routers หรือ domains ต่างๆ ที่เชื่อมต่อกันด้วยไฟเบอร์ออฟติก เวิลด์ไวด์เว็บเป็นเครือข่ายของเว็บเพจที่เชื่อมต่อกันด้วย hyperlinks เข้าด้วยกัน

การวัดและศึกษาทฤษฎี complex network ได้แก่

- Average path length : ค่าเฉลี่ยของระยะทางระหว่าง node ซึ่งสำหรับเครือข่ายซับซ้อนนั้น ส่วนใหญ่มีค่าค่อนข้างน้อย
- Clustering coefficient : node 2 nodes ที่เชื่อมต่อมายัง node เดียวกัน ย่อมมีโอกาสที่จะเชื่อมต่อกันโดยตรงได้ เครือข่ายในโลกแห่งความจริงที่มีขนาดใหญ่ มีแนวโน้มที่จะรวมกลุ่มเป็น clustering และไม่ใช่เป็นการ random
- Degree distribution : node ที่มีค่า degree สูงจะเป็น node ที่มีความสำคัญ การค้นพบ small-world effect และลักษณะ scale-free ของเครือข่ายซับซ้อน ทำให้เกิดทฤษฎีเครือข่ายซับซ้อน (complex network theory) และมีผู้ทำการศึกษาวิจัยในเรื่องนี้กันอย่างกว้างขวางในปัจจุบัน

ในประเทศไทยมีระบบเครือข่ายการศึกษาและวิจัยซึ่งเครือข่ายที่ใช้ในการศึกษาวิจัย (Research Education Network) ใช้คำย่อว่าREN [2] เป็นเครือข่ายขนาดใหญ่และมีความซับซ้อนครอบคลุมมากกว่า 15 ประเทศ มีสมาชิกทั้งประเทศในทวีปยุโรป อเมริกา เอเชียใต้ ญี่ปุ่น จีน และ

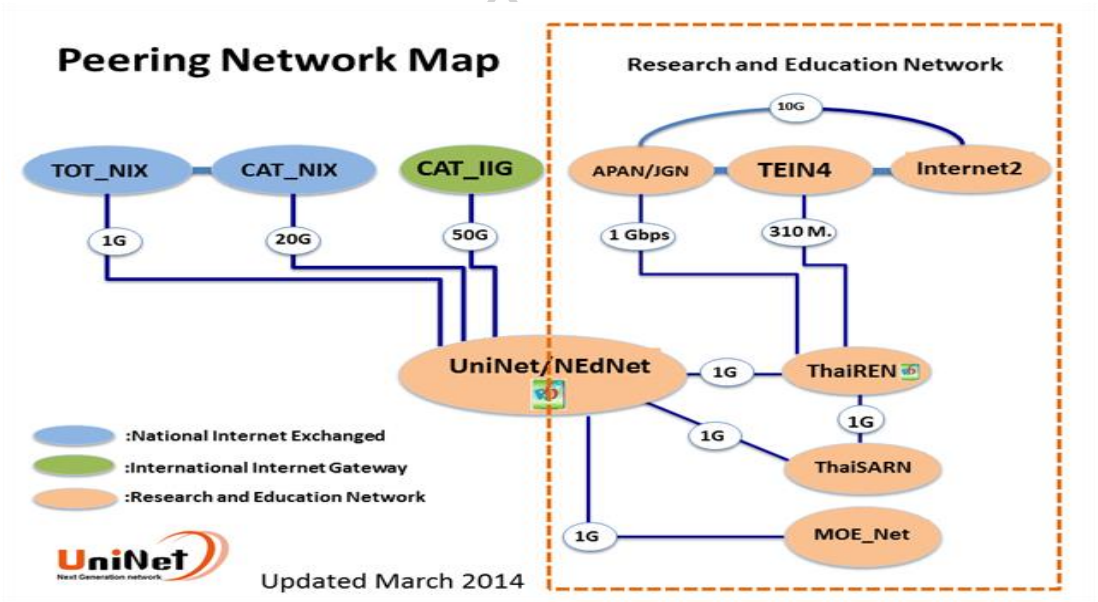
ประเทศในแถบเอเชียตะวันออกเฉียงใต้ รวมทั้งประเทศไทย มีการเชื่อมโยงเครือข่ายของสถาบันการศึกษาต่างๆ เพื่อดำเนินการวิจัยร่วมกัน สำหรับประเทศไทยหน่วยงานที่ดำเนินการวิจัยร่วมกันมีชื่อว่า ThaiREN นอกเหนือจากการสนับสนุนเครือข่ายเพื่อการศึกษาวิจัยแล้ว ทางสมาคมโดยหน่วยงาน UniNet ได้ดำเนินการสร้างเครือข่ายความเร็วสูงเชื่อมโยงมหาวิทยาลัย/สถาบันระดับอุดมศึกษาในประเทศไทย รวมถึงเชื่อมโยงเครือข่ายไปยังเครือข่ายเพื่อการศึกษาวิจัยในต่างประเทศ ทำให้สถาบันการศึกษาสามารถจัดการเรียนการสอนและการวิจัยร่วมกันกับสถาบันการศึกษาในประเทศได้โดยสะดวกและรวดเร็วผ่านเครือข่ายอินเทอร์เน็ตความเร็วสูง อีกทั้งยังพัฒนาแหล่งความรู้โดยพัฒนาความร่วมมือระหว่างห้องสมุดสถาบันอุดมศึกษาในประเทศไทย โดยดำเนินโครงการ ThaiLIS รวมถึงพัฒนากลุ่มวิจัยด้านต่างๆ เพื่อสนับสนุนพัฒนาและปรับปรุงเครือข่ายให้เหมาะสมสำหรับใช้งานเพื่อการศึกษาวิจัยโดยเฉพาะ ซึ่งมีสมาชิก ดังนี้ มหาวิทยาลัยจำนวน 293 แห่ง เชื่อมต่อกันที่ความเร็ว 1-10 Gbps สถาบันอาชีวะจำนวน 415 แห่ง เชื่อมต่อกันด้วยความเร็ว 100-1,000 Mbps และ โรงเรียนจำนวน 9,000 แห่ง เชื่อมต่อกันที่ความเร็ว 10-100 ดังภาพที่ 2.1



ภาพที่ 2.1 แสดงรูปแบบการเชื่อมต่อของเครือข่าย UniNet [2]

UniNet ได้พัฒนาและขยายโครงสร้างพื้นฐานเครือข่ายแกนหลัก และระบบเครือข่ายกระจาย โดยสร้างเครือข่ายเคเบิลใยแก้วนำแสง และเครือข่ายอุปกรณ์หลักและอุปกรณ์อื่นที่เกี่ยวข้อง

ให้สามารถครอบคลุมและรองรับการขยายการเชื่อมโยงไปยังสถาบันการศึกษาทั้งระบบครอบคลุมทุกจังหวัด เพื่อเพิ่มขีดความสามารถในการเข้าถึงและนำเทคโนโลยีสารสนเทศและการสื่อสารมาใช้ประโยชน์ในการจัดการศึกษาและวิจัยของประเทศ นอกจากการสร้างเครือข่ายด้วยสื่อใยแก้วนำแสงดังกล่าวข้างต้นแล้ว UniNet ยังได้ทำหน้าที่เชื่อมโยงเครือข่ายออกสู่ภายนอก ทั้งในส่วนอินเทอร์เน็ตเพื่อการศึกษาวิจัยเชื่อมโยงเข้ากับเครือข่ายเพื่อการศึกษาวิจัยทั่วโลก อาทิ เครือข่าย Internet2, TEIN, JGN เป็นต้น นอกจากนี้ยังเชื่อมโยงใช้งานเครือข่ายอินเทอร์เน็ตทั่วไปผ่าน CAT และ TOT รายละเอียดดังภาพ



ภาพที่ 2.2 แสดงโครงข่าย UniNet [2]

โครงการเครือข่ายสารสนเทศเพื่อพัฒนาการศึกษา (Inter University Network) หรือที่เรียกว่า เครือข่าย “UniNet” จัดตั้งขึ้นตามมติคณะรัฐมนตรี เมื่อวันที่ 8 ตุลาคม 2539 เห็นชอบให้จัดตั้งองค์กรกลางดำเนินโครงการในลักษณะการจัดหาวงจรสื่อสัญญาณความเร็วสูงเพื่อใช้สำหรับการเชื่อมโยงเครือข่ายสารสนเทศและการสื่อสารของสถาบันอุดมศึกษา และจัดตั้งเป็นสำนักงานบริหารเทคโนโลยีสารสนเทศเพื่อพัฒนาการศึกษา ตั้งแต่วันที่ 25 มิถุนายน 2540 ทำหน้าที่บริหารจัดการโครงการเครือข่ายสารสนเทศเพื่อพัฒนาการศึกษาซึ่งเป็นการดำเนินการขยายโอกาสอุดมศึกษาสู่ภูมิภาค โดยการนำเทคโนโลยีสารสนเทศมาช่วยในการจัดการเรียนการสอน สำนักงานฯ ได้เชื่อมโยงเครือข่ายเทคโนโลยีสารสนเทศของมหาวิทยาลัย/สถาบันในสังกัดทบวงมหาวิทยาลัยในขณะนั้น 24

แห่ง และวิทยาเขตสารสนเทศ 37 แห่ง ตั้งแต่ พ.ศ. 2539 เชื่อมโยงอยู่บนเครือข่ายสารสนเทศเพื่อพัฒนาการศึกษา (UniNet) เพื่อให้สถาบันการศึกษาระดับอุดมศึกษาทั้งในส่วนกลางและส่วนภูมิภาคสามารถเข้าถึงเทคโนโลยีสารสนเทศ ที่เหมาะสมและเพียงพอต่อการจัดการศึกษา สามารถเชื่อมต่อแลกเปลี่ยนข้อมูลระหว่างกันทั้งภายในและต่างประเทศ

ต่อมาปี พ.ศ. 2553-2555 กระทรวงศึกษาธิการมีการบูรณาการเครือข่ายภายในกระทรวงศึกษาธิการเข้าด้วยกันเป็นเครือข่ายเดียว รองรับการศึกษาทุกระดับ (ระดับอุดมศึกษา ระดับอาชีวศึกษา ระดับการศึกษาขั้นพื้นฐาน และอื่นๆ) ตามโครงการพัฒนาเครือข่ายสารสนเทศเพื่อพัฒนาการศึกษา (UniNet) เพื่อรองรับการศึกษาทั้งระบบ โดยการพัฒนาโครงสร้างพื้นฐานโครงข่ายเคเบิลใยแก้วนำแสงขึ้นเอง เชื่อมต่อไปยังสถานศึกษา จำนวน 3,000 แห่งทั่วประเทศ

และปี พ.ศ. 2555-2557 มีการพัฒนาต่อยอดในโครงการเครือข่ายการศึกษาแห่งชาติ (National Education Network : NedNet) ดำเนินการขยายโครงข่ายเคเบิลใยแก้วนำแสง เชื่อมต่อไปยังโรงเรียนอีกจำนวน 7,606 แห่งทั่วประเทศ ซึ่งเมื่อดำเนินโครงการแล้วเสร็จ จะมีสมาชิกเครือข่ายทั้งหมดกว่า 10,000 แห่งทั่วประเทศและในปี 2562 การขยายขอบเขตและบูรณาการเป็นระบบเครือข่ายการศึกษาแห่งชาติ National Education Network (NedNet) มีสมาชิกเครือข่าย 10,630 แห่ง จากโรงเรียนในสังกัด สพฐ. มหาวิทยาลัยในสังกัด สกอ. หน่วยงานวิจัยและสถานศึกษาอื่น ๆ เพื่อรองรับสถาบันการศึกษาทั่วประเทศ ด้วยความเร็ว 100 Gbps

เครือข่ายเพื่อการศึกษาวิจัย เป็นเครือข่ายเฉพาะกิจ สำหรับสนับสนุนสถาบันการศึกษาดำเนินกิจกรรมการเรียนการสอน และการวิจัย เชื่อมต่อไปยังสถาบันการศึกษาซึ่งมีสมาชิกเครือข่ายดังนี้

ลำดับ	ประเภทสถาบันการศึกษา	จำนวนสถาบัน (แห่ง)	จำนวนโหนด ที่เชื่อมต่อทั้งหมด (โหนด)
1	สถาบันอุดมศึกษาในกำกับของรัฐ	14	55
2	สถาบันอุดมศึกษาของรัฐ	66	127
3	วิทยาลัยชุมชน	20	20
4	สถาบันอาชีวศึกษา	427	427
5	สำนักงานเขตพื้นที่การศึกษา	225	225
6	โรงเรียน	9568	9568
7	ห้องสมุดสังกัด กศน.	152	152
8	โรงเรียนเอกชนสังกัด สช.	142	142
9	หน่วยงานอื่นๆ	16	16
รวม		**10630**	**10732**

ตารางที่ 2.1 แสดงสมาชิกเครือข่ายของ UniNet [2]

งานวิจัยที่เกี่ยวข้องเกี่ยวกับการเพิ่มประสิทธิภาพของระบบเครือข่ายที่มีความซับซ้อนสูง

การเปรียบเทียบเครื่องมือในการประมวลผลข้อมูลขนาดใหญ่บนระบบเครือข่ายที่ความซับซ้อนสูงระหว่าง MapReduce กับ Aphache Spark เพื่อหาความเหมาะสมกับระบบเครือข่ายและปริมาณข้อมูลที่มีอยู่บนระบบเครือข่าย ผลการศึกษาพบว่า Aphache Spark มีความเร็วมากกว่าหากปริมาณข้อมูลมีจำนวนน้อยกว่าพื้นที่หน่วยความจำหลัก แต่เนื่องจากเครือข่ายที่มีความซับซ้อนสูงนี้ มีหลายลำดับชั้น และมีปริมาณข้อมูลจำนวนมาก จึงทำให้การเลือกใช้ MapReduce มีความเหมาะสมมากกว่า [4] และมีการศึกษางานวิจัยเกี่ยวกับการนำเสนอโมเดลเกี่ยวกับการจัดการข้อมูลขนาดใหญ่บนระบบเครือข่ายที่มีความซับซ้อนสูง โดยใช้เครื่องมือในการจัดการข้อมูลขนาดใหญ่ที่ได้รับความนิยมมากที่ชื่อว่า Hadoop โดยอาศัยหลักการทำงานของ Software – Defined Networking (SDN) ใช้ในการกำหนดเส้นทางและให้ลำดับความสำคัญกับข้อมูล เพื่อลดความแออัดของการจราจรในระบบเครือข่าย [5]

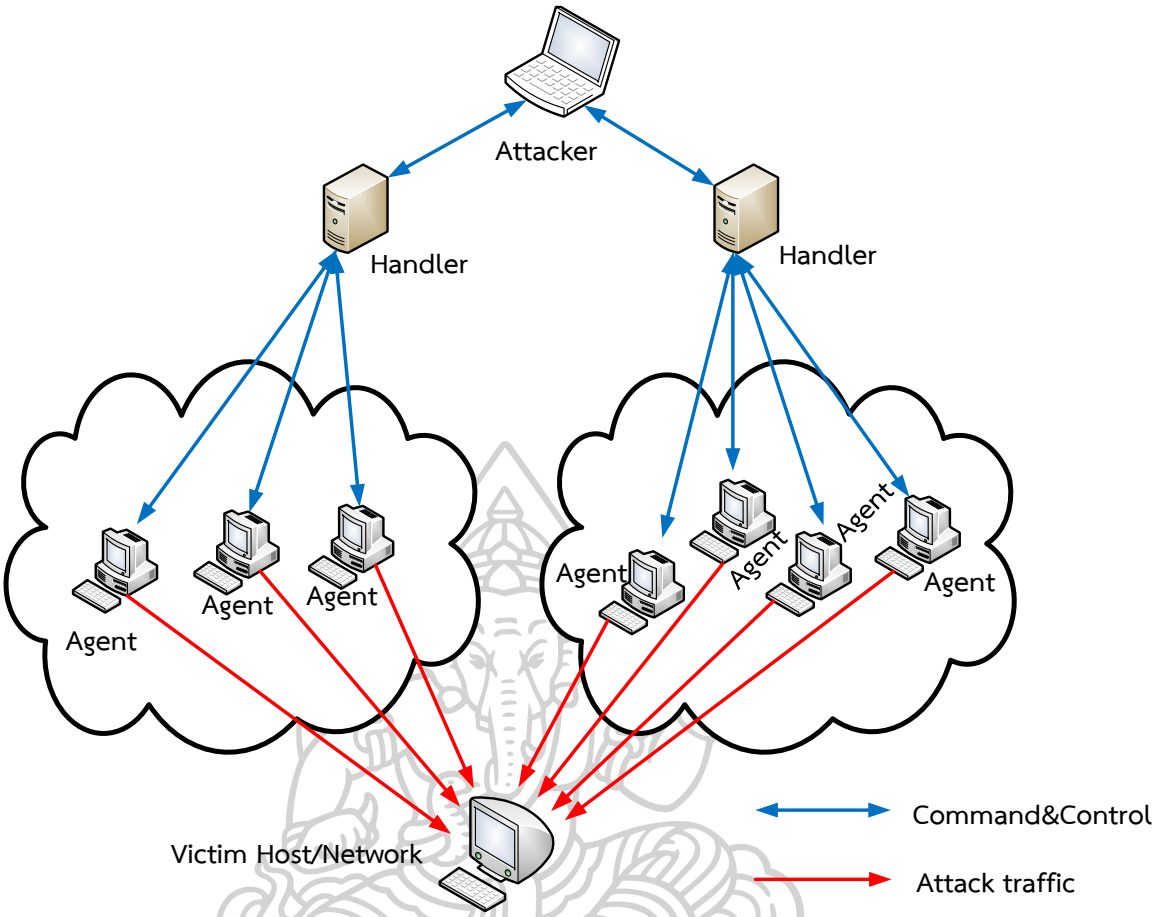
2.2 ปัญหาของระบบเครือข่าย

ระบบเครือข่ายที่มีความซับซ้อน ปัญหาส่วนใหญ่มักเป็นปัญหาที่เกี่ยวกับเครือข่ายไม่สามารถทำงานได้เต็มประสิทธิภาพ ในงานวิจัยนี้จะศึกษาเกี่ยวกับปัญหาการถูกโจมตีเครือข่ายแบบกระจาย (DDoS) ผ่านทางอินเทอร์เน็ต เนื่องจากการโจมตีจะมีความประสงค์ให้เซิร์ฟเวอร์หรือเครือข่ายที่ถูกโจมตีไม่สามารถตอบสนองได้

การโจมตีเพื่อไม่ให้เซิร์ฟเวอร์ตอบสนองได้มี 2 รูปแบบคือ การโจมตีแบบ DoS และการโจมตีแบบ DDoS การโจมตีดังกล่าวสามารถแบ่งวิธีการโจมตีได้เป็น 2 วิธี วิธีที่หนึ่งเป็นการโจมตีไปยังช่องโหว่ในโปรแกรมระบบของอุปกรณ์ เพื่อทำให้อุปกรณ์นั้นหยุดทำงาน วิธีที่สอง คือ การโจมตีแบบฟลัดดิ้งไปยังอุปกรณ์ทำให้อุปกรณ์ถูกใช้ทรัพยากร (ได้แก่ หน่วยประมวลผล หน่วยความจำอุปกรณ์จัดเก็บข้อมูล และช่องทางรับส่งข้อมูล) เป็นจำนวนมากจนไม่สามารถให้บริการได้ตามปกติ ข้อแตกต่างระหว่างการโจมตีแบบ DoS และการโจมตีแบบ DDoS คือการโจมตีแบบ DoS มีเส้นทางในการโจมตีเพียงเส้นทางเดียว สวนการโจมตีแบบ DDoS มีเส้นทางในการโจมตีมาจากหลายเส้นทางจึงเป็นอันตรายและรวดเร็วมากกว่า การหาความผิดปกติในระบบเครือข่าย จากการศึกษาโปรโตคอลมาตรฐานที่ใช้ได้แก่ SYN Flood, ICMP Flood และ UDP Flood โดยการตรวจจับการรับชุดข้อมูลจำนวนมากที่ผิดปกติของเครือข่ายปลายทาง ระบบเครือข่ายที่มีเซิร์ฟเวอร์ไว้ให้บริการ อาจไม่

สามารถให้บริการได้ เป้าหมายของการโจมตีเซิร์ฟเวอร์ก็เพื่อที่จะทำให้ระบบบริการหยุดการทำงาน รูปแบบที่พบบ่อยที่สุดในปัจจุบัน คือ ผู้ไม่ประสงค์ดีจะใช้มัลแวร์ที่ตนสร้างขึ้นเข้าไปฝังตัวยังอุปกรณ์ต่างๆ ที่มีคุณสมบัติเชื่อมต่อระบบเครือข่ายได้ และหากอุปกรณ์นั้นสามารถสื่อสารผ่านระบบเครือข่ายมายังเซิร์ฟเวอร์ได้ ผู้ไม่ประสงค์ดีก็จะสั่งการให้โจมตีมายังเซิร์ฟเวอร์นั้นๆ ได้ โดยอุปกรณ์บนระบบเครือข่ายที่ถูกฝังมัลแวร์เหล่านี้จะมีชื่อเรียกในทางเทคนิคว่า “ซอมบี้” (Zombie)

อุปกรณ์ที่เป็นเป้าหมายในการโจมตี ได้แก่ คอมพิวเตอร์ สมาร์ทโฟน แท็บเล็ต หรืออุปกรณ์ IoT (Internet of Things) โดยผู้ไม่ประสงค์ดีจะทำการแฝงตัวเพื่อควบคุมอุปกรณ์ให้ได้จำนวนมากที่สุด จากนั้นจะสั่งการให้อุปกรณ์เหล่านั้นระดมยิงข้อมูลปริมาณมากไปยังเซิร์ฟเวอร์ที่เป็นเป้าหมาย จนทำให้เซิร์ฟเวอร์ทำงานหนักเนื่องจากการได้รับข้อมูลปริมาณมากเกินไปจนเกิดการท่วมของข้อมูล (Data Flood) ทำให้เซิร์ฟเวอร์ทำงานช้าลงเรื่อยๆ จนหยุดการทำงาน ซึ่งส่งผลกระทบต่อผู้ใช้งานที่กำลังเชื่อมต่อไปยังเซิร์ฟเวอร์ในขณะนั้นจนไม่สามารถทำธุรกรรมใดๆ ต่อได้ การโจมตีเซิร์ฟเวอร์ในลักษณะนี้ เป็นอีกหนึ่งรูปแบบของการโจมตีที่เรียกกันโดยทั่วไปว่า DDoS Attacks (Distributed Denial of Service) ดังภาพที่ 2.3



ภาพที่ 2.3 แสดงการโจมตีแบบ DDoS

DDoS คือการโจมตีโดยไม่ได้ใช้เครื่องคอมพิวเตอร์เพียงเครื่องเดียวเป็นตัวโจมตีแบบ DoS แต่จะเป็นการใช้เครื่องคอมพิวเตอร์หรืออุปกรณ์ที่มีการเชื่อมต่อกับระบบอินเทอร์เน็ตหลายๆ เครื่องช่วยกันโจมตีเพื่อให้ระบบเสียหาย โดยส่งผลทำให้เกิดขนาดในการโจมตีที่ใหญ่และป้องกันได้ยาก Protocol ที่นิยมใช้ในการทำ DDoS มากที่สุดคือ UDP เพราะสามารถปลอมแปลงได้ ไม่จำเป็นต้องมีการพิสูจน์ตัวตนแบบ TCP เช่นเทคนิค Amplification/Reflection กับช่องโหว่ของ Service พวก NTP, DNS, SNMP, SSDP เป็นต้น ที่กล่าวข้างต้นเป็น UDP ทั้งหมด ที่มีความรุนแรงและตรวจจับยาก

งานวิจัยที่เกี่ยวข้องเกี่ยวกับปัญหาของระบบเครือข่าย

ระบบตรวจสอบสำหรับการโจมตีแบบ Denial of Service (DDoS) บน Apache Hadoop และระบบ HBase โดยมีวิธีการตรวจจับ DDoS ตามหลักการทำงานสองแบบหลัก ๆ คือความสามารถในการเรียนรู้ของระบบตรวจจับ DDoS และความสามารถในการประมวลผลชุดข้อมูลขนาดใหญ่ที่ไม่มี

โครงสร้าง ระบบมีความสามารถในการเรียนรู้เพื่อปรับให้เข้ากับ DDoS ชนิดใหม่ การโจมตีและความสามารถในการจัดเก็บและวิเคราะห์โครงสร้างที่ไม่มีโครงสร้างขนาดใหญ่ ชุดข้อมูลจะถูกเก็บรวบรวมจากบันทึกของเครือข่าย ที่ถูกออกแบบมาสำหรับการตรวจจับ DDoS ระบบจะทำการเรียนรู้เปรียบเสมือนเครือข่ายประสาทเทียม (Neural network) วิธีนี้ได้รับ การตรวจสอบกับชุดของการสร้างสถานการณ์ที่แตกต่างกัน แสดงให้เห็นว่าระบบที่มีเครือข่ายประสาทเทียมผ่านการเรียนรู้อย่างดีสามารถตรวจจับการโจมตี DDoS ได้อย่างมีประสิทธิภาพและประสบความสำเร็จ [6] ในปี 2017 มีการนำเสนอการวิเคราะห์ Big Data เพื่อตรวจจับความผิดปกติบนระบบเครือข่าย จาก Netflow data โดยใช้ PCA เพื่อวิเคราะห์พฤติกรรมของระบบเครือข่าย ซึ่งทั้งสองอย่างสามารถที่จะระบุตัวการโจมตีและเพิ่มความปลอดภัยในระบบเครือข่ายได้ วิธีข้างต้นจะใช้ Apache Spark Cluster ใน Azure HDInsight ได้ผลค่าความถูกต้องออกมามากถึงร้อยละ 96 [7] มีงานวิจัยที่ทำบนพื้นฐานของการคาดคะเน DDoS ตามรูปแบบ Periodic โดยการจำนวนของแพ็คเกต และจำแนกตาม Source IP ส่วน ARIMA Model และ BOXCOX เทคนิค ในที่นี้จะเป็นการแยก Time Series ออกจากกัน BOXCOX เทคนิคมีหน้าที่เตรียม Constant data ก่อนที่จะใช้หลักการ ARIMA Model ด้วยโฟล์วที่ทดสอบแล้วมากกว่า 100 โฟล์ว การโจมตีจะใช้เวลา 54 นาที ซึ่งสามารถตรวจจับได้ 51 นาที มากไปกว่านั้นค่าความแปรผันของ Packet Time Series ถูกตั้งให้มีค่าคงที่โดยการใช้ Boxcox technique ซึ่งผล accuracy ออกมาได้ 99.5% [1] และมีการตรวจจับ DDoS นั้นจะมีเทคนิคสำคัญที่ชื่อว่า “MDRA Algorithm” ใน Big Data Dataset ที่ชื่อ KDD CUP 1999 จะถูกนำมาใช้บน testbed และสรุปได้ว่า MDRA Algorithm ให้ผลที่ดีกว่า MCA Algorithm [6]

2.3 การวิเคราะห์ระบบเครือข่าย (Network Analysis)

การบริหารระบบเครือข่าย เปรียบเสมือนกรอบที่วางแนวทางที่ต้องการ หลังจากการกำหนดวัตถุประสงค์ให้ชัดเจนแล้ว ขั้นตอนการปฏิบัติเพื่อให้บรรลุวัตถุประสงค์คือการวิเคราะห์ระบบเครือข่าย การวิเคราะห์ระบบเครือข่าย มีวัตถุประสงค์เพื่อรักษาระดับความน่าเชื่อถือ (Reliability) และบำรุงรักษาให้ระบบสามารถใช้งานได้ตามปกติ (Availability) องค์ประกอบที่สำคัญสองประการคือ ค่าสถิติเกี่ยวกับการใช้งานและการปรับปรุงระบบเครือข่ายจะถูกนำมาใช้ในระบบเครือข่ายให้เป็นไปอย่างมีประสิทธิภาพสูงสุด ข้อมูลเกี่ยวกับระบบเครือข่ายจะถูกเก็บรวบรวมไว้ในรูปของค่าสถิติเกี่ยวกับการใช้งานระบบเครือข่าย (Network Statistics) เพื่อนำมาใช้ประกอบการวิเคราะห์ประสิทธิภาพ ข้อมูลสถิติจะถูกเก็บรวบรวมผ่านทางฮาร์ดแวร์เรียกว่าอุปกรณ์ตรวจสอบประสิทธิภาพ (Performance Monitor) ซึ่งจะใช้ซอฟต์แวร์ในการสร้างรายงานสรุปผลและแสดงภาพกราฟิกของระบบในขณะที่กำลังทำงาน

ซอฟต์แวร์บันทึกเหตุการณ์ (Log files) ทำหน้าที่เก็บรวบรวมข้อมูลที่เกิดขึ้นจริงในระบบเครือข่ายไว้เพื่อการวิเคราะห์ในภายหลัง เหตุการณ์ที่ทำการบันทึกมีหลายประเภท เช่น บันทึกรายการทำงาน (Transaction logs) บันทึกการส่งข้อความ (Message logs) และบันทึกเส้นทางเดินข้อมูล (Line traces) โดยปกติข้อมูลที่จะสามารถนำไปใช้ในการวิเคราะห์ได้อย่างมีประสิทธิภาพนั้นจะต้องมีปริมาณมากพอสมควร ดังนั้นจึงต้องจัดเตรียมอุปกรณ์บันทึกข้อมูลไว้ให้สามารถเก็บข้อมูลปริมาณมาก โดยไม่กระทบต่อการทำงานอื่นของระบบเครือข่าย

การวิเคราะห์ระบบเครือข่ายมักเกี่ยวข้องกับข้อมูลของเหตุการณ์หลายชนิดที่เกิดขึ้นในระบบเครือข่าย เช่น ความเร็วในการถ่ายทอดข้อมูล ความถี่ในการเกิดข้อมูลผิดพลาด และความถี่ในการถ่ายทอดข้อมูลซ้ำ การวิเคราะห์ระบบในเบื้องต้นสามารถดูได้จากสิ่งบอกเหตุต่าง ๆ ซึ่งระยะเวลาการตอบสนองควรจะอยู่ภายในขอบเขตที่กำหนด ถ้าหากระยะเวลาตอบสนองสูงเกินกว่าขอบเขต จะต้องเพิ่มการตรวจสอบเกี่ยวกับระยะเวลาให้มากขึ้น และถ้าเกิดขึ้นเป็นประจำแสดงว่ามีปัญหาเกิดขึ้นแล้ว โดยปกติข้อมูลที่ถ่ายทอดผ่านระบบเครือข่ายจะเป็นแบบผสม (Transaction mix) ซึ่งหมายถึงข้อมูลหลากหลายรูปแบบมีจุดส่งและเป้าหมายแตกต่างกันออกไป การตรวจดูอัตราส่วนประเภทของข้อมูลก็เป็นสิ่งที่นักวิเคราะห์ควรให้ความสนใจ ผู้บริหารระบบเครือข่ายต้องตรวจสอบการใช้ประโยชน์วงจรสื่อสาร (Circuit utilization) ผ่านซอฟต์แวร์ตรวจสอบระบบเครือข่าย ปริมาณข้อมูลที่ส่งผ่านสายสื่อสารเป็นตัวบอกการใช้ที่เกิดขึ้น วงจรสื่อสารที่มีการใช้มากเกินไปนอกจากจะทำให้เกิดปัญหาในปัจจุบันแล้วยังอาจทำให้เกิดปัญหารุนแรงในอนาคตก็ได้ โปรแกรมบางชนิดอาจถูกเรียกใช้ไม่บ่อยนัก แต่ว่าเมื่อถูกเรียกใช้จะทำให้เกิดการถ่ายเทข้อมูลปริมาณมาก สามารถใช้โปรแกรมตรวจสอบระบบในการหาโปรแกรมและความถี่ในการใช้งานเพื่อจัดการแก้ไขได้

ซอฟต์แวร์ตรวจสอบระบบเครือข่ายสามารถใช้บอกตำแหน่งที่เกิดปัญหาในการหาเส้นทางการสื่อสารข้อมูล ซอฟต์แวร์นี้จะค้นหาโหนดที่ส่งข้อมูลออกไปผิดเส้นทางเพื่อจะได้แก้ไขให้ทำงานได้อย่างถูกต้อง ข้อมูลที่ถูกส่งออกไปผิดเส้นทางอาจทำให้เกิดปัญหาอื่นตามมา เช่น ข้อมูลนั้นจะอยู่ในระบบนานขึ้น ทำให้ระยะเวลาในการตอบสนองนานเกินความเป็นจริง ข้อมูลอาจสูญหายทำให้ต้องส่งข้อมูลนั้นซ้ำ ข้อมูลอาจถูกส่งผิดเส้นทางเนื่องจากเส้นทางบางส่วนไม่สามารถใช้การได้หรือมีการใช้งานสูงมากผิดปกติ โหนดที่ทำหน้าที่ค้นหาเส้นทางให้กับโหนดอื่นๆ อาจล้มเหลว ทำงานผิดปกติ หรือการจัดตั้งระบบเครือข่ายมีความผิดพลาดเกิดขึ้น จึงเป็นหน้าที่ของผู้ชำนาญการระบบเครือข่ายจะต้องจัดการแก้ไขปัญหานี้

อัตราการเกิดข้อมูลผิดพลาดที่เกิดขึ้นในส่วนหนึ่งของระบบเครือข่ายจะส่งผลกระทบต่อประสิทธิภาพโดยรวมของระบบเครือข่าย การตรวจข้อมูลผิดพลาด ที่เกิดขึ้นเป็นประจำหรือเกิดขึ้นถี่มากกว่าปกติจะช่วยให้พบจุดที่เกิดปัญหา ปัญหาหลายอย่างที่เกิดขึ้นในระบบเครือข่ายอาจเป็นที่โปรแกรมที่เลือกใช้อาจใช้เวลาในการทำงานนานมาก ดังนั้นระยะเวลาการตอบสนองที่ยาวนานจึง

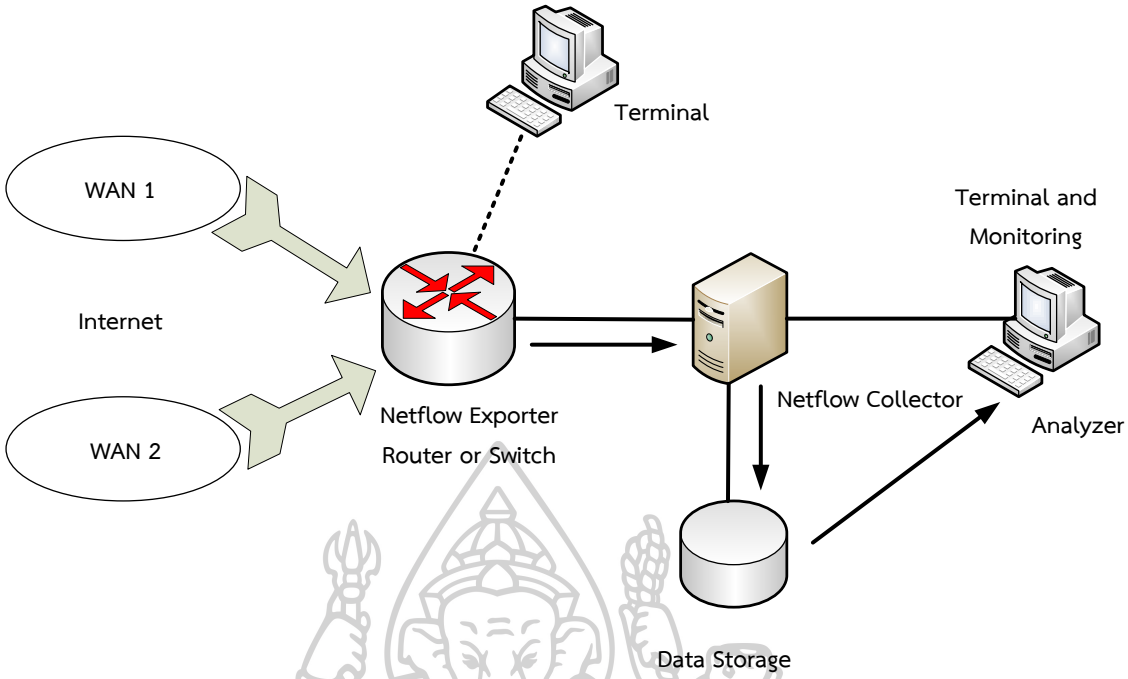
ไม่ใช่ปัญหาที่เกิดจากระบบเครือข่ายหรือข้อมูล ดังนั้นการตรวจสอบโปรแกรมการวิเคราะห์ระบบเครือข่ายที่ใช้งานจึงเป็นส่วนหนึ่งที่จะต้องทำเพื่อค้นหาสาเหตุของปัญหา

ประสิทธิภาพของเครือข่ายสามารถประเมินได้จาก การวัดประสิทธิภาพระบบเครือข่ายด้วย Confusion Matrix เป็นการประเมินผลลัพธ์การจากโปรแกรมเปรียบเทียบกับผลลัพธ์จริง

- True Positive (TP) คือ สิ่งที่โปรแกรมทำนายว่าจริง และผลลัพธ์เป็นจริง
- True Negative (TN) คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง และผลลัพธ์ไม่จริง
- False Positive (FP) คือ สิ่งที่โปรแกรมทำนายว่าจริง แต่ผลลัพธ์ไม่จริง
- False Negative (FN) คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง แต่ผลลัพธ์เป็นจริง

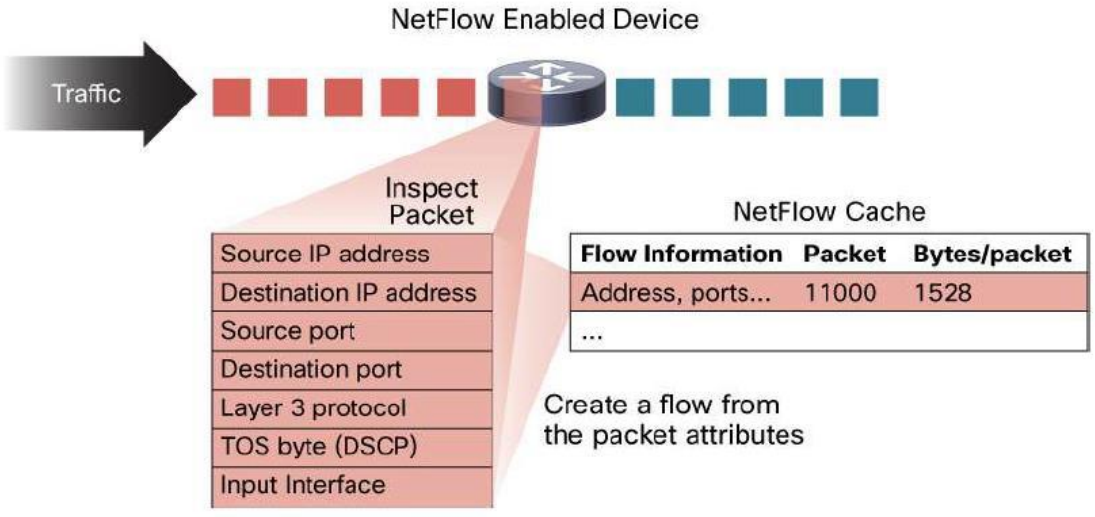
ในการนำข้อมูลมาวัดประสิทธิภาพของเครือข่ายใช้ข้อมูลการจราจร (log file) บนระบบเครือข่าย UniNet ซึ่งจำเป็นต้องใช้แอฟพลิเคชันในการสร้างและจัดเก็บ log file คือ NetFlow

Netflow เป็นเครื่องมือที่มีอยู่ในซอฟต์แวร์ Cisco IOS ซึ่งเป็นเครื่องมือสาหรับตรวจสอบและเฝ้าระวังของปริมาณทราฟฟิก (Traffic) ที่วิ่งอยู่บนระบบเครือข่าย ซึ่ง Netflow [8] ถูกออกแบบมาเพื่อการประยุกต์ใช้และการใช้งานในระบบเครือข่าย NetFlow คือเทคโนโลยีการเฝ้าดูรูปแบบของทราฟฟิก ได้รับการพัฒนาโดย Darren Kerr และ Barry Burins ของ Cisco Systems ในปี 1996 NetFlow ได้อธิบายวิธีการสำหรับRouter (Router) ในการส่งสถิติเกี่ยวกับความสัมพันธ์เป็นคู่ของ Routed Socket ออกมา และบรรจุเข้าไปในRouter ของ Cisco ทั้งหมด ความสามารถของ NetFlow คืออธิบายลักษณะการจราจรของ IP Address และสามารถเข้าใจวิธีการไหลของข้อมูลในระบบเครือข่าย การนำ NetFlow มาใช้งานในระบบช่วยให้การตรวจสอบการจราจรในระบบเครือข่ายได้รับความสะดวกยิ่งขึ้น



ภาพที่ 2.4 สถาปัตกรรม Netflow

ภาพที่ 2.4 แสดงถึงสถาปัตยกรรมการเชื่อมต่อและการไหลของข้อมูลโฟล์วที่ได้รับจากอุปกรณ์ต่างๆ ข้อมูลจะถูกส่งเข้ามาจากแหล่งต่าง ๆ มายัง NetFlow collector เพื่อ Dump ข้อมูลเหล่านั้นเข้าสู่ฐานข้อมูลเพื่อรอให้ Netflow analyzer ทำการดึงข้อมูลในส่วนนี้ไปทำการตรวจสอบการใช้งาน



ภาพที่ 2.5 ภาพแสดง NetFlow Cache [8]

ภาพที่ 2.5 แสดงการสร้างโฟล์วใน NetFlow Cache เป็นการสร้างแพ็คเกตทั้งหมดที่มี Source address เดียวกัน / Destination address เดียวกัน / พอร์ตปลายทางโปรโตคอล อินเตอร์เฟสและระดับของบริการที่จัดกลุ่มเป็นโฟล์วแล้ว จำนวนแพ็คเกตและไบต์จะมีมาก วิธีการตรวจสอบหรือการพิจารณาโฟล์วจะสามารถตรวจสอบได้ไม่ว่าข้อมูลจะมีขนาดใหญ่ โดยข้อมูลในเครือข่ายจะรวมตัวเป็นฐานข้อมูล NetFlow ที่เรียกว่า NetFlow Cache

Flow เป็นความต่อเนื่องไปของแพ็คเกตในทิศทางเดียว คือมีทิศทางของการไหลของข้อมูลในทิศทางใดทิศทางหนึ่ง เช่น ข้อมูลจาก Server ไปยัง Client และข้อมูลจาก Clientไปยัง Server ระหว่างสองจุดปลายทาง การระบุโฟล์วสามารถระบุจากส่วนย่อยหลัก 7 ส่วน เรียกว่า “Key-field” ดังต่อไปนี้

1) Source IP Address คือ IP Address ของเครื่องส่งข้อมูล

2) Destination IP Address คือ IP Address ของเครื่องที่รับข้อมูล

3) Source port คือ พอร์ตของเครื่องส่งข้อมูล

4) Destination port คือ พอร์ตของเครื่องที่รับข้อมูลุ

5) Layer 3 protocol type คือ ประเภทของโปรโตคอลที่ใช้ในการรับส่งข้อมูล

6) TOS bytes คือ ข้อตกลงการใช้บริการ เช่น DHCP

7) Input logical interface (ifIndex) คือ เลข index ของแต่ละ interface บนอุปกรณ์

หลังจากรับแพ็คเกตเข้ามา ตัวRouterจะตรวจสอบจาก 7 ส่วนนี้ หลังจากนั้นจะตั้งเงื่อนไขว่าถ้าแพ็คเกตเป็นสมาชิกของโฟล์วที่มีอยู่ในขณะนั้น สถิติรวมทราฟฟิกของโฟล์วที่สอดคล้องกันจะมีการเพิ่มขึ้น ไม่เช่นนั้นแล้วโฟล์วใหม่จะถูกสร้างขึ้นแทนโดยแนวความคิดของเทคโนโลยีของ Cisco, โฟล์วใหม่ๆ จะถูกสร้างขึ้นอย่างต่อเนื่องเมื่อโฟล์วที่บันทึกไว้หมดอายุลงมันจะถูกส่งออกมาเป็น UDP Packet ไปยังสถานีเฝ้าดูหรือตัวเลือกรับข้อมูลที่ผู้ใช้งานได้กำหนดไว้ ถ้าเงื่อนไขดังต่อไปนี้ได้เกิดขึ้น เงื่อนไขการหมดอายุของโฟล์วนี้คือ

1) Transport Protocol ได้ระบุว่าการเชื่อมต่อนั้นได้เสร็จสมบูรณ์แล้ว (TCP FIN) และมีความล่าช้าเล็กน้อยที่ยอมรับได้สำหรับความสำเร็จของการประสานงานกันเพื่อรับรู้และยอมรับกระบวนการการเชื่อมต่อที่สมบูรณ์

2) ทราฟฟิกไม่มีการเคลื่อนไหวเกิน 15 นาที

3) สำหรับโฟล์วที่มีการเคลื่อนไหวอย่างต่อเนื่อง การบันทึกโฟล์วของหน่วยความจำสำรองจะหมดอายุลงทุกๆ 30 นาที มีการนำ NetFlow มาใช้งานแต่ละรุ่นอย่างหลากหลาย ทุก UDP

Datagram บรรจุไปด้วย Flow Header และ 30 Flow records ทุกๆ Flow records จะสร้างไว้หลายๆส่วน ซึ่งประกอบด้วย

1) หมายเลขตำแหน่งของ IP ต้นทางและปลายทาง
2) หมายเลขตำแหน่งของ Next Hop
3) หมายเลข Interface ขาเข้าและขาออก
4) จำนวนของ Packet ในโฟล์วนั้น
5) จำนวน Bytes สุทธิของโฟล์วนั้น
6) Source Port and Destination Port
7) Protocol
8) Type of Service
9) หมายเลข AS ต้นทางและปลายทาง และ TCP flags (ตัวเดียวหรือหลายตัวรวมกันของ TCP flags)

บนเครื่องรับข้อมูล (Collector) การวิเคราะห์โฟล์วที่ได้รับมาจำเป็นต้องดำเนินการตามเวลาจริงคือกระทำทันที ในNetFlow export datagram บรรจุไว้ด้วยส่วนเริ่มต้น (Header) และข้อมูลในการเรียงลำดับของบันทึกที่เข้ามา ส่วนเริ่มต้นประกอบไปด้วยข้อมูลอย่างเช่น Sequence number ,Record count , และ System uptime , ข้อมูลการไหลเวียนที่บันทึกจะประกอบไปด้วยข้อมูลข่าวสารของการไหลเวียน ตัวอย่างเช่น IP Address, Ports และ Routing information รุ่นที่ 1 มีรูปแบบที่เริ่มต้นออกมารองรับพื้นฐานของ Cisco IOS software ซึ่งบรรจุไปด้วยหน้าที่ของ NetFlow และไม่ค่อยจะใช้บ่อยนักในปัจจุบัน รุ่นที่ 5 มีรูปแบบที่พัฒนาเพิ่มขึ้นโดยเพิ่มในส่วนของข้อมูล Border Gateway Protocol (BGP) Autonomous system และข้อมูลหมายเลขลำดับของการไหลเวียน รุ่นที่ 7 มีรูปแบบที่พัฒนาเพิ่มขึ้นโดยเพิ่มในส่วนการรองรับการทำงานของ Cisco Catalyst Switches ตระกูลที่ใช้งานเป็น Hybrid หรือ Native mode รุ่นที่ 2 ถึงรุ่นที่ 4 และรุ่นที่ 6 ไม่ได้ถูกนำออกมาใช้งานหรือไม่ได้รับการยอมรับ รุ่นที่ 8 มีรูปแบบที่ออกใช้เมื่อเราใช้งานจำเพาะบน Routerที่มีการร่วมกันทางานกับ Cisco IOS router ด้วยกันเอง และรุ่นปัจจุบันคือรุ่นที่ 9เป็นรุ่นที่มีใช้บนRouterรุ่นล่าสุดส่วนมากในรายงานโฟล์ว เช่น IPv6, MPLS หรือแม้กระทั่งIPv4

Byte 3	Byte 2	Byte 1	Byte 0
Source IP address			
Destination IP address			
Next-hop IP address			
Input ifIndex		Output ifIndex	
Packets			
Bytes			
Start time of flow			
End time of flow			
Source port		Destination port	
Padding	TCP flags	IP protocol	TOS
Source AS		Destination AS	
Source mask length	Dest. mask length	Padding	

ภาพที่ 2.6 แสดงส่วนประกอบของ Flow record [7]

จากภาพที่ 2.6 ส่วนประกอบของ Flow record ประกอบด้วย ดังนี้

- srcaddr หมายเลข IP address ต้นทาง
- dstaddr หมายเลข IP address ปลายทาง
- nexthop หมายเลข IP address ของRouterตัวถัดไปของโฟล์วนั้น
- input ตัวบ่งชี้ SNMP ของ interface ที่รับข้อมูล
- output ตัวบ่งชี้ SNMP ของ interface ที่ส่งข้อมูล
- dPkys ปริมาณแพ็คเกตในการส่งข้อมูล
- dOctets จำนวนสูงสุดของ Layer 3 นับเป็น Bytes ในปริมาณแพ็คเกตของการส่งข้อมูล
- first เวลาเริ่มต้นตั้งแต่ส่งข้อมูล
- last เวลานับตั้งแต่ที่ได้รับแพ็คเกตล่าสุดเข้ามาของข้อมูลที่ได้รับแล้ว
- srcport หมายเลข port ต้นทางของ TCP/UDP หรือที่มีความหมายเหมือนกัน
- dstport หมายเลข port ปลายทางของ TCP/UDP หรือที่มีความหมายเหมือนกัน

- pad 1 ไม่ถูกใช้งาน (มีค่าเป็น 0)
- tcp_flags Cumulative OR ของ TCP flags
- Protocol หมายเลขชนิดของโปรโตคอล (ตัวอย่างเช่น TCP = 6 ; UDP = 17) Tos IP type of service (ToS)
- Src_as หมายเลข AS ของต้นทาง จะเป็น origin หรือ peer อย่างใดอย่างหนึ่ง
- dst_as หมายเลข AS ของปลายทาง จะเป็น origin หรือ peer อย่างใดอย่างหนึ่ง
- src_mask prefix bits ของ address ต้นทาง
- dst_mask prefix bits ของ address ปลายทาง
- pad2 ไม่ถูกใช้งาน (มีค่าเป็น 0)

งานวิจัยที่เกี่ยวข้องกับการจัดการเครือข่าย

ในปี 2018 มีการนำเสนอการคาดการณ์การถูกโจมตีจาก DDoS และการตรวจจับที่ได้ประสิทธิภาพเร็วกว่า โดยการใช้ MapReduce และ Time Series ซึ่งการวิเคราะห์นี้จะเป็นประโยชน์ต่อการวิเคราะห์พฤติกรรมที่ผิดปกติ ขั้นตอนแรก MapReduce จะถูกใช้ในการแยก Log file ออกมาเป็น Time Series และใน Time Series นั้นจะแบ่งเป็น window ให้มีการแบ่งเพียง 5 นาที ซึ่งเราสามารถเลือกช่วงที่มีการ access และ traffic ที่มีปริมาณมากเข้ามายัง Server ต่อมาจะได้ IP Address และขนาดของข้อมูล จะถูกนำมาหาค่าความปกติของเครือข่าย ทำให้สามารถหยุดการทำงานของระบบก่อนที่ระบบจะ Shut down ด้วยเหตุนี้การวิเคราะห์ Time series ตามรูปแบบ MapReduce สามารถสรุปผลออกมาได้ว่า ต้นแบบที่มีการใช้ MapReduce ให้ผลลัพธ์ที่ดีกว่าต้นแบบที่ไม่ทำงานร่วมกับ MapReduce [9]

2.4 การจัดการข้อมูลขนาดใหญ่

การจะนำเอาข้อมูล Big Data [10] มาใช้งานให้เกิดประโยชน์ต้องมีความพร้อมหลายด้าน อาทิเช่น ด้านเทคโนโลยี ด้านเทคนิค และด้านบุคลากรเพื่อให้ได้รับประโยชน์อย่างเป็นรูปธรรมจากข้อมูล Big Data เราจำเป็นต้องมีเครื่องมือที่เหมาะสมสำหรับการบันทึก และจัดระเบียบข้อมูลหลายประเภทจากแหล่งต่างๆ และต้องสามารถวิเคราะห์ข้อมูลดังกล่าวได้

การจัดการข้อมูลที่เป็น Big Data มีองค์ประกอบดังต่อไปนี้

(1) การจัดเก็บ (Storage)

(2) การประมวลผล (Processing)

(3) การวิเคราะห์ (Analysis Algorithm)

(4) การทำรายงานสรุป (Visualization)

ในคลังข้อมูลแบบเก่าการจัดระเบียบข้อมูลจะเป็นการผนวกรวมข้อมูลเข้าด้วยกัน (Data Integration) แต่ในปัจจุบันข้อมูล Big Data มีจำนวนมหาศาล จึงมีแนวโน้มที่จะทำการจัดระเบียบข้อมูลในตำแหน่งที่จัดเก็บดั้งเดิมเพื่อประหยัดทั้งเวลาและค่าใช้จ่าย เพราะไม่ต้องย้ายข้อมูลจำนวนมหาศาลไปมา โครงสร้างพื้นฐานที่จำเป็นสำหรับการจัดระเบียบข้อมูล Big Data ต้องสามารถประมวลผลและจัดการข้อมูลในตำแหน่งที่จัดเก็บเดิม โดยรองรับอัตราการรับส่งที่สูงมากเพื่อจัดการกับขั้นตอนการประมวลผลข้อมูลจำนวนมาก และจัดการกับข้อมูลหลากหลายรูปแบบ ตั้งแต่ข้อมูลที่ไม่มีโครงสร้างไปจนถึงข้อมูลที่มีโครงสร้าง

การเคลื่อนย้ายกลุ่มของข้อมูลเข้าสู่ระบบฐานข้อมูลเพื่อการประมวลผลและนำไปเก็บในส่วนจัดเก็บข้อมูล ซึ่งจะเก็บปริมาณข้อมูลมหาศาลที่ไม่สามารถเคลื่อนย้ายได้ ในปัจจุบันได้มีการพัฒนาซอฟต์แวร์เข้ามาช่วยจัดการปริมาณข้อมูลมหาศาลนี้

การพัฒนาอย่างต่อเนื่องของระบบเทคโนโลยีสารสนเทศและการสื่อสารทำให้องค์กรต่าง ๆ มีความสามารถในการเก็บข้อมูลในด้านต่าง ๆ ได้มหาศาล ซึ่งเมื่อนำข้อมูลเหล่านั้นมาวิเคราะห์และประมวลผล จะมีประโยชน์ต่อการตัดสินใจของผู้บริหารองค์กร ดังนั้น Big Data จึงเป็นแนวคิด ที่จะช่วยให้เกิดการบริหารจัดการข้อมูลให้ได้ประโยชน์สูงสุดอย่างมีประสิทธิภาพ

Big Data คือการนำข้อมูลที่มีปริมาณมาก มาผ่านการประมวลผลการวิเคราะห์ และแสดงผลด้วยวิธีที่เหมาะสมไม่ว่าจะเป็นข้อมูลด้านการเงิน ข้อมูลการดำเนินงานข้อมูลเกี่ยวกับผู้รับบริการ ข้อมูลเกี่ยวกับบุคลากร รวมไปถึงข้อมูลที่ได้มีการจัดเก็บในระบบฐานข้อมูลซึ่งจะมีปริมาณที่เพิ่มมากขึ้นเรื่อย ๆ จนมหาศาลทำให้ไม่สามารถใช้วิธีการจัดการทั่วไปได้อย่างมีคุณภาพ จึงต้องใช้แนวคิด Big Data ในการจัดการ

Big Data มีประโยชน์อยู่ 2 ประการ คือ

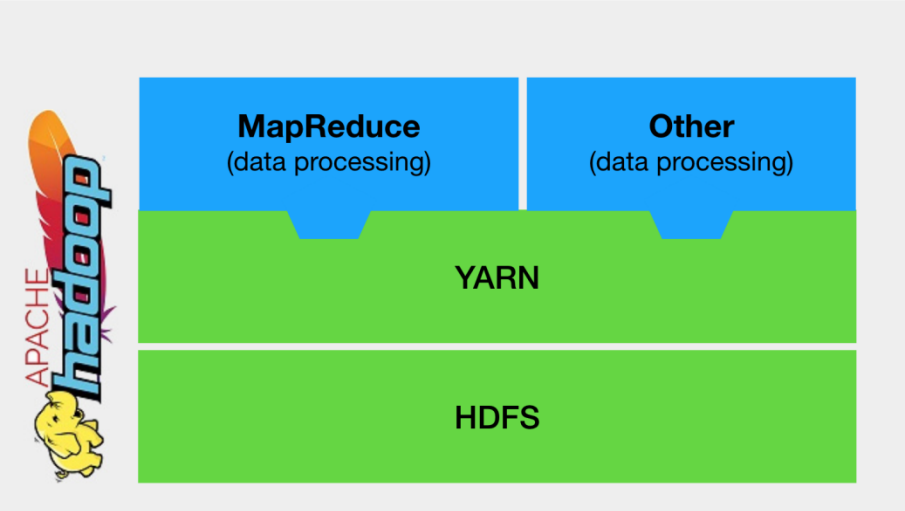
1. การวิเคราะห์ข้อมูลที่ทำให้เห็นความรู้ที่ซ่อนอยู่ เช่น ข้อมูลสภาพอากาศ ข้อมูลการจราจร ข้อมูลทางการศึกษา ข้อมูลทางการแพทย์ เป็นต้น

2. การเกิดผลิตภัณฑ์หรือบริการใหม่ ๆ ที่เหมาะสมตามความต้องการของผู้ใช้ จะเห็นได้ว่าข้อมูลด้านต่าง ๆ ที่กระจัดกระจายที่มีอยู่มากเมื่อนำเอาแนวคิด Big Data มาวิเคราะห์ประมวลผล

ทำให้เกิดประโยชน์ต่อองค์กรและผู้รับบริการ ด้วยจำนวนปริมาณข้อมูลที่มีมากมหาศาลในการจัดเก็บข้อมูลไม่สามารถใช้ฐานข้อมูลเดิมได้จึงนำฐานข้อมูล NoSQL มาใช้แทน เช่น MySQL Cluster, Amazon RDS, Azure SQL, MongoDB หรือ Cassandra และเครื่องมืออย่าง Hadoop ที่ใช้สำหรับจัดการ Unstructured Data ที่เป็น PetaByte ซอฟต์แวร์ที่นำมาใช้ในการจัดการข้อมูลขนาดใหญ่เช่น Hadoop ซึ่งเป็น Software แบบ Open Source ที่ได้รับการออกแบบมาเพื่อทำงานบนระบบคอมพิวเตอร์แบบกระจาย (Distributed computing) และสนับสนุนการทำงานแบบขนาน (Parallel) โดยมีชุดคำสั่ง (API) ที่ช่วยอำนวยความสะดวกในการสร้างระบบค้นหาหรือวิเคราะห์ข้อมูลขนาดใหญ่ให้แก่นักพัฒนาแอปพลิเคชัน [11]

Hadoop [12] เป็น Open source Project ของ Apache สำหรับการเก็บและบริหารข้อมูลขนาดใหญ่ พัฒนาด้วยโปรแกรมภาษาจาวา มีความสามารถในการทนทานต่อผิดพลาด (Fault Tolerance) เนื่องจากจะเก็บข้อมูลซ้ำกันในหลายๆที่ และเป็นระบบที่เป็น Horizontal Scale ที่รันบนเครื่อง commodity server จำนวนมาก Hadoop ประกอบไปด้วยส่วนประกอบหลักที่สำคัญอยู่ 4 ส่วน ได้แก่

1. Hadoop Common เป็น libraries และ utilities ส่วนกลางที่ช่วยสนับสนุนการทำงานของ modules อื่นๆ ใน Hadoop
2. Hadoop Distributed File System (HDFS) เป็นระบบไฟล์แบบกระจายที่ช่วยทำให้ผู้ใช้สามารถจัดการกับไฟล์ขนาดใหญ่ได้อย่างสะดวกและรวดเร็ว
3. Hadoop YARN (Yet Another Resource Negotiator) เป็น Framework ที่ใช้ในการจัดการ Job scheduling และบริหารจัดการทรัพยากรต่างๆบนระบบ Hadoop cluster
4. Hadoop MapReduce ซึ่ง MapReduce เป็น programming model หรือ programming paradigm ที่ออกแบบมาเพื่อเขียนโปรแกรมสำหรับการประมวลผลแบบขนาน คือสามารถเขียนโปรแกรมเพื่อให้ทำงานบนเครื่องคอมพิวเตอร์หลายๆเครื่องพร้อมกัน ในระบบ Hadoop cluster ทำให้การประมวลผลมีความรวดเร็วขึ้น



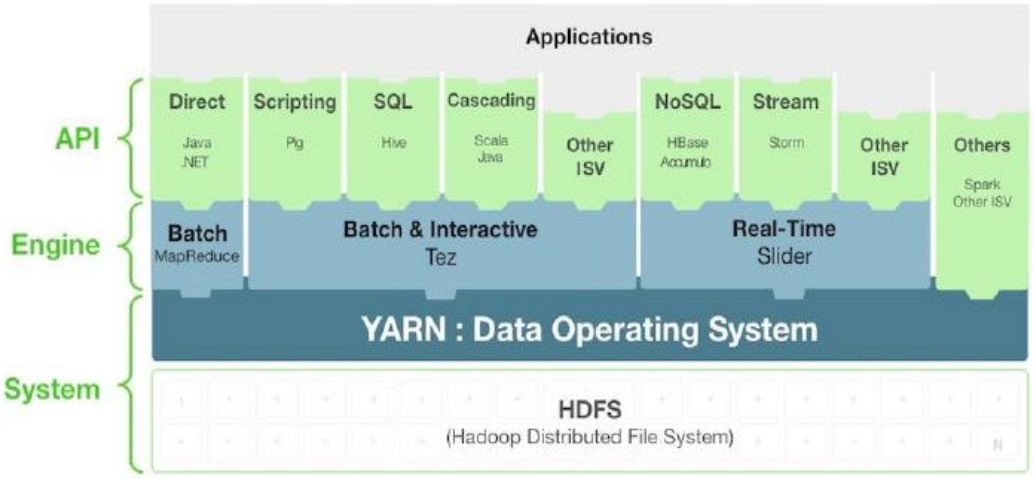
ภาพที่ 2.7 Hadoop Architecture [13]

HDFS (Hadoop Distributed File System) [14] เป็นระบบไฟล์ที่ทำหน้าที่บริหารจัดการไฟล์ข้อมูลภายในระบบ Cluster ของ Hadoop ซึ่งจะทำการแบ่งไฟล์ออกเป็นชิ้นๆ (block) แล้วกระจาย (distribute) แต่ละ block ไปเก็บในเครื่องต่างๆใน cluster และทำการสำเนาแต่ละ block ไว้ในเครื่องต่างๆ ในระบบ cluster เพื่อการันตีว่าถ้าเกิดเหตุการณ์ที่เครื่องใดเครื่องหนึ่งในระบบ cluster เกิดขัดข้อง Hadoop จะยังสามารถเข้าถึง block นั้น ได้จากเครื่องอื่นที่ทำสำเนาไว้และยังสามารถใช้งานต่อไปได้ การใช้งาน ผู้ใช้ระบบจะมองเห็น drive หรือเข้าถึงไฟล์ใน HDFS ด้วย location เพียงที่เดียว ผู้ใช้ไม่จำเป็นต้องรู้ว่าในระบบ cluster มีจำนวนเครื่องคอมพิวเตอร์มากน้อยเพียงใด เมื่อเก็บไฟล์ไว้มากจนเหลือพื้นที่จัดเก็บข้อมูลน้อยลงก็สามารถเพิ่มเครื่องคอมพิวเตอร์เครื่องใหม่เข้าไปใน Hadoop cluster (Horizontal Scaling) เพื่อทำให้พื้นที่จัดเก็บไฟล์เพิ่มโดยไม่ต้องทำการ shut down ระบบ

สถาปัตยกรรมฮาร์ดแวร์ของระบบ Hadoop จะประกอบด้วยเครื่อง Server จำนวนมาก โดยจะมีเครื่องหนึ่งทำหน้าที่เป็น Master และจะมีเครื่องลูกอีกจำนวนมากทำหน้าที่เป็น Slave โดยปกติ Hadoop จะกำหนดให้ข้อมูลที่เก็บในเครื่อง Slave มีการเก็บข้อมูลซ้ำกันสามแห่ง ดังนั้นเครื่อง Slave ควรจะมีอย่างน้อยสามเครื่อง ส่วนเครื่อง Master ก็จะทำหน้าที่หลักในการระบุตำแหน่งของข้อมูลและ Task ที่กระจายในการประมวลผลของ Map/Reduce ดังนั้นเครื่อง Master จึงมีความสำคัญอย่างมาก และต้องมีเครื่อง Secondary Master ในการที่จะสำรองไว้ ในกรณีเครื่อง Master ตายไป ดังนั้นระบบ Hadoop โดยทั่วไปจะเริ่มต้นที่เครื่อง Server 5 เครื่อง สำหรับ Master หนึ่งเครื่อง, Secondary Master หนึ่งเครื่อง และ Slave สามเครื่อง โดยหากต้องการเก็บข้อมูลมากขึ้นหรือต้องการประมวลผลข้อมูลให้เร็วขึ้นจำเป็นต้องเพิ่มจำนวนเครื่อง Slave ให้มากขึ้น ทั้งนี้ขนาด

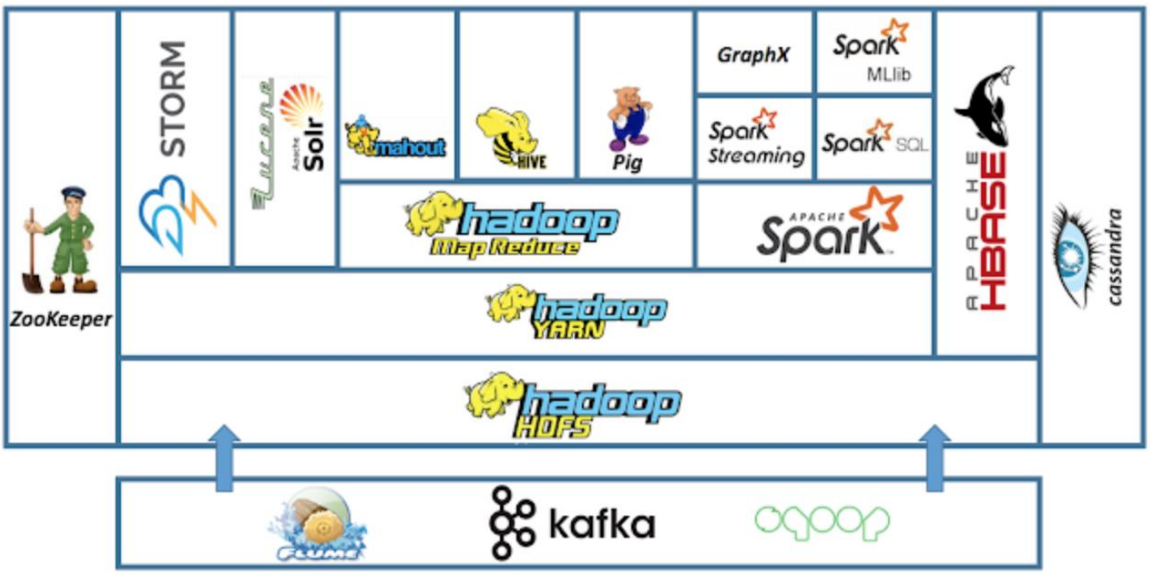
ของข้อมูลที่เก็บได้ก็จะขึ้นอยู่กับขนาดความจุข้อมูลของเครื่อง Slave รวมกันหารด้วยจำนวนข้อมูลที่ต้องการเก็บซ้ำ (default คือ 3) ซึ่งการเก็บข้อมูลจำนวนเป็น Petabyte ได้ต้องมีเครื่องเป็นจำนวนมากกว่าร้อยเครื่อง โดยปัจจุบัน Yahoo เป็น site ที่มี Hadoop Cluster ใหญ่ที่สุด โดยมีเครื่องจำนวนมากถึง 40,000 เครื่อง [15]

Hadoop เวอร์ชั่นแรกมีข้อจำกัดหลายประการอาทิเช่น ระบบการสำรองของ Secondary Master เป็นแบบ Passive และไม่สามารถทำ Multiple Master ได้จึงจำกัดเครื่อง Slave ไว้ไม่เกิน 4,000 เครื่อง และข้อสำคัญการประมวลผลต้องใช้ Map/Reduce ที่เป็นแบบ Batch ดังนั้นจึงมีการพัฒนา Hadoop 2.0 ที่จะลดข้อจำกัดต่างๆ Hadoop เวอร์ชั่น2.0 [13] จะมีสถาปัตยกรรมดังรูปที่ 2.8 โดยมีการนำ Data Operating System ที่เรียกว่า YARN (Yet Another Resource Negotiator) เข้ามาใช้เพื่อเพิ่มประสิทธิภาพ



ภาพที่ 2.8 แสดงคุณสมบัติที่เพิ่มขึ้นจาก Hadoop V.1 [16]

Ecosystem ของ Hadoop มี software หลายตัวที่ถูกพัฒนาขึ้นมาเป็น ecosystem ของ Hadoop โดย software เหล่านี้จะทำหน้าที่ดึงความสามารถในแต่ละด้านของ Hadoop ออกมาเพื่อให้เกิดประสิทธิภาพมากที่สุดในการทำงาน ดังที่แสดงในภาพที่ 2.9



ภาพที่ 2.9 แสดง Hadoop Ecosystem [13]

- ZooKeper ทำหน้าที่ในการบริหารจัดการในการรับส่งข้อมูลใน service ที่กระจายอยู่บน Hadoop Cluster
- STORM คือหน่วยประมวลผลที่คอยจัดการกับข้อมูลแบบ streaming หรือข้อมูลแบบ real-time
- Solr Lucene เป็นระบบที่ใช้ทำ full-text search หรือ search engine ที่ทำงานบน HDFS
- Mahout คือชุดเครื่องมือที่ใช้สำหรับจัดการงานทางด้าน machine learning และโมเดลทางคณิตศาสตร์ต่างๆ
- Hive เป็น SQL engine ที่ใช้ query ข้อมูลด้วยคำสั่งภาษา sql ทำงานบน MapReduce
- Pig เป็น platform ที่ใช้สำหรับบริการจัดการข้อมูล โดยมีคำสั่งหรือภาษาโปรแกรมที่เป็นระบบ High-level language ที่เข้าใจได้ง่าย อีกทั้งยังอาศัยความสามารถในการประมวลจาก MapReduce framework หรือการทำงานในโหมดของ Spark
- Spark เป็น Cluster computing framework ใช้สำหรับสร้าง application ในการประมวลผลข้อมูลขนาดใหญ่ มีหน้าที่เหมือนกับ MapReduce แต่ Spark ทำงานแบบ in-memory ไม่เหมือน MapReduce ที่ต้องใช้ disk ร่วมด้วย

- HBase คือระบบฐานข้อมูลแบบ non-RDBMS จัดเก็บข้อมูลลงบน HDFS และมีฟังก์ชั่นต่างๆที่อาศัยความสามารถในการประมวลผลจาก MapReduce
- Cassandra คือระบบฐานข้อมูลแบบ NoSQL ที่สามารถรับมือกับข้อมูลขนาดใหญ่และยังรองรับความสามารถในการขยายระบบ
- Flume เป็นเครื่องมือที่มีความสามารถในการจัดการกับ Log data ทั้งการทำ collection การทำ aggregation และการถ่ายโอนข้อมูลของ log ขนาดใหญ่ที่มีลักษณะเป็น streaming
- Kafka คือตัวช่วยสำหรับการรับและส่งข้อมูลแบบ real-time ที่สามารถจัดการกับข้อมูล streaming ขนาดใหญ่
- Sqoop เป็นตัวช่วยในการโอนย้ายข้อมูล ระหว่าง RDBMS และ Hadoop สามารถ import ข้อมูลจากฐานข้อมูลเดิมเข้ามายัง Hadoop หรือจะ export ข้อมูลจาก Hadoop ไปยังฐานข้อมูลได้

.

งานวิจัยที่เกี่ยวข้องกับการจัดการข้อมูลขนาดใหญ่

การศึกษาวิธีการแก้ปัญหาความไม่เพียงพอของแบนด์วิดท์ในเครือข่ายเมื่อมีการประมวลผลรูปแบบโปรแกรมสำหรับการคำนวณแบบกระจาย (MapReduce) เนื่องจากจะต้องใช้แบนด์วิดท์จำนวนมาก จึงอาจทำให้มีแบนด์วิดท์เหลือไม่เพียงพอต่อการทำงานในระบบเครือข่าย จึงมีการนำมาตรฐานเปิดที่อนุญาตให้ผู้ใช้งานสามารถนำโปรโตคอลที่สร้างขึ้นเองนำมาใช้งานได้ในระบบเครือข่าย (Openflow) เข้ามาช่วยเพื่อเพิ่มประสิทธิภาพในการทำงานของเครื่องมือในการจัดการกับข้อมูลขนาดใหญ่ที่ชื่อว่า Hadoop ในการทำ MapReduce ด้วยการที่มีระบบคอมพิวเตอร์มีจำนวนมากและอยู่สถานที่ต่างกันกลายเป็นเครือข่ายที่มีขนาดกว้าง เพื่อให้ระบบเครือข่ายที่มีขนาดกว้างสามารถใช้งานเกี่ยวกับการประมวลข้อมูลได้อย่างมีประสิทธิภาพและสามารถปรับเปลี่ยนได้ตามความต้องการ ซึ่งหากโครงสร้างของเครือข่าย สามารถปรับได้ตามต้องการของประสิทธิภาพที่ต้องใช้งานจะทำให้การประมวลผลบน Hadoop ประสบความสำเร็จ ในการออกแบบการทดลองจะทำการวางคอมพิวเตอร์ไว้หลายๆ สถานที่ ซึ่ง จะมีการใช้ซอฟต์แวร์ หรือฮาร์ดแวร์ที่ทำหน้าที่ส่งผ่านแพ็คเกต (OpenFlow Switch) เพื่อจัดลำดับความสำคัญของงานที่กำลังดำเนินการอยู่ ให้ได้รับความสำคัญที่สุด โดยจะควบคุมการเปลี่ยนแปลงเส้นทางการไหลแบบพลวัต (Dynamic) จากส่วนควบคุมเส้นทางการไหลของข้อมูล (Flow) เริ่มจากการตั้งค่าลำดับความสำคัญที่แตกต่างกัน และจำลองการทำงานใน

สภาวะที่มีความแออัด โดยเปรียบเทียบระหว่างการเปิดใช้งาน OpenFlow และไม่ได้เปิดใช้ OpenFlow ผลลัพธ์นี้ได้แสดงให้เห็นว่า งานที่ทำการเรียงลำดับความสำคัญนั้นมีประสิทธิภาพดีกว่า [17] ปี 2014 มีการใช้โปรโตคอลที่สร้างขึ้นเองโดยสามารถนำมาใช้งานได้ในระบบเครือข่าย (Openflow) และระบบบริหารจัดการเครือข่ายที่ใช้ซอฟต์แวร์ซึ่งทำหน้าที่จัดการ (Configure) และควบคุมการทำงานของระบบเครือข่ายได้แบบอัตโนมัติจากจุดเดียว (SDN) เพื่อทำการปรับปรุงประสิทธิภาพในการทำงานของเครื่องมือในการจัดการกับข้อมูลขนาดใหญ่ที่ชื่อว่า Hadoop ในการประมวลผลให้เสร็จสมบูรณ์ เมื่อเกิดข้อผิดพลาดขึ้น โดยทำการใช้การเปลี่ยนเส้นทาง โดยใช้หลักการกำหนดระยะเวลาในการไม่ได้รับสัญญาณการเชื่อมต่อ (Heart Beat) จากเครื่องคอมพิวเตอร์ที่เชื่อมต่อกัน โดยมีการทดสอบคือทำการประมวลผลไปที่ 50 เปอร์เซ็นต์ แล้วสั่งให้เกิดความผิดพลาดขึ้นเพื่อทำการสลับเส้นทาง (Switch) ไปยังเครื่องคอมพิวเตอร์สำรองเพื่อให้การประมวลผลนั้นสามารถทำงานได้จนเสร็จสมบูรณ์ ซึ่งจะช่วยลดความผิดพลาดในการส่งต่อข้อมูลของ Hadoop และช่วยลดความล่าช้าในการกู้คืนที่เกิดจากความล้มเหลวได้ถึง 99 เปอร์เซ็นต์ [11] ในการสร้างการทนทานต่อความล้มเหลว (Fault Tolerance) ในรูปแบบโปรแกรมสำหรับการคำนวณแบบกระจาย (MapReduce) โดยอาศัยหลักการทำงานบนยูดีพี (UDP) แต่ละเครื่องคอมพิวเตอร์จะมีหมายเลขประจำตัวไม่ซ้ำกัน เราเรียกมันว่า node ID ซึ่งถูกสุ่มมาจากการกำหนดค่า (DHT algorithm) เพื่อช่วยในเรื่องการทำงานให้สำเร็จ หากมีคอมพิวเตอร์หลักที่ทำหน้าที่ควบคุม (Master node) เกิดความเสียหาย หรือไม่สามารถใช้งานได้ จะการสลับไปยัง Master node ที่เตรียมพร้อมอยู่ โดยที่ผู้ใช้ไม่สามารถทราบเลยว่าเกิดความล้มเหลวขึ้น [18] ซึ่งมีขั้นตอนในการทดลองคือ ทำการสร้างเครือข่ายของคอมพิวเตอร์ตัวหลัก (Distributed master network) ขึ้นมาและใช้หลักการของ DHT algorithm ทำการเชื่อมตัวของคอมพิวเตอร์หลักบนระบบเครือข่าย หากคอมพิวเตอร์หลักไม่สามารถใช้งานได้คอมพิวเตอร์หลักตัวสำรองก็ทำงานเป็นคอมพิวเตอร์หลักแทน โดยหลักการที่จะทำคอมพิวเตอร์ตัวสำรองเปลี่ยนสถานะเป็นคอมพิวเตอร์หลักนั้นจะทำเรียงลำดับจากน้อยไปหามากโดยเรียงจาก Node ID ซึ่งในงานวิจัยนี้ Node ID จะใช้เป็นหมายเลขไอพีแอดเดรส (IP address) โดยใช้หลักการประเมินผลวัดจากการเข้าถึงเข้ามูล (Latency) สามารถลดได้ถึง 0.51 µs ในระบบเครือข่ายกิกะบิต (Gbps network) เพื่อให้ระบบมีการเข้าถึงข้อมูลที่เร็วขึ้นมีการใช้กระบวนการย่อยข้อมูล (Fragment) ออกมาทั้งหมด N ชิ้น จากนั้น กระจายชิ้นส่วนย่อย N ชิ้น ไปตามคอมพิวเตอร์ในระบบ ซึ่งอาจจะเป็นไปได้ว่าชิ้นส่วนย่อยของไฟล์เดียวกันมากกว่า 1 ชิ้นอาจจะอยู่บนคอมพิวเตอร์เครื่อง

เดียวกัน และเมื่อไหร่ก็ตามที่ผู้ใช้ต้องการเข้าถึงไฟล์นี้ ผู้ใช้จำเป็นต้องดาวน์โหลดชิ้นส่วนให้ได้ทั้งหมด M ชิ้นที่ไม่ซ้ำกัน (ไม่จำเป็นต้องให้ได้ N ชิ้น) เมื่อได้มา M ชิ้นแล้ว ต้องนำ M ชิ้นมาประกอบใหม่ (Reconstruct) เพื่อให้ได้ไฟล์ต้นฉบับ โดยสรุปคือการสร้างชิ้นส่วนย่อย N ชิ้น โดยต้องการเพียง M ชิ้นที่แตกต่างกันเพื่อประกอบเป็นไฟล์ต้นฉบับ และ M ≤ N (Erasure code) ร่วมกับระบบการจัดเก็บข้อมูลแบบกระจายของของซอฟต์แวร์ Hadoop (HDFS) มาช่วยเพิ่มประสิทธิภาพในการจัดเก็บข้อมูลบนระบบการประมวลแบบกลุ่มเมฆ (Cloud Storage) ด้วยการทำให้เกิดความทนทานต่อความล้มเหลว (Fault Tolerance) ถือว่ามีความจำเป็นอย่างมาก เนื่องจากข้อมูลนั้นมีความสำคัญ หากเกิดความเสียหายขึ้น จะทำให้ไม่สามารถใช้งานข้อมูลนั้นได้เลย ดังนั้นเพื่อให้เกิดความน่าเชื่อถือ จึงมีการนำ HDFS เข้ามาใช้ในการจัดเก็บข้อมูลบน Cloud Storage เนื่องจากมีความน่าเชื่อถือในการทนทานต่อความล้มเหลวของข้อมูล เพราะมีการทำสำเนาของข้อมูล แต่ในการนำ HDFS มาใช้บนระบบการประมวลแบบกลุ่มเมฆนั้น จะเสียค่าใช้ในการจัดเก็บข้อมูล งานวิจัยนี้จึงนำ Erasure code มาใช้ร่วมกับ HDFS เพื่อประหยัดพื้นที่ในการจัดเก็บข้อมูล และผลจากการทดลองสามารถลดพื้นที่ในการจัดเก็บข้อมูลได้ถึง 33 % [19] และมีการศึกษาการสร้างการตรวจสอบ (Checkpoint) ที่ MapReduce ในส่วนของงานที่กำลังประมวลผลแบบจับคู่ข้อมูล (Map task) เพื่อใช้ในการฟื้นคืนสภาพปกติ (Recover) เมื่อเกิดการล้มเหลวขึ้นที่งานที่กำลังประมวลผลอยู่ โดยงานวิจัยก่อนหน้านี้ มีการประยุกต์การจัดทำดรรชนี (Indexing) โดยแบ่งเป็นส่วนย่อย ทำให้เร็วขึ้นในการเข้าถึงข้อมูล เป็นการสร้าง Checkpoint เพื่อจัดการกับความล้มเหลว (Failures) ด้านต่างๆ ใน MapReduce การทำ Checkpoint ที่ Map task จะทำการจับคู่ค่าของข้อมูล เพื่อให้ได้ค่าผลลัพธ์ออกมา และเก็บไว้ที่พักข้อมูล (Buffer) แต่หากผลลัพธ์มีจำนวนมากจนเกินหน่วยความจำของ buffer ทำให้ผลลัพธ์ล้นออกมา เราเรียกว่า "Spill" เมื่อเกิด Spill ขึ้น คอมพิวเตอร์มีหน้าที่จับคู่ค่าข้อมูล (Map worker) โดยนำไปเก็บไว้ใน Local disk และรอรวมจนกระทั่งเสร็จสิ้นกระบวนการจับคู่ เพื่อทำการรวมเป็นไฟล์เดียวกัน แล้วส่งให้กับคอมพิวเตอร์หลักที่ทำหน้าที่ควบคุม (Master) แต่ในงานวิจัยนี้จะไม่ทำการรวม Spill เป็นก้อนเดียวกัน เพื่อลด Overhead และลดภาระของ Master กล่าวคือ เมื่อเกิดการล้มเหลวขึ้น จึงไม่ต้องทำการฟื้นคืนสภาพทั้ง Map task เพราะสามารถกำหนดได้ว่า จะทำ Checkpoint ที่จำนวนเท่าใดของ Map task ในการประเมินผลของงานวิจัยนี้ ในด้านของ Overhead และ ประสิทธิภาพดีกว่า MapReduce เท่ากับ 9.0 % ในขณะที่ไม่เกิดความล้มเหลว แต่ในกรณีที่เกิดความล้มเหลวขึ้นจะดีกว่า MapReduce เท่ากับ 29.4 % โดยทำการทดลองที่เกิด

ความล้มเหลว 3 ครั้ง โดยหั่นขนาดของข้อมูลเป็น 256 MB และทำการ Checkpoint ที่ 256 MB เพื่อใช้ในการทำความทนทานต่อความล้มเหลว [8]

จากการศึกษางานวิจัยที่เกี่ยวข้อง ผู้วิจัยได้ทำการศึกษาและแบ่งประเภทของการเพิ่มประสิทธิภาพในการพัฒนาต้นแบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ โดยจะอธิบายดังตารางที่ 2.2

การเพิ่มประสิทธิภาพ / วรรณกรรมที่เกี่ยวข้อง	**ลดเวลาในการเข้าถึงข้อมูล**	**การจัดการข้อมูล**	**ความทนทานต่อความผิดพลาด**	**การวิเคราะห์ระบบเครือข่าย**	**เจ้าของผลงาน**
OpenFlow Enabled Hadoop Over Local and Wide Area Clusters	✓				Sandhya et al., 2012
Experiments on Networking of Hadoop	✓	✓			Abdul et al., 2014
Distributed MapReduce Engine With Fault Tolerance	✓				Song et al., 2014
Fault Tolerant Erasure Coded Replication for HDFS Based Cloud Storage		✓	✓		Aye and Wint, 2014
Improving MapReduce Performance under Failures with Resilient Checkpointing Tactics	✓	✓	✓		Wang et al., 2014
A Neural-Network Based DDoS Detection System Using Hadoop And HBase				✓	Zhao et al., 2015
Improving Big Data on Research and Education Networks using Future Internet Approach		✓			Tantatsanawong and Dontongdang, 2015

การเพิ่มประสิทธิภาพ / วรรณกรรมที่เกี่ยวข้อง	ลดเวลาในการเข้าถึงข้อมูล	การจัดการข้อมูล	ความทนทานต่อความผิดพลาด	การวิเคราะห์ระบบเครือข่าย	เจ้าของผลงาน
Big Data Testbed for Research and Education Networks Analysis	✓	✓			Dontongdang et al. 2015
Faster Detection and Prediction of DDoS attacks using MapReduce and Time Series Analysis	✓			✓	Maheshwari et al. 2018
Big Data Analytics for Network Anomaly Detection from Netflow Data				✓	Terzi et al. 2017
A Novel DoS and DDoS Attacks Detection Algorithm Using ARIMA Time Series Model and Chaotic System in Computer Networks				✓	Tabatabaie Nezhad et al. 2016
A Novel Real-Time DDoS Attack Detection Mechanism Based on MDRA Algorithm in Big Data				✓	Jia et al. 2016

ตารางที่ 2.2 แสดงงานวิจัยที่เกี่ยวข้องโดยแบ่งเป็นการเพิ่มประสิทธิภาพในด้านต่าง ๆ

บทที่ 3
วิธีดำเนินการวิจัย

เนื่องจากระบบเครือข่ายที่มีความซับซ้อนสูง มีการเชื่อมต่อกันของโหนดจำนวนมากที่มีการเพิ่มและลดของโหนดและลิงค์อยู่เสมอ ในแต่ละโหนดก็มีความแตกต่างกันทั้งในรูปแบบการเชื่อมต่อและการประมวลผล เช่น เครือข่าย UniNet ซึ่งมีการเชื่อมต่อเป็นลำดับชั้น (Layer) และมีการเชื่อมต่อในความเร็วที่แตกต่างกัน งานวิจัยนี้จึงได้ทำการศึกษารูปแบบและทำการวิเคราะห์ปัญหาบนระบบเครือข่ายที่มีความซับซ้อนสูง เพื่อพัฒนาระบบเครือข่ายให้มีประสิทธิภาพโดยการนำข้อมูลขนาดใหญ่มาใช้ในการวิเคราะห์ด้วยความรวดเร็ว ปลอดภัยและสามารถแจ้งเตือนเมื่อระบบเกิดความผิดปกติ ช่วยให้ผู้ดูแลระบบรับรู้ถึงสถานการณ์ของเครือข่ายได้อย่างทันท่วงที ดังนั้นเพื่อให้การวิจัยบรรลุวัตถุประสงค์ที่ตั้งไว้ จึงมีลำดับการทำงานประกอบด้วยขั้นตอนหลักดังต่อไปนี้

3.1 เครื่องมือและอุปกรณ์ที่ใช้ในการทำวิจัย

การวิจัยนี้ดำเนินการโดยการพัฒนาต้นแบบเพื่อแก้ปัญหาการวิเคราะห์ระบบเครือข่ายด้วยข้อมูลขนาดใหญ่ของเครือข่าย UniNet มีดังนี้

3.1.1 ฮาร์ดแวร์ที่ใช้ในการติดตั้งระบบทดสอบ

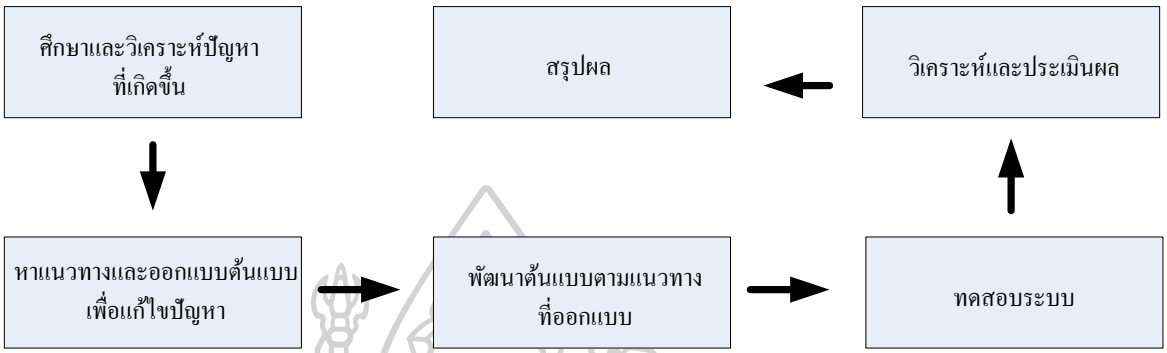
- เครื่องคอมพิวเตอร์แม่ข่าย (Server) จำนวน 5 เครื่อง โดยมีทรัพยากรดังนี้
  - หน่วยประมวลผลความเร็วขั้นต่ำ 1.0 GHz
  - หน่วยความจำขั้นต่ำ 2 GB
  - การ์ดเชื่อมต่อเครือข่าย 1 ชุด
  - ฮาร์ดดิสก์เก็บข้อมูลขั้นต่ำ 20 GB
  - สวิทช์สำหรับเชื่อมต่ออุปกรณ์คอมพิวเตอร์

3.1.2 ซอฟต์แวร์ ใช้ในการติดตั้งทดสอบระบบ มีดังต่อไปนี้

- ระบบปฏิบัติการลีนุกซ์ (Linux Operating System) : Ubuntu 16.04
- ซอฟต์แวร์ที่ช่วยในการจัดการข้อมูลขนาดใหญ่ : Apache Hadoop เวอร์ชัน 2.7.2
- ซอฟต์แวร์ที่ใช้ในการสร้างข้อมูลการจราจร (log file) : Netflow version9

3.2 ขั้นตอนการดำเนินการวิจัย

ขั้นตอนทั้งหมดของการวิจัยการพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่มีทั้งหมด 6 ขั้นตอนดังภาพที่ 3.1 ด้านล่าง โดยเริ่มตั้งแต่ขั้นตอนการศึกษาและวิเคราะห์ปัญหาจนถึงขั้นตอนการสรุปผล



ภาพที่ 3.1 แสดงขั้นตอนในการดำเนินการวิจัย

จากภาพที่ 3.1 ขั้นตอนการทำงานเบื้องต้น ผู้วิจัยได้ดำเนินการทดสอบภายใต้เครือข่าย UniNet โดยอุปกรณ์และสถานที่ติดตั้งจะกล่าวถึงรายละเอียดต่อไป โดยเริ่มจากการศึกษาและวิเคราะห์ปัญหาเกี่ยวกับการวิเคราะห์ระบบเครือข่ายด้วยข้อมูลขนาดใหญ่ เพื่อออกแบบต้นแบบในการจัดการปัญหาดังกล่าว รวมทั้งดำเนินการออกแบบและพัฒนาระบบต้นแบบเพื่อนำไปทดสอบระบบและนำผลที่ได้รับมาวิเคราะห์เพื่อประเมินประสิทธิภาพในการทำงานและสรุปผล

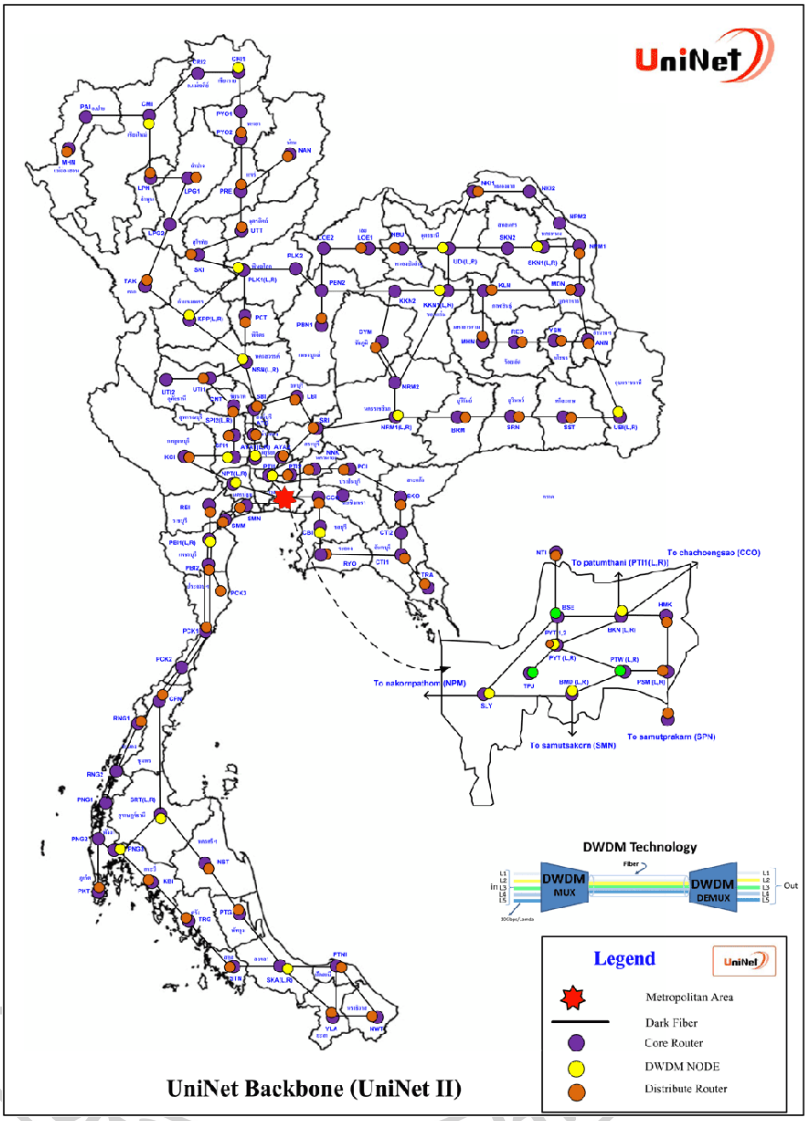
3.2.1 ศึกษาและวิเคราะห์ปัญหาที่เกิดขึ้นกับระบบเครือข่ายที่มีความซับซ้อนสูง

เมื่อได้ทำการศึกษาและวิเคราะห์ปัญหาของระบบเครือข่ายที่มีความซับซ้อนสูงแล้วนั้น พบว่าระบบจะมี ลิงค์และโหนด ที่เชื่อมต่อกันเป็นจำนวนมาก มีการเพิ่มหรือลดของโหนดและlinkอยู่เสมอ การที่มี ลิงค์และ โหนด จำนวนมากก่อให้เกิดข้อมูลที่เกิดจากติดต่อสื่อสารกันเป็นจำนวนมากตามไปด้วย ส่วนการเพิ่มหรือลดของ ลิงค์หรือ โหนด จะส่งให้ทำให้การทำดูแลระบบทำได้ยาก เกิดค่าใช้จ่ายที่เพิ่มขึ้น ตัวอย่างเช่น ในการลดลงของโหนด ที่ผู้ดูแลระบบไม่สามารถทราบสาเหตุที่แน่ชัดว่าการลดลงของโหนดนั้นเกิดจากสาเหตุใด เช่น การถูกโจมตี การยกเลิกโหนดนั้น หรืออุปกรณ์เครือข่ายเกิดความเสียหายที่ทำให้ไม่สามารถเชื่อมต่อโหนดนั้นๆ ได้ ทำให้ไม่สามารถจัดการหรือวางแผนเส้นทางการติดต่อสื่อสารของโหนดภายในระบบได้ ส่งผลให้การใช้งานระบบเกิดความล่าช้าหลังมีโหนดลดลง

ในกรณีศึกษาของงานวิจัยนี้ได้ทำการศึกษาและวิเคราะห์ปัญหาในระบบเครือข่ายที่มีความซับซ้อนสูงในประเทศไทย (UniNet) โดยได้ทำการแยกปัญหาออกเป็น ข้อๆ ดังนี้

- ปัญหาข้อมูลมีขนาดใหญ่

ข้อมูลที่นำมาใช้ในการวิจัยครั้งนี้คือข้อมูลการจราจรของระบบเครือข่าย (log file) และมีขนาดใหญ่เกินไปที่ระบบปัจจุบันจะจัดเก็บและประมวลผลได้ UniNet มีรูปแบบการเชื่อมต่อของเครือข่ายมีการเชื่อมต่อเป็น 2 ลำดับชั้น คือ Backbone ลิงค์เชื่อมต่อกันที่ความเร็ว 50 Gbps และ Distribution ลิงค์เชื่อมต่อกันที่ความเร็ว 10 Gbps โดยรูปแบบการเชื่อมต่อจะเชื่อมต่อกันด้วยอุปกรณ์เลือกเส้นทาง (Router) 120 ตัว เมื่อต้องการใช้ข้อมูลการจราจร (Log file) โดยในแต่ละตัวจะสร้าง Log file ออกมาประมาณ 3.5 GB ในเวลาชั่วโมง แต่ถ้ารวมกันทั้งหมดใน 1 วัน จะมีปริมาณข้อมูล 9.8 TB และในระยะเวลา 1 ปี จะมีปริมาณ Log file ของเครือข่ายเท่ากับ 3.4 PB ซึ่งถือว่าปริมาณข้อมูลที่ถูกสร้างออกมาเป็นจำนวนมากหากต้องการที่จะนำข้อมูลทั้งหมด มาใช้เพื่อทำการประมวลผลเพื่อนำไปใช้ในการวิเคราะห์ระบบเครือข่าย อาจจะทำเกิดปัญหาด้านพื้นที่ในการจัดเก็บข้อมูลที่มีอยู่อย่างจำกัด และความเร็วในการประมวลผลเพื่อทำการวิเคราะห์ระบบเครือข่ายแสดงให้เห็นการเชื่อมต่อของอุปกรณ์หาเส้นทาง (Router) ดังภาพ 3.2



ภาพที่ 3.2 แสดงจุดการเชื่อมต่ออุปกรณ์หาเส้นทางในระบบ UniNet [2]

UniNet เป็นระบบเครือข่ายที่มีความซับซ้อนสูง ที่มีการเชื่อมต่อขนาดใหญ่มีผู้ใช้จำนวนมาก เครือข่าย UniNet ที่มีผู้ใช้มากกว่า 6 ล้านผู้ใช้ [20] หากระบบเครือข่ายเกิดความเสียหายหรือไม่สามารถใช้งานได้จะส่งกระทบต่อผู้ใช้ทั้งทางตรงและทางอ้อม ตัวอย่างเช่น เมื่อผู้ใช้กำลังใช้งานเครือข่ายอยู่และเกิดความเสียหายขึ้นกับระบบเครือข่าย ระบบไม่สามารถใช้งานได้ตามปกติอาจก่อให้เกิดความเสียขึ้นกับการใช้งานของผู้ใช้ที่กำลังดำเนินการอยู่ทำให้เกิดค่าใช้จ่ายที่ไม่จำเป็นตามมา

ได้แก่ การเสียเวลา หรืออาจเกิดผลกระทบที่รุนแรงมากกว่านั้น เช่นการเชื่อมต่อทางไกลด้านการศึกษา หรือ ด้านการแพทย์ เป็นต้น

- ปัญหาการเก็บสถิติการเกิด DDoS

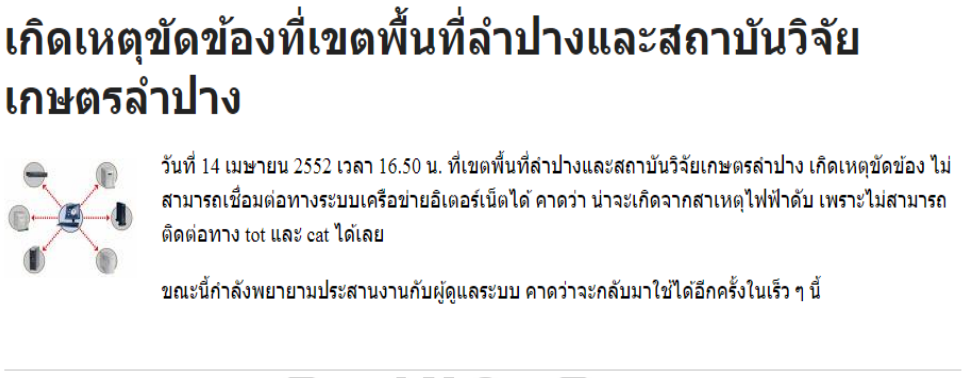
การบันทึกสถิติเปรียบเสมือนการเก็บประวัติของระบบนั้นๆ เราสามารถนำข้อมูลในอดีตมาวิเคราะห์เพื่อเป็นแนวทางในการแก้ไขปัญหาที่เกิดขึ้นในปัจจุบัน เพื่อลดความเสี่ยงและความเสียหาย เนื่องจากความเสียหายนั้นไม่ได้มีผลกระทบกับเพียงระบบเท่านั้น ยังมีผลกับปัจจัยภายนอกต่างๆ เช่น แรงงาน ค่าใช้จ่าย และในปัจจุบัน UniNiet ยังไม่มีการเก็บสถิติการเกิด DDoS ที่เป็นการรวมข้อมูลไว้ที่ศูนย์กลาง เนื่องจากระบบการวิเคราะห์ระบบเครือข่ายเดิมเป็นเพียงการวิเคราะห์เฉพาะจุด แยกไปตามโหนด (Backbone Node/Distribution Node) ไม่มีการส่งข้อมูลหรือรวบรวมสถิติการเกิด DDoS ไว้ที่ศูนย์กลาง หากต้องการที่จะนำข้อมูลมาใช้หรือทำการเรียกดู ต้องมีการรวบรวมข้อมูลจากแต่ละโหนด ทำให้เกิดความล่าช้า และข้อมูลอาจไม่ครบถ้วนสมบูรณ์ได้ เนื่องจากมีเพียงบางโหนดที่เก็บข้อมูลการโดนโจมตีไว้ และบางโหนดอาจไม่ได้ทำการเก็บบันทึกไว้ เมื่อนำข้อมูลมาประมวลผลเพื่อวิเคราะห์ระบบเครือข่าย อาจทำให้ได้ผลลัพธ์ที่ผิดพลาด

ภาพที่ 3.3 IP Address ในระบบ UniNet มีการปิดกั้นเนื่องจากเป็นภัยคุกคาม [2]

จากภาพที่ 3.3 แสดงให้เห็นได้ว่าระบบ UniNet มีการบันทึกหมายเลข IP ที่เป็นตัวคุกคามไว้ตั้งแต่ปี ค.ศ. 2016 และหลังจากนั้นจะไม่สามารถแสดงหรือค้นหาได้ ในระยะเวลาเกือบ 4 ปีนั้น ไม่มีการเก็บบันทึกหรือแจ้งเตือนเกี่ยวกับหมายเลข IP Address ที่เป็นภัยคุกคามใดๆเลย

- ความล่าช้าในการตรวจพบ DDoS

ปัญหาในระบบ UniNet ปัจจุบันการตรวจ DDoS ในระบบเครือข่าย UniNet นั้นจะพบได้ก็ต่อเมื่อระบบเกิดความเสียหาย หรือ ระบบเกิดการล่ม ไม่สามารถใช้งานระบบได้ ระบบ UniNet เป็นระบบที่มีความซับซ้อนสูง มีขนาดใหญ่ และผู้ใช้จำนวนมาก ทำให้การแยกระหว่างพฤติกรรมของการโจมตีกับพฤติกรรมผู้ใช้ปกติออกจากกันได้ยาก เช่น เหตุการณ์ที่เกิดขึ้นเมื่อวันที่ 25 พฤศจิกายน พ.ศ. 2557 วงจรต่างประเทศ CAT IIG มีปัญหา ทำให้การใช้งานเครือข่ายต่างประเทศของสมาชิกเครือข่าย UniNet/NEdNet ไม่สามารถใช้งานได้ในบางช่วงเวลา และวันที่ 15 กันยายน พ.ศ. 2557 สายเคเบิลใต้น้ำเกิดความเสียหาย ส่งผลกระทบให้ไม่สามารถใช้งานอินเทอร์เน็ตได้ เป็นต้น



เกิดเหตุขัดข้องที่เขตพื้นที่ลำปางและสถาบันวิจัยเกษตรลำปาง

วันที่ 14 เมษายน 2552 เวลา 16.50 น. ที่เขตพื้นที่ลำปางและสถาบันวิจัยเกษตรลำปาง เกิดเหตุขัดข้อง ไม่สามารถเชื่อมต่อทางระบบเครือข่ายอิเตอร์เน็ตได้ คาดว่า น่าจะเกิดจากสาเหตุไฟฟ้าดับ เพราะไม่สามารถติดต่อทาง tot และ cat ได้เลย

ขณะนี้กำลังพยายามประสานงานกับผู้ดูแลระบบ คาดว่าจะกลับมาใช้ได้อีกครั้งในเร็ว ๆ นี้

ภาพที่ 3.4 แจ้งประชาสัมพันธ์เหตุขัดข้องที่ไม่สามารถจะระบุสาเหตุได้ [2]

จากภาพที่ 3.4 จะเห็นข้อความแจ้งประชาสัมพันธ์ ว่ามีเหตุขัดข้องของระบบเครือข่ายเนื่องจากไม่สามารถเชื่อมต่อระบบได้ และไม่สามารถทราบได้แน่ชัดว่าสาเหตุที่ไม่สามารถเชื่อมต่อระบบได้เกิดจากสาเหตุใด

- ระบบการเฝ้าระวัง

โครงสร้างเครือข่ายบนระบบ UniNet ในปัจจุบันเป็นการเชื่อมต่อแบบกระจาย ในแต่ละโหนดจะมี NetFlow Application อยู่เนื่องจากเป็น Application ที่ติดมากับอุปกรณ์หาเส้นทาง (Router) ซึ่งทำหน้าที่ในการมอนิเตอร์เครือข่ายในแต่ละโหนด เมื่อทำการประมวลผลเสร็จจะทิ้งข้อมูล Log file ทั้งหมด เนื่องจาก log file มีจำนวนมากไม่สามารถที่จะเก็บไว้ในอุปกรณ์การจัดเก็บข้อมูลของ UniNet ได้


UniNet
Network operation center

เครื่องแม่ข่ายที่ติดตั้งตามภูมิภาค

ภาคเหนือ	ภาคตะวันออกเฉียงเหนือ	ภาคตะวันออก
จังหวัดกำแพงเพชร	จังหวัดกาฬสินธุ์	จังหวัดฉะเชิงเทรา
จังหวัดเชียงราย	จังหวัดขอนแก่น	จังหวัดชลบุรี
จังหวัดเชียงใหม่	จังหวัดชัยภูมิ	จังหวัดจันทบุรี
จังหวัดน่าน	จังหวัดนครพนม	จังหวัดนครนายก
จังหวัดนครสวรรค์	จังหวัดนครราชสีมา	จังหวัดปราจีนบุรี
จังหวัดตาก	จังหวัดบุรีรัมย์	จังหวัดระยอง
จังหวัดพะเยา	จังหวัดมหาสารคาม	จังหวัดสระแก้ว
จังหวัดแพร่	จังหวัดมุกดาหาร	จังหวัดสมุทรปราการ
จังหวัดแม่ฮ่องสอน	จังหวัดยโสธร	จังหวัดตราด
จังหวัดลำปาง	จังหวัดร้อยเอ็ด	
จังหวัดลำพูน	จังหวัดเลย	
จังหวัดพิจิตร	จังหวัดสกลนคร	
จังหวัดพิษณุโลก	จังหวัดสุรินทร์	
จังหวัดเพชรบูรณ์	จังหวัดศรีสะเกษ	
ภาคตะวันตก	**ภาคกลาง**	**ภาคใต้**
จังหวัดกาญจนบุรี	จังหวัดอ่างทอง	จังหวัดกระบี่
จังหวัดนครปฐม	จังหวัดพระนครศรีอยุธยา-1	จังหวัดชุมพร
จังหวัดเพชรบุรี-1	จังหวัดพระนครศรีอยุธยา-2	จังหวัดตรัง
จังหวัดเพชรบุรี-2	จังหวัดสระบุรี	จังหวัดนครศรีธรรมราช
จังหวัดประจวบคีรีขันธ์-1	จังหวัดสิงห์บุรี	จังหวัดนราธิวาส
จังหวัดประจวบคีรีขันธ์-2	จังหวัดนนทบุรี	จังหวัดปัตตานี
จังหวัดราชบุรี	จังหวัดชัยนาท	จังหวัดพังงา
จังหวัดสมุทรสงคราม	จังหวัดปทุมธานี-1	จังหวัดพัทลุง
จังหวัดสมุทรสาคร	จังหวัดปทุมธานี-2	จังหวัดภูเก็ต
จังหวัดสุพรรณบุรี-1	จังหวัดลพบุรี	จังหวัดระนอง
จังหวัดสุพรรณบุรี-2	กรุงเทพมหานคร	จังหวัดสตูล
		จังหวัดสงขลา

ภาพที่ 3.5 แสดงเครื่องแม่ข่ายที่ติดตั้งตามภูมิภาคที่ติดตั้ง NetFlow Application [2]

จากภาพที่ 3.5 แสดงเครื่องแม่ข่ายที่ติดตั้ง NetFlow Application ตามภูมิภาคต่างๆ ซึ่งในแต่ละโหนด สามารถที่จะ Monitor ได้เพียงโหนดของตัวเองเท่านั้น ทำให้ผลลัพธ์กระจายอยู่ตามโหนดต่างๆ หากต้องการนำข้อมูลมาใช้ จะทำให้เสียเวลาในการรวบรวมข้อมูล

ปัญหาของ NetFlow Application ในการนำมาใช้ในระบบเครือข่ายที่มีความซับซ้อนสูงคือในการทำงานของ NetFlow Monitor นั้นทำงาน 2 ส่วนคือ

1) รับโฟล์ว คือ รับข้อมูลโฟล์วแล้วมาจัดเก็บบันทึกไว้ในฐานข้อมูล

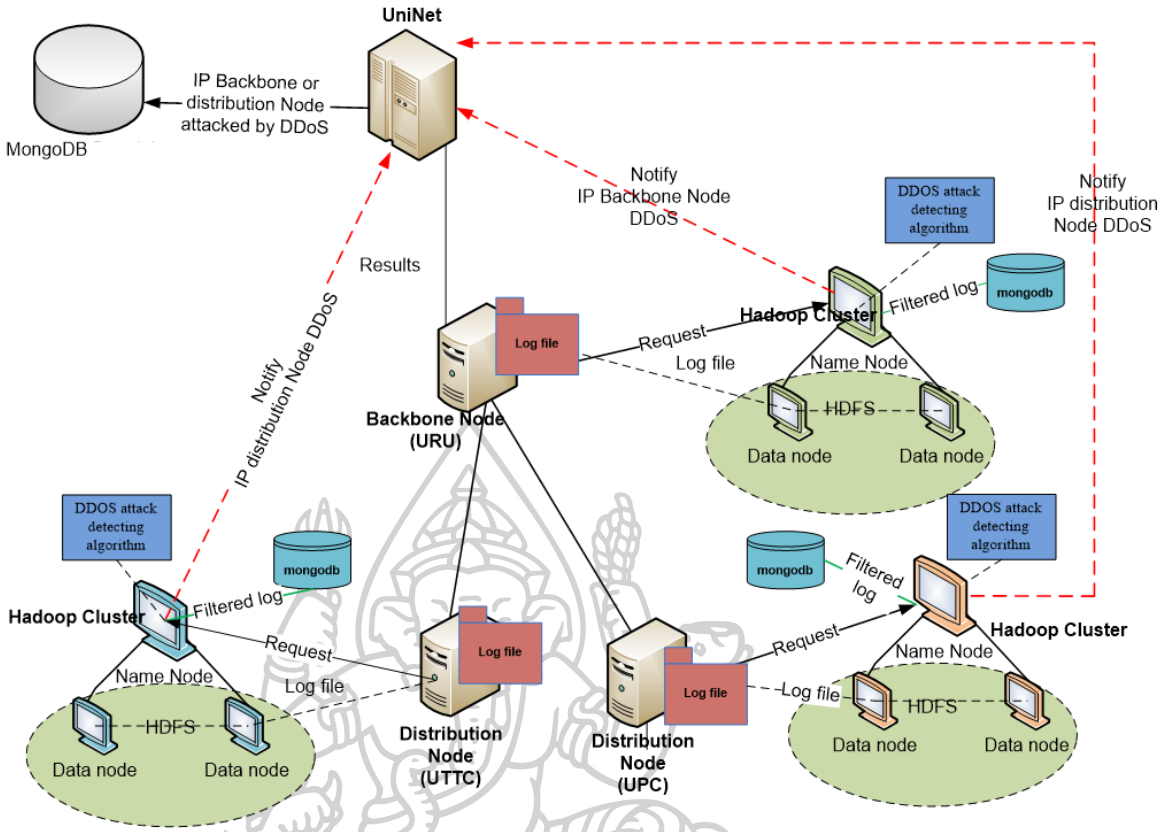
2) ประมวลผล จะรับโฟล์วจากทุกๆ Router แล้วจึงจะประมวลผลตามเงื่อนไขที่ได้กำหนดไว้

จากการทำงานข้างต้นจะเห็นได้ว่า หากต้องนำข้อมูลโฟล์วทั้งหมดมารวบรวมเพื่อทำการประมวลผลบนระบบเครือข่ายที่มีความซับซ้อนสูงนี้ จะมี ลิงค์และ โหนด จำนวนมาก หากต้องส่งข้อมูลจากทุกโหนดมายังส่วนกลางที่มี NetFlow Application อยู่เพื่อทำการประมวลผลจะทำให้เส้นทางการจราจรเกิดความแออัดได้และด้วยปริมาณข้อมูลมีจำนวนมากจะทำให้เกิดสภาวะคอขวดเมื่อมีการรอคิวในการประมวลผลจาก NetFlow Application ปัญหาอีกประการหนึ่งของ NetFlow Application คือ ในการวิเคราะห์ระบบเครือข่าย จะไม่ได้เก็บข้อมูล Payload (ข้อมูลข่าวสารของระดับชั้นสูงๆ) มีเพียงการเก็บรูปแบบต่างๆ ของ ทราฟฟิกเท่านั้น ทำให้การวิเคราะห์หรือตรวจจับ DDoS ไม่ละเอียดเพียงพอ ที่จะทำให้ระบบมีความปลอดภัยจากการโจมตีด้วย DDoS ได้

3.2.2 ศึกษาและออกแบบต้นแบบเพื่อแก้ปัญหาที่เกิดขึ้น

ในการออกแบบต้นแบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่เพื่อแก้ปัญหาข้างต้น มีดังนี้

1. ออกแบบเพื่อจัดการกับข้อมูลที่มีขนาดใหญ่
2. ออกแบบการคัดกรองข้อมูลเพื่อลดขนาดของข้อมูล
3. เลือกอัลกอริทึมที่มีความเหมาะสมกับ ข้อมูลที่นำมาประมวลผลและรูปแบบของระบบเครือข่าย
4. ออกแบบวิธีการส่งผลลัพธ์ไปยังศูนย์กลาง



ภาพที่ 3.6 แสดงต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่

โครงสร้างเครือข่ายทดสอบระบบที่พัฒนาขึ้น ให้มี Hadoop Cluster เพื่อนำมาใช้ในการจัดเก็บและคัดกรองข้อมูล โดยในแต่ละโหนดจะจัดเก็บข้อมูลไว้ที่ HDFS หลังจากนั้นจะนำข้อมูลจาก HDFS มาทำการคัดกรองข้อมูล เพื่อลดขนาดของข้อมูลลง ด้วย Map Function แล้วนำไปเก็บไว้ในฐานข้อมูล MongoDB ซึ่งมีคุณสมบัติเป็นฐานข้อมูลที่ยืดหยุ่น ปรับขนาดได้ มีประสิทธิภาพเหมาะสมสำหรับข้อมูลแบบกึ่งมีโครงสร้าง (Semi-Structured) และไม่มีโครงสร้าง (Unstructured) และเข้าถึงรูปแบบที่เปิดใช้งานประสิทธิภาพที่สูงกว่า มี API การทำงานและสามารถสร้างตามวัตถุประสงค์สำหรับโมเดลข้อมูลแต่ละโมเดลที่สอดคล้องกันมักถูกออกแบบมาให้ปรับขนาดได้โดยใช้คลัสเตอร์แบบกระจายของฮาร์ดแวร์แทนการปรับขนาดขึ้น

3.2.3 พัฒนาต้นแบบตามที่ได้ออกแบบไว้ข้างต้น

- การเลือก Dataset [21] ในการทดลองครั้งนี้ จะทำการทดสอบเพื่อหาความแม่นยำในการตรวจสอบ DDoS โดยจะทำการทดสอบกับ Dataset ที่ชื่อว่า CTU-13 Dataset [22] ซึ่งเป็นป้ายกำกับของแต่ละโฟล์วว่า เป็น DDoS หรือไม่แสดงให้เห็นในภาพที่ 3.7

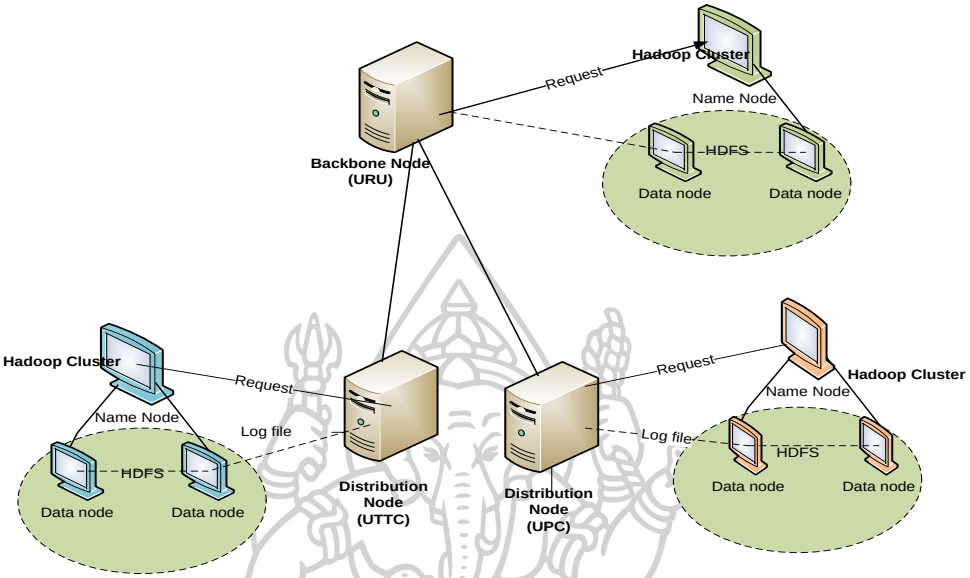
```
#StartTime,Dur,Proto,SrcAddr,Sport,Dir,DstAddr,Dport,State,sTos,dTos,TotPkts,TotBytes,SrcBytes,Label(Normal:CC:Background),CCDetector(Normal:CC:Unknown)
2011/08/18 11:46:16.889182,2.991484,tcp,147.32.84.118,3038,   ->,2.215.205.93,6881,S_,0,,2,124,124,flow=Background-TCP-Attempt,Unknown
2011/08/18 11:48:31.567539,8.933183,tcp,147.32.84.118,3168,   ->,2.215.205.93,6881,S_,0,,3,186,186,flow=Background-TCP-Attempt,Unknown
2011/08/18 11:51:49.645845,9.010275,tcp,147.32.84.118,3312,   ->,2.215.205.93,6881,S_,0,,3,186,186,flow=Background-TCP-Attempt,Unknown
2011/08/18 14:56:38.633552,0.000921,udp,46.217.68.251,20962,  <->,147.32.86.116,19083,CON,0,0,2,136,75,flow=Background-UDP-Established,Unknown
2011/08/18 13:24:42.033424,2570.226318,udp,147.32.84.229,13363,  <->,208.88.186.4,34033,CON,0,0,16,2241,1643,flow=Background-UDP-Established,Unknown
2011/08/18 13:47:19.434012,4.106750,icmp,195.47.235.10,0x0303,   ->,147.32.84.59,0xd9df,URP,0,,2,140,140,flow=Background-Attempt-cmpgw-CVUT,Unknown
2011/08/18 10:47:32.644518,3034.190674,udp,147.32.84.229,13363,  <->,87.51.144.31,64920,CON,0,0,12,798,438,flow=Background-UDP-Established,Unknown
2011/08/18 12:29:34.758297,3092.267822,udp,147.32.84.229,13363,  <->,87.51.144.31,64920,CON,0,0,9,1416,736,flow=Background-UDP-Established,Unknown
2011/08/18 14:30:27.465577,161.721420,udp,147.32.84.229,13363,  <->,87.51.144.31,64920,CON,0,0,4,273,153,flow=Background-UDP-Established,Unknown
2011/08/18 14:36:52.579506,0.000000,icmp,147.32.84.118,0x0303,   ->,89.111.65.198,0xe11a,URP,0,,1,135,135,flow=Background,Unknown
2011/08/18 13:40:49.034963,3337.479492,udp,200.146.28.30,15934,  <->,147.32.84.229,13363,CON,0,0,4,268,148,flow=Background-UDP-Established,Unknown
2011/08/18 11:41:26.268829,364.333130,udp,98.154.238.196,28226,  <->,147.32.84.229,13363,CON,0,0,6,404,224,flow=Background-UDP-Established,Unknown
2011/08/18 11:05:22.925055,0.001998,udp,112.204.148.197,52435,  <->,147.32.84.229,13363,CON,0,0,2,127,67,flow=Background-UDP-Established,Unknown
2011/08/18 11:26:42.805220,832.729248,udp,112.204.148.197,52434,  <->,147.32.84.229,13363,CON,0,0,4,273,153,flow=Background-UDP-Established,Unknown
2011/08/18 11:48:08.017588,245.990387,tcp,147.32.84.164,40223,   ->,74.125.232.218,80,FSPA_FSRPA,0,0,33,11785,1553,flow=From-Normal-V51-Grill,Unknown
2011/08/18 11:48:08.017596,15.977654,tcp,147.32.84.164,40224,   ->,74.125.232.218,80,FSA_FSA,0,0,6,412,272,flow=From-Normal-V51-Grill,Unknown
2011/08/18 11:48:09.476772,244.530823,tcp,147.32.84.164,40239,   ->,74.125.232.218,80,FSPA_FSRPA,0,0,28,5899,2885,flow=From-Normal-V51-Grill,Unknown
2011/08/18 12:11:49.414857,249.324158,tcp,147.32.84.164,39306,   ->,74.125.232.218,80,FSPA_FSRPA,0,0,21,3617,2130,flow=From-Normal-V51-Grill,Unknown
2011/08/18 11:38:53.108272,0.026932,tcp,147.32.86.89,2699,   ->,62.168.44.115,80,FSPA_FSPA,0,0,12,1942,1294,flow=Background-TCP-Established,From-Botnet-V54-TCP-CC1-HTTP-Not-Encrypted
2011/08/18 11:39:07.193113,0.028531,tcp,147.32.86.89,2737,   ->,62.168.44.115,80,FSPA_FSPA,0,0,12,1937,1289,flow=Background-TCP-Established,From-Botnet-V54-TCP-CC1-HTTP-Not-Encrypted
2011/08/18 11:39:16.198135,0.024877,tcp,147.32.86.89,2775,   ->,62.168.44.115,80,FSPA_FSPA,0,0,11,1881,1233,flow=Background-TCP-Established,From-Botnet-V54-TCP-CC1-HTTP-Not-Encrypted
2011/08/18 11:39:31.484314,0.024750,tcp,147.32.86.89,2805,   ->,62.168.44.115,80,FSPA_FSPA,0,0,12,1938,1290,flow=Background-TCP-Established,From-Botnet-V54-TCP-CC1-HTTP-Not-Encrypted
2011/08/18 11:39:45.441439,0.026138,tcp,147.32.86.89,2841,   ->,62.168.44.115,80,FSPA_FSPA,0,0,12,1942,1294,flow=Background-TCP-Established,From-Botnet-V54-TCP-CC1-HTTP-Not-Encrypted
2011/08/18 13:13:25.388534,0.108374,tcp,147.32.86.89,3895,   ->,62.168.44.115,80,FSPA_FSPA,0,0,12,4450,758,flow=Background-TCP-Established,From-Botnet-V54-TCP-CC1-HTTP-Not-Encrypted
2011/08/18 10:54:23.545398,34.097092,udp,147.32.84.229,13363,  <->,84.51.195.16,11957,CON,0,0,8,544,304,flow=Background-UDP-Established,Unknown
2011/08/18 11:28:58.121357,2446.322021,udp,89.117.191.89,4452,  <->,147.32.84.229,13363,CON,0,0,4,266,146,flow=Background-UDP-Established,Unknown
2011/08/18 12:53:30.288946,2464.322266,udp,89.117.191.89,4452,  <->,147.32.84.229,13363,CON,0,0,4,266,146,flow=Background-UDP-Established,Unknown
2011/08/18 13:31:44.247569,15.653972,tcp,147.32.86.194,1069,   ->,147.32.240.57,80,FSPA_FSPA,0,0,21,8338,2358,flow=Background-TCP-Established,Unknown
2011/08/18 13:31:44.295946,14.605665,tcp,147.32.86.194,1071,   ->,147.32.240.57,80,FSPA_FSPA,0,0,12,2318,1130,flow=Background-TCP-Established,Unknown
2011/08/18 13:32:21.027950,14.857969,tcp,147.32.86.194,1118,   ->,147.32.240.57,80,FSPA_FSPA,0,0,13,4724,1346,flow=Background-TCP-Established,Unknown
2011/08/18 13:32:21.029930,14.856419,tcp,147.32.86.194,1120,   ->,147.32.240.57,80,FSPA_FSPA,0,0,10,1459,850,flow=Background-TCP-Established,Unknown
2011/08/18 13:35:16.182480,2207.006836,udp,90.176.239.129,41349,  <->,147.32.86.165,12114,CON,0,0,14,942,522,flow=Background-UDP-Established,Unknown
2011/08/18 10:57:58.734053,2.945069,tcp,147.32.84.118,4684,   ->,217.68.187.17,6881,S_,0,,2,124,124,flow=Background-TCP-Attempt,Unknown
2011/08/18 11:01:49.689788,8.970511,tcp,147.32.84.118,4834,   ->,217.68.187.17,6881,S_,0,,3,186,186,flow=Background-TCP-Attempt,Unknown
2011/08/18 11:05:16.381671,8.984080,tcp,147.32.84.118,1053,   ->,217.68.187.17,6881,S_,0,,3,186,186,flow=Background-TCP-Attempt,Unknown
2011/08/18 13:42:37.258515,2.946277,tcp,147.32.84.118,2103,   ->,217.68.187.17,6881,S_,0,,2,124,124,flow=Background-TCP-Attempt,Unknown
2011/08/18 13:46:37.272727,8.965590,tcp,147.32.84.118,2323,   ->,217.68.187.17,6881,S_,0,,3,186,186,flow=Background-TCP-Attempt,Unknown
2011/08/18 13:49:56.203334,8.895177,tcp,147.32.84.118,2540,   ->,217.68.187.17,6881,S_,0,,3,186,186,flow=Background-TCP-Attempt,Unknown
```

ภาพที่ 3.7 แสดง NetFlow log file ของ CTU-13 Dataset [23]

ภาพที่ 3.7 แสดงให้เห็นถึงป้ายกำกับ ของโฟล์วที่มีทั้งปกติและเป็นภัยคุกคาม โฟล์วที่มีความปกติจะมี ป้ายกำกับเป็น “Unknown” ส่วนที่ผิดปกติ จะมีตัวอย่างเช่น โฟล์วที่ถูกทำไฮไลท์ คือโฟล์วที่ผิดปกติ และจะถูกกำกับด้วยคำว่า “Botnet”

- ขั้นตอนการจัดเก็บ Log File ใน HDFS ในการจัดเก็บ log file ที่มีขนาดใหญ่ ผู้วิจัยเลือก Hadoop ในงานวิจัยนี้ เนื่องจากมีคุณสมบัติที่สามารถจัดการกับ ข้อมูลขนาดใหญ่ได้อย่างมีประสิทธิภาพ Hadoop เป็นที่นิยมในการนำมาใช้กับ Big Data เนื่องจากสาเหตุดังต่อไปนี้
  - Low cost computing system คือ Hadoop เป็น open-source software ที่เราสามารถดาวน์โหลดมาใช้ได้ฟรี ด้วยเงื่อนไขของ Apache License 2.0 และ Hadoop สามารถรันบนเครื่องคอมพิวเตอร์ทั่วไป เช่นเครื่อง PC หรือ notebook ก็สามารถนำมาทำเป็น Hadoop Cluster ได้ ไม่จำเป็นต้องเป็น Server ราคาแพง
  - Effective Ecosystem คือ Hadoop มีชุดโปรแกรมใน Ecosystem ที่ทรงประสิทธิภาพ และยังมีโปรแกรมตัวช่วยแถมมาให้อีกเป็นจำนวนมาก

- Easy-to-Scale คือ เราสามารถเพิ่มจำนวน node เข้าไปใน Hadoop Cluster ได้ง่ายไม่ต้องเสียเวลาและค่าใช้จ่ายมาก



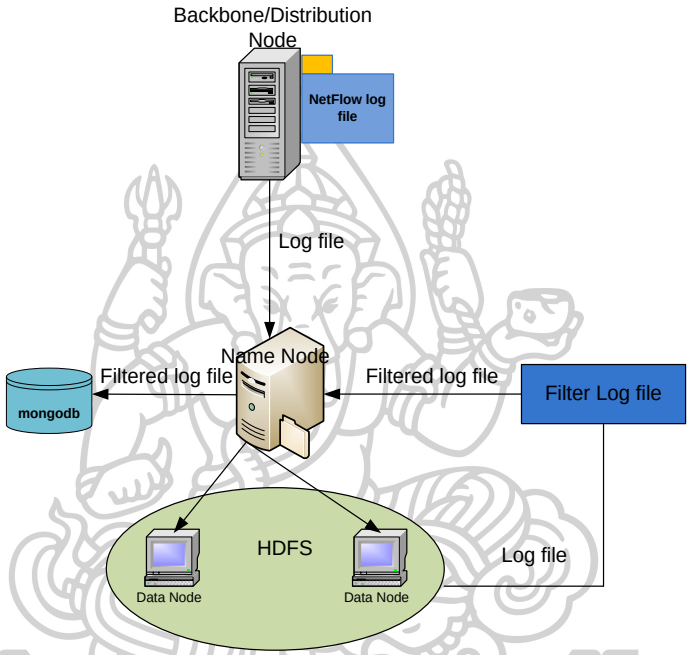
ภาพที่ 3.8 แสดงขั้นตอนการจัดเก็บ log file ลงใน HDFS

จากภาพที่ 3.8 แสดงให้เห็นขั้นตอนการทำงานในการจัดเก็บ log file เข้าสู่ HDFS มีดังต่อไปนี้

1. Backbone Node/Distribution Node ทำการโอน NetFlow log file ที่มีขนาดใหญ่ลงใน HDFS
2. NetFlow log file บน HDFS จะถูกหั่นเป็นชิ้นเล็กๆ เพื่อกระจายไปเก็บไว้บน Cluster
3. ไฟล์ที่ถูกหั่นเป็นชิ้นเล็ก สามารถในการประมวลผลแบบขนานได้ (Parallel Processing) ทำให้การอ่านไฟล์ขนาดใหญ่เร็วกว่าการเปิดไฟล์ในเครื่องคอมพิวเตอร์ทั่วไป
4. ไฟล์ที่ถูกแบ่งเป็นชิ้นเล็กๆ แล้ว จะถูกทำสำเนาอย่างน้อย 3 สำเนาเพื่อให้เกิดความน่าเชื่อถือว่าไฟล์จะไม่มีการสูญหายไปได้
5. หากมีการเพิ่มจำนวนเครื่องคอมพิวเตอร์บน Cluster ไม่ส่งผลให้การทำงานช้าลงแต่สามารถเพิ่มทรัพยากรให้การประมวลผลมีประสิทธิภาพมากขึ้น

- การคัดกรองข้อมูล (Data Filtration) เพื่อลดจำนวน Data ที่ไม่จำเป็นในการประมวลผล สามารถทำได้โดยการ คัดกรองข้อมูล เพื่อให้เกิดความมีประสิทธิภาพ ในการประมวลผล ส่งผลให้การประมวลผลทำได้เร็วและการบริโภคทรัพยากรที่น้อยลง โดยมีหลักการทำงานดังนี้คือ เมื่อ

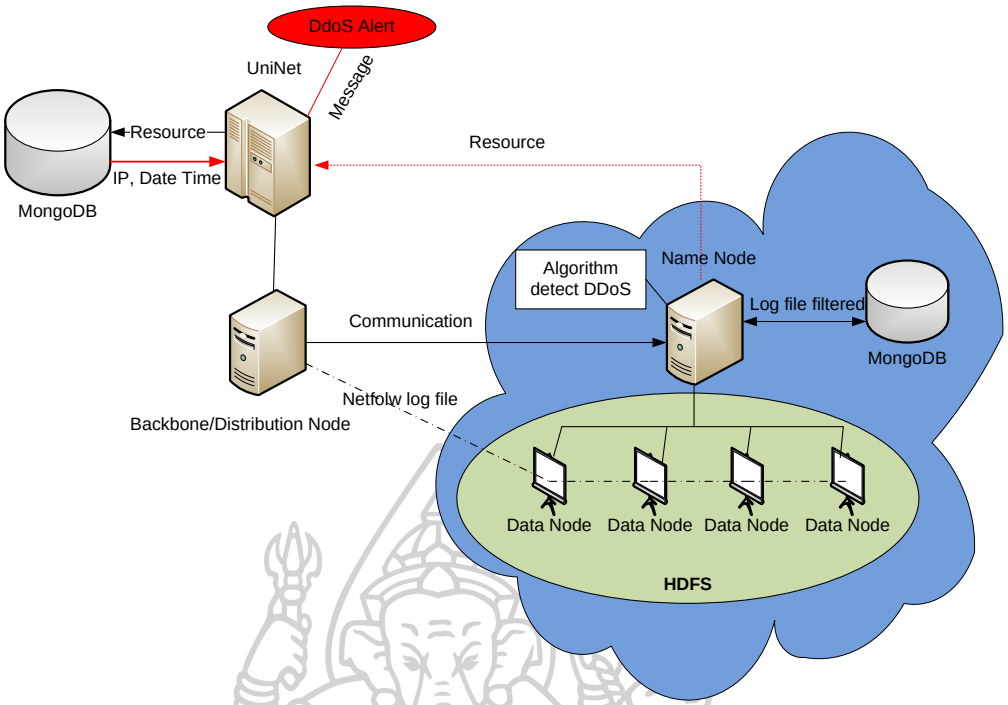
Log file ที่ถูกส่งมาจาก Backbone Node/Distribution Node เพื่อจัดเก็บใน HDFS โดย HDFS จะทำหน้าที่แบ่งข้อมูลออกเป็น Block และทำสำเนา แล้วทำการกระจายไปเก็บไว้ที่ Data Node ต่างๆ เมื่อทำการคัดกรองข้อมูลด้วย Map Function แล้ว จะนำข้อมูลจาก Data Node มาประมวลผลเพื่อให้ได้ log file ที่ลดจำนวน Field แล้ว (IP Address, Timestamp, Total_Fw_Packet) จากนั้นนำไปเก็บบันทึกไว้ที่ ฐานข้อมูล MongoDB ในเครื่อง Name Node แสดงให้เห็นในภาพที่ 3.9



ภาพที่ 3.9 แสดงการคัดกรองข้อมูล

การใช้ Map Function เพื่อใช้ในการคัดกรองข้อมูลให้เหลือเฉพาะที่จำเป็นเท่านั้น โดย Map Function จะทำหน้าที่ในการับคู่ข้อมูลระหว่าง Input เป็น Key กับ value ได้ออกมาเป็น Intermediate key/value pair คือ key กับ value ที่ได้ออกมาจากหลัง function Map จากนั้นจะทำการรวบรวมค่าที่มี key เหมือนกันไว้ด้วยกัน แสดงให้เห็นดังภาพที่ 3.12

- การส่งผลลัพธ์และข้อความแจ้งเตือน ไปยังศูนย์กลางซึ่งได้จากการที่เมื่อพบการโจมตีแบบDDoS จะมีข้อความแจ้งเตือนกับผู้ดูแลระบบ เพื่อสามารถลดความแออัดของการจราจรบนระบบเครือข่ายลง เนื่องจากในการส่งผลลัพธ์ไปยังศูนย์กลางจะส่งไปเป็นในรูปแบบข้อมูลแบบ Text ที่มีขนาดของข้อมูลที่เล็กจะไม่ส่งผลกระทบต่อการจราจรของระบบเครือข่าย



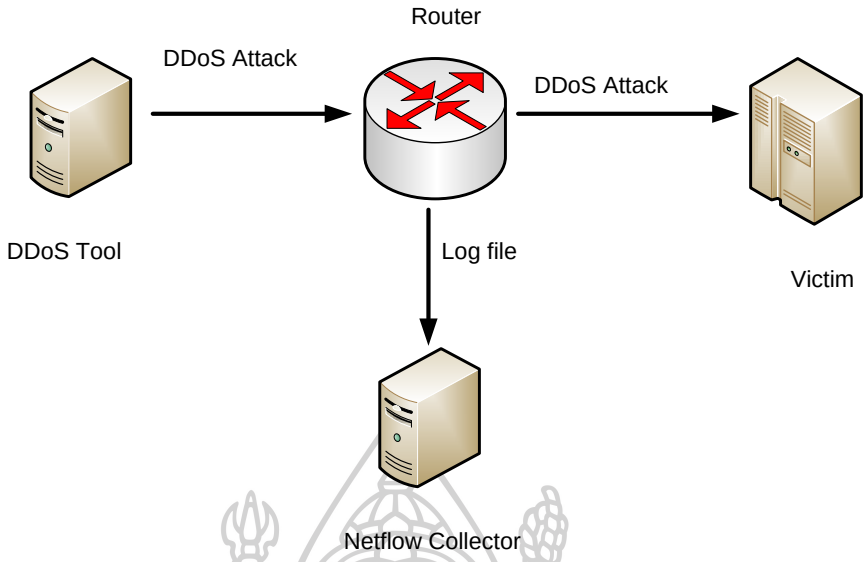
ภาพที่ 3.10 แสดงการส่งข้อความแจ้งเตือนและผลลัพธ์ไปยังศูนย์กลาง

จากภาพที่ 3.10 แสดงให้เห็นถึงการจัดส่งผลลัพธ์และการส่งข้อความแจ้งเตือนไปยังศูนย์กลาง โดยจะทำงานดังนี้คือ เมื่อมีการประมวลด้วยอัลกอริทึมตรวจจับ DDoS ที่โหนดใดก็ตาม หากมีการตรวจพบ DDoS ระบบที่พัฒนาขึ้นจะทำการส่งข้อความแจ้งเตือน พร้อมกับส่งผลลัพธ์ที่ได้จากการประมวลผลเช่น หมายเลข IP Address และ วันเวลาที่โดนโจมตี ส่งไปยังศูนย์กลาง เพื่อให้ทราบและแก้ปัญหาในกรณีที่โหนดในระบบเครือข่ายถูกโจมตี

3.2.4 ทดสอบระบบ

การทดสอบระบบต้นแบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ตามที่ได้ออกแบบไว้ มีขั้นตอนดังนี้

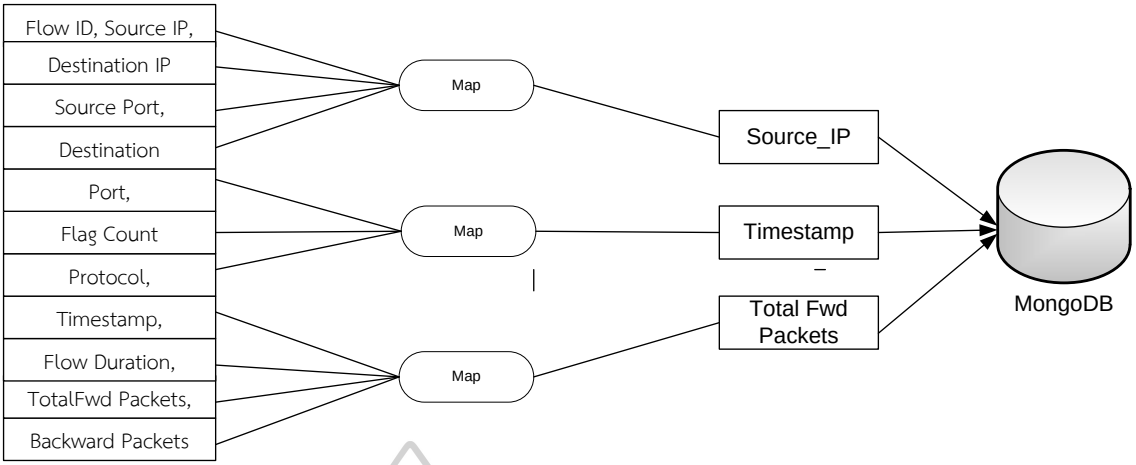
- นำ Dataset CTU-13 ทำการทดสอบ เพื่อหาความค่า Accuracy ของ อัลกอลิทึม ที่จะนำมาใช้ในการพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่
- จัดเก็บ Log file ที่เป็น DDoS เพื่อใช้ในการทดสอบระบบ ดังแสดงให้เห็นในภาพที่ 3.11



ภาพที่ 3.11 การจัดเก็บ log file DDoS Attack

ภาพที่ 3.11 แสดงการเก็บ log file ที่เป็น DDoS บนระบบจริงของเครือข่าย UniNet ดังต่อไปนี้

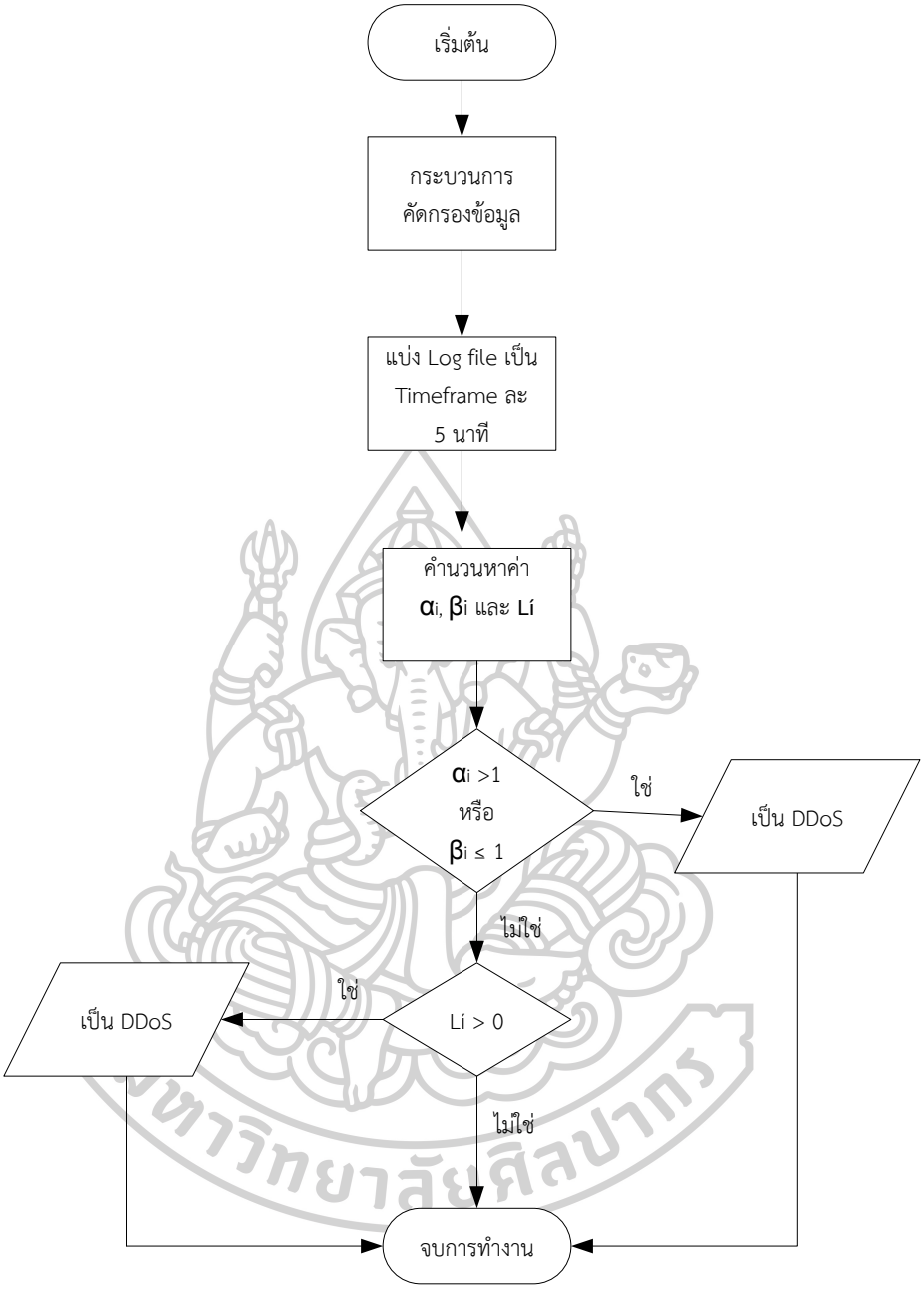
- โดยการตั้งเครื่องเซิฟเวอร์ที่เป็นเป้าหมายในการโจมตี (Victim) บนระบบเครือข่าย UniNet หลังจากนั้นทำการโจมตีด้วย DDoS Tool โดยมีเป้าหมายการโจมตีคือเครื่อง Victim ดังภาพที่ 3.11 และทำการโจมตี เป็นเวลา 1 ชั่วโมง แล้วทำการหยุดโจมตีและเก็บ log file ปกติ เป็นเวลา 1 ชั่วโมงเช่นกัน
- ตั้งค่าการจัดเก็บ log file ให้บันทึกที่เครื่อง Netflow Collector ด้วย Netflow Application ที่ติดตั้งบน Router ของระบบเครือข่าย UniNet
- การจัดเก็บข้อมูลที่เป็น NetFlow log file จาก Backbone Node / Distribution Node เข้าสู่ HDFS ที่มีคุณสมบัติในการจัดการข้อมูลขนาดใหญ่ โดยจะให้แสดงให้เห็นขั้นตอนการนำเข้าและบันทึกข้อมูล ดังภาพที่ 3.11
- การคัดกรองข้อมูลทำการคัดกรองข้อมูลโดยใช้ MapReduce โดยสั่งให้ MapReduce ทำงานทุกๆ 5 นาที ดังภาพ 3.12



ภาพที่ 3.12 แสดงการทำงานMap Function เพื่อทำการคัดกรองข้อมูล

จากภาพที่ 3.12 แสดงถึงการคัดกรองข้อมูลเพื่อให้เหลือเพียง field ที่จำเป็นต่อการประมวลผล จะช่วยลดเวลาและทรัพยากรในการประมวลผลลง Netflow log file มี field อยู่จำนวนหลาย field ซึ่งบางตัวไม่มีความจำเป็นในการนำมาประมวลผล จากภาพที่ 3.12 ใน Netflow log file จะมี field ดังนี้ Flow ID, Source IP, Source Port, Destination IP, Destination Port, Protocol, Timestamp, Flow Duration, Total Fwd Packets, Total Backward Packets, Total Length of Fwd Packets, Total Length of Bwd Packets, Fwd Packet Length Max, Fwd Packet Length Min, Fwd packet Length Mean, Fwd Packet Length Std, Bwd Packet ฯลฯ หากมีจำนวนมากเกินไป ทำให้การประมวลช้าลงตามไปด้วย

การตรวจจับหาการโจมตี DDoS จะใช้อัลกอริทึมการตรวจจับ DDoS ของ Vishal Maheshwari โดยนำข้อมูลที่คัดกรองด้วย MapReduce แล้วทำการประมวลผลด้วยอัลกอริทึมตรวจจับ DDoS จะทำการทดสอบระบบโดยทำการประมวลผลในช่วงเวลาที่ถูกโจมตีด้วย DDoS Tool และ ในช่วงเวลาที่มีการใช้งาปกติ เพื่อทดสอบว่าระบบที่พัฒนาขึ้นจะแจ้งผล อย่างไร โดยมีผังการทำงานของอัลกอริทึมการตรวจจับ DDoS ดังต่อไปนี้



ภาพที่ 3.13 แสดงการทำงานของอัลกอริทึมในการตรวจจับ DDoS

จากภาพที่ 3.12 แสดงให้เห็นหลักการทำงานของ อัลกอริทึมในการตรวจ DDoS มีดังต่อไปนี้

- อัลกอริทึมในการตรวจ DDoS ติดตั้งไว้ที่เครื่อง Name Node
- เมื่อทำการประมวลผลจะนำข้อมูลที่ผ่านการคัดกรองแล้วจาก ฐานข้อมูล MongoDB ที่ได้ติดตั้งไว้บนเครื่อง Name Node เพื่อทำการประมวลผล

- ในการประมวลผลจะใช้อัลกอริทึมในการตรวจ DDoS [9] ซึ่งจะทำงานตามเงื่อนไขทั้งหมด 3 เงื่อนไขดังนี้

1. $a_i = ln(\left(\frac{Y_i}{Y_1}\right)$ ถ้า $a_i > 1$ เป็น DDoS
   - Y_i = จำนวนแพ็คเกต ที่ถูกส่งด้วย IP Address เดียวกันใน 1 Window
   - Y_1 = ค่าของ Y_i ที่เป็นตัวแรกของ window ที่ 1
2. $\beta_i = ln(\left(\frac{Y_i}{Y_{i-1}}\right)$ ถ้า $\beta_i \leq 1$ เป็น DDoS
   - Y_{i-1} = ค่าตัวก่อนหน้า Y_i
3. $L_i = \frac{1}{t_i} ln \frac{\Delta(X_i)}{\Delta(X_1)}$. ถ้า $L_i > 0$ เป็น DDoS
   - t_i = ค่าความคาดเคลื่อน
   - X_i = จำนวนแพ็คเกตทั้งหมดที่ถูกส่งใน 1 window
   - X_1 = ค่าใน window ที่ 1 ที่เป็นจำนวนแพ็คเกตทั้งหมดที่ถูกส่งใน 1 window

โดยเงื่อนไขข้างต้นจะผ่านการประมวลผลโดยการตรวจสอบเงื่อนไขทีละข้อ เช่น ตรวจสอบเงื่อนไข a_i >1 หากไม่ตรงกับเงื่อนไขนี้ ต้องตรวจสอบกับเงื่อนไข 2 และ 3 ตามลำดับต่อไป หากไม่ตรงกับเงื่อนใดเลยทั้ง 3 เงื่อนไขโฟล์วนั้นๆ ถือว่าไม่เป็น DDoS

3.2.5 การประเมินผล

ในการประเมินผลของงานวิจัยนี้จะทำการประเมินผลโดยการวัดผล คือ Delay และ Accuracy

- ค่า Delay หาจากเวลาที่ทำการประมวลผลแล้วพบ DDoS เปรียบเทียบกับ เวลาของการโจมตี DDoS ที่เกิดขึ้นจริง ว่าใช้เวลา เท่าใดที่จะตรวจพบ DDoS เนื่องจากในการทำ MapReduce นั้นมี Delay ในการดึงข้อมูลมากจาก HDFS เพื่อทำการ MapReduce
- วัดประสิทธิภาพระบบเครือข่ายด้วย Confusion Matrix โดยจะทำการวัดประสิทธิภาพดังต่อไปนี้

1. Accuracy คือ ค่าที่บอกว่าโปรแกรมสามารถได้ผลแม่นยำเท่าใด

2. Recall (True Positive Rate) คือ ค่าที่บอกว่าโปรแกรมทำนายได้ว่าจริง เป็นอัตราส่วนเท่าใดของจริงทั้งหมด
3. True Negative Rate (TNR) คือ ค่าที่บอกว่าโปรแกรมทำนายได้ว่าไม่จริง เป็นอัตราส่วนเท่าใดของจริงทั้งหมด
4. False Positive Rate (TPR) คือ ค่าที่บอกว่าโปรแกรมทำนายว่าจริง เป็นอันตราส่วนเท่าใดของไม่จริงทั้งหมด
5. False Negative Rate (FNR) คือ ค่าที่บอกว่าโปรแกรมทำนายว่าไม่จริง เป็นอันตราส่วนเท่าใดของจริงทั้งหมด
6. Precision คือ ค่าที่บอกว่าโปรแกรมทำนายว่าจริง ถูกต้องเท่าใด

3.2.6 สรุปผล

เมื่อนำ หลักการของอัลกอริทึม Vishal Maheshwari มาใช้ร่วมกันแล้ว จะทำให้ผู้ดูแลระบบทราบว่าระบบโดนโจมตีด้วย DDoS ก่อนที่ระบบจะเกิดความเสียหายหรือไม่สามารถใช้งานระบบเครือข่ายได้ สามารถนำข้อมูลที่บันทึกไว้ใน Data lake มาวิเคราะห์เครือข่ายเพื่อหาแนวทางในการป้องกันและแก้ไขระบบเครือข่ายได้

บทที่ 4
ผลการดำเนินการวิจัย

จากการพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ ซึ่งมีการรายงานผลการดำเนินงานของงานวิจัยในครั้งนี้ได้แจกแจงออกมาเป็นข้อๆ ได้ดังนี้

4.1 ผลการดำเนินงานที่ได้จากการศึกษาและวิเคราะห์ปัญหา

ปัญหาของระบบ UniNet ประเด็นหลักคือ การที่มีปริมาณข้อมูลมากเกินไปจนไม่สามารถที่จะนำมาเก็บบันทึกเพื่อทำการวิเคราะห์และประมวลผลได้ทำให้เกิดปัญหาตามมาอีกหลายอย่างเช่น ระบบไม่สามารถที่จะนำข้อมูล (log file) มาใช้ในการวิเคราะห์ระบบเครือข่ายเพื่อป้องกันการโจมตี หรือไม่สามารถนำข้อมูลมาเพื่อใช้ในการตัดสินในด้านต่างๆ ได้ เนื่องจากไม่มีข้อมูลดิบที่ทำการบันทึกไว้เพื่อนำมาวิเคราะห์เป็นข้อมูลที่ช่วยในการตัดสิน

ข้อมูล (log file) ในระบบเครือข่าย UniNet มีจำนวนมากและต้องใช้พื้นที่จำนวนมากในการจัดเก็บ เพื่อที่จะนำมาประมวลผล โดยในแต่ละ Interface ของ Router จะสร้าง log file เพื่อมาบันทึกไว้ในเครื่อง Netflow Collector โดยแสดงขนาดของข้อมูลดังตารางที่ 4.1

nfcapd.201412020000	0:05	201412020000 File	1,685 KB
nfcapd.201412020005	0:10	201412020005 File	1,383 KB
nfcapd.201412020010	0:15	201412020010 File	1,480 KB
nfcapd.201412020015	0:20	201412020015 File	1,330 KB
nfcapd.201412020020	0:25	201412020020 File	1,393 KB
nfcapd.201412020025	0:30	201412020025 File	1,381 KB
nfcapd.201412020030	0:35	201412020030 File	1,492 KB
nfcapd.201412020035	0:40	201412020035 File	1,354 KB
nfcapd.201412020040	0:45	201412020040 File	1,621 KB
nfcapd.201412020045	0:50	201412020045 File	1,723 KB
nfcapd.201412020050	0:55	201412020050 File	1,313 KB
nfcapd.201412020055	1:00	201412020055 File	1,250 KB
nfcapd.201412020100	1:05	201412020100 File	1,211 KB

ตารางที่ 4.1 ขนาดของ log file ในเวลา 1 ชั่วโมง/Interface ใน Router

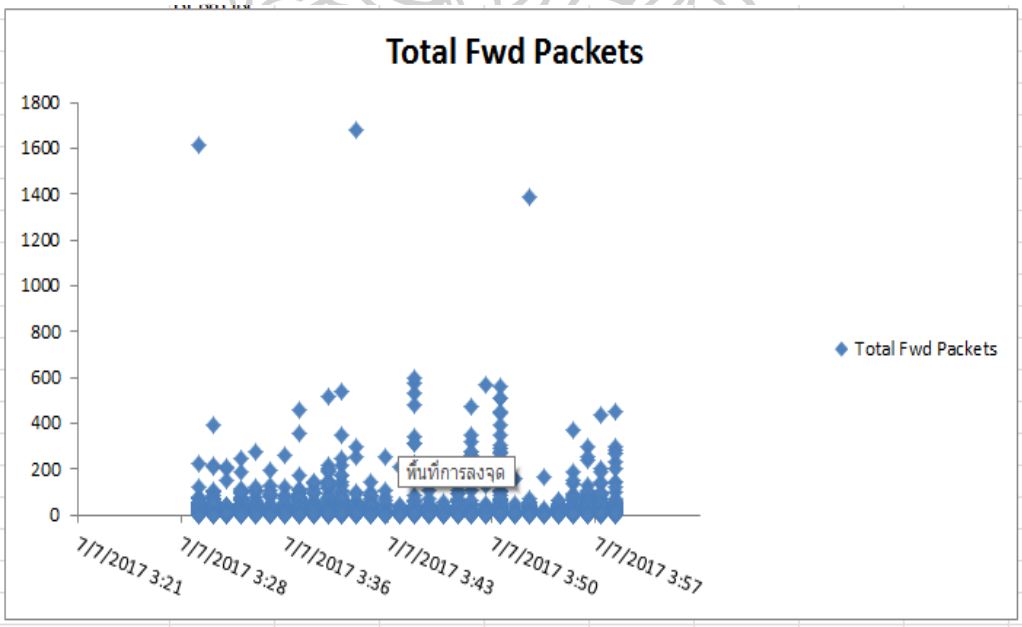
จากตารางที่ 4.1 ผู้วิจัยได้ทำการเก็บ Log file ใน 1 interface ของ Router เป็นเวลา 1 ชั่วโมงซึ่งใช้พื้นที่ในการจัดเก็บข้อมูล 8 Mb แต่ในสถาปัตยกรรมเครือข่ายของระบบ UniNet มี Router 120 ตัวและแต่ละตัวมีจำนวนของ interface ที่มากจึงทำให้ log file นั้นมีจำนวนมากเกินกว่าที่จะเก็บไว้ที่ระบบฐานข้อมูลเชิงสัมพันธ์ได้

4.2 ผลการดำเนินการในการศึกษาเพื่อออกแบบต้นแบบ

การออกแบบต้นแบบจะทำการทดสอบอัลกอริทึมที่ใช้ในการพัฒนาต้นแบบระบบการวิเคราะห์วิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ เพื่อทดสอบหาค่าความถูกต้องในการตรวจจับการโจมตีแบบ DDoS โดยมีขั้นตอนการตรวจสอบดังนี้

4.2.1 การเลือก Dataset ที่ใช้ในการทดสอบระบบ

งานวิจัยนี้ใช้หลักการโดยแบ่งการประมวลผลเป็นแบบกระจายไปยังแต่ละโหนดเพื่อลดการทำงานของเครื่องคอมพิวเตอร์ในการประมวลผล และหลีกเลี่ยงความแออัดบนระบบเครือข่าย ในการทดสอบระบบที่พัฒนาขึ้นมีการใช้หลักการวิเคราะห์อนุกรมเวลา (Time Series Analysis) ด้วยการนำ CTU-13 Dataset มาเป็นชุดข้อมูลในการทดสอบอัลกอริทึมการตรวจจับ DDoS เพื่อหาค่าความถูกต้อง และได้ทำการสร้างกราฟเพื่อตรวจสอบกลุ่มของข้อมูล ดังภาพที่ 4.1



ภาพที่ 4.1 แสดงการสร้างกราฟเพื่อแสดงค่าคงที่ (Stationary) ของข้อมูล

จากภาพ 4.1แสดงให้เห็นรูปการกระจายตัวของข้อมูลในลักษณะที่ไม่สามารถคาดเดาได้ว่าในช่วงเวลาถัดไปแนวโน้มของข้อมูลจะไปในทิศทางใด ในชุดของข้อมูล CTU-13 Dataset เป็นชุดข้อมูล

ที่มีขนาดใหญ่ มีรูปแบบการโจมตีที่สามารถทำการแยกออกมาได้ทั้งหมด 13 สถานการณ์ (Scenarios) เพื่อให้การทดลองมีประสิทธิภาพทางผู้วิจัยเลือกใช้ สถานการณ์ที่ 10 ของชุดข้อมูล CTU-13 Dataset เนื่องจากเป็นชุดข้อมูล Netflow log file ที่มีป้ายกำกับของแต่ละโฟล์วว่าโฟล์วใดที่เป็นการโจมตีแบบ DDoS โดยแสดงให้เห็นในตารางที่ 4.2 ซึ่งในแต่ละสถานการณ์จะมีรูปแบบการโจมตีที่แตกต่างกันไป

Scen.	Total Flows	Botnet Flows	Normal Flows	C&C Flows	Background Flows
1	2,824,636	39,933(1.41%)	30,387(1.07%)	1,026(0.03%)	2,753,290(97.47%)
2	1,808,122	18,839(1.04%)	9,120(0.5%)	2,102(0.11%)	1,778,061(98.33%)
3	4,710,638	26,759(0.56%)	116,887(2.48%)	63(0.001%)	4,566,929(96.94%)
4	1,121,076	1,719(0.15%)	25,268(2.25%)	49(0.004%)	1,094,040(97.58%)
5	129,832	695(0.53%)	4,679(3.6%)	206(1.15%)	124,252(95.7%)
6	558,919	4,431(0.79%)	7,494(1.34%)	199(0.03%)	546,795(97.83%)
7	114,077	37(0.03%)	1,677(1.47%)	26(0.02%)	112,337(98.47%)
8	2,954,230	5,052(0.17%)	72,822(2.46%)	1,074(2.4%)	2,875,282(97.32%)
9	2,753,884	179,880(6.5%)	43,340(1.57%)	5,099(0.18%)	2,525,565(91.7%)
10	1,309,791	106,315(8.11%)	15,847(1.2%)	37(0.002%)	1,187,592(90.67%)
11	107,251	8,161(7.6%)	2,718(2.53%)	3(0.002%)	96,369(89.85%)
12	325,471	2,143(0.65%)	7,628(2.34%)	25(0.007%)	315,675(96.99%)
13	1,925,149	38,791(2.01%)	31,939(1.65%)	1,202(0.06%)	1,853,217(96.26%)

ตารางที่ 4.2 แสดง Scenarios ทั้งหมดของ CTU-13 Dataset

เมื่อเลือกชุดข้อมูลของ Dataset เรียบร้อยแล้วก่อนที่จะนำ Dataset เข้าสู่การประมวลผล ซึ่งแสดงรูปแบบของ NetFlow log file ออกมาเป็นคอลัมน์ ดังตารางที่ 4.3

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Flow ID	Source IP	Source Po	Destinatio	Destinatio	Protocol	Timestam	Flow Dura	Total Fwd	Total Bac	Total Leng	Total Leng	Fwd Packe	Fwd Packe	Fwd Packe	Fwd Packe	Bwd Packe	Bwd Packe	Bwd Packe	Bwd Packe	Flow Bytes
2	192.168.1(	104.24.11(	443	192.168.1(	9443	6	######	233	2	0	12	0	6	6	6	0	0	0	0	0	51502.15
3	192.168.1(	104.244.4(	443	192.168.1(	58610	6	######	1	2	0	37	0	31	6	18.5	17.67767	0	0	0	0	37000000
4	192.168.1(	104.25.21(	443	192.168.1(	9456	6	######	3	2	0	12	0	6	6	6	0	0	0	0	0	4000000
5	192.168.1(	104.28.24.	443	192.168.1(	9455	6	######	3	2	0	12	0	6	6	6	0	0	0	0	0	4000000
6	192.168.1(	104.87.80.	443	192.168.1(	58627	6	######	3	2	0	37	0	31	6	18.5	17.67767	0	0	0	0	12300000
7	192.168.1(	104.97.71.	443	192.168.1(	58710	6	######	203	2	1	37	6	31	6	18.5	17.67767	6	6	6	0	211822.7
8	192.168.1(	104.97.71.	443	192.168.1(	58708	6	######	49	3	0	43	0	31	6	14.33333	14.43376	0	0	0	0	877551
9	192.168.1(	108.161.1(	443	192.168.1(	58645	6	######	17262	2	0	12	0	6	6	6	0	0	0	0	0	695.1686
10	192.168.1(	108.161.1(	443	192.168.1(	58646	6	######	13464	2	0	12	0	6	6	6	0	0	0	0	0	891.2656
11	192.168.1(	108.161.1(	443	192.168.1(	58641	6	######	16874	2	0	12	0	6	6	6	0	0	0	0	0	711.1533
12	192.168.1(	108.161.1(	443	192.168.1(	58644	6	######	1	2	0	12	0	6	6	6	0	0	0	0	0	12000000
13	192.168.1(	108.161.1(	443	192.168.1(	58643	6	######	87	1	1	6	6	6	6	6	0	6	6	6	0	137931
14	149.202.2(	149.202.2(	443	192.168.1(	58607	6	######	1	2	0	59	0	53	6	29.5	33.23402	0	0	0	0	59000000
15	149.202.2(	149.202.2(	443	192.168.1(	58607	6	######	15	1	1	6	6	6	6	6	0	6	6	6	0	800000
16	172.16.0.1	172.16.0.1	49650	192.168.1(	80	6	######	1293792	3	7	26	11607	20	0	8.666667	10.2632	5840	0	1658.143	2137.297	8991.399
17	172.16.0.1	172.16.0.1	49650	192.168.1(	80	6	######	4421382	4	0	24	0	6	6	6	0	0	0	0	0	5.428167
18	172.16.0.1	172.16.0.1	51684	192.168.1(	80	6	######	1083538	3	6	26	11601	20	0	8.666667	10.2632	4380	0	1933.5	1757.79	10730.59
19	172.16.0.1	172.16.0.1	51684	192.168.1(	80	6	######	80034360	8	4	56	11601	20	0	7	5.656854	8760	0	2900.25	4128.319	145.6499
20	172.16.0.1	172.16.0.1	51686	192.168.1(	80	6	######	642654	3	6	26	11607	20	0	8.666667	10.2632	5840	0	1934.5	2538.919	18101.5
21	172.16.0.1	172.16.0.1	51686	192.168.1(	80	6	######	79731718	8	5	56	11601	20	0	7	5.656854	5840	0	2320.2	2436.833	146.2028
22	172.16.0.1	172.16.0.1	51687	192.168.1(	80	6	######	306157	3	6	26	11607	20	0	8.666667	10.2632	5840	0	1934.5	2538.919	37996.84
23	172.16.0.1	172.16.0.1	51687	192.168.1(	80	6	######	79780371	8	4	56	11607	20	0	7	5.656854	11595	0	2901.75	5795.501	146.1888
24	172.16.0.1	172.16.0.1	51688	192.168.1(	80	6	######	683575	3	5	26	11607	20	0	8.666667	10.2632	10135	0	2321.4	4413.202	17042.82

ตารางที่ 4.3 แสดงการจัดการแยก log file ออกเป็นคอลัมน์

ตารางที่ 4.3 แสดงการนำข้อ log file ที่เป็น Text เข้ามาจัดเรียงให้เป็นคอลัมน์ เพื่อแสดงชุดของข้อมูลก่อนที่จะนำเข้าการจัดเก็บใน HDFS เพื่อเตรียมพร้อมในการคัดกรองข้อมูลเพื่อลดปริมาณข้อมูลให้เหลือเฉพาะที่จำเป็นต่อการประมวลผลเท่านั้น

4.2.2 การคัดกรองข้อมูล **(Filtering Data)**

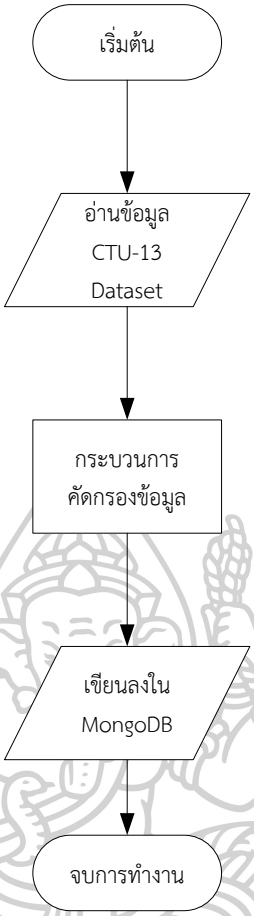
การประมวลผลข้อมูลการตรวจจับ DDoS โดยใช้ NetFlow log file ซึ่งมีจำนวน field ที่ไม่จำเป็นต่อการประมวลผลจึงทำให้ใช้พื้นที่ในการจัดเก็บข้อมูลมาก การลดปริมาณข้อมูลให้เหลือเพียงข้อมูลที่จำเป็นต่อการประมวลผลนั้นสามารถลดพื้นที่ในการจัดเก็บข้อมูล และลดภาระการทำงานของคอมพิวเตอร์เซิร์ฟเวอร์ การคัดกรองข้อมูลใช้เฉพาะสิ่งที่ต้องการด้วย MapReduce ซึ่งจะทำการลดปริมาณ Field จากที่มีทั้งหมด ดังตารางที่ 4.4

Flow ID	Source IP	Source Port	Destination IP	Destination Po	Protocol	Timestamp	Flow Duration	Total Fwd Pac
Total Backward	Total Length of	Total Length of	Fwd Packet Len	Fwd Packet Len	Fwd Packet Len	Fwd Packet Len	Bwd Packet Len	Bwd Packet Le
Bwd Packet Le	Bwd Packet Le	Flow Bytes/s	Flow Packets/s	Flow IAT Mean	Flow IAT Std	Flow IAT Max	Flow IAT Min	Fwd IAT Total
Fwd IAT Mean	Fwd IAT Std	Fwd IAT Max	Fwd IAT Min	Bwd IAT Total	Bwd IAT Mean	Bwd IAT Std	Bwd IAT Max	Bwd IAT Min
Fwd PSH Flags	Bwd PSH Flags	Fwd URG Flags	Bwd URG Flags	Fwd Header Le	Bwd Header Le	Fwd Packets/s	Bwd Packets/s	Min Packet Len
Max Packet Le	Packet Length	Packet Length	Packet Length	FIN Flag Count	SYN Flag Count	RST Flag Count	PSH Flag Count	ACK Flag Count
URG Flag Coun	CWE Flag Coun	ECE Flag Count	Down/Up Ratio	Average Packet	Avg Fwd Segme	Avg Bwd Segm	Fwd Header Le	Fwd Avg Bytes/
Fwd Avg Packe	Fwd Avg Bulk R	Bwd Avg Bytes/	Bwd Avg Packe	Bwd Avg Bulk R	Subflow Fwd Pa	Subflow Fwd B	Subflow Bwd P	Subflow Bwd B
Init_Win_bytes_	Init_Win_bytes_	act_data_pkt_fw	min_seg_size_f	Active Mean	Active Std	Active Max	Active Min	Idle Mean
Idle Std	Idle Max	Idle Min	Label					

ตารางที่ 4.4 Field ทั้งหมดของ CTU-13 Dataset

ตารางที่ 4.4 แสดง field ทั้งหมดของ Netflow log file ใน CTU-13 Dataset ซึ่งเป็น Dataset ที่นำมาใช้ในการทดสอบเพื่อหาประสิทธิภาพของอัลกอริทึมการตรวจจับ DDoS

ในการคัดกรองข้อมูลเพื่อลดปริมาณการใช้พื้นที่การจัดเก็บข้อมูล ลดเวลาในการประมวลผลข้อมูล เพิ่มประสิทธิภาพให้กับระบบ โดยแสดงผังการทำงานดังภาพที่ 4.2



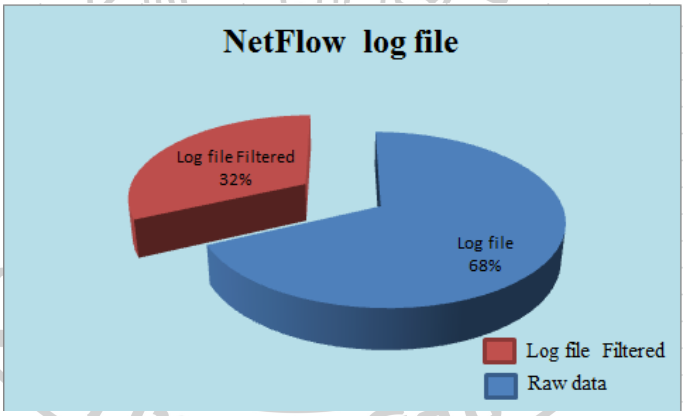
ภาพที่ 4.2 แสดงผังงานในการคัดกรองข้อมูล

จากภาพแสดงการคัดกรองข้อมูลโดยใช้ MapReduce Function เมื่อเริ่มกระบวนการคัดกรองข้อมูล ทางผู้วิจัยจะทำการนำ CTU-13 Dataset ซึ่งเป็น Netflow log file ที่มีป้ายกำกับโฟล์วเพื่อใช้ในการทดลองหาประสิทธิภาพของอัลกอริทึมการตรวจจับ DDoS ถือว่าเป็นการนำข้อมูลเข้าระบบ เพื่อทำการ MapReduce ที่มีการพัฒนาโปรแกรมคำสั่งให้ลดปริมาณของ field ของ Dataset ที่มีจำนวน 85 คอลัมน์ ให้เหลือ 3 คอลัมน์ และนำไปจัดเก็บไว้ในฐานข้อมูล MongoDB ดังภาพที่ 4.3

	Timestamp	Source_IP	Total_Fwd_Packets
0	7/7/2017 3:30	104.16.207.165	2
1	7/7/2017 3:30	104.16.28.216	1
2	7/7/2017 3:30	104.16.28.216	1
3	7/7/2017 3:30	104.17.241.25	1
4	7/7/2017 3:30	104.19.196.102	2
...	...	...	...
225740	7/7/2017 5:02	72.21.91.29	1
225741	7/7/2017 5:02	72.21.91.29	1
225742	7/7/2017 5:02	72.21.91.29	1
225743	7/7/2017 5:02	8.41.222.187	2
225744	7/7/2017 5:02	8.43.72.21	1

ภาพที่ 4.3 แสดงผลลัพธ์ที่ได้จากการคัดกรองข้อมูลเพื่อลดขนาดลง

จากภาพที่ 4.3 อธิบายได้ว่า เมื่อเข้าสู่กระบวนการ MapReduce Function เพื่อทำการคัดกรองข้อมูลเรียบร้อยแล้ว ปริมาณ field ที่มีจำนวน 85 คอลัมน์จะลดลงเหลือ 3 คอลัมน์ คือ Timestamp Source_IP และ Total_Fwd_Packets



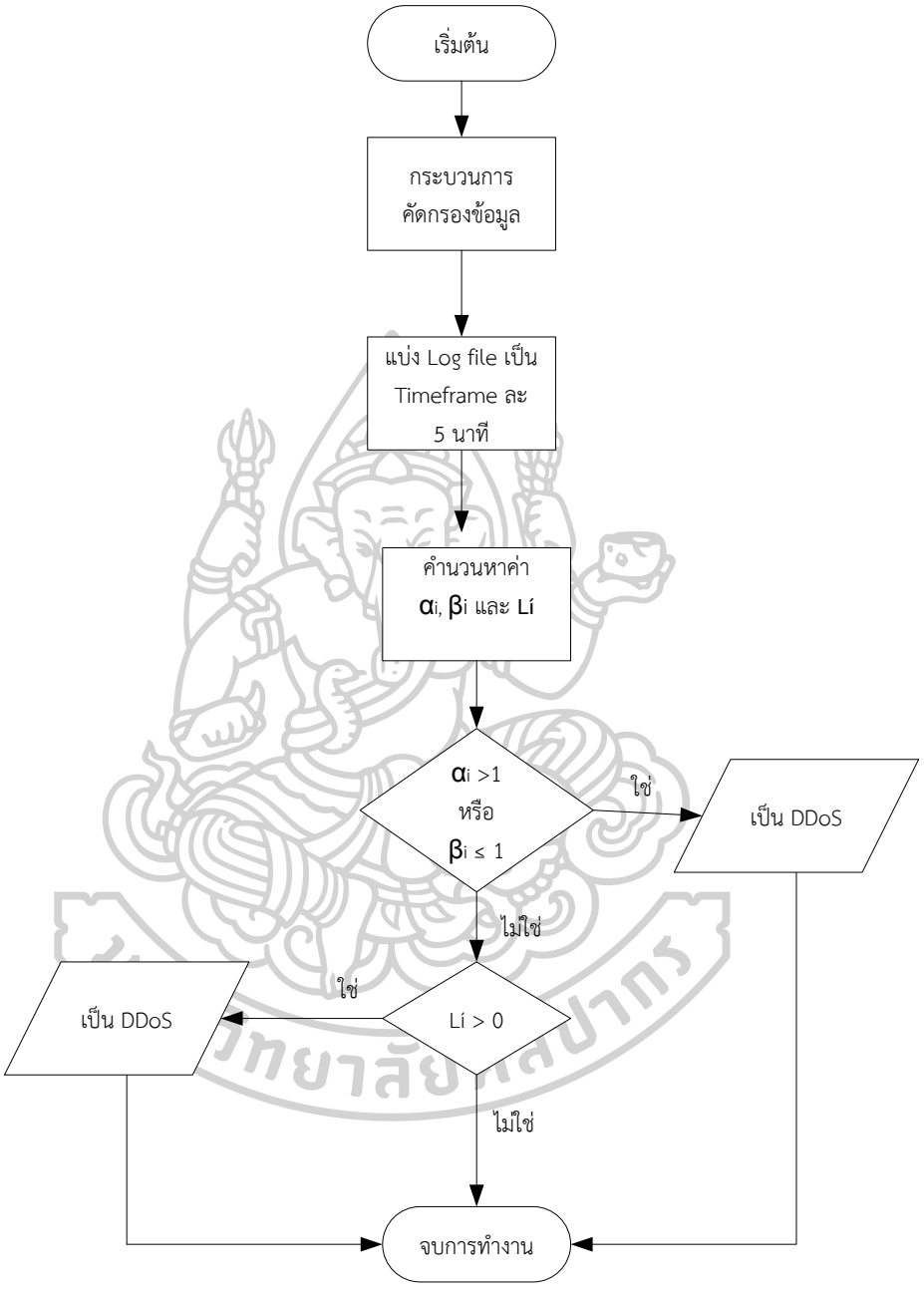
ภาพที่ 4.4 แสดงจำนวนข้อมูลที่ลดลงหลังจากการคัดกรองข้อมูล

จากภาพที่ 4.4 แสดงให้เห็นขนาดของข้อมูลดิบที่ลดลงจาก 8.1 MB แต่เมื่อผ่านกระบวนการคัดกรองข้อมูลแล้วจะเหลือเพียง 3.8 MB จึงทำให้ลดพื้นที่ในการจัดเก็บข้อมูลและเพิ่มประสิทธิภาพในการประมวลผล

4.2.3 ทดสอบอัลกอริทึมในการทดสอบระบบ

การแยกโฟล์วที่เป็นภัยคุกคามกับโฟล์วปกติสามารถทำได้ยาก เนื่องจากต้นแบบที่พัฒนาขึ้นเมื่อได้ทำการทดลองแล้ว ผลลัพธ์ที่ออกมาสามารถแยกโฟล์วปกติและโฟล์วที่เป็นภัยคุกคามออกจากกันได้อย่างชัดเจน ด้วยอัลกอริทึมการตรวจจับ DDoS เมื่อทำการคัดกรองข้อมูลให้เหลือเฉพาะที่

จำเป็นต่อการประมวลผลซึ่งสามารถจัดการกับปัญหาเหล่านี้ได้ จะช่วยในการวิเคราะห์และจัดการระบบเครือข่ายให้มีประสิทธิภาพ ขั้นตอนในการทำงานของอัลกอริทึมทำงานดังภาพที่ 4.5



ภาพที่ 4.5 แสดงการทำงานอัลกอริทึมการตรวจจับ DDoS

เมื่อทำการคัดกรองข้อมูลด้วย MapReduce Function และจัดเก็บผลลัพธ์ที่ได้ไว้ในฐานข้อมูล MongoDB แล้ว จะทำการทดสอบอัลกอริทึมด้วยการนำข้อมูลที่ได้จากการคัดกรองแล้วนำมาประมวลผลด้วยอัลกอริทึมการตรวจจับ DDoS ดังนี้

1. แบ่ง log file ตามช่วงเวลา ช่วงละ 10 นาที เรียกว่า 1 window ในแต่ละ window จะแยกเป็น window ย่อย คือ window ละ 1 นาที โดยจะแสดงให้เห็นโดยการแยกเป็นแต่ละ window โดยจะทำการแสดงให้เห็นรูปแบบการแบ่งข้อมูลตาม Timestamp ในรูปแบบของ Microsoft Excel ดังตารางที่ 4.5

Timestamp	Running	Source_IP	Unique_IPs	Total_Fwd_Packets
2017-07-07 03:30:00.000	1		122	20663
2017-07-07 03:31:00.000	2		159	13567
2017-07-07 03:32:00.000	3		66	2030
2017-07-07 03:33:00.000	4		123	8190
2017-07-07 03:34:00.000	5		105	5388
2017-07-07 03:35:00.000	1		79	3136
2017-07-07 03:36:00.000	2		75	3474
2017-07-07 03:37:00.000	3		80	8595
2017-07-07 03:38:00.000	4		110	5485
2017-07-07 03:39:00.000	5		87	5219
2017-07-07 03:40:00.000	1		94	10600
2017-07-07 03:41:00.000	2		88	5687
2017-07-07 03:42:00.000	3		81	4603
2017-07-07 03:43:00.000	4		39	1250
2017-07-07 03:44:00.000	5		29	674
2017-07-07 03:45:00.000	1		37	5839
2017-07-07 03:46:00.000	2		41	3065
2017-07-07 03:47:00.000	3		29	908
2017-07-07 03:48:00.000	4		24	2588
2017-07-07 03:49:00.000	5		33	4051

ตารางที่ 4.5 แสดงรูปแบบการแบ่งข้อมูลตาม Timestamp

จากตารางที่ 4.5 แสดงการแบ่ง window ของ log file โดยแบ่งออกเป็น window ละ 5 นาที และภายใน window จะแบ่งออกเป็น window ย่อย 5 window โดยจะแบ่งแต่ละ window ย่อยเป็น window ละ 1 นาที ด้วย Timestamp

2. ใน window แต่ละ window จะมีการลดจำนวน field ของ log file ที่จำเป็นต่อการประมวลให้เหลือเพียง 3 field เท่านั้น คือ Timestamp, Source_IP และ Total_Fwd_Packets โดยในแต่ละ field จะบ่งบอกลักษณะข้อมูลใน NetFlow log fileดังนี้

- TimeStamp คือ ช่วงเวลาที่มีการส่งข้อมูล
- Source_IP คือ IP Address ที่ทำส่งข้อมูลไปยังปลายทาง
- Total_Fwd_Packet คือ จำนวนแพ็คเกตที่ส่งไปยังปลายทาง

3. เมื่อเตรียมข้อมูลที่จำเป็นในการประมวลผล ในแต่ละ window แล้วตัวแปรที่สำคัญที่จะนำมาใช้ประมวลผลในอัลกอริทึมการตรวจจับ DDoS จะมีทั้งหมด 3 ตัวแปร คือ Unique, Total packet และ γ_i ซึ่งจะคำนวณหาค่าดังกล่าวมาจาก field Source_IP และ Total_Fwd_Packet ซึ่งจะทำการอธิบายดังรายละเอียดด้านล่างต่อไปนี้

- Unique คำนวณหาได้จาก ในเวลา 1 นาที มีหมายเลย IP Address เดียวกันทำการส่งข้อมูลทั้งหมดกี่ครั้ง
- Total packets คำนวณหาได้โดย นำจำนวนการส่งข้อมูลทั้งหมดของ Unique IP ในเวลา 1 นาทีมีหาค่ารวมมีการสั่งจำนวนกี่แพ็คเกต
- $\boldsymbol{\gamma_i}$ คำนวณหาได้จาก จำนวน Total packets หารด้วย จำนวน Unique

$$\left(\frac{\textbf{Total Packets}}{\textbf{Unique}}\right)$$

ในการคำนวณหาค่า Unique, Total Packets และ $\boldsymbol{\gamma_i}$ ของ window ย่อยจะแสดงให้เห็นในตารางที่ 4.6

Files(minute)	Unique	Total Packets	$\boldsymbol{\gamma_i}$
1st	106	40216	379
2nd	71	38759	545
3rd	79	35389	447
4th	56	36960	660
5th	36	37774	1049

ตารางที่ 4.6 แสดงการคำนวณหาค่า Unique, Total Packets และ γ_i

4. ก่อนการประมวลผลตามเงื่อนไขของอัลกอริทึม ต้องหาค่า a_i β_i และ L_i สามารถคำนวณได้ดังนี้

- a_i คำนวณหาได้จาก สมการ $ln\left(\left(\frac{Y_i}{Y_1}\right)\right.$
- β_i คำนวณหาได้จาก สมการ $ln\left(\left(\frac{Y_i}{Y_{i-1}}\right)\right.$
- L_i คำนวณหาได้จาก สมการ $\frac{1}{t_i} ln \frac{\Delta(X_i)}{\Delta(X_1)}$

การคำนวณหาค่า a_i β_i และ L_i สามารถคำนวณให้เห็นเป็นตัวอย่างได้ดังตารางที่ 4.7

Files(minute)	Unique	Total Packets	$\boldsymbol{\gamma_i}$	$\boldsymbol{\beta_i}$	$\boldsymbol{\alpha_i}$	$\boldsymbol{L_i}$
1st	106	40216	379	0	0	0
2nd	71	38759	545	0.363	0.363	-0.0005
3rd	79	35389	447	-0.198	0.165	-0.001
4th	56	36960	660	0.389	0.554	-0.001
5th	36	37774	1049	0.463	1.018	-0.001

ตารางที่ 4.7 แสดงการคำนวณหาค่า a_i β_i และ L_i.

5. ในการพิจารณา log file มีเงื่อนไขที่นำมาตัดสินใจว่าเป็นการโจมตี DDoS หรือไม่ มี 3 เงื่อนไขดังนี้คือ

- เงื่อนไขที่ 1 คือ a_i >1 หากได้ค่าการประมวลผลแล้วผลลัพธ์ตามเงื่อนไขเป็น DDoS
- เงื่อนไขที่ 2 คือ $\beta_i \leq 1$ หากได้ค่าการประมวลผลแล้วผลลัพธ์ตามเงื่อนไขเป็น DDoS
- เงื่อนไขที่ 3 คือ L_i >0 หากได้ค่าการประมวลผลแล้วผลลัพธ์ตามเงื่อนไขเป็น DDoS

การพิจารณาเงื่อนไขของอัลกอริทึมการตรวจจับ DDoS คือตรวจสอบเงื่อนไขตามลำดับคือเมื่อทำการประมวลผลแล้วได้ผลลัพธ์ตรงกับเงื่อนไขที่ 1 ก็เป็น DDoS แต่หากไม่ตรงให้ไปพิจารณาที่เงื่อนไขที่ 2 ถ้าตรงกับเงื่อนที่ 2 ก็เป็น DDoS ถ้าไม่ตรงให้ไปพิจารณาที่เงื่อนไขที่ 3

6. ผลลัพธ์ที่ได้จากการประมวลผลด้วยอัลกอริทึมการตรวจจับ DDoS โดยแบ่งออกเป็นเงื่อนไข ดังตารางที่ 4.8, 4.9 และ 4.10

การตรวจจับพบ DDoS ด้วยอัลกอริทึมตามเงื่อนไข a_i >1 แสดงให้เห็นในเวลาที่ 5 ซึ่งค่าในคอลัมน์ a_i มีค่าเท่ากับ 1.018 ตรงกับเงื่อนไขที่ 1 คือ a_i >1 ซึ่งสังเกตได้จากแพ็คเกตที่ถูกส่งโดย Unique IP ใน window นั้นมีปริมาณที่ต่ำ

Files(minute)	Unique	Total Packets	γ_i	β_i	α_i	L_i	DDoS
1st	106	40216	379	0	0	0	No
2nd	71	38759	545	0.363	0.363	-0.0005	No
3rd	79	35389	447	-0.198	0.165	-0.001	No
4th	56	36960	660	0.389	0.554	-0.001	No
5th	36	37774	1049	0.463	1.018	-0.001	Yes

ตารางที่ 4.8 การตรวจจับพบ DDoS ด้วยอัลกอริทึมตามเงื่อนไข a_i >1

จากตารางที่ 4.8 ในการหาค่า a_i มีต้องใช้ค่าตัวแปร 2 ตัวที่ใช้ในการคำนวณ คือ

- γ_i ในนาทีที่ 5 ของ window เท่ากับ 1049
- Y_1 ในนาทีที่ 5 ของ window เท่ากับ 379

มีวิธีคำนวณดังนี้ $a_i = ln\left(\left(\frac{Y_i}{Y_1}\right)\right)$

$$a_i = ln\left(\left(\frac{1049}{379}\right)\right)$$

$$a_i = 1.018$$

จากสมการข้างต้นแสดงให้เห็นว่าผลลัพธ์ของ a_i = 1.018 ซึ่งตรงกับเงื่อนไขที่ 1 คือ a_i >1 ดังนั้นสรุปได้ว่า window นาทีที่ 5 เป็น DDoS

การตรวจจับพบ DDoS ด้วยอัลกอริทึมตามเงื่อนไข $\beta_i \leq 1$ แสดงให้เห็นในเวลาที่ 4 ซึ่งค่าในคอลัมน์ β_i มีค่าเท่ากับ 0.435 ไม่ตรงกับเงื่อนไข $\beta_i \leq 1$ สังเกตได้จาก window นาทีที่ 4 มี 2 ค่าที่ค่อนข้างสูงคือ Unique IP และ Total Packets ส่งผลให้ค่า γ_i มีค่าสูง

Files(minute)	Unique	Total Packets	γ_i	β_i	α_i	L_i	DDoS
1st	70	6861	98	0	0	0	No
2nd	39	3123	80	-0.202	-0.202	-0.020	No
3rd	81	4489	55	-0.374	-0.577	-0.005	No
4th	97	8287	85	0.435	-0.142	0.001	Yes
5th	62	768	12	-1.957	-2.100	-0.035	No

ตารางที่ 4.9 การตรวจจับพบ DDoS ด้วยอัลกอริทึมตามเงื่อนไข $\beta_i \leq 1$

จากตารางที่ 4.9 ในการหาค่า β_i มต้องใช้ค่าตัวแปร 2 ตัวที่ใช้ในการคำนวณ คือ

- Y_i ในนาทีที่ 4 ของ window เท่ากับ 85
- Y_{i-1} ในนาทีที่ 4 ของ window เท่ากับ 55

คือมีวิธีคำนวณดังนี้ $\beta_i = ln\left(\left(\frac{Y_i}{Y_{i-1}}\right)\right)$

$$\beta_i = ln\left(\left(\frac{85}{55}\right)\right)$$

$$\beta_i = 0.435$$

จากสมการข้างต้นแสดงให้เห็นว่าผลลัพธ์ของ β_i = 0.435 ทำการตรวจสอบกับเงื่อนไขที่ 1 ซึ่งไม่ตรงกัน แต่ตรงกันกับเงื่อนไขที่ 2 คือ $\beta_i \leq 1$ ดังนั้นสรุปได้ว่า window นาทีที่ *4* เป็น DDoS

การตรวจจับพบ DDoS ด้วยอัลกอริทึมตามเงื่อนไข L_i >0 แสดงให้เห็นในเวลาที่ 4 ซึ่งค่าในคอลัมน์ L_i มีค่าเท่ากับ 0.026 ไม่ตรงกับเงื่อนไขที่ 1และ 2 แต่ตรงกับเงื่อนไขที่ 3 คือ L_i >0 ฉะนั้นสรุปได้ว่า window นาทีที่ 4 เป็น DDoS

Files(minute)	Unique	Total Packets	γ_i	β_i	α_i	L_i	DDoS
1st	32	1566	48	0	0	0	No
2nd	65	11431	175	1.293	1.293	0.030	Yes
3rd	29	583	20	-2.169	-0.875	-0.034	No
4th	47	5425	115	1.749	0.873	0.026	Yes
5th	56	877	15	-2.036	-1.163	-0.010	No

ตารางที่ 4.10 การตรวจจับพบ DDoS ด้วยอัลกอริทึมตามเงื่อนไข L_i >0

จากตาราง 4.10 ตาราง ในการหาค่า L_i มีต้องใช้ค่าตัวแปร 3 ตัวที่ใช้ในการคำนวณ คือ

- X_i ในนาทีที่ 4 ของ window เท่ากับ 5425
- X_1 ในนาทีที่ 4 ของ window เท่ากับ 583
- t_i ในนาทีที่ 4 ของ window เท่ากับ 4

มีวิธีคำนวณดังนี้ $L_i = \frac{1}{t_i} ln \frac{\Delta(X_i)}{\Delta(X_1)}$

$$L_i = \frac{1}{t_i} ln \frac{\Delta(5425)}{\Delta(583)}$$

$$L_i = 0.026$$

จากสมการ L_i มีค่าเท่ากับ 0.026 ไม่ตรงกับเงื่อนไขที่ 1และ 2 แต่ตรงกับเงื่อนไขที่ 3 คือ $L_i > 0$ ฉะนั้นสรุปได้ว่า window นาทีที่ 4 เป็น DDoS

4.2.4 วิเคราะห์และประเมินผลการทดสอบ อัลกอริทึม

ในการทดสอบระบบต้นแบบที่ได้พัฒนาขึ้น การหาค่าความถูกต้องแม่นยำ (accuracy) เป็นการสร้างความมั่นใจได้ว่า เมื่อนำระบบที่พัฒนาขึ้นไปใช้ในระบบจริงแล้วสามารถที่จะตรวจจับ DDoS ได้จริง และสามารถที่จะช่วยให้ผู้ดูแลระบบทราบก่อนที่ระบบจะเกิดความเสียหายได้ โดยการหาค่าความถูกต้องแม่นยำ หาได้จากการนำจำนวนของโฟล์วที่ทำนายด้วยระบบที่พัฒนาขึ้นพบว่าเป็น DDos หารกับ จำนวน Flow ที่มีป้ายกำกับว่าเป็น DDoS ทั้งหมดใน Dataset โดยใช้สมการ ดังนี้

ค่าความถูกต้อง (Accuracy) = $\frac{predict}{actual}$ x 100

Actual = จำนวน Flow ที่เป็น DDoS ทั้งหมดใน Dataset

Predict = โฟล์วที่เป็น DDos ซึ่งได้มาจากการประมวลผลด้วยอัลกอริทึมตรวจจับ DDoS

เมื่อประมวลผลด้วยอัลกอริทึมการตรวจจับ DDoS เสร็จเพื่อต้องการทราบความถูกต้องของระบบต้นแบบที่พัฒนาขึ้นด้วยการหาค่า Accuracy โดยคำนวณจากโฟล์วที่ตรวจพบด้วยอัลกอริทึมตรวจจับว่าเป็น DDoS เปรียบเทียบกับ โฟล์วที่เป็น DDoS ใน Dataset ที่นำมาทดสอบ โดยแสดงให้เห็นในภาพที่ 4.6

```
Running Time: 0:00:00.164993
Predict:30842, Actual:37375, Accuracy:0.8252040133779264
```

ภาพที่ 4.6 แสดงการหาค่าความถูกต้องแม่นยำด้วยค่า Accuracy

จากภาพที่ 4.6 แสดงให้เห็นว่าเมื่อเอาผลลัพธ์ มาทำการประมวลผล จะได้ผลลัพธ์ดังต่อไปนี้

Actual = จำนวน โฟล์วทั้งหมด ที่ติดป้ายกำกับ (label) DDoS เท่ากับ 37375 โฟล์ว

Predict = จำนวนโฟล์วที่ตรวจพบด้วยอัลกอริทึมตรวจจับ DDoS เท่ากับ 30842 โฟล์ว

$$\text{ค่าความถูกต้อง} = \frac{30842}{37375} \times 100$$

ฉะนั้นค่าความถูกต้อง = 82.5 %

4.3 ผลการดำเนินงานในการทดสอบระบบที่พัฒนาขึ้น

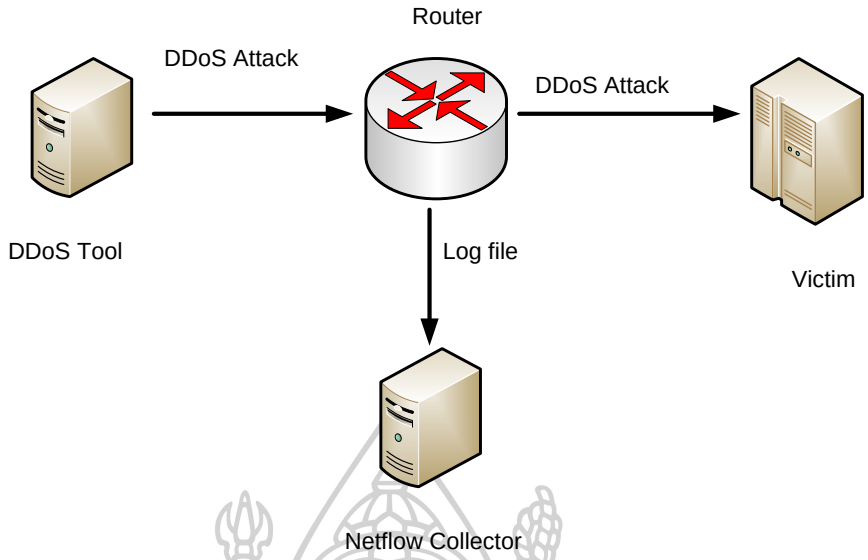
ขั้นตอนการทดสอบระบบต้นแบบที่พัฒนาขึ้นสามารถแบ่งเป็นขั้นตอนได้ดังต่อไปนี้

4.3.1 จัดเก็บข้อมูล **log file** จาก ระบบเครือข่าย

1. ติดตั้งคอมพิวเตอร์เซิร์ฟเวอร์ จำนวน 3 เครื่อง บนระบบเครือข่าย UniNet เพื่อทำการเก็บ log file ที่ถูกโจมตีด้วย DDoS และ log file ที่เป็นการใช้งานปกติ ดังภาพที่ 4.7

ภาพที่ 4.7 อุปกรณ์ที่ใช้ในการทดสอบระบบ

จากภาพที่ 4.7 แสดงเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ 3 เครื่องที่ทำการเชื่อมกันด้วย Router ที่มี Netflow ซึ่งคอมพิวเตอร์ในแต่ละตัวจะทำหน้าต่างกันออกดังที่เคยแสดงให้เห็นในภาพที่ 3.11 ดังนี้



ภาพที่ 4.8 แสดงผังการจัดอุปกรณ์ที่ใช้ในการทดลอง

จากภาพที่ 4.8 สามารถอธิบายรายละเอียดได้ดังต่อไปนี้

- คอมพิวเตอร์เซิร์ฟเวอร์เครื่องที่หนึ่ง คือเครื่อง Collector ซึ่งทำหน้าในการจัดเก็บข้อมูล log file จาก Router โดยจะทำการตั้งค่าให้ Netfolw ส่ง log file จาก Interface ของ router ที่ต้องการให้ส่งมาที่เครื่อง IP Address หมายเลข 202.28.194.243 ดังภาพที่ 4.9

```
flow exporter exportnetflow
 destination 202.28.194.243
 source Loopback0
 transport udp 9995
!
```

ภาพที่ 4.9 ตั้งค่า Netflow ให้ส่ง log file มาที่เครื่อง Collector

ในการรับ log file จาก Router เพื่อทำการบันทึกในเครื่อง Collector นั้นใช้โปรแกรมการจัดเก็บ log file คือ NfSen ในการรับ log file ซึ่งจะทำการบันทึก log file เป็นไฟล์นามสกุล nfcapd. แล้วตามด้วยวันเวลาที่ทำการบันทึก ดังภาพที่ 4.10

```
-rw-r--r-- 1 root root 204344 Jun 21 01:56 nfcapd.202006210151
-rw-r--r-- 1 root root 209912 Jun 21 02:01 nfcapd.202006210156
-rw-r--r-- 1 root root 206968 Jun 21 02:06 nfcapd.202006210201
-rw-r--r-- 1 root root 223032 Jun 21 02:11 nfcapd.202006210206
-rw-r--r-- 1 root root 214328 Jun 21 02:16 nfcapd.202006210211
-rw-r--r-- 1 root root 196536 Jun 21 02:21 nfcapd.202006210216
-rw-r--r-- 1 root root 218552 Jun 21 02:26 nfcapd.202006210221
-rw-r--r-- 1 root root 188792 Jun 21 02:31 nfcapd.202006210226
-rw-r--r-- 1 root root 217848 Jun 21 02:36 nfcapd.202006210231
-rw-r--r-- 1 root root 224632 Jun 21 02:41 nfcapd.202006210236
-rw-r--r-- 1 root root 210360 Jun 21 02:46 nfcapd.202006210241
-rw-r--r-- 1 root root 197112 Jun 21 02:51 nfcapd.202006210246
-rw-r--r-- 1 root root 221688 Jun 21 02:56 nfcapd.202006210251
-rw-r--r-- 1 root root 207800 Jun 21 03:01 nfcapd.202006210256
```

ภาพที่ 4.10 ไฟล์ nfcap

- คอมพิวเตอร์เซิร์ฟเวอร์เครื่องที่สอง คือเครื่องที่มีการติดตั้งโปรแกรมจำลองการโจมตี Slowloris DDoS Tool [24] พัฒนาโดย Robert Hanse เป็นการโจมตีในชั้น application layer โดยการใช้ HTTP Request การโจมตีจะทำได้โดยการเปิดการเชื่อมต่อไปยังเว็บเซิร์ฟเวอร์ เป้าหมาย และทำการเก็บการเชื่อมต่อนั้นให้เปิดไว้นานที่สุดเท่าที่จะทำได้

การโจมตีประเภทนี้ จะใช้ปริมาณ bandwidth ต่ำ และ ใช้การเพิ่มทรัพยากรของ เซิร์ฟเวอร์ แทน ซึ่งจะทำให้ traffic อยู่ในรูปแบบปกติ การโจมตีประเภทนี้มักเรียกกันว่า “ต่ำ และ ช้า” เซิร์ฟเวอร์เป้าหมายจะมีเส้นทางการขอทำการเชื่อมต่อจำนวนมาก และจะทำการเปิดเส้นทางนั้นไว้ในขณะที่รอให้ request นั้นเสร็จสิ้นแต่ request จะเกิดขึ้นอย่างต่อเนื่องไม่สิ้นสุด เมื่อการเชื่อมต่อถึงจุดสูงสุดของ เซิร์ฟเวอร์ แล้ว การเชื่อมต่อที่เพิ่มเข้ามาแต่ละครั้ง จะไม่ถูกตอบกลับ และ DoS จะเกิดขึ้นมีขั้นตอนการทำงานดังนี้

1. ผู้โจมตีเริ่มแรกจะเปิดการเชื่อมต่อจำนวนมากไปยัง เซิร์ฟเวอร์ เป้าหมาย โดยการส่ง header ของ HTTP request บางส่วน

2. เป้าหมายจะทำการเปิดการเชื่อมต่อสำหรับ request ที่กำลังเข้ามา ถ้าการเชื่อมต่อใช้เวลานานเกินไป เซิร์ฟเวอร์ จะ timeout การเชื่อมต่อที่ใช้เวลานาน จะทำให้เกิดการขอเส้นทางเชื่อมต่อจำนวนมากขึ้นใน request ถัดไป

3. ผู้โจมตีมีการป้องกันเป้าหมายจาก time out ของการเชื่อมต่อ ผู้โจมตีจะส่ง header ของ request บางส่วนเป็นระยะๆ เพื่อให้ request ยังอยู่

4. หากมีการร้องขอการเชื่อมต่อจำนวนมากถูกใช้ เซิร์ฟเวอร์จะไม่สามารถตอบรับ request เพิ่มเติมได้จาก traffic ปกติ จึงทำให้เกิด DoS

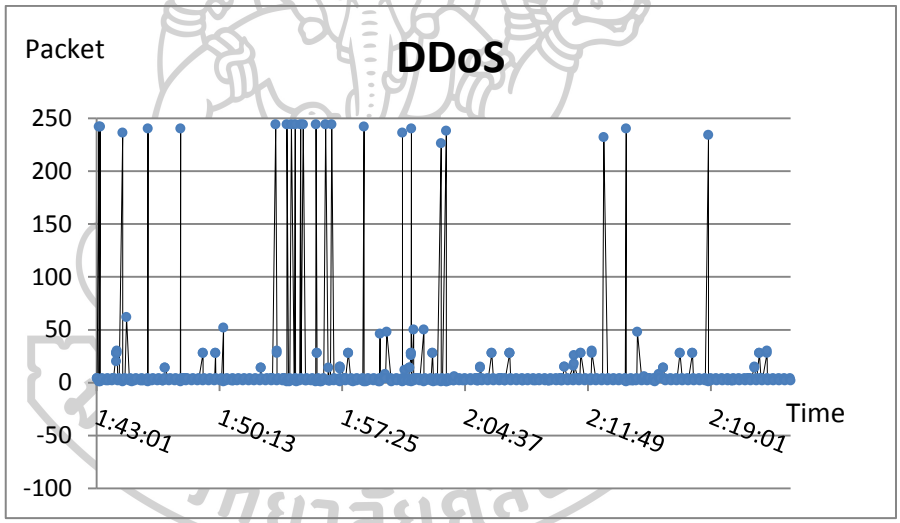
```
test4@4:~/slowloris$ python3 slowloris.py --sleep 10 http://202.28.194.245
```

ภาพที่ 4.11 คำสั่งในการใช้ Slowloris DDoS Tool

- คอมพิวเตอร์เซิร์ฟเวอร์เครื่องที่ 3 คือเครื่องที่เป็นเว็บเซิร์ฟเวอร์ Web server ทำหน้าที่ในการรับการโจมตีจาก DDoS (Victim) โดยทำการตั้งค่าให้ Netflow ทำการจับ log file ใน interface ของ Router เพื่อทำการคัดลอก log file ไปที่เครื่อง Netflow collector

2. จัดเก็บ log file เพื่อใช้ในการทดสอบระบบ

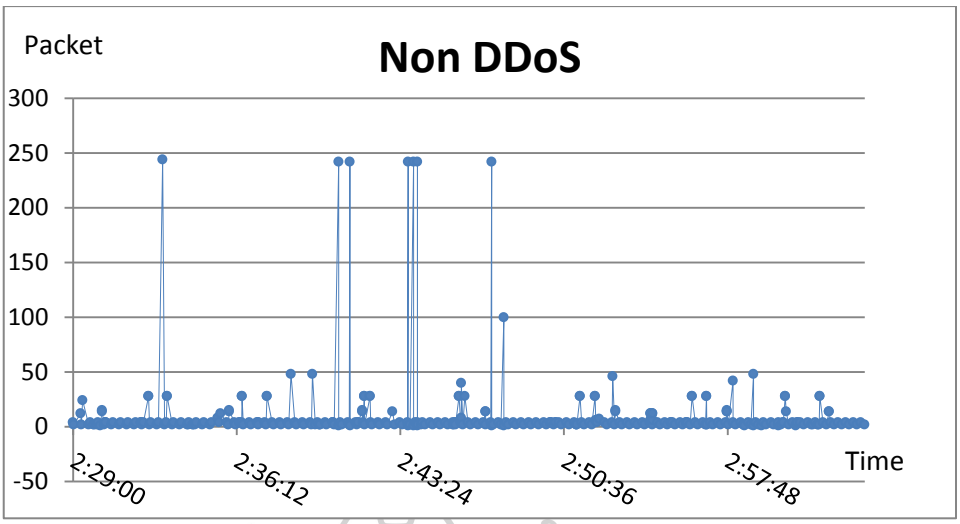
- สร้างการโจมตี DDoS ด้วย Slowloris DDoS Tool ไปยังเครื่องที่เป็นเว็บเซิร์ฟเวอร์เป็นเวลา 30 นาที เพื่อใช้เป็นข้อมูลในการประมวลด้วยอัลกอริทึมการตรวจจับ DDoS โดย log file ที่เป็นการโจมตีด้วย DDoS จะนำมาสร้างกราฟดังภาพที่ 4.12



ภาพที่ 4.12 กราฟแสดงปริมาณแพ็คเกตที่มีการโจมตี DDoS

จากภาพที่ 4.12 แสดงให้เห็นว่าเมื่ออัตราแพ็คเกตสูงเกิน 200 แพ็คเกต คือช่วงเวลาที่ Slowloris DDoS Tool ทำการสร้างแพ็คเกตจำนวนมากเพื่อใช้ในการโจมตี และส่งเป็นเวลาติดต่อกัน

- จัดเก็บ log file ที่ใช้งานปกติเป็นระยะเวลา 30 นาที โดยเริ่มจัดเก็บหลังจากที่ได้ทำการโจมตีด้วย DDoS เสร็จแล้ว โดยจะนำ log file มาสร้างกราฟดังภาพที่ 4.13

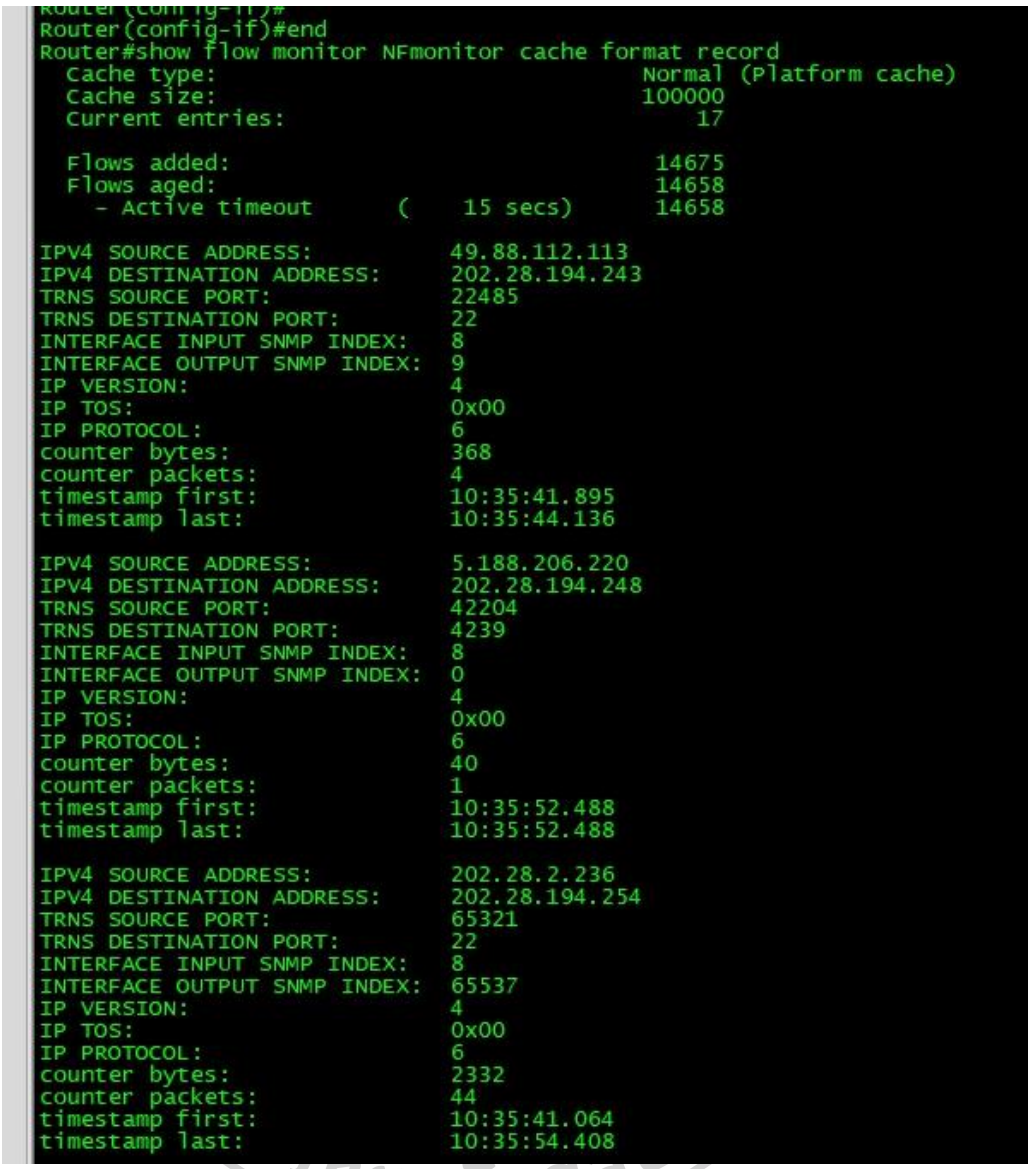


ภาพที่ 4.13 กราฟแสดงปริมาณแพ็คเกตที่มีการใช้งานปกติ

จากภาพที่ 4.13 แสดงให้เห็นว่าเมื่ออัตราแพ็คเกตที่มีการใช้งานปกติจะมีการส่งแพ็คเกตในอัตราที่ไม่สูงมาก ดังภาพที่แสดงให้เห็นบางช่วงเวลามีอัตราการส่งแพ็คเกตมากกว่า 200 แพ็คเกต สันนิษฐานได้ว่าอาจเป็นส่งรับส่งไฟล์ข้อมูลขนาดใหญ่ระหว่างผู้ใช้ เป็นต้น

3. สร้าง Netflow Collector เพื่อใช้ในการเก็บ Log file

- เมื่อมีการส่งแพ็คเกตผ่านอินเตอร์เฟสไปยังRouterจะมีการตั้งค่าไว้ จะทำการส่ง log file มาที่เครื่องที่ Netflow collector โดย แพ็คเกตในการรับส่งข้อมูลจะแสดงให้เห็นดังภาพที่ 4.14



```
Router(config-if)#end
Router#show flow monitor NFmonitor cache format record
  Cache type:                               Normal (Platform cache)
  Cache size:                               100000
  Current entries:                              17

  Flows added:                               14675
  Flows aged:                                14658
    - Active timeout      (    15 secs)      14658

IPV4 SOURCE ADDRESS:          49.88.112.113
IPV4 DESTINATION ADDRESS:     202.28.194.243
TRNS SOURCE PORT:             22485
TRNS DESTINATION PORT:        22
INTERFACE INPUT SNMP INDEX:   8
INTERFACE OUTPUT SNMP INDEX:  9
IP VERSION:                   4
IP TOS:                       0x00
IP PROTOCOL:                  6
counter bytes:                368
counter packets:              4
timestamp first:              10:35:41.895
timestamp last:               10:35:44.136

IPV4 SOURCE ADDRESS:          5.188.206.220
IPV4 DESTINATION ADDRESS:     202.28.194.248
TRNS SOURCE PORT:             42204
TRNS DESTINATION PORT:        4239
INTERFACE INPUT SNMP INDEX:   8
INTERFACE OUTPUT SNMP INDEX:  0
IP VERSION:                   4
IP TOS:                       0x00
IP PROTOCOL:                  6
counter bytes:                40
counter packets:              1
timestamp first:              10:35:52.488
timestamp last:               10:35:52.488

IPV4 SOURCE ADDRESS:          202.28.2.236
IPV4 DESTINATION ADDRESS:     202.28.194.254
TRNS SOURCE PORT:             65321
TRNS DESTINATION PORT:        22
INTERFACE INPUT SNMP INDEX:   8
INTERFACE OUTPUT SNMP INDEX:  65537
IP VERSION:                   4
IP TOS:                       0x00
IP PROTOCOL:                  6
counter bytes:                2332
counter packets:              44
timestamp first:              10:35:41.064
timestamp last:               10:35:54.408
```

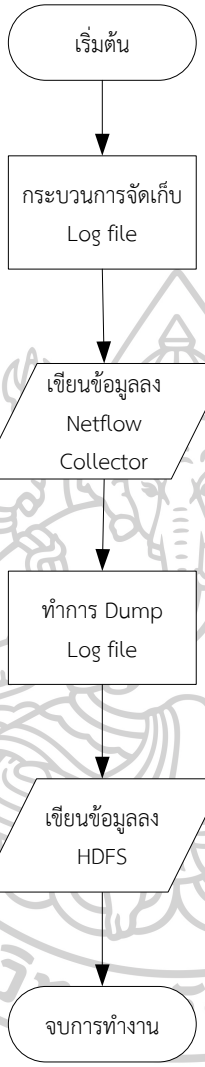
ภาพที่ 4.14 log file ที่อยู่ใน Router

ภาพที่ 4.14 จะแสดงถึงรายละเอียด Log file ที่อยู่ในอุปกรณ์ Router ซึ่งมี IP Address ที่เกี่ยวข้องกับงานวิจัยดังนี้

- 202.28.194.243 : เครื่องที่เป็นเว็บเซิร์ฟเวอร์ Web server ทำหน้าที่ในการรับการโจมตีจาก DDoS (Victim)
- 202.28.194.245 : เครื่อง Collector ซึ่งทำหน้าในการจัดเก็บข้อมูล log file จาก Router
- 202.28.194.248 : เครื่องที่มีการติดตั้งโปรแกรมจำลองการโจมตี Slowloris DDoS Tool

4. นำ log file จาก Netflow collector เข้าสู่ HDFS

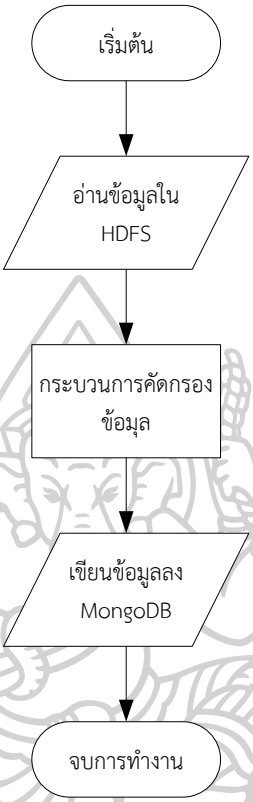
- นำ log file ที่ได้จาก Netflow collector จัดเก็บใน HDFS เพื่อให้สามารถที่จะนำข้อมูลมาใช้เพื่อการประมวลผล ซึ่งจะอธิบายขั้นตอนการทำงานด้วยผังงานตามภาพที่ 4.15



ภาพที่ 4.15 แสดงผังงานการนำ Log file ไปจัดเก็บใน HDFS

จากภาพที่ 4.15 สามารถอธิบายการทำงานได้คือ ตั้งค่า Netflow ให้ทำการจับข้อมูลการจราจร (log file) จาก Interface ของ Router จากนั้น Netflow จะทำการคัดลอก log file นั้นไปยังเครื่อง Netflow collector และส่งต่อไปทำการเก็บไว้ที่ HDFS

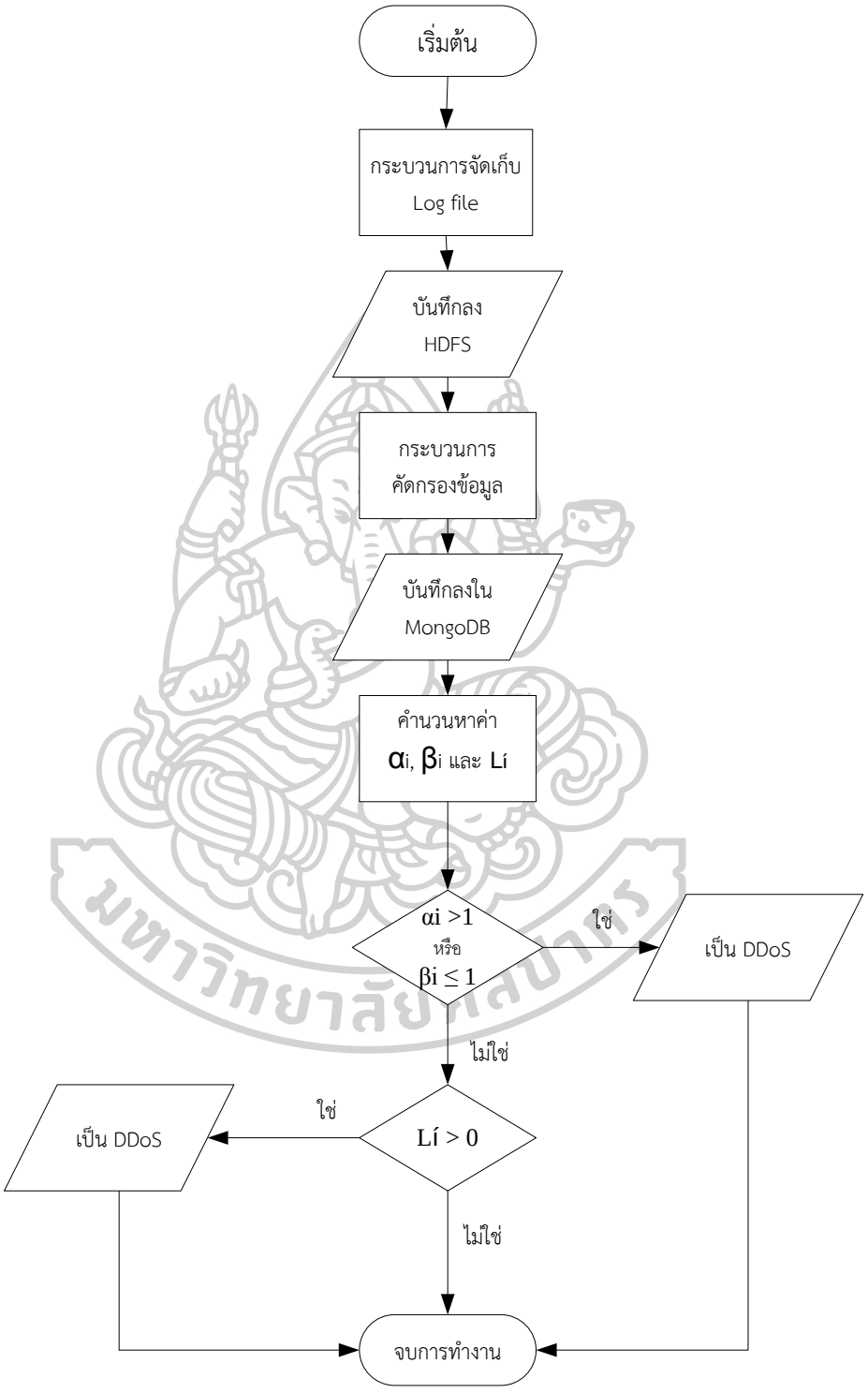
5. การคัดกรองข้อมูล ด้วย Map Reduce Function จะทำงานโดยการดึงบล็อกข้อมูลมาทำการประมวลผลโดยมีขั้นตอนตามผังงานในภาพที่ 4.16



ภาพที่ 4.16แสดงการทำงานของการคัดกรองข้อมูล

จากภาพที่ 4.16 อธิบายการใช้ MapReduce Function เพื่อการลดปริมาณข้อมูล โดยทำการดึงบล็อกข้อมูล จาก HDFS เพื่อมาทำการ MapReduce เพื่อให้ได้ log file ที่คัดกรองเหลือเพียงข้อมูลที่จำเป็นต่อการประมวลผลและนำไปบันทึกไว้ในฐานข้อมูล MongoDB

6. ประมวลผลเพื่อตรวจจับ DDoS ด้วยข้อมูลใน MongoDB



ภาพที่ 4.17 แสดงการทำงานของระบบ

จากภาพที่ 4.17 แสดงขั้นตอนการทำงานของระบบดังนี้

1. จัดเก็บ log file เป็นเวลา 1 ชั่วโมง โดยแบ่งเป็นจัดเก็บ log file ที่ถูกโจมตีด้วย DDoS เป็นเวลา 30 นาที และ log file ที่เป็นการใช้งานปกติ 30 นาที เช่นกัน

2. นำ log file ที่อยู่ใน Netflow Collector เข้ามาจัดเก็บที่ HDFS

3. เรียกใช้ block ข้อมูลใน HDFS มาทำการ MapReduce เพื่อลดปริมาณของข้อมูลให้เหลือแต่ข้อมูลที่จำเป็นต่อการประมวลผล

4. เมื่อคัดกรองข้อมูลด้วย MapReduce แล้ว จะจัดเก็บข้อมูลนั้นไว้ที่ฐานข้อมูล MongoDB

5. ขั้นตอนการประมวลผลจะใช้ข้อมูลที่ผ่านการคัดกรองข้อมูลมาแล้วใน MongoDB มาประมวลผลด้วยอัลกอริทึม การตรวจจับ DDoS โดยจะทำการประมวลตามเงื่อนไขที่ได้กำหนดไว้ในอัลกอริทึม

6. ผลลัพธ์จากการทดสอบระบบ

- ทดสอบและประมวลผลจาก log file ที่เป็น DDoS เป็นเวลา 30 นาทีแสดงดังตารางที่ 4.11

Time	Normal	DDoS	TP	FP	TN	FN
1:51	✓					✓
1:52		✓	✓			
1:53		✓	✓			
1:54		✓	✓			
1:55		✓	✓			
1:56		✓	✓			
1:57		✓	✓			
1:58	✓					✓
1:59		✓	✓			
2:00	✓					✓
2:01		✓	✓			
2:02		✓	✓			
2:03	✓					✓
2:04		✓	✓			
2:05		✓	✓			
2:06		✓	✓			
2:07		✓	✓			
2:08	✓					✓
2:09		✓	✓			
2:10		✓	✓			
2:11		✓	✓			
2:12		✓	✓			
2:13	✓					✓
2:14		✓	✓			
2:15		✓	✓			
2:16		✓	✓			
2:17		✓	✓			
2:18	✓					✓
2:19		✓	✓			
2:20	✓					✓

ตารางที่ 4.11 ผลลัพธ์การประมวลผล DDoS

จากตารางที่ 4.11 แสดงผลลัพธ์ที่ได้จากการประมวลผลข้อมูล log file ที่เป็นการโจมตีด้วย DDoS บนระบบเครือข่าย UniNet เป็นเวลา 30 นาที โดยจะประมวลผลเป็นช่วงเวลา (Timeframe) ช่วงเวลาละ 10 นาที่เพื่อให้มีปริมาณข้อมูลที่เพียงพอต่อการประมวลผลด้วยอัลกอริทึมตรวจจับ DDoS โดยแบ่งออกในแต่ช่วงเวลาให้เป็นวินโดว์ย่อยวินโดว์ละ 1 นาที ในการโจมตีด้วย DDoS ระบบที่พัฒนาขึ้นสามารถตรวจพบ 22 วินโดว์ ซึ่งหมายความว่า log file ที่เป็น DDoS ระบบสามารถตรวจพบว่าเป็น DDoS ได้ 22 วินโดว์ จากเวลาทั้งหมด 30 วินโดว์ และมีอีก 8 วินโดว์ ที่ตรวจไม่พบว่าเป็น DDoS

- ผลลัพธ์จากโฟล์วการทำงานปกติโดยทำการบันทึก log file เป็นเวลา 30 นาทีหลังการหยุดการโจมตีแบบ DDoS

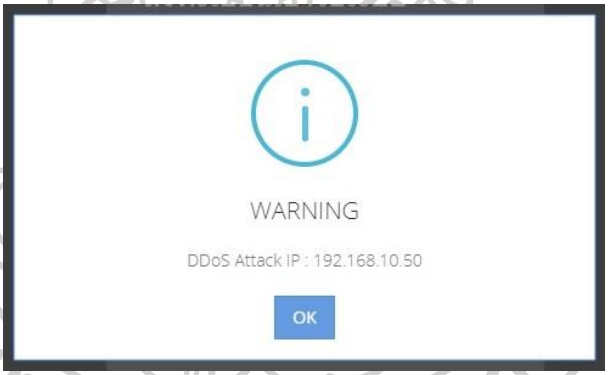
Time	Normal	DDoS	TP	FP	TN	FN
2:21	✓				✓	
2:22	✓				✓	
2:23	✓				✓	
2:24	✓				✓	
2:25		✓		✓		
2:26	✓				✓	
2:27		✓		✓		
2:28		✓		✓		
2:29	✓				✓	
2:30		✓		✓		
2:31	✓				✓	
2:32	✓				✓	
2:33	✓				✓	
2:34	✓				✓	
2:35	✓				✓	
2:36	✓				✓	
2:37		✓		✓		
2:38	✓				✓	
2:39	✓				✓	
2:40	✓				✓	
2:41		✓		✓		
2:42	✓				✓	
2:43	✓				✓	
2:44		✓		✓		
2:45	✓				✓	
2:46		✓		✓		
2:47	✓				✓	
2:48	✓				✓	
2:49		✓		✓		
2:50	✓				✓	

ตารางที่ 4.12 ผลลัพธ์การประมวลผลที่เป็นปกติ

จากตารางที่ 4.12 แสดงผลลัพธ์ที่ได้จาการประมวลผลข้อมูลที่เป็น log file การใช้งานปกติ บนระบบเครือข่าย UniNet เป็นเวลา 30 นาทีโดยผ่านการประมวลผลเป็นช่วงเวลา ช่วงเวลาละ 10 นาที และแบ่งออกเป็น วินโดว์ ย่อย วินโดว์ ละ 1 นาที ซึ่งผลที่ได้คือ ในการใช้งานปกติที่มีทั้งหมด 30 วินโดว์ ระบบตรวจและแจ้งว่าเป็น DDoS อยู่ 9 วินโดว์ อีก 21 วินโดว์ ไม่มีการแจ้งเตือน

7. การส่งผลลัพธ์ไปสู่ศูนย์กลาง

เมื่อทำการประมวลข้อมูลที่ได้ทำการคัดกรองเรียบร้อยแล้ว เมื่อประมวลผล log file ด้วยอัลกอริทึมการตรวจจับ DDoS และเมื่อมีตรวจพบ การโจมตีแบบ DDoS ในทุกครั้ง ระบบจะมีข้อความแจ้งเตือนที่ประกอบไปด้วยหมายเลข IP Address ที่ถูกโจมตี และ วันเวลา ที่ถูกโจมตี ไปยังศูนย์กลางซึ่งมีผู้ดูแลระบบอยู่ เพื่อให้ทราบและหาวิธีป้องกันและแก้ไขก่อนที่ระบบจะเกิดความเสียหาย โดยมีหลักการทำงานดังนี้คือ เมื่อทำการประมวลผลด้วยอัลกอริทึมการตรวจจับ DDoS ใน Backbone Node/Distribution Node และเมื่อการตรวจพบการโจมตี DDoS ระบบจะทำการส่งข้อความแจ้งเตือนไปยังศูนย์กลางในทันที เพื่อให้ผู้ดูแลระบบทราบในทันทีว่าระบบถูกโจมตีด้วย DDoS ก่อนที่จะทำให้ระบบเกิดความเสียหาย เมื่อมีการตรวจพบการถูกโจมตีด้วย DDoS ระบบจะทำการส่งข้อความแจ้งเตือนไปยังศูนย์กลาง เพื่อให้ผู้ดูแลระบบสามารถทราบว่าระบบถูกโจมตี โดยจะมีข้อความแจ้งเตือนที่จะแสดงให้ ผู้ดูแลระบบได้เห็น มีลักษณะดังภาพที่ 4.18



ภาพที่ 4.18 แสดงกล่องข้อความแจ้งเตือนไปยังผู้ดูแลระบบเมื่อพบ DDoS

4.3.2 การประเมินผล

ในการประเมินผลในการทดลองการพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่จะทำการหาค่าความล่าช้า (Delay) และค่าความถูกต้องแม่นยำ (Accuracy)

1. Delay

เป็นค่าที่สามารถวัดได้หลังจากการเกิด DDoS กับเครื่องคอมพิวเตอร์เป้าหมาย ซึ่งคือเหยื่อของการโจมตีบนระบบเครือข่าย ระบบที่ถูกพัฒนาขึ้น สามารถทำการตรวจจับ DDoS ได้ ซึ่งพบได้ว่าการตรวจจับนั้นสามารถตรวจจับได้ช้ากว่าเวลาที่มีการโจมตีจริงประมาณ 7 นาที โดยนับจากเวลาการ

โจมตีด้วย DDoS ที่เวลา 01.50 น. แต่ต้องใช้เวลาในการรอเพื่อที่จะให้ Netflow ทำการส่ง log file ไปที่ Netflow collector เพื่อบรรจุเป็นไฟล์นามสกุล nfcapd ซึ่งจะเสร็จ ณ เวลา 01.56 น. ถึงจะสามารถนำ log file ไปเข้าสู่กระบวนการการตรวจจับ DDoS ได้

2. Accuracy

เป็นการตรวจสอบจาก log file ที่เป็น DDoS เป็นเวลา 30 นาที และการใช้งานแบบปกติ 30 นาที โดยมีจำนวนโฟล์ว ทั้งหมด 19,972 โฟล์ว และผลลัพธ์ที่ได้จากการทำนายในระบบมีดังนี้

True Positive (TP) = 22

True Negative (TN) = 21

False Positive (FP) = 9

False Negative (FN) = 8

- Accuracy คือ ค่าที่บอกถึงความถูกต้องแม่นยำของระบบที่พัฒนาขึ้น

  หาได้จาก (TP+TN)/(TP+TN+FP+FN)

  (22+21)/(22+21+9+8) = ร้อยละ 71

- True Positive Rate (TPR) คือ ค่าที่บอกว่าโปรแกรมทำนายได้ว่าจริง เป็นอัตราส่วนเท่าใดของจริงทั้งหมด

  หาได้จาก TP(TP+FN)

  22/(22+8) = ร้อยละ 73

- True Negative Rate (TNR) คือ ค่าที่บอกว่าระบบทำนายได้ว่าไม่จริง เป็นอัตราส่วนเท่าไรของจริงทั้งหมด

  หาได้จาก TN/(TN+FP)

  21/(21+9) = ร้อยละ 70

- False Positive Rate (TPR) คือ ค่าที่บอกว่าระบบทำนายว่าจริง เป็นอัตราส่วนเท่าไรของไม่จริงทั้งหมด

  หาได้จาก FP/(TN+FP)

  9/(21+9) = ร้อยละ 30

- False Negative Rate (FNR) คือ ค่าที่บอกว่าโปรแกรมทำนายว่าไม่จริง เป็นอัตราส่วนเท่าไรของจริงทั้งหมด

  หาได้จาก FN/(TP+FN)

  8/(22+8) = ร้อยละ 26

โฟล์ว	TP	TN	FP	FN	Accuracy
DDoS	73%			26%	
Normal		70%	30%		
Total					71%

ตารางที่ 4.13 แสดงผลค่าความถูกต้องในการทดลอง

4.4 สรุปผล

สรุปผลการทดลองในครั้งนี้ เป็นการทดลองการพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ ซึ่งนำเอาแนวคิดอัลกอริทึมของ Vishal Maheshwari ที่ใช้หลักการ การวิเคราะห์อนุกรมเวลาในการประมวลผลเพื่อตรวจจับการโจมตีแบบ DDoS ซึ่งทำการทดสอบกับ CTU-13 Dataset เพื่อทดสอบประสิทธิภาพเพื่อนำมาใช้จริงบนระบบเครือข่ายที่มีความซับซ้อนสูง (UniNet) ซึ่งจากการทดสอบได้ค่าความถูกต้องที่ร้อยละ 82.5 และเมื่อนำอัลกอริทึมดังกล่าวมาใช้จริงกับระบบ UniNet ทำให้เกิดค่าความถูกต้องลดลงเหลือร้อยละ 71 และมีค่าความล่าช้าที่ 7 นาที เนื่องจากกระบวนการในการจัดเก็บ log file มีหลายขั้นตอนมากกว่าการนำข้อมูลจาก Dataset มาทำการประมวลผล

บทที่ 5
สรุปผลการดำเนินวิจัย

จุดประสงค์ในการพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่เพื่อแก้ไขปัญหา โดยสามารถระบุออกมาได้ดังนี้

5.1 แก้ปัญหาที่พบในการวิเคราะห์และศึกษาระบบ

5.1.1 แก้ปัญหาข้อมูลที่มีจำนวนมากเกินไปที่ระบบจะทำการจัดเก็บได้

ปัญหาที่มีข้อมูลขนาดใหญ่แก้ไขด้วยการนำ Big data technique ในงานวิจัยนี้ใช้ HDFS ในการจัดเก็บข้อมูล และการประมวลผลข้อมูลขนาดใหญ่ด้วย MapReduce ซึ่งผลที่ได้จากการพัฒนาต้นแบบเพื่อแก้ปัญหาดังกล่าวนี้สรุปได้ว่า สามารถที่จะนำข้อมูลที่มีขนาดใหญ่ มาจัดเก็บและทำการลดปริมาณข้อมูลลงให้เหลือเพียงข้อมูลที่จำเป็นต่อการประมวลผล มีประโยชน์กับระบบเครือข่ายในด้านการใช้ข้อมูลที่มีอยู่ให้เกิดประโยชน์สูงสุด ต้นแบบที่ได้พัฒนาขึ้นนี้เหมาะสำหรับระบบเครือข่ายที่มีความซับซ้อนเนื่องจากสามารถที่จะรองรับข้อมูลที่มีขนาดใหญ่ตามการเพิ่มหรือลดของจำนวนลิงค์และโหนดในระบบเครือข่าย อีกทั้งระบบการเชื่อมต่อของระบบเครือข่ายที่มีความซับซ้อนสูงที่มีโครงสร้างการเชื่อมต่อที่หลากหลาย คือ มีการเชื่อมต่อทั้งเป็น Small world และ Scale free ที่มีรูปแบบความสัมพันธ์ของโหนดและลิงค์ที่ต่างกัน ทำให้ปริมาณข้อมูลที่เกิดขึ้นคำนวณได้ยากลำบาก การนำ Big Data Technique มาใช้กับต้นแบบที่ได้พัฒนาขึ้นนี้ เพื่อให้สามารถที่จะขยายพื้นที่การจัดเก็บข้อมูลตามขนาดของข้อมูลที่เกิดขึ้น โดยไม่ส่งผลกระทบกับระบบเครือข่ายหรืองานที่กำลังดำเนินการอยู่

5.1.2 แก้ปัญหาการจัดเก็บข้อมูลที่อยู่แบบกระจาย

แก้ไขด้วยการออกแบบการประมวลเป็นแบบกระจายโดยจะมีหลักการทำงานคือ ในแต่ละ Backbone Node/Distribution Node จะทำการประมวลผลที่โหนดนั้นๆ เมื่อดำเนินการเสร็จ จะส่งผลลัพธ์ไปเก็บบันทึกไว้ที่ Data lake บนศูนย์กลาง การแก้ปัญหาจากระบบเดิมที่ไม่มีการจัดเก็บบันทึกข้อมูลการโดนโจมตีจาก DDoS ในระบบต้นแบบที่พัฒนาขึ้นนี้ สามารถที่จะนำผลลัพธ์จากการประมวลผลไปจัดเก็บบันทึกไว้ที่ศูนย์กลางเพื่อให้สามารถที่จะนำข้อมูลไปใช้ในอนาคตได้ ไม่ว่าจะเป็นการนำไปใช้ในการวางแผนพัฒนาระบบเครือข่าย หรือใช้ดูสถิติการโจมตีเพื่อนำไปสร้างระบบการป้องกันให้กับระบบเครือข่าย หากระบบเครือข่ายที่มีความซับซ้อนสูง มีผู้ใช้จำนวนมาก มีลิงค์และโหนดที่เชื่อมต่อกันเป็นจำนวนมาก หากไม่มีการจัดเก็บผลลัพธ์ไว้ที่ศูนย์กลางที่เดียว อาจก่อเกิดปัญหาตามมาได้ เช่น หากเก็บผลลัพธ์ไว้ที่โหนด (Backbone Node/Distribution Node) เมื่อต้องการประมวลผลหรือเรียกใช้ข้อมูลจะต้องทำการร้องขอไปยังทุกๆ โหนด เพื่อให้ส่งข้อมูลมา แต่

เนื่องจากระบบมีขนาดใหญ่มาก เส้นทางในการติดต่อสื่อสารกันจะไม่เท่ากัน ทำให้ในการประมวลผลจะต้องร้องขอข้อมูล เพื่อให้ข้อมูลของทุกโหนดมาครบ อีกทั้งมีความเสี่ยงที่จะไม่ได้ข้อมูลตามจำนวนที่ต้องการเพราะหากเก็บข้อมูลที่โหนดในระบบเครือข่ายหากโหนดใดเสียหายก็จะไม่สามารถใช้ข้อมูลของโหนดนั้นได้ ต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ได้มีการพัฒนาและออกแบบให้มีการจัดเก็บผลลัพธ์ไว้ที่ศูนย์กลางเพื่อลดความไม่ครบถ้วนของการนำข้อมูลมาใช้ ลดระยะเวลาในการรอข้อมูลเพื่อประมวลผล ฉะนั้นต้นแบบที่พัฒนาขึ้นจึงมีความเหมาะสมที่จะนำมาใช้ในระบบเครือข่ายที่มีความซับซ้อนสูง ที่ช่วยในส่วนของการรวบรวมผลลัพธ์และจัดเก็บไว้ที่ศูนย์กลาง

5.1.3 ความล่าช้าในในการประมวลผล

ในการป้องกันการถูกโจมตีด้วย DDoS ทำได้ยาก เนื่องจาก การโจมตีนั้นจะมีลักษณะการคล้ายกับการใช้งานปกติ ดังนั้นงานวิจัยนี้จึงได้นำอัลกอริทึมในการตรวจจับ DDoS มาทำการแยกระหว่างการใช้งานปกติกับการถูกโจมตี ต้นแบบที่ได้พัฒนาขึ้นได้แก้ปัญหาในส่วนของการแยกแยะการใช้งานปกติกับการโดนโจมตีด้วยDDoS ออกจากกันโดยใช้อัลกอริทึมการตรวจจับ DDoS เป็นตัวแยกแยะ ซึ่งอาศัยหลักการของการวิเคราะห์อนุกรมเวลา (Time Series Analysis) ที่ออกแบบให้เหมาะสมกับการใช้งานกับระบบเครือข่ายที่มีความซับซ้อนสูง คือ ด้วยจำนวนปริมาณข้อมูล (log file) ที่มหาศาลของระบบเครือข่าย ในการประมวลผลโดยใช้ช่วงเวลาเพื่อให้เกิดความถูกต้องแม่นยำ เมื่อเราแบ่งการประมวลผลการตรวจจับ DDoS โดยวิเคราะห์เป็นช่วงเวลา (window) โดยจะทำการเปรียบเทียบแบบมีเงื่อนไขในเวลานาทีปัจจุบันและนาทีก่อนหน้า เพื่อให้เห็นถึงความแตกต่างในเรื่องของการส่งข้อมูลจาก IP ต้นทางว่ามีความแตกต่างกัน และตรงกับเงื่อนไขของอัลกอริทึมหรือไม่ ส่งผลให้การประมวลผลในการตรวจจับ DDoS มีความรวดเร็วหากมีการโจมตีจริงสามารถที่จะตรวจพบได้ในเวลาไม่นาน จึงสร้างความมั่นใจได้ว่า หากนำมาใช้จริงบนระบบเครือข่ายที่มีความซับซ้อนสูงสามารถช่วยในการป้องกันหรือแก้ไขเพื่อให้ระบบเครือข่ายอยู่ในความปลอดภัย อีกทั้งต้นแบบที่พัฒนาขึ้นยังมี Feature ในการแจ้งเตือนไปยังศูนย์กลางหากพบ DDoS ในทุกครั้งของทุกโหนดเพื่อให้ผู้ดูแลระบบได้รับทราบ และสามารถจัดการแก้ปัญหานั้นได้ทันเวลา ในการส่งข้อความแจ้งเตือนไปยังศูนย์กลางจะไม่ก่อให้เกิดผลกระทบเรื่องการสร้างความแออัดของการจราจรในระบบได้เลย เพราะในการส่งข้อความแจ้งเตือนนั้นจะส่งเฉพาะ IP Address และเวลาที่เกิดการโจมตี เท่านั้น

5.2 สรุปผลการออกแบบต้นแบบที่จะทำการพัฒนา

ในการออกต้นแบบได้ทำการทดสอบประสิทธิภาพของอัลกอริทึมการตรวจจับ DDoS ของ Vishal Maheshwari โดยนำมาประมวลผลกับชุดข้อมูล CTU-13 Dataset เพื่อหาค่าความถูกต้อง โดยจะนำ CTU-13 Dataset เข้าสู่กระบวนการคัดกรองข้อมูลเพื่อลดพื้นในการจัดเก็บข้อมูล แล้วจึงทำการประมวลผลซึ่งผลลัพธ์ที่ได้จากทดลองมีความความถูกต้องเป็นร้อยละ 82.5

5.3 สรุปผลการทดลองต้นแบบที่พัฒนาขึ้น

การพัฒนาต้นแบบระบบการวิเคราะห์ระบบเครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ จะมีการประเมินผลการทดลองในการวิจัยแบ่งออกเป็น 2 ส่วนหลักดังนี้

1.ประเมินด้วยค่าความถูกต้องแม่นยำ (Accuracy)

2.ประเมินด้วยค่าความล่าช้า (Delay) ในการจัดเก็บ log file เพื่อที่จะทำการประมวลผล

- ประเมินด้วยค่าความถูกต้องแม่นยำ

การวัดค่าความถูกต้องแม่นยำของต้นแบบที่พัฒนาขึ้น โดยคำนวณหาค่าความถูกต้องด้วยการนำ NetFlow log file มาหาผลลัพธ์ด้วยอัลกอริทึมการตรวจจับ DDoS จะได้ค่าความถูกต้อง ในการพัฒนาต้นแบบนี้เมื่อนำมาวัดค่าความถูกต้องได้ผลออกมาเป็นร้อยละ 71 ซึ่งถือได้ว่ามีความถูกต้องที่ไม่สูงมากนักเมื่อนำมาใช้กับระบบเครือข่ายที่มีความซับซ้อนสูง เนื่องจากในการจัดเก็บ log file เพื่อใช้ในการประมวลผลต้องแบ่งเป็นช่วงเวลาตาม Timeframe ของอัลกอริทึม ต่างจากการทดสอบอัลกอริทึมด้วย CTU-13 Dataset ที่เป็นการนำข้อมูลเข้าไปในฐานข้อมูล แล้วทำการประมวลผลให้เสร็จภายในรอบเดียว จึงทำให้มีค่าความถูกต้องมากกว่า

- ประเมินด้วยค่าความล่าช้า (Delay)

ในการประมวลผลเพื่อตรวจจับ DDoS นั้นต้องมีการจัดเก็บและคัดกรองข้อมูลเพื่อให้ข้อมูลนั้นซึ่งถือว่าเป็นการเตรียมข้อมูลเพื่อการประมวลผล โดยค่าความล่าช้าจากขั้นตอนเริ่มแรกคือการเก็บข้อมูล จะถึงกระบวนการในการประมวลเพื่อตรวจจับ DDoS จะมีค่าความล่าช้าประมาณ 7 นาที ซึ่งถือว่าเป็นค่าความล่าช้าในการจัดการข้อมูลเพื่อการประมวลผล

เมื่อทำการพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่สำเร็จแล้วได้ทำการสัมภาษณ์ผู้เชี่ยวชาญ คือ นายเกรียงศักดิ์ เหล็กดี ตำแหน่งหัวหน้าฝ่ายวิจัยและพัฒนาระบบสำนักบริหารเทคโนโลยีสารสนเทศเพื่อการพัฒนาการศึกษาเกี่ยวกับระบบที่พัฒนาขึ้น โดยมีหัวข้อในการสัมภาษณ์ดังนี้

1. ความเป็นไปได้ในการนำต้นแบบที่พัฒนาขึ้นมาใช้กับระบบงานจริง
2. ความสามารถของระบบต้นแบบที่พัฒนาขึ้นในการรองรับปริมาณข้อมูลที่เพิ่มมากขึ้นในการกรณีที่มีการเพิ่มของจำนวน ลิงค์และโหนด

3. ในการตรวจจับการจู่โจมของ DDoS ตามต้นแบบที่พัฒนาขึ้นสามารถช่วยแก้ปัญหาจากระบบเดิมได้หรือไม่
4. ประโยชน์ที่ เครือข่าย UniNet จะได้รับเมื่อนำต้นแบบมาใช้ในระบบจริง
5. โครงสร้างในการนำต้นแบบมาใช้ควรเป็นอย่างไร
6. ข้อเสนอแนะ
7. ปัญหาและอุปสรรคในการนำมาใช้กับระบบจริง

ความเป็นไปได้ในการนำต้นแบบที่พัฒนาขึ้นมาใช้กับระบบงานจริง เนื่องจากจำนวนผู้ใช้ของ UniNet มีจำนวนมากจึงทำให้ *log file* มีจำนวนมากตามไปด้วย ซึ่งทางผู้ถูกสัมภาษณ์แจ้งว่า ระบบปัจจุบัน UniNet ใช้ NetFlow collector เป็นตัวเก็บข้อมูลและมีข้อจำกัดในเรื่องของพื้นที่ในการจัดเก็บที่มีอยู่อย่างจำกัด การนำระบบต้นแบบที่พัฒนาขึ้นมาใช้โดยนำหลักการของ Big Data มาใช้ในการจัดเก็บข้อมูล ที่สามารถเพิ่มพื้นที่ในการจัดเก็บข้อมูลได้ขึ้นเรื่อยๆ โดยไม่ส่งผลกระทบกับระบบงานที่กำลังดำเนินการอยู่ จะส่งผลให้สามารถจัดเก็บข้อมูล log file มาใช้ประโยชน์ได้ ไม่ว่าจะเป็นการนำมาใช้ในการวิเคราะห์ระบบเครือข่าย หรือนำมาประมวลเพื่อใช้ในการป้องกันการโจมตีจากผู้ไม่ประสงค์ดี ได้อีกด้วย

- ความสามารถในการรองรับปริมาณข้อมูลที่เพิ่มมากขึ้นในกรณีที่มีการเพิ่มของจำนวนลิงค์และโหนด

ผู้ให้สัมภาษณ์แจ้งว่าในระบบของ UniNet ในปัจจุบันยังไม่มีการจัดเก็บ log file ของระบบไว้ และนำมาใช้ในการวิเคราะห์เครือข่าย ด้วย NetFlow Application และทิ้งข้อมูลนั้นไป เพราะไม่สามารถบันทึกเก็บเอาไว้ได้เนื่องจากจำนวนพื้นที่ในการจัดเก็บข้อมูลมีอยู่จำกัด และอัตราข้อมูลก็มีแนวโน้มที่เพิ่มขึ้นเรื่อยๆ จากการเพิ่มจำนวนลิงค์และ โหนดของระบบ หากนำระบบต้นแบบที่พัฒนาขึ้นมาใช้ในการจัดเก็บข้อมูลสามารถเพิ่มพื้นที่ในการจัดเก็บข้อมูลได้โดยไม่ส่งผลกระทบต่อกระบวนการทำงานที่ทำอยู่ในระบบ

- ในการตรวจจับการจู่โจมของ DDoS ตามต้นแบบที่พัฒนาขึ้นสามารถช่วยแก้ปัญหาจากระบบเดิมได้หรือไม่

ผลที่ได้จากการสัมภาษณ์ผู้เชี่ยวชาญได้รับข้อมูลว่า ต้นแบบที่พัฒนาขึ้นสามารถที่จะช่วยหรือให้ผู้ดูแลระบบตัดสินใจในการตัดการเชื่อมต่อโหนดที่มีปัญหาได้อย่างมีเหตุผลมากขึ้น เนื่องจากบางครั้งไม่สามารถเชื่อมต่อโหนดในระบบเครือข่ายได้ ทางผู้ดูแลระบบก็ไม่อาจทราบได้ว่า โหนดนั้นเกิดอะไรขึ้น ติดต่อไม่ได้เนื่องจากอะไร *ลิงค์*เกิดความเสียหาย หรือ โดนโจมตีจากผู้ไม่ประสงค์ดี

- ประโยชน์ที่ เครือข่าย UniNet จะได้รับเมื่อนำต้นแบบมาใช้ในระบบจริง

1. สามารถที่จะนำข้อมูล ขนาดใหญ่ (log file) มาใช้ในการวิเคราะห์ระบบเครือข่ายและป้องกันการโจมตีจากผู้บุกรุกได้
2. มีข้อมูลเกี่ยวกับการโดนโจมตีเก็บบันทึกไว้เพื่อให้เป็นข้อมูลในการพัฒนาระบบเครือข่ายต่อไปในอนาคต
3. ลดภาวะเสี่ยงที่จะเกิดความเสียหายกับระบบเครือข่าย
4. สามารถทำการวิเคราะห์แยกแยะการใช้งานปกติและการโดนโจมตีออกจากกันได้

- โครงสร้างในการนำต้นแบบมาใช้ควรเป็นอย่างไร

โครงสร้างของระบบต้นแบบที่พัฒนาขึ้น ได้นำเสนอการออกแบบที่สามารถช่วยลดความแออัดในการจราจรบนระบบเครือข่ายได้ เพราะเป็นการส่งแต่เพียงผลลัพธ์ ที่เกิด DDoS เท่านั้นโดยผลลัพธ์ที่ส่งไปจะเป็น Text ทำให้ไม่ส่งผลกระทบต่อการจราจรของระบบเครือข่าย ซึ่งทางผู้เชี่ยวชาญได้แสดงความคิดเห็นเกี่ยวกับโครงสร้างที่ออกแบบซึ่งเป็นแนวคิดที่ดี เนื่องจากระบบ UniNet มีผู้ใช้จำนวนมาก หากนำระบบต้นแบบที่พัฒนาขึ้นมาใช้งานจริง จะไม่ก่อให้เกิดผลกระทบต่อการจราจรของระบบ

- ข้อเสนอแนะ

ผู้เชี่ยวชาญได้เสนอแนะว่า หากนำระบบต้นแบบที่พัฒนาขึ้นมาใช้กับระบบเครือข่าย UniNet ต้องทำการเลือกจับข้อมูล log file จาก *Interface* ของ Router เพื่อที่จะทำให้ได้ข้อมูลที่ครบถ้วน ดังนั้นควรจะเลือกเก็บข้อมูลจาก *interface* ของ Router

- อุปสรรค

ทางผู้ให้สมภาษณ์แจ้งว่าระบบเครือข่าย UniNet ปัจจุบันการเก็บ log file ใช้ NetFlow เป็นตัวสร้างและ เก็บไปยัง NetFlow collector โดยมีลักษณะ ในการจัดเก็บ log file คือ จะจัดเก็บในลักษณะสุ่ม ซึ่งหมายความว่าในการจัดเก็บจะไม่มีการเก็บทุกโฟล์ว เช่น 1/1000, ความหมายคือ ใน 1000 โฟล์ว จะเก็บเพียง 1 โฟล์วเท่านั้น ปัญหาคืออาจก่อให้เกิดการสูญเสียข้อมูลที่อาจมีประโยชน์ต่อการประมวลผลก็ได้

5.4 สรุปผลการวิจัย

- ปัญหาและอุปสรรค

ในการวิจัยนี้การมีปัญหาและอุปสรรคที่เกิดขึ้นในการวิจัยสามารถแยกออกมาได้ดังนี้คือ

1. ปัญหาเกี่ยวกับระบบที่ใช้ในการทดสอบ

ในการวิจัยครั้งนี้ ได้ทำการโจมตี DDoS กับระบบ UniNet จริงอาจก่อให้เกิดความเสียหายขึ้นกับระบบได้

2. ปัญหาเกี่ยวกับการนำข้อมูลจริงของระบบเครือข่ายมาใช้งาน

ในการนำข้อมูล (log file) จริงมาใช้นั้นมีความเป็นไปได้ยากเนื่องจาก ในการใช้ข้อมูลจริงไม่สามารถทราบได้ว่า โฟล์วใดที่เป็น DDoS เพราะระบบเครือข่าย UniNet ไม่ได้ทำการบันทึกข้อมูลที่เกิด DDoS ไว้

5.5 แนวทางการวิจัยในอนาคต

ในการพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ ได้มีการทดสอบค่าความถูกต้องในการตรวจจับ DDoS ซึ่งได้ผลค่าความถูกต้องไม่มากนัก อาจเนื่องมาจาก อัลกอริทึมที่นำมาทดสอบยังไม่เหมาะกับรูปแบบการจัดเก็บข้อมูลในระบบที่พัฒนาขึ้น ในการป้องกันการถูกโจมตีด้วย DDoS โดยใช้ข้อมูลขนาดใหญ่นั้น ยังมีนักวิจัยจำนวนมากให้ความสนใจ และพยายามที่จะศึกษาพฤติกรรมในการโจมตี เพื่อสร้างอัลกอริทึมที่สามารถทราบล่วงหน้าได้ก่อนที่จะมีการโจมตี ซึ่งในทางเดียวกันผู้วิจัยก็มีความคาดหวังเป็นอย่างยิ่งที่จะศึกษาและทำการต่อยอดการพัฒนาต้นแบบระบบการวิเคราะห์เครือข่ายที่มีความซับซ้อนสูงด้วยข้อมูลขนาดใหญ่ที่สามารถแจ้งเตือนล่วงหน้าก่อนที่จะเกิดภัยคุกคามกับระบบได้

การวิจัยในครั้งนี้จะเป็นประโยชน์ระบบเครือข่ายที่มีความซับซ้อนสูงเนื่องจากการป้องกันการโจมตีด้วย DDoS ทำได้ยาก หากผู้ดูแลระบบมีเครื่องมือที่สามารถทราบได้ว่าถูกโจมตีด้วย DDoS ก่อนที่จะทำให้ระบบเกิดความเสียหาย จะช่วยให้ระบบมีความน่าเชื่อถือ และมีเสถียรภาพมากขึ้น โดยการประเมินต้นแบบที่พัฒนาขึ้นนี้ ด้วยการนำเสนอต้นแบบที่พัฒนาขึ้นให้กับผู้เชี่ยวชาญขอระบบเครือข่าย UniNet และทำการสัมภาษณ์เกี่ยวกับความความเป็นไปได้ในการนำต้นแบบที่พัฒนาขึ้นไปใช้จริง ผลสรุปจากผู้เชี่ยวชาญให้ความเห็นว่า ต้นแบบที่ทำการพัฒนาขึ้นมีประโยชน์และสามารถนำมาใช้กับระบบจริงของ UniNet ได้

รายการอ้างอิง

1. Tabatabaie Nezhad, S.M., M. Nazari, and E.A. Gharavol, *A Novel DoS and DDoS Attacks Detection Algorithm Using ARIMA Time Series Model and Chaotic System in Computer Networks.* IEEE Communications Letters, 2016. 20(4): p. 700–703.
2. UniNet. *UniNet*. 2014; Available from: http://www.UniNet.th.
3. Wang, X.F. and G. Chen, *Complex networks: small-world, scale-free and beyond.* IEEE Circuits and Systems Magazine, 2003(First Quarter): p. 6-20.
4. Dontongdang, S., P. Tantatsanawong, and A. Saeung. *Big Data Testbed for Research and Education Networks Analysis*. in *Proceedings of the APAN – Network Research Workshop*. 2015.
5. Tantatsanawong, P., S. Dontongdang, and P. U-Aroon. *Improving Big Data on Research and Education Networks using Future Internet Approach: A Case Study of Networks Analysis*. 2015.
6. Jia, B., et al., *A Novel Real-Time DDoS Attack Detection Mechanism Based on MDRA Algorithm in Big Data.* Hindawi Publishing Corporation Mathematical Problems in Engineering, 2016: p. 1-10.
7. Terzi, D.S., R. Terzi, and S. Sagiroglu, *Big Data Analytics for Network Anomaly Detection from NetFlow Data.* IEEE 2nd International Conference on Computer Science and Engineering, 2017: p. 592-597.
8. Zhenqi, W. and W. Xinyu, *NetFlow Based Intrusion Detection System.* International Conference on MultiMedia and Information Technology, 2008: p. 825-828.
9. Maheshwari, V., A. Bhatia, and K. Kumar, *Faster Detection and Prediction of DDoS attacks using MapReduce and Time Series Analysis.* IEEE 12th International Conference on Information Technology - New Generations, 2018: p. 556–561.
10. Mohanty, H., P. Bhuyan, and D. Chenthati, *Big Data A Primer*. 2015, India: Spinger
11. Navaz, A., G. Velusamy, and D. Gurkan, *Experiments on Networking of Hadoop.* IEEE 22nd International Conference on Network Protocols, 2014: p. 544 – 547.
12. Pandey, K., A. Gadwal, and P. Lakkadwala, *Hadoop Multi Node Cluster Resource*

Analysis. IEEE Symposium on Colossal Data Analysis and Networking (CDAN), , 2016. 167-172.

13. Vohra, D., *Practical Hadoop Ecosystem*. 2016: Apress.
14. Lakhe, B., *Practical Hadoop Migration*. 2016: Springer.
15. White, T., *Hadoop: The Definitive Guide*. 3 ed. 2012: O'Reilly.
16. Khan, M.A., Z.A. Memon, and S. Khan, *Highly Available Hadoop NameNode Architecture.* 2012 International Conference on Advanced Computer Science Applications and Technologies, 2012: p. 168-172.
17. Narayan, S., et al., *OpenFlow Enabled Hadoop Over Local and Wide Area Clusters.* IEEE, 2012: p. 1625-1628.
18. Wu, S., et al., *Distributed MapReduce Engine with Fault Tolerance.* IEEE ICC 2014 - Selected Areas in Communications Symposium, 2014. 3626-3630.
19. Ko, A.C. and W.T. Zaw, *Fault Tolerant Erasure Coded Replication for HDFS Based Cloud Storage.* IEEE Fourth International Conference on Big Data and Cloud Computing, 2014: p. 104-109.
20. UniNet. 2019; Available from: http://wunca.uni.net.th/wunca_regis/wunca39_doc/26/001_WUNCA39-a.somsak.pdf.
21. Sammut, C. and G.I. Webb, *Data Set*, in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G.I. Webb, Editors. 2017, Springer US: Boston, MA. p. 327-327.
22. Garcia, S., et al., *An empirical comparison of botnet detection methods.* Computers and Security Journal, 2014. 45: p. 100-123.
23. Laboratory, S.R. *CTU-13 Dataset A Labeled Dataset with Botnet, Normal and Background traffic*. 2014; Available from: https://www.stratosphereips.org/atasets-ctu13.
24. Dev, D. and R. Patgiri, *Performance Evaluation of HDFS in Big Data Management.* IEEE 12th International Conference on Information Technology - New Generations, 2014.

มหาวิทยาลัยศิลปากร

ประวัติผู้เขียน

ชื่อ-สกุล	สมเกียรติ ดอนทองแดง
วัน เดือน ปี เกิด	04 เมษายน 2523
สถานที่เกิด	นครปฐม
ที่อยู่ปัจจุบัน	49 ม. 2 ต.สระสี่มุม อ.กำแพงแสน จ.นครปฐม 73140