สำนักงานวิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

วิธีดำเนินการวิจัย

ในบทนี้จะกล่าวถึงการจัดเก็บข้อมูลปริมาณแทรฟฟิคเครือข่ายการจัดเก็บข้อมูลปริมาณการให้บริการของตัวอุปกรณ์แต่ละตัวบนเครือข่ายคอมพิวเตอร์จุฬาลงกรณ์มหาวิทยาลัย ซึ่งได้ทำการศึกษาโครงสร้างการเชื่อมต่อเครือข่ายจุฬาฯในการทำวิจัยลักษณะจากปลายทางถึงปลายทางคือเริ่มจากภาควิชาวิศวกรรมคอมพิวเตอร์เชื่อมต่อไปยังคณะวิศวกรรมศาสตร์และเชื่อมต่อไปยังสำนักเทคโนโลยีสารสนเทศซึ่งเป็นทางออกในการเชื่อมต่ออินเตอร์เน็ตดังนั้นได้ทำการเก็บข้อมูลตัวอุปกรณ์สวิตช์ในการเชื่อมต่อของแต่ละหน่วยงานได้ดังนี้

3.1 การเก็บข้อมูลปริมาณเครือข่ายบนอุปกรณ์สวิตช์

3.1.1 ภาควิชาวิศวกรรมคอมพิวเตอร์ ซึ่งเป็นภาควิชาส่วนหนึ่งในคณะวิศวกรรมศาสตร์อยู่อาคาร4 ตั้งแต่ชั้น 17 ถึงชั้น 20 ได้มีการเชื่อมต่อระบบเครือข่ายแลนถึงกันทั้งหมดโดยมีตัวอุปกรณ์หลักๆดังนี้

3.1.1.1 Main Switch Catalyst 3550 อยู่ชั้น 19 ทำหน้าที่เชื่อมต่อไปยังห้อง LAB ต่างๆทุกห้อง ห้องพักอาจารย์ทุกท่าน ห้องคอมพิวเตอร์ชั้น17 ถึงชั้น 20 ทำให้ระบบเครือข่ายภาคคอมพิวเตอร์สามารถติดต่อกันได้ตลอดเวลา ด้วความเร็ว 100 Mbps.ที่ Port Interface FastEthernet0/43 และใช้ Port Interface FastEthernet0/36 Uplink ไปยังSwitch-Catalyst 2926

3.1.1.2 Gateway Switch Catalyst2926 อยู่ชั้น 19 ทำหน้าที่เป็นประตูทางออกในการเชื่อมต่อไปอินเตอร์เน็ตโดยจะเชื่อมต่อไปยังคณะวิศวกรรมศาสตร์ด้วยความเร็ว 1 Gbps.ที่ Port Interface Gigabit0/1 วงจรเดียว และเชื่อมต่อไปยังอุปกรณ์สวิตช์ตัวอื่นๆ เช่น Catalyst3550 ด้วยความเร็ว 100 Mbps. ที่ Port Interface Gigabit0/2

3.1.2 คณะวิศวกรรมศาสตร์ ซึ่งมีสำนักบริการคอมพิวเตอร์ให้บริการด้านเทคโนโลยีสารสนเทศให้กับภาควิชาต่างๆ ในคณะ อีกทั้งมีศูนย์คอมพิวเตอร์ให้บริการนิสิตด้วย ดังนั้นที่คณะจะมีห้องศูนย์คอมพิวเตอร์ในการเชื่อมต่อวงจรสื่อสัญญาณไปยังอาคารต่างๆ สาขาต่างๆ ด้วยกันหลายวงจรแต่ในขอบเขตงานวิจัยนี้ทางผู้วิจัยจะสนใจวงจรการเชื่อมต่อมายังภาควิชาวิศวกรรมคอมพิวเตอร์ซึ่งมีวงจรการเชื่อมต่อเป็นแบบสายใยนำแสง 1 วงจรด้วยตัวอุปกรณ์ดังนี้

3.1.2.1 Switch Catalyst5500 (ENG04) อยู่อาคาร 4 ชั้น 2 ทำหน้าที่เป็นตัวเชื่อมวงจรสายใยแก้วนำแสงไปยังภาควิชาคอมพิวเตอร์ ด้วยความเร็ว 1 Gbps. ที่ Port Interface Gigabit9/8 และมี Port Interface Gigabit1/2 เชื่อมติดต่อกลับไปยังศูนย์คอมพิวเตอร์ที่คณะวิศวกรรมศาสตร์ด้วยความเร็ว20 Gbps.

3.1.2.2 Switch Catalyst5500 (ENG03) อยู่อาคาร 3 ชั้น 1 ห้องศูนย์คอมพิวเตอร์ทำหน้าที่เป็นตัวเชื่อมวงจรสายใยแก้วนำแสงไปยังอาคาร 4 ด้วยความเร็ว 20 Gbps.ที่ Port Interface Gigabit1/2 และมี Port Interface

FastEthernet3/6 ติดต่อไปยังเกตเวย์อุปกรณ์สวิตช์ของคณะวิศวกรรมศาสตร์ผ่านสายตีเกลียวคู่(UTP CAT5) จำนวน 1 วงจรด้วยความเร็ว 100 Mbps.

3.1.2.3 Gateway Switch Catalyst3550 อยู่อาคาร 3 ชั้น 1 ห้องศูนย์คอมพิวเตอร์ทำหน้าที่เป็นประตูทางออกในการเชื่อมต่อไปเกตเวย์อินเตอร์เน็ตที่สำนักเทคโนโลยีสารสนเทศด้วยความเร็ว 1 Gbps.ที่ Port Interface Gigabit0/1 วงจรเดียว และเชื่อมต่อไปยังอุปกรณ์สวิตช์ตัวอื่นๆ ในอาคารต่างๆ รวมทั้งภาคคอมพิวเตอร์โดยผ่านตัว Switch Catalyst5500 (ENGO3) ที่ Port Interface FastEthernet0/36 ด้วยความเร็ว 100 Mbps.

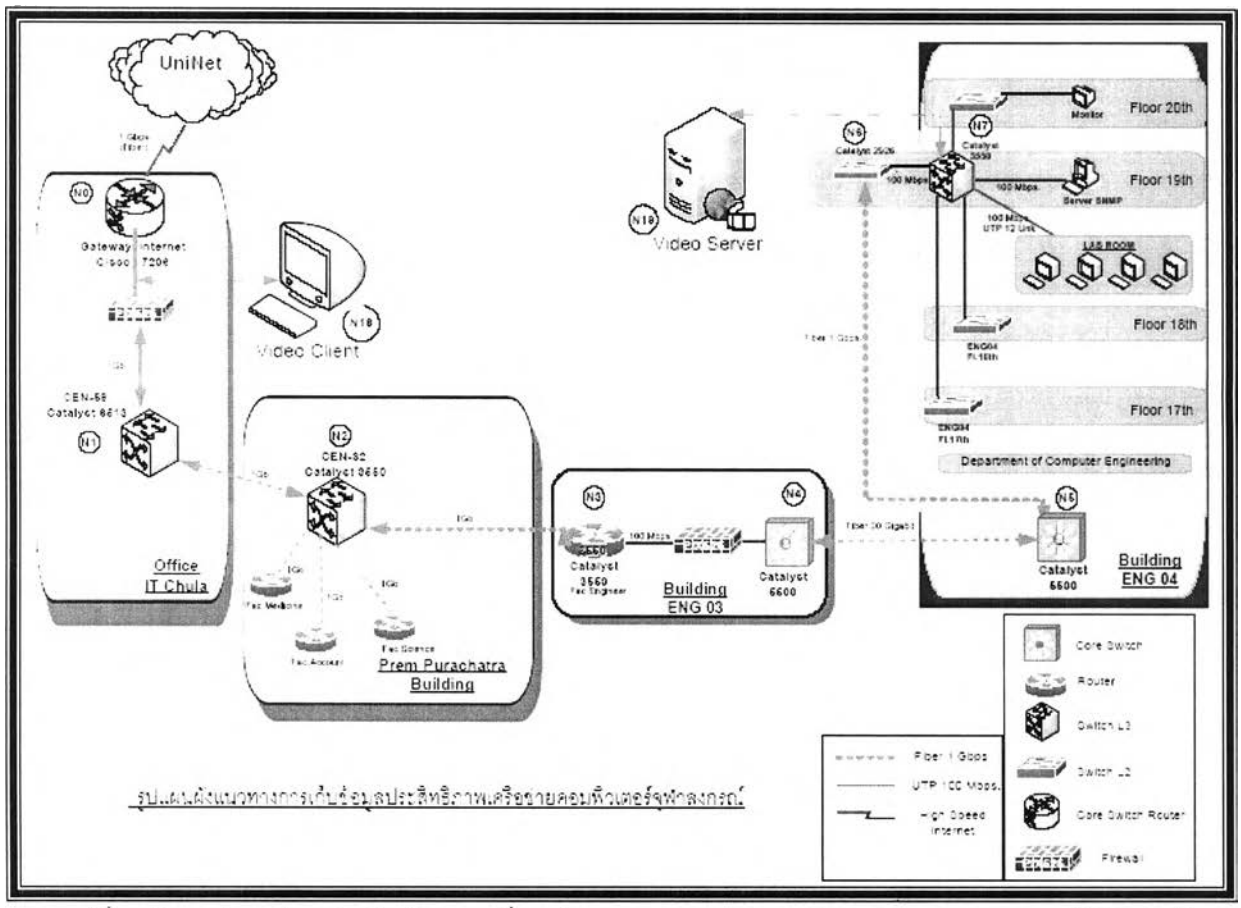
3.1.3 สำนักเทคโนโลยีสารสนเทศ ซึ่งเป็นหน่วยงานให้บริการหลักระบบสารสนเทศและเป็นทางออกการให้บริการระบบอินเตอร์เน็ตทุกบริการ เช่น เว็บเพจ ,จดหมายอิเลคทรอนิคส์ , การส่งไฟล์เป็นต้น ให้กับคณะ สาขา และหน่วยงานต่างๆ ในมหาวิทยาลัยทั้งหมด ซึ่งมีอุปกรณ์ที่ใช้ในการติดต่อเครือข่ายไปยังสถานที่ต่างๆ ก็จะมีอยู่หลายรูปแบบทั้งตัวอุปกรณ์สวิตช์และฮับ ในงานวิจัยนี้เส้นทางการเชื่อมต่อจากคณะวิศวกรรมศาสตร์เข้ามายังสำนักเทคโนโลยีสารสนเทศเพื่อใช้บริการอินเตอร์เนตมีอุปกรณ์เชื่อมต่อดังนี้

3.1.3.1 Switch Catalyst3550 (CEN-32) อยู่อาคารเปรมบุรฉัตร ทำหน้าที่เชื่อมต่อวงจรสายใยแก้วนำแสง ไปยังคณะวิศวกรรมศาสตร์ด้วยความเร็ว 1 Gbps.ที่ Port Interface Gigabit0/5 และมี Port Interface Gigabit0/1 เชื่อมต่อไปยังห้องศูนย์คอมพิวเตอร์ของสำนักเทคโนโลยีสารสนเทศด้วยความเร็ว1Gbps.

3.1.3.2 Switch Catalyst6513 (CEN-59) อยู่อาคารจามจุรี 3 ชั้น 4 ห้องศูนย์คอมพิวเตอร์ทำหน้าที่เป็นแกนกลางระบบเครือข่ายคอมพิวเตอร์ของจุฬาลงกรณ์มหาวิทยาลัยในการผ่านเข้าออกเครือข่ายระบบสารสนเทศทั้งหมด ซึ่งมี Port Interface Gigabit1/22 ติดต่อไปยังคณะวิศวกรรมศาสตร์ด้วยความเร็ว 1 Gbps. และมี Port Interface Gigabit1/24 ที่ความเร็ว 1 Gbps.เป็นport ที่เชื่อมต่อตัวอุปกรณ์ไฟร์วอล์ลทำหน้าที่ตรวจสอบความปลอดภัยทุกอย่างในการเข้าออกระบบเครือข่ายอินเตอร์เน็ตต่อไป

3.1.3.3 Internet Gateway Router7206 (CEN-59) อยู่อาคารจามจุรี 3 ชั้น 4 ห้องศูนย์คอมพิวเตอร์ทำหน้าที่เป็นอุปกรณ์สวิตช์ในการเชื่อมต่อระบบเครือข่ายอินเตอร์เน็ตทั้งเครือข่ายยูนิเนต(Uninet)และผู้ให้บริการอินเตอร์เน็ตอื่นๆ (Internet Service Provider) ด้วยความเร็ว 1 Gbps. ที่ Port Interface Gigabit0/1 และมี Port Interface Gigabit0/2 เชื่อมต่อเข้าไปยังตัวอุปกรณ์ไฟร์วอล์ลทำหน้าที่ตรวจสอบความปลอดภัยข้อมูลเข้าออกก่อนเข้าสู่เครือข่ายคอมพิวเตอร์จุฬาลงกรณ์มหาวิทยาลัยต่อไป

อุปกรณ์ต่างๆทั้งหมดที่ได้กล่าวมานี้จะเป็นโหนดในการเก็บข้อมูลปริมาณแทรฟฟิคผ่านเข้าออกตัวอุปกรณ์สวิตช์ ซึ่งทั้งหมดนี้จะถูกนำใช้ในแบบจำลองเครือข่ายการทดสอบการให้บริการวีดีโอตามสั่งผ่านเครือข่ายจุฬาฯ โดยผู้วิจัยได้ทำการเก็บข้อมูลจากอุปกรณ์สวิตช์ในการเชื่อมต่อเครือข่ายตามรูปที่ 3-1



รูปที่3-1 โครงข่ายอุปกรณ์สวิตช์ในการเชื่อมต่อจากภาคคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ไปยังสำนักเทคโนโลยีสารสนเทศ

3.2การจัดเก็บข้อมูลการใช้งานสายสัญญาณโดยใช้โพรโทคอลเอสเอ็นเอ็มพี (SNMP Protocol)

ข้อมูลในอุปกรณ์เครือข่ายส่วนใหญ่แล้วจะมีการเก็บค่ามิบ (MIB:Management Information Base) ซึ่งจะเก็บค่าสถานะเพื่อนำไปใช้ในการบริหารจัดการเครือข่ายโดยใช้โพรโตคอลเอสเอ็นเอ็มพี โดยในค่ามิบนี้จะมีค่าอยู่ชนิดหนึ่งเรียกตัวนับการลำเลียงข้อมูล(Traffic Counter) ซึ่งจะทำหน้าที่ในการตรวจนับปริมาณข้อมูลที่ถูกส่งเข้าและออกจากช่องสัญญาณต่างๆ ในอุปกรณ์สื่อสาร เราสามารถนำเอาค่านี้มาทำการเก็บสถิติปริมาณการใช้งานแทรฟฟิคเข้าออกในเครือข่ายคอมพิวเตอร์ได้เพื่อดูปริมาณการใช้งานช่องสัญญาณผ่านอุปกรณ์สวิตช์ในช่วงเวลาต่างๆได้

อุปกรณ์ในแต่ละโหนดจะมีการตั้งค่าพารามิเตอร์ต่างๆ ดังต่อไปนี้

- IP Address คือหมายเลขประจำตำแหน่งของอุปกรณ์
- Subnet Mak คือหมายเลขประจำเครือข่ายของอุปกรณ์
- SNMP Community String Name ของแต่ละอุปกรณ์ว่ามีชื่ออย่างไรบ้างในการให้เข้ามาอ่านค่ามิบของตัวอุปกรณ์นั้นๆ
- SNMP Port Service 161 คือ การทำงานของโปรโตคคอลเอสเอ็นเอ็มพีคือช่องทางสื่อสารหมายเลข

หอสมุดกลาง สำนักงานวิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ช่อง 161 สำหรับการดึงและรับข้อมูลค่าการทำงานของอุปกรณ์สวิตช์
ดังนั้นอุปกรณ์ในเครือข่ายคอมพิวเตอร์ที่นำมาวิจัยจะมีค่าพารามิเตอร์ต่างๆสรุปได้ดังต่อไปนี้ดัง ตารางที่ 3-1 และ ตารางที่3-2

ตารางที่ 3-1 หมายเลขประจำตำแหน่งและหมายเลขประจำเครือข่ายของอุปกรณ์สวิตช์

No.Device	Description	IP Address	Mask	Gateway	DNS	Hostname Name Device	SNMP Community String Name	Port Interface VLAN ID No.
19	Server Monitor	161.200.92.6	255.255.255.192	161.200.92.62	161.200.93.17 161.200.93.11	BOONLUE		
7	Catalyst 3550 (Fl.19)	161.200.93.252	255.255.255.0				cpeng	In=0/43, Out=0/36 VLAN 3
6	Catalyst 2926 (Fl.19)	161.200.93.241	255.255.255.0				cpeng	In=1/2, Out=SL0
5	Catalyst 5500 (ENG04)	161.200.80.229	255.255.255.0				readonly	In=9/8, Out=1/2 VLAN28
4	Catalyst 5500 (ENG03)	161.200.80.239	255.255.255.0				readonly	In=1/2, Out=3/6 VLAN28
3	Catalyst 3550 (Fac.ENG)	161.200.80.227	255.255.255.0				readonly	In=0/6, Out=GI0/1
2	Catalyst 3550 (CEN-32)	161.200.255.6	255.255.255.252				public	In=GI0/5, Out=GI0/1 VLAN TRUNK 691
1	Catalyst 6513 (CEN-59)	161.200.255.10	255.255.255.252				public	In=GI1/22, Out=GI1/24 VLAN691
0	GW Router 7206 (CEN-59)	161.200.255.218	255.255.255.252				public	In=GI0/2, Out=0/1

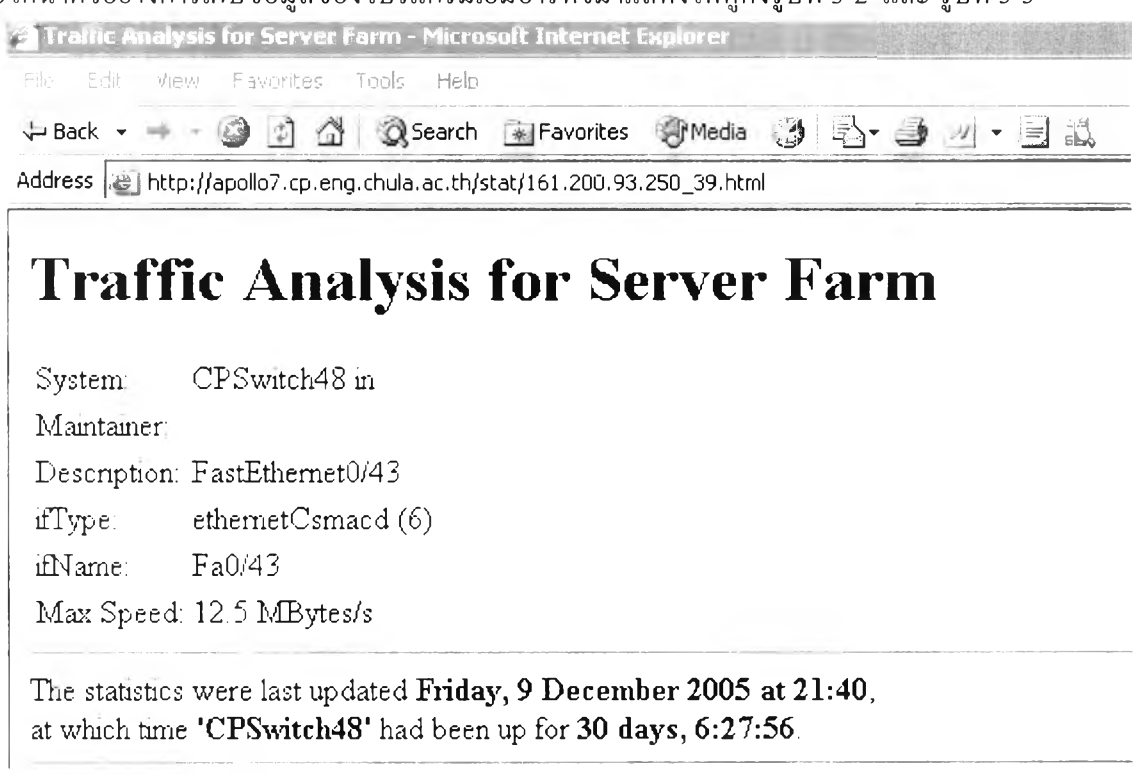
ตารางที่ 3-2 พารามิเตอร์อินเตอร์เฟสขาเข้าและขาออกของอุปกรณ์สวิตช์

No.Device	Description	Interface Port Device	Address Interface MRTG Traffic
19	Server Monitor	In FastEthernet 6 Out : FastEthernet 6	http://161.200.92.6:8000/
7	Catalyst 3550 (Fl.19)	In FastEthernet 0/43 Out FastEthernet 0/36	http://apollo7.cp.eng.chula.ac.th/stat/
6	Catalyst 2926 (Fl.19)	In FastEthernet 1/2 (Traffic Analysis for 3 -- CP-CAT2926-1) Out FastEthernet 1/1 (Traffic Analysis for 4 -- CP-CAT2926-	http://161.200.92.7/cisco2926/
5	Catalyst 5500 (ENG04)	In : Traffic Analysis for 9/8 -- ENG04_CAT5500 Out Traffic Analysis for 1/2 -- ENG04_CAT5500	http://161.200.80.2/eng04_5500.html
4	Catalyst 5500 (ENG03)	In Traffic Analysis for 1/2 -- (VLAN28) Out : Traffic Analysis for 3/6 -- (VLAN28)	http://161.200.80.2/eng03_5500.html
3	Catalyst 3550 (Fac.ENG)	In FastEthernet 0/6 VLAN 28 for CP on Fl. 17 - 19 BLDG4 Out : GigabitEthernet 0/1 Uplink to CUNET	http://161.200.80.2/eng03_3550.html
2	Catalyst 3550 (CEN-32)	In Traffic Analysis for GigabitEthernet0/5 Out : Traffic Analysis for GigabitEthernet0/1	http://www.it.chula.ac.th/statistics/cu-traffic/3550-cen32/
1	Catalyst 6513 (CEN-59)	In 3550-cen32 GigabitEthernet 1/22 Out : fw1-cen59 GigabitEthernet 1/24	http://www.it.chula.ac.th/statistics/cu-traffic/6513-cen59/
0	GW Router 7206 (CEN-59)	In fw1-cen59(1Gbps Full-duplex): GigabitEthernet0/2 Out Link to UniNet 23Apr2005(1Gbps Full-Duplex) : Gi0/1	http://www.it.chula.ac.th/statistics/cu-traffic/7206-cen59/

3.2.1 โปรแกรมเอ็มอาร์ทีจี (Multi Router Traffic Grapher)

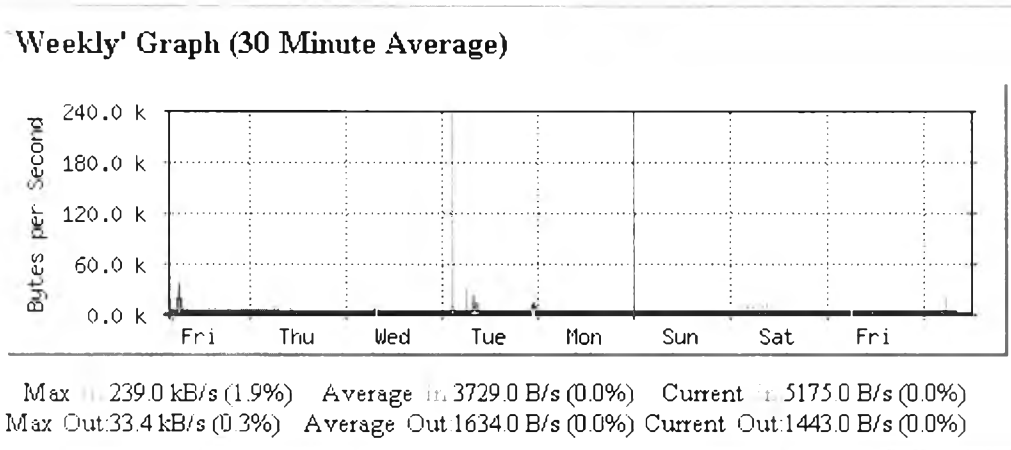
เป็นโปรแกรมซึ่งจะอ่านค่าการลำเลียงข้อมูล (Traffic Counter) นี้โดยผ่านโพรโทคอลเอสเอ็นเอ็มพี โดยทุกๆ ช่วงเวลาหนึ่งเช่นทุกๆ 5 นาที จะทำการอ่านค่าตัวนับการลำเลียงข้อมูล และมาเปรียบเทียบกับเมื่อ 5 นาทีที่ผ่านมาว่าใน 5 นาทีนี้มีปริมาณข้อมูลถูกรับและส่งผ่านสัญญาณนั้นๆ เป็นเท่าไร และจะทำการคำนวณหาอัตราการใช้งานช่องสัญญาณแบนด์วิดท์และทำการวาดกราฟ เพื่อแสดงผลได้

ดังนั้นผู้วิจัยจึงได้ทำการเก็บข้อมูลปริมาณแทรฟฟิคขาเข้าและขาออกผ่านตัวอุปกรณ์ที่สนใจตามตารางที่ 3-2 โดยทำการเก็บข้อมูลแทรฟฟิคเครือข่ายตั้งแต่วันที่ 1 ธันวาคม 2548 ถึงวันที่ 31 ธันวาคม 2548 เป็นระยะเวลาหนึ่งเดือนในการเก็บข้อมูลมาวิจัยซึ่งร่วมมือกับทางเจ้าหน้าที่ดูแลระบบคอมพิวเตอร์ของสำนักเทคโนโลยีสารสนเทศ คณะวิศวกรรมศาสตร์ และภาคคอมพิวเตอร์ทำการติดตั้งโปรแกรมเอ็มอาร์ทีจีที่เครื่องเซอร์เวอร์ของแต่ละหน่วยงานนั้นและกำหนดสิทธิในการเข้าไปดึงข้อมูลจากอุปกรณ์สวิตช์ตามรายการอุปกรณ์ที่สนใจได้อีกทั้งผู้วิจัยสามารถเข้าไปดูตรวจสอบการทำงานการของโปรแกรมเอ็มอาร์ทีจีได้ตลอดเวลา อนึ่งผู้วิจัยได้นำตัวอย่างการเก็บข้อมูลของโปรแกรมเอ็มอาร์ทีจีมาแสดงให้ดูดังรูปที่ 3-2 และ รูปที่ 3-3



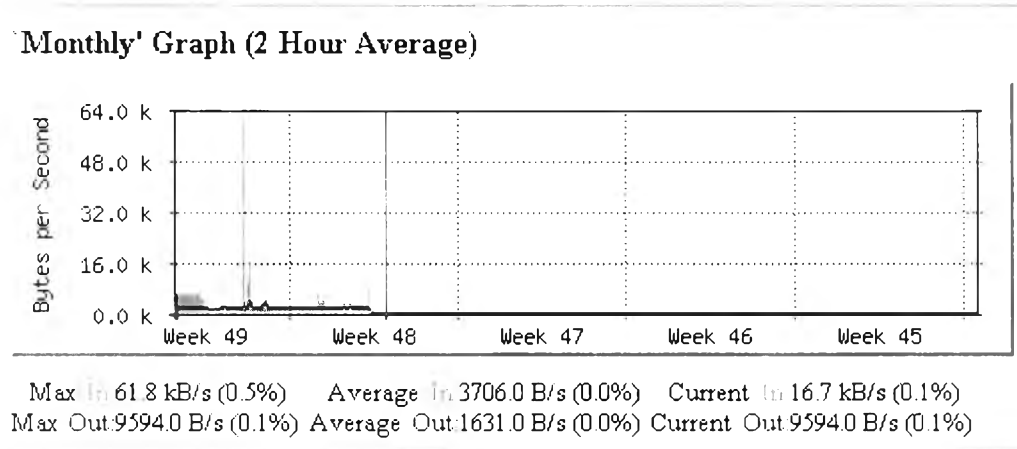
รูปที่ 3-2 รูปแบบการเก็บข้อมูลจากโปรแกรมเอ็มอาร์ทีจีจากอุปกรณ์โหนด7 คือCatalyst 3550 (Fl.19) :
ด้านขาเข้า(In) = FastEthernet0/43

แสดงพารามิเตอร์ของอุปกรณ์โหนด 7 คือตัวสวิตช์ Catalyst3550 ซึ่งอยู่ที่ภาคคอมพิวเตอร์ชั้น 19 อาคาร 4 คณะวิศวกรรมศาสตร์ เป็นสวิตช์ 48 พอร์ท ซึ่งพอร์ทที่ 0/43 เป็นพอร์ทที่เชื่อมต่อเข้ามายังเครือข่ายภายในผู้วิจัยสนใจจึงทำการเก็บข้อมูลที่หมายเลขพอร์ทเชื่อมต่อ ซึ่งมีความเร็วสูงสุดที่รับส่งได้ในวันที่ 9 ธันวาคม 2548 เวลา 21:40 มีค่า 12.5 MB. ตามรูปที่ 3-2



รูปที่ 3-2-1 รูปแสดงกราฟอัตราแทรฟฟิคการใช้งานเข้าออกรายสัปดาห์

จากรูปที่3-2-1 ผู้วิจัยจะทำการตรวจเช็คปริมาณแทรฟฟิคเข้าและออกประจำสัปดาห์ว่ามีค่าสูงสุดเท่าไร และค่าเฉลี่ยเท่าไรในรอบรายสัปดาห์



รูปที่ 3-2-2 รูปแสดงกราฟอัตราแทรฟฟิคการใช้งานเข้าออกรายเดือน

จากรูปที่3-2-2 ผู้วิจัยจะทำการตรวจเช็คปริมาณแทรฟฟิคเข้าและออกประจำเดือนว่ามีค่าสูงสุดเท่าไร และค่าเฉลี่ยเท่าไรในรอบรายเดือน

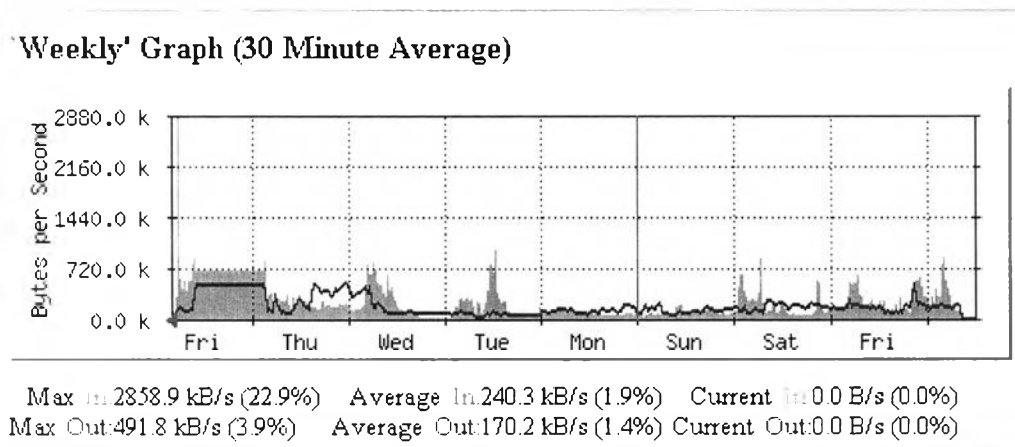
Traffic Analysis for UpLink

System: CPSwitch48 in
Maintainer:
Description: FastEthernet0/36
ifType: ethernetCsmacd (6)
ifName: Fa0/36
Max Speed: 12 5 MBytes/s

The statistics were last updated **Friday, 9 December 2005 at 20:26**,
at which time **'CPSwitch48'** had been up for **30 days, 5:14:34**.

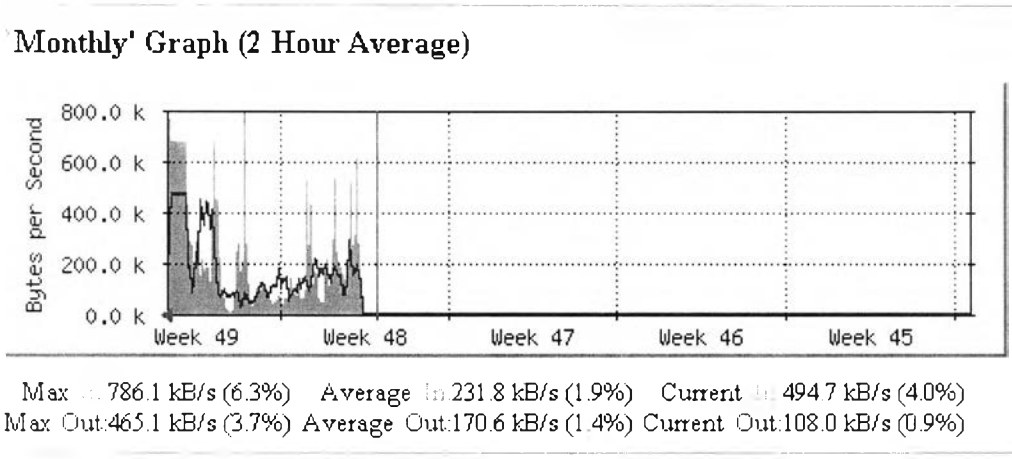
รูปที่ 3-3 รูปแบบการเก็บข้อมูลจากโปรแกรม MRTG จากอุปกรณ์โหนด 7 คือ Catalyst 3550 (Fl.19) :
ด้านขาออก (Out) = FastEthernet0/36

แสดงพารามิเตอร์ของอุปกรณ์โหนด 7 คือตัวสวิตช์ Catalyst3550 ซึ่งอยู่ที่ภาคคอมพิวเตอร์ชั้น 19 อาคาร 4 คณะวิศวกรรมศาสตร์ เป็นสวิตช์ 48 พอร์ท ซึ่งพอร์ทที่ 0/36 เป็นพอร์ทที่เชื่อมต่อออกไปยังเครือข่ายอินเตอร์เน็ตผู้วิจัยสนใจจึงทำการเก็บข้อมูลที่หมายเลขพอร์ทเชื่อมต่อนี่ ซึ่งมีความเร็วสูงสุดที่รับส่งได้ในวันที่ 9 ธันวาคม 2548 เวลา 20:26 มีค่า 12.5 MB. ตามรูปที่ 3-3



รูปที่ 3-3-1 รูปแสดงกราฟอัตราแทรฟฟิคการใช้งานเข้าออกรายสัปดาห์

จากรูปที่3-3-1 ผู้วิจัยจะทำการตรวจเช็คปริมาณแทรฟฟิคเข้าและออกประจำสัปดาห์ว่ามีค่าสูงสุดเท่าไร และค่าเฉลี่ยเท่าไรในรอบรายสัปดาห์



รูปที่ 3-3-2 รูปแสดงกราฟอัตราแทรฟฟิคการใช้งานเข้าออกรายเดือน

จากรูปที่3-3-2 ผู้วิจัยจะทำการตรวจเช็คปริมาณแทรฟฟิคเข้าและออกประจำเดือนว่ามีค่าสูงสุดเท่าไร และค่าเฉลี่ยเท่าไรในรอบรายเดือน

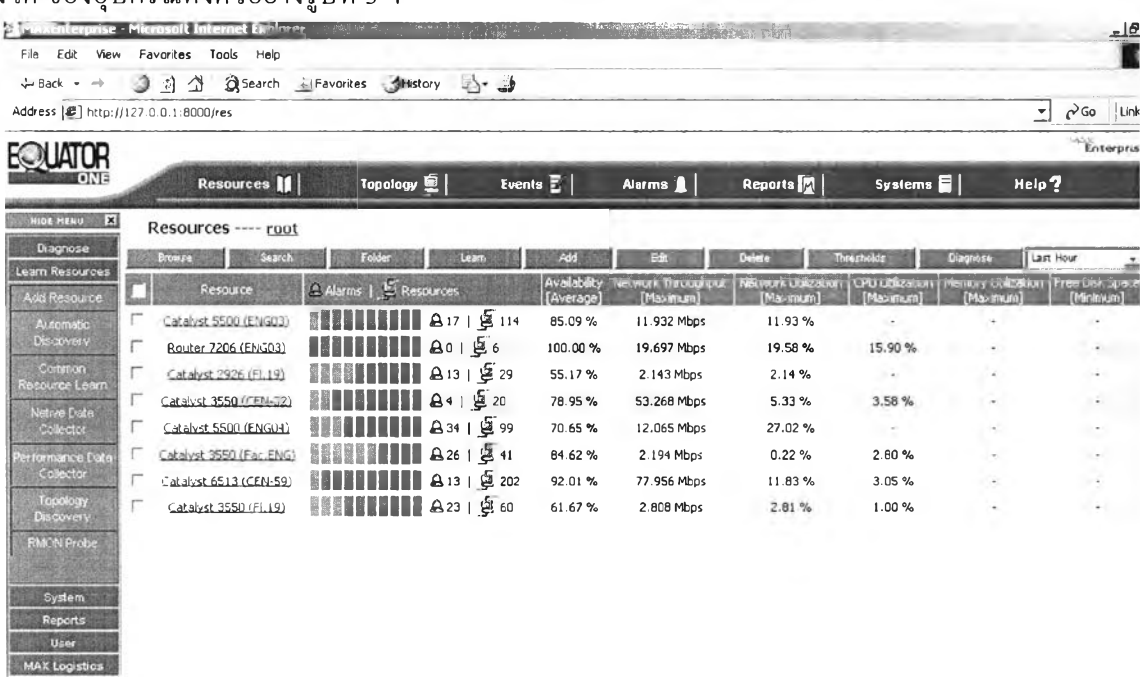
โดยการเก็บข้อมูลจากโปรแกรมเอ็มอาร์ทีจี ผู้วิจัยได้เก็บค่าสูงสุด(Maximum) และค่าฉลี่ย(Average) แทรฟฟิคของแต่ละด้านขาเข้าและขาออกของแต่ละพอร์ทอินเตอร์เฟสในเส้นทางของการเชื่อมต่อของทั้ง8 โหนดได้เป็นเวลาหนึ่งเดือนได้ดังตารางที่3-3

ตารางที่ 3-3 แสดงผลการเก็บข้อมูลแทรฟฟิคผ่านเข้าออกอุปกรณ์สวิตช์

Code Node No.	Node Device	MAX E1 Throughput Outside Traffic link to Device	MRTG Traffic				
		Network Throughput	Interface Resource	IN		OUT	
		Max		Max	Average	Max	Average
19	Video Server		In : Ethernet 100 Mbps Out : Ethernet 100 Mbps				
7	Catalyst 3550 (FI.19)	3.340 Mbps	In : FastEthernet 0/43 Out : FastEthernet 0/36	0.239 Mbps 2.8589 Mbps	0.003729 Mbps 0.2403 Mbps	0.0334 Mbps 0.4918 Mbps	0.001635 Mbps 0.1702 Mbps
6	Catalyst 2926 (FI.19)	2.260 Mbps	In : FastEthernet 1/2 Out : FastEthernet 1/1	0.4919 Mbps 2.8596 Mbps	0.1717 Mbps 0.2451Mbps	2.8588 Mbps 0.4926 Mbps	0.2459 Mbps 0.1817 Mbps
5	Catalyst 5500 (ENG04)	12.449 Mbps	In : FastEthernet9/8 Out : FastEthernet1/2	27.0 Mbps 40.5 Mbps	1.8823 Mbps 8.5295 Mbps	33.2 Mbps 32.4 Mbps	2.4065 Mbps 12.7 Mbps
4	Catalyst 5500 (ENG03)	7.095 Mbps	In : FastEthernet1/2 Out : FastEthernet3/6	56.7 Mbps 37.7 Mbps	13.6 Mbps 3.5094 Mbps	47.9 Mbps 45.6 Mbps	9.6128 Mbps 3.4796 Mbps
3	Catalyst 3550 (Fac.ENG)	2.444 Mbps	In : FastEthernet 0/6 Out : GigabitEthernet 0/1	10.6 Mbps 15.3 Mbps	1.6453 Mbps 1.6776 Mbps	19.1 Mbps 15.4 Mbps	2.5517 Mbps 1.7195 Mbps
2	Catalyst 3550 (CEN-32)	100.324 Mbps	In : GigabitEthernet0/5 Out : GigabitEthernet0/1	39.0 Mbps 154 Mbps	22.0 Mbps 67.3 Mbps	61.8 Mbps 145.6 Mbps	16.6 Mbps 78.1 Mbps
1	Catalyst 6513 (CEN-59)	110.835 Mbps	In : GigabitEthernet 1/22 Out : GigabitEthernet 1/24	145.8 Mbps 238.9 Mbps	78.2 Mbps 98.8 Mbps	153.5 Mbps 205 Mbps	67.4 Mbps 105.8 Mbps
18	Video Client		In : Ethernet 100 Mbps Out : Ethernet 100 Mbps				
0	GW Router 7206 (CEN-59)	111.234 Mbps	In : GigabitEthernet0/2 Out : GigabitEthernet0/1	202.2 Mbps 237.3 Mbps	101.1Mbps 98.0 Mbps	243.5 Mbps 201.9 Mbps	99.5 Mbps 100.9 Mbps
Update Collect Data Weekly 2/12/2005 to 9/12/2005 Via Monitor MRTG and E1 Program use port service SNMP (161)							

3.2.2 โปรแกรมอีควอเตอร์วัน (Program EquartorOne)

เป็นโปรแกรมที่ช่วยในการจัดเก็บค่าปริมาณช่องสัญญาณที่ส่งได้ของอุปกรณ์แต่ละโหนดว่าสามารถรองรับประสิทธิภาพ ณ ช่วงเวลานั้นเป็นเท่าไร โดยตัวโปรแกรมทำงานบนพื้นฐานของโพรโทคอลเอสเอ็นเอ็มพี ซึ่งช่วยให้โปรแกรมสามารถติดต่อขอข้อมูลกับอุปกรณ์ต่างๆ ในระบบเครือข่ายได้ โดยข้อมูลเหล่านี้จะช่วยให้ทราบถึงสถานะของแต่ละอุปกรณ์ในระบบเครือข่ายได้ โดยโปรแกรมจะทำหน้าที่เป็นตัวจัดการ(Manager) ทำการดึงข้อมูลจากตัวแทน(Agent)ซึ่งทำหน้าที่เก็บข้อมูลที่สนใจของอุปกรณ์สวิตช์นั้นจะถูกจัดเก็บในรูปของมิบ และใช้ Abstract Syntax Notation1 (ASN.1) เป็นตัวที่กำหนดวิธีการอธิบายโครงสร้างของข้อมูลที่ส่งข้ามไปมาของโพรโตคอล โดยค่าที่ทำการจัดเก็บจากตัว Agent ของแต่ละโหนดคือค่าเครือข่ายช่องสัญญาณที่ส่งได้และค่าเครือข่ายการใช้งานจากโปรแกรมอีควอเตอร์วัน ซึ่งจะทำการติดตั้งตัวเองเป็นเอสเอ็นเอ็มพีเซอร์เวอร์ในการทำการวิเคราะห์อุปกรณ์เครือข่ายต่างๆ ที่ได้ทำการเพิ่มโหนดไว้ทั้งหมด 8 ตัว ตามค่า IP Address และ Subnet Mask ของแต่ละโหนด โดยแต่ละ โหนด จะทำการเปิด บริการเอสเอ็นเอ็มพีหมายเลขพอร์ท161 และทำการเซทค่าชื่อประกาศสาธารณะของเอสเอ็นเอ็มพี(SNMP Commmunity String Name) ของอุปกรณ์โหนดนั้นๆ ให้ตรงกับชื่อประกาศสาธารณะของเอสเอ็นเอ็มพีเซอร์เวอร์บนโปรแกรมอีควอเตอร์วัน(SNMPSever EquartorOne) จากนั้นตัวโปรแกรมอีควอเตอร์วันจะถูกตั้งค่าให้ทำการวิเคราะห์อุปกรณ์ทุกๆ 180 วินาทีในการไปดึงข้อมูลจากตัวแทนเอสเอ็นเอ็มพีกลับมาทำการประมวลผลดูค่าปริมาณช่องสัญญาณที่ส่งได้ ของแต่ละอุปกรณ์ซึ่งจะคำนวณมาจากทุกๆ Port Interface ที่มีแทรฟฟิคผ่านเข้าออก ซึ่งจะแยกออกมาเป็น ช่องสัญญาณที่รับได้ขาเข้าและช่องสัญญาณที่ส่งได้ขาออก แล้วนำมารวมกันมาคำนวณมาเป็นปริมาณการใช้งานช่องสัญญาณที่ส่งได้ ของอุปกรณ์ตังตัวอย่างรูปที่ 3-4

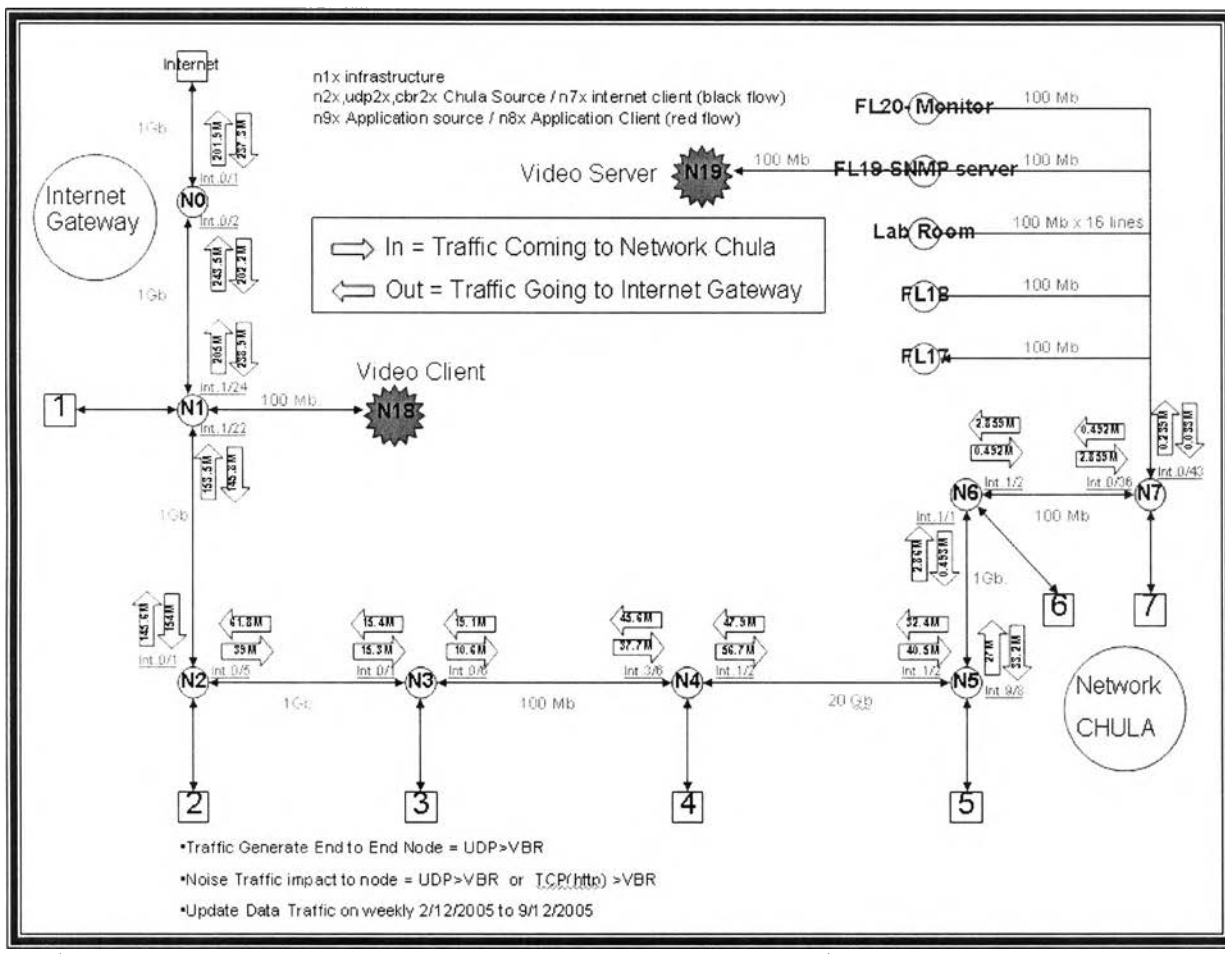


รูปที่ 3-4 แสดงรูปแบบการเก็บข้อมูลของโปรแกรมอีควอเตอร์วัน

3.3 การออกแบบโมเดลในเครื่องมือทดสอบเครือข่ายเอ็นเอสทู

หลังจากที่ผู้วิจัยได้ทำการเก็บข้อมูลจากอุปกรณ์ในเครือข่ายจริงทั้งหมด8 ตัวแล้วจากนั้นก็จะนำมาออกแบบโมเดลในการทำวิจัยบนเครื่องมือแบบทดสอบเครือข่ายเอ็นเอสทูซึ่งจะออกแบบมาได้โมเดลได้ดังต่อไปนี้

- โมเดลที่1 : การจำลองเหตุการณ์จริงเครือข่ายจริงทั้งหมดในการเชื่อมต่อทั้งหมด 8 โหนดดังรูปที่3-5



รูปที่3-5 แสดงแบบโมเดล 8 โหนดโครงสร้างในการนำไปทดสอบบนเครื่องมือแบบทดสอบเครือข่ายเอ็นเอสทู

เป็นการออกแบบโครงสร้างโมเดลเครือข่ายจุฬาฯหลังจากทำการเก็บข้อมูลปริมาณแทรฟฟิคทั้งเข้าและออกของแต่ละโหนดอุปกรณ์และโครงสร้างการเชื่อมต่อของเครือข่ายซึ่งจะให้บริการรูปแบบวีดีโอตามสั่งจากเครื่องแม่ข่ายวีดีโอซึ่งผู้วิจัยให้อยู่ที่ภาคคอมพิวเตอร์ คณะวิศวกรรมศาสตร์คือโหนดที่19 และเครื่องลูกข่ายวีดีโอคือโหนดที่18ให้อยู่ที่สำนักเทคโนโลยีสารสนเทศสามารถเข้ามาใช้เปิดรับชมดูวีดีโอนการสอนของอาจารย์ที่ภาคคอมพิวเตอร์ได้ทันทีโดยผ่านเครือข่ายการเชื่อมต่อดังรูปที่ 3-5

3.3.1 รูปแบบวิธีการทำโปรแกรมคำสั่งบนการทำงานของเครื่องมือแบบทดสอบเครือข่ายเอ็นเอสทู

เอ็นเอสทูสามารถทำงานได้ใน 2 ลักษณะ คือ

- แบบเชิงโต้ตอบ (Interactive Mode)

เมื่อผู้ใช้กระทำข้อความสั่งใดๆ ผลลัพธ์ก็จะแสดงออกมาในทันที ดังตัวอย่าง

$ ns

% set ns [new Simulator]

O1

% $ns at 1 "puts \"Hello World\""

1

% $ns at 1.5 "exit"

2

% $ns run

Hello World

$

- แบบกลุ่ม (Batch Mode)

ผู้ใช้จะต้องสร้างไฟล์ขึ้นมาโดยในไฟล์นั้นจะมีข้อความสั่งต่างๆที่ใช้ในการทำงานตามที่ผู้ใช้ต้องการดังตัวอย่าง

simple.tcl

set ns [new Simulator]

$ns at 1 "puts \"Hello World\""

$ns at 1.5 "exit"

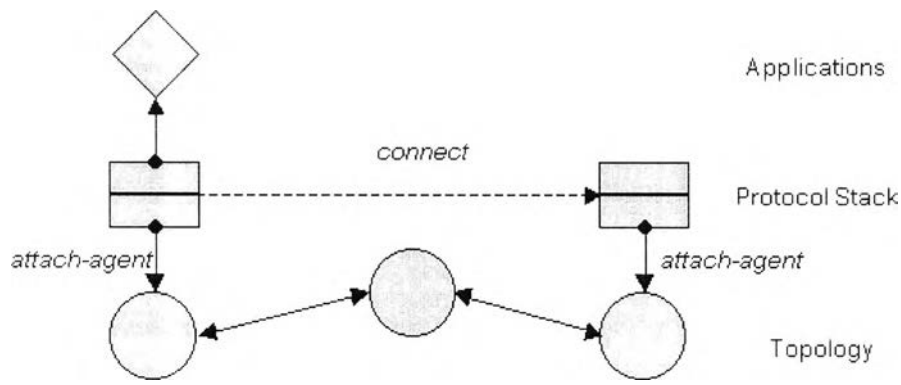
$ns run

$ ns simple.tcl

Hello World

$

โดยส่วนใหญ่การจำลองการทำงานจะทำในแบบกลุ่มเนื่องจากการจำลองการทำงานแต่ละแบบจำเป็นต้องเขียนคำสั่งเพื่อสร้างโครงแบบของการทำงาน การเชื่อมต่อของส่วนประกอบต่างๆ เช่น โหนด ลิงค์ และโพรโทคอล เป็นต้น การกำหนดเวลาตามเหตุการณ์ที่จะจำลองการทำงาน และส่วนอื่นๆ ดังรูปที่3-6 เพื่อใช้ในการจำลองการทำงานนั้นๆ ซึ่งจะสะดวกและกระทำได้ง่ายกว่าการทำงานแบบเชิงโต้ตอบ



รูปที่ 3-6 แสดงการเชื่อมต่อส่วนประกอบต่าง ๆ ของเอ็นเอสทูสำหรับจำลองการทำงาน

สำหรับส่วนประกอบพื้นฐานที่มีอยู่ในเอ็นเอสทู และวิธีการในการเรียกใช้ มีดังนี้

3.3.2 การสร้างตัวกำหนดเหตุการณ์ต่างๆ (Creating Event Scheduler)

- สร้างตัวกำหนดเหตุการณ์

  - *set ns [new Simulator]*

- กำหนดเหตุการณ์

  - *$ns at <time> <event>*

  - *<event>* เป็นคำสั่งของเอ็นเอสทูหรือทีซีแอลที่ถูกต้อง

- เริ่มการทำงาน

  - *$ns run*

3.3.3 การติดตาม (Tracing)

- การติดตามกลุ่มข้อมูลบนลิงค์ทั้งหมด

- *$ns trace-all [open test.out w]*

☐ การติดตามกลุ่มข้อมูลบนแต่ละลิงค์ที่กำหนด

- *$ns trace-queue $n0 $n$tr*

☐ การติดตามกลุ่มข้อมูลบนลิงค์ทั้งหมดตามรูปแบบของนามรุ่น 1 (Nam-1)

- *set nf [open test.nam w]*
- *$ns namtrace-all $nf*

☐ การติดตามกลุ่มข้อมูลบนแต่ละลิงค์ตามรูปแบบของนามรุ่น 1 (Nam-1)

- *$ns namtrace-queue $n0 $n1*

☐ รูปแบบฟอร์แมตของTrace File

- <Event><time><from><to><pkt><size>....<fid><src><dst><seq><attr>

เช่น	+	1	0	2	cbr	210	0	0.0	3.1	0	0
	-	1	0	2	cbr	210	0	0.0	3.1	0	0

3.3.4 การสร้างเครือข่าย (Creating Network)

☐ โหนด

- *set n0 [$ns node]*

- *set n1 [$ns node]*

☐ การเชื่อมโยง

- *$ns duplex-link $n0 $n1 <bandwidth> <delay> <queuetype>*

- *<queuetype>: DropTail, RED, CBQ, FQ, SFQ, DRR*

☐ แลน (LAN)

- *$ns make-lan <nodelist> <bandwidth> <delay> <lltype> <ifqtype> <mactype> <channeltype>*

- *<lltype>: LL*

- *<ifqtype>: Queue/DropTail,*

- *<mactype>: MAC/8023*

- *<channeltype>: Channel*

3.3.5 การใส่ความผิดพลาดต่างๆ (Inserting errors)

- สร้างมอดูลความผิดพลาด

  - *set lossmodule [new ErrorModel]*

  - *$lossmodule set rate 0.01*

  - *$lossmodule unit pkt*

  - *$lossmodule ranvar [new RandomVariable/Uniform]*

  - *$lossmodule drop-target [new Agent/Null]*

- การใส่มอดูลความผิดพลาด

  - *$ns lossmodel $lossmodule $n0 $n1*

3.3.6 การจัดเตรียมการจัดเส้นทาง (Setup Routing)

- ยูนิคาสท์

  - *$ns rtproto <type>*

  - *<type>: Static, Session, DV, cost, multi-path*

- มัลติคาสท์

  - *$ns multicast*

  - *$ns mrtproto <type>*

  - *<type>: CtrMcast, DM, ST, BST*

3.3.7 การสร้างการเชื่อมต่อ (Creating Connection)

- ทรานสปอร์ต (Transports) คือ ทีซีพี และยูดีพี

☐ ยูดีพี

จุดเริ่มต้นและจุดปลายทาง

แบบที่1 - *set usrc [new Agent/UDP]*

- *set udst [new Agent/NULL]*

แบบที่2 - *set udp [new Agent/UDP]*

- *set null [new Agent/Null]*

การติดต่อกันระหว่างโหนดต่างๆ กัน

แบบที่1 - *$ns attach-agent $n0 $usrc*

- *$ns attach-agent $n1 $udst*

- *$ns attach-agent $n0 $udp*

แบบที่2 - *$ns attach-agent $n1 $null*

- *$ns connect $usrc $udst*

- *$ns connect $udp $null*

☐ ทีซีพี

- *set tsrc [new Agent/TCP]*

- *set tdst [new Agent/TCPSink]*

- *set tcp [new Agent/TCP]*

- *set tcpsink [new Agent/TCPSink]*

การติดต่อกันระหว่างโหนดต่างๆ กัน

- *$ns attach-agent $n0 $tsrc*

- *$ns attach-agent $n1 $tdst*

- *$ns attach-agent $n0 $tcp*

- *$ns attach-agent $n1 $tcpsink*

- *$ns connect $tsrc $tdst*
- *$ns connect $tcp $tcpsink*

3.3.8 การสร้างแทรฟฟิค (Creating Traffic)

- บนยูดีพี
  - ซีบีอาร์ (CBR)
    - *set src [new Application/Traffic/CBR]*
    - *$cbr0 attach-agent $usrc*
    - *$ns at 0.5 "$cbr0 start"*
    - *$ns at 4.5 "$cbr0 stop"*
  - วีบีอาร์ (VBR)
    - *set src [new Application/Traffic/VBR]*
    - *$vbr0 attach-agent $usrc*
    - *$ns at 0.5 "$vbr0 start"*
    - *$ns at 4.5 "$vbr0 stop"*
  - เอ็กซ์โพเนนเชียล (Exponential) หรือ พาเรโต (Pareto)
    - *set src [new Application/Traffic/Exponential]*
    - *set src [new Application/Traffic/Pareto]*
- บนทีซีพี
  - เอฟทีพี (FTP)
    - *set ftp [new Application/FTP]*
    - *$ftp attach-agent $tcp*
    - *$ftp attach-agent $tsrc*
    - *ns at <time> "$ftp start"*
  - *เทลเน็ต (Telnet)*
    - *set telnet [new Application/Telnet]*

- *$telnet attach-agent $tcp*

- ใช้ข้อมูลจริง (Trace Driven)

  - ใช้ข้อมูลจริง

    - *set tfile [new Tracefile]*

    - *$tfile filename <file> ;# <file> : Binary format*

      : Inter-packet time (msec) and packet size(byte)

    - *set src [new Application/Traffic/Trace]*

    - *$src attach-tracefile $tfile*

3.3.9 การควบคุมพารามิเตอร์อ๊อบเจคต่างๆ ในแทรฟฟิค,ซีบีอาร์,วีบีอาร์และระดับงานประยุกต์

- *$set cbr0 [new Application/Traffic/CBR]*
- *$cbr0 set packetSize_500*
- *$cbr0 set interval_0.005*

3.3.10 การจำลองการทำงานในระดับงานประยุกต์ (Application-Level Simulation)

- ลักษณะสำคัญของการจำลองการทำงานในระดับงานประยุกต์

  - ทำงานอยู่บนโพรโทคอลนำส่ง (Transport protocol)

  - ใช้ส่งผ่านข้อมูลผู้ใช้ เช่น ส่วนหัวของเอชทีทีพี (HTTP header)

- วิธีการที่ใช้ส่งผ่านข้อมูล

  - บนทีซีพี ใช้ Agent/Message

  - บนยูดีพี ใช้ Application/TcpApp

- เว็บแคช (Web Cache)

  - *set cache [new HttpCache $ns $node]*

  - *$cache connect $server*

- เว็บไคลแอนท์ (Web Client)

- *set client [new Http/Client $ns $node]*

- *$client connect $server*

- *$client set-page-generator $pgp*

- *$client start-session $cache $server*

☐ เว็บเซิร์ฟเวอร์ (Web Server)

- *set server [new Http/Server $ns $node]*

- *$server set-page-generator $pgp*

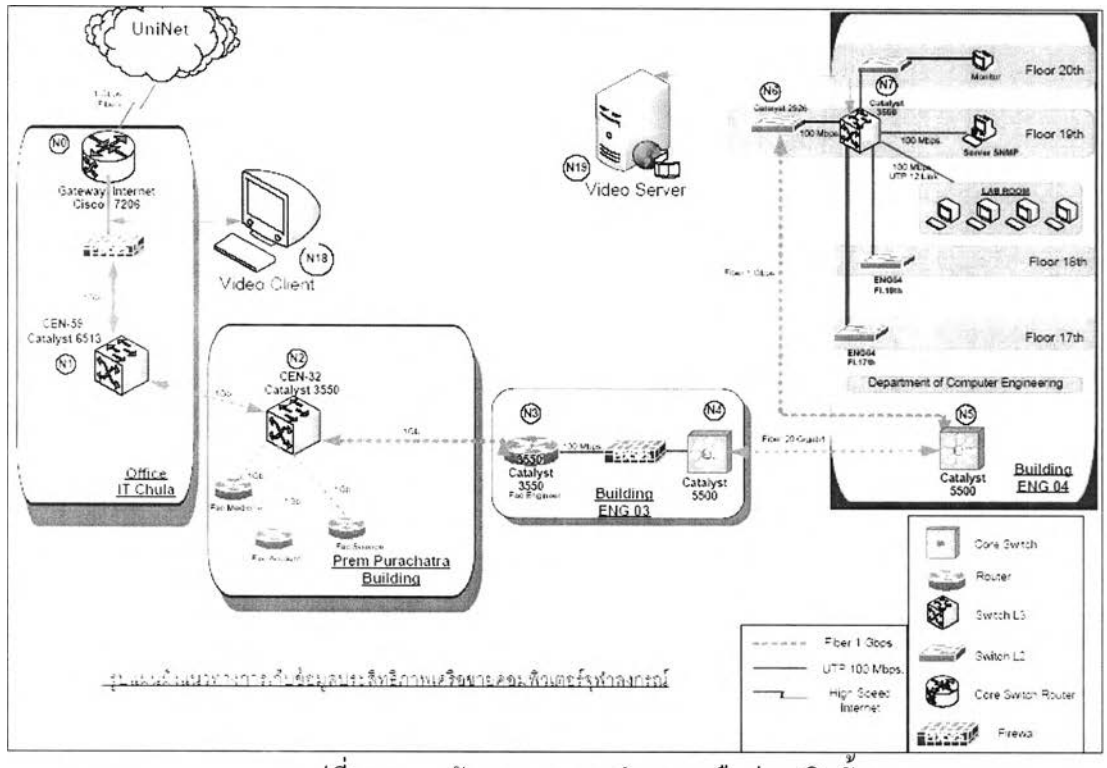
3.3.11 การปิดขั้นตอนการทำงานของคำสั่ง (Finishing up the script)**และสั่งทำงาน Nam**

```
proc finish { }   {
global ns nf
$ns flush-trace
close $nf
exec nam test.nam &
exit0  }
$ns at 5.0 "finish"
```

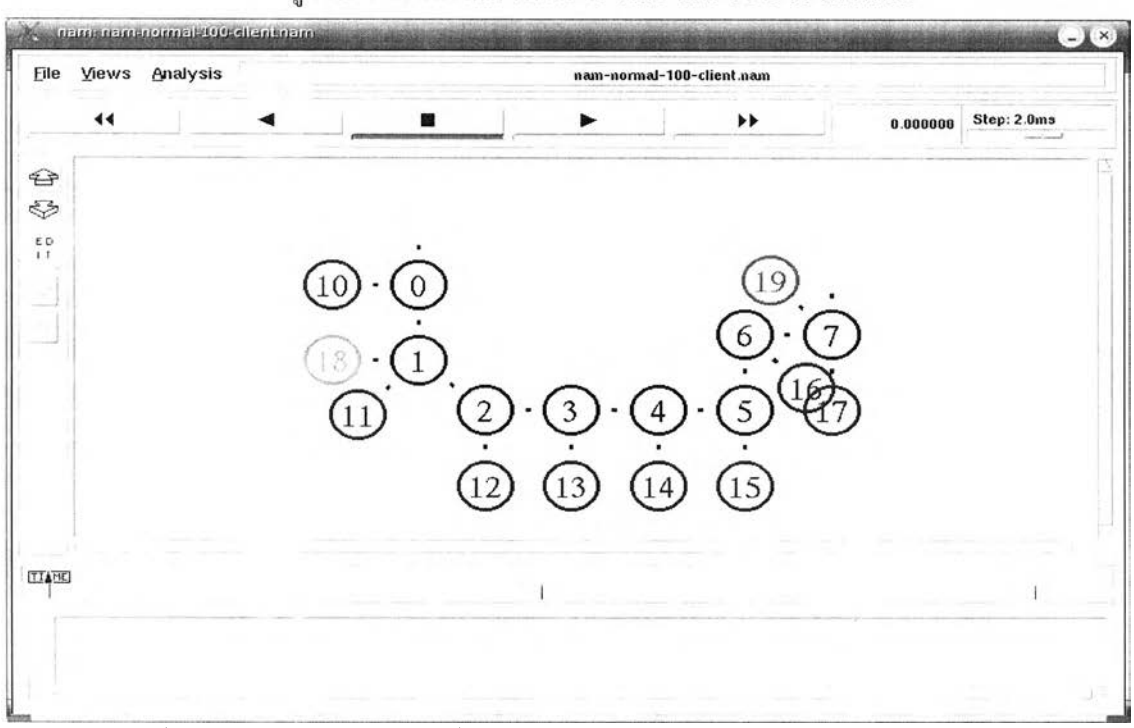
I 9 3 8 4 1 8 0

3.4 การพัฒนาคำสั่งบนการจำลองเหตุการณ์เครือข่ายจริงทั้งหมด

การจำลองเหตุการณ์ทำได้โดยการกำหนดโครงสร้างเครือข่ายที่สนใจศึกษาดังนั้นเราจึงแทนโหนดด้วยอุปกรณ์ต่างๆ ดังรูปที่ 3-7



รูปที่3-7 แผนผังแนวทางการจำลองเครือข่ายจริงทั้งหมด



รูปที่ 3-8 รูปการทำงานของเครื่องมือแบบทสอบเครือข่ายเอ็นเอสทูในโมเดลการจำลองเหตุการณ์เครือข่ายจริงทั้งหมด

ตารางที่ 3-4 กำหนดชื่อตัวแปรโหนดต่างๆ ในการทำวิจัยเครือข่ายจุฬาทั้งหมด

โหนด	อุปกรณ์ในการทำวิจัยเครือข่ายจุฬาฯ
N19	เครื่องแม่ข่ายให้บริการวีดีโอ
N7	อุปกรณ์Catalyst 3550 (Fl.19)
N6	อุปกรณ์Catalyst 2926 (Fl.19)
N5	อุปกรณ์Catalyst 5500 (ENG04)
N4	อุปกรณ์Catalyst 5500 (ENG03)
N3	อุปกรณ์Catalyst 3550 (Fac.ENG)
N2	อุปกรณ์Catalyst 3550 (CEN-32)
N1	อุปกรณ์Catalyst 6513 (CEN-59)
N0	อุปกรณ์GW Router 7206 (CEN-59)
N18	เครื่องลูกข่ายรับชมวีดีโอ
N10	โหนดสร้างข้อมูลของอุปกรณ์หมายเลข N0
N11	โหนดสร้างข้อมูลของอุปกรณ์หมายเลข N1
N12	โหนดสร้างข้อมูลของอุปกรณ์หมายเลข N2
N13	โหนดสร้างข้อมูลของอุปกรณ์หมายเลข N3
N14	โหนดสร้างข้อมูลของอุปกรณ์หมายเลข N4
N15	โหนดสร้างข้อมูลของอุปกรณ์หมายเลข N5
N16	โหนดสร้างข้อมูลของอุปกรณ์หมายเลข N6
N17	โหนดสร้างข้อมูลของอุปกรณ์หมายเลข N7

การจำลองปริมาณข้อมูลระหว่างอุปกรณ์คู่หนึ่งในการรับและส่งกลุ่มข้อมูลตามตารางที่ 3-5 ซึ่งนำข้อมูลสูงสูดด้านขาเข้าและขาออกของแต่ละโหนดในการเก็บข้อมูลจากโปรแกรมเอ็มอาร์ทีจีเป็นรายสัปดาห์มาใช้ ซึ่งผู้วิจัยต้องการให้ทดสอบการจัดการคุณภาพบริการวีดีโอตามสั่งผ่านปริมาณแทรฟฟิคสูงที่สุดที่เกิดขึ้นจริงในเครือข่ายจุฬาฯ ซึ่งจะทำให้ผลการจำลองมีผลใกล้เคียงความจริงมากที่สุดในการทำโมเดลแบบจำลองขึ้นมาระหว่างเครือข่ายภาควิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์และสำนักเทคโนโลยีสารสนเทศ

ตารางที่ 3-5 แสดงผลจากการเก็บข้อมูลจากอุปกรณ์สวิตช์บนเครือข่ายจริงทั้งหมด

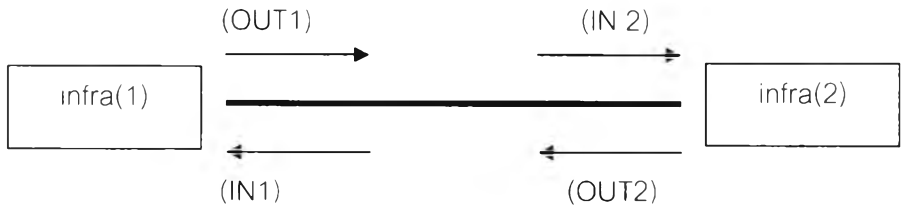
โหนด	อุปกรณ์ในเครือข่ายจุฬาฯ	หมายเลขช่องการติดต่อเข้า(In) และออก(Out)	ค่าสูงสุดขาเข้า(Mbps)	ค่าสูงสุดขาออก(Mbps)
N7	Catalyst 3550 (Fl.19)	In : FastEthernet 0/43	0.239	0.033
		Out : FastEthernet 0/36	2.859	0.492
N6	Catalyst 2926 (Fl.19)	In : FastEthernet ½	0.492	2.859
		Out : FastEthernet 1/1	2.860	0.493
N5	Catalyst 5500 (ENG04)	In : FastEthernet9/8	27.000	33.200
		Out : FastEthernet1/2	40.500	32.400
N4	Catalyst 5500 (ENG03)	In : FastEthernet1/2	56.700	47.900
		Out : FastEthenet3/6	37.700	45.600
N3	Catalyst 3550 (Fac.ENG)	In : FastEthernet 0/6	10.600	19.100
		Out : GigabitEthernet 0/1	15.300	15.400
N2	Catalyst 3550 (CEN-32)	In : GigabitEthernet0/5	39.000	61.800
		Out : GigabitEthernet0/1	**154.000**	**145.600**
N1	Catalyst 6513 (CEN-59)	In : GigabitEthernet 1/22	**145.800**	**153.500**
		Out : GigabitEthernet 1/24	238.900	205.000
N0	GW Router 7206 (CEN-59)	In : GigabitEthernet0/2	202.200	243.500
		Out : GigabitEthernet0/1	237.300	201.900

3.4.1 การสร้างโหนด

อุปกรณ์แต่ละตัวจะถูกแทนที่ด้วยตัวแปรอาเรย์ชื่อ infra ดังสคริปต์

```
For {set i 0} {$i < 8 } {incr i}
{
set infra($i) [$ns node]
}
```

ระหว่างอุปกรณ์คู่หนึ่ง เราทำการเฉลี่ยปริมาณข้อมูลตามข้อมูลจริงดังนี้ เช่น จากสถิติปริมาณข้อมูลที่เก็บได้ระหว่าง infra(1)และ infra(2)



รูปที่ 3-9 แสดงรูปแบบการไหลของกลุ่มข้อมูลเข้าออกของอุปกรณ์สวิตช์ระหว่างโหนดแต่ละโหนด

ดังนั้น ปริมาณข้อมูลเฉลี่ยที่วิ่งทั้งหมดระหว่าง infra(1) กับ infra(2) คือ

$$T1 = (OUT1 + IN2) /2$$

$$T2 = (OUT2 + IN1)/2$$

แทนค่าได้คือ

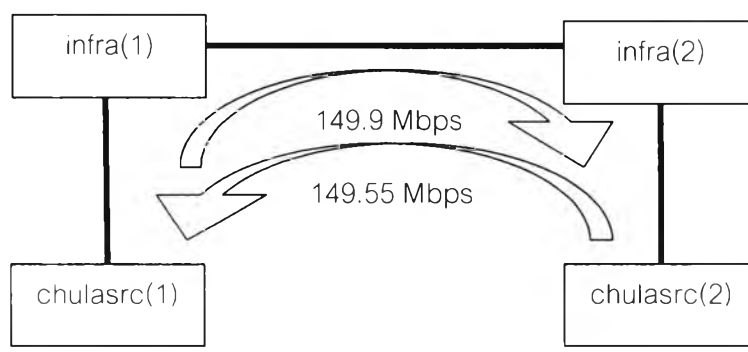
$$T_{1-2} = (145.8 + 154) / 2$$

$$= 149.9 \text{ Mbps.}$$

$$T_{2-1} = (145.6 + 153.5) / 2$$

$$= 149.55 \text{ Mbps.}$$

ในการจำลองเหตุการณ์นั้นการทำให้เครือข่ายมีปริมาณข้อมูลตามที่คำนวณ ผู้วิจัยจึงทำการเพิ่มโหนดสร้างข้อมูลที่เป็นผู้ส่งและผู้รับคือที่โหนด chulasrc(1)และchulasrc(2) โดยต่อเข้ากับโหนดที่เป็นโครงสร้างพื้นฐานทุกตัวและส่งข้อมูลไปมาหากันตามที่คำนวน



รูปที่ 3-10 แสดงการไหลของแทรฟฟิครบกวนระหว่างโหนด

3.4.2 การสร้างข้อมูล

ในการจำลองเหตุการณ์ด้วยเอ็นเอสทู ผู้วิจัยได้สร้างโหนดที่จะทำการสร้างข้อมูลขึ้นโดยให้ชื่อตัวแปรอาเรย์ว่า chulasrc ดังนี้

```
For {set i 0} {$i < 8 } {incr i} {
        set chulasrc($i) [$ns node]
        $chulasrc($i) color "blue"

        set chulasrc_null($i) [new Agent/Null]
        $ns attach-agent $chulasrc($i) $chulasrc_null($i)
}
```

โหนดที่จะทำการส่งข้อมูลจะประกอบไปด้วยการเชื่อมต่อทรานสปอร์ตเลเยอร์(Transport layer) และแอพลิเคชั่นเลเยอร์(Application Layer)เข้าด้วยกันซึ่งในที่นี้คือยูดีพี และตัวสร้างข้อมูลแบบ Exponential ตามลำดับ ส่วนโหนดที่จะทำการรับข้อมูลต้องสร้าง NULL obj และแนบเข้ากับโหนดเช่นกัน

จากรูปที่3-11 โหนด chulasrc(1) จะต้องทำการส่งข้อมูลในทิศทางเข้าสู่เครือข่ายของมหาวิทยาลัย ดังนั้นเราจึงตั้งชื่อตัวแปร ยูดีพี ที่มีทิศทางเข้าสู่มหาวิทยาลัยให้ลงท้ายด้วย IN และในทางกลับกันโหนด chulasrc(2) จะต้องทำการส่งข้อมูลในทิศทางออกสู่อินเตอร์เน็ตเราจึงตั้งชื่อตัวแปร ยูดีพีให้ลงท้ายด้วย OUT ดังตัวอย่างการสร้างตัวแปร ยูดีพี (ตัวแปร $i คือโหนดที่ตัวแปร ยูดีพี จะแนบเข้าด้วยกันในที่นี้คือ chulasrc(1) และ chulasrc(2))

```
set chulasrc_udp_in($i) [new Agent/UDP]
set chulasrc_udp_out($i) [new Agent/UDP]
```

วิธีการแนบ ยูดีพี

โหนด chulasrc(1) จะต้องทำหน้าที่ในการส่งข้อมูลทั้งเข้าและออก จึงต้องแนบทั้ง UDP IN และ UDP OUT เข้าที่โหนดเดียวกัน

```
$ns attach-agent $chulasrc($i) $chulasrc_udp_in($i)
$ns attach-agent $chulasrc($i) $chulasrc_udp_out($i)
```

จากนั้น ทำการสร้างตัวแปรสร้างข้อมูล และเลือกใช้ Exponential ดังตัวอย่าง

```
set chulasrc_exp_in($i) [new Application/Traffic/Exponential]
set chulasrc_exp_out($i) [new Application/Traffic/Exponential]
$chulasrc_exp_in($i) set packetSize_ 1000
$chulasrc_exp_out($i) set packetSize_ 1000
$chulasrc_exp_in($i) set burst-time 7.5s
$chulasrc_exp_out($i) set burst-time 7.5s
```

การตั้งค่า packetSize เท่ากับ 1000 ไบต์เนื่องจากเป็นค่าสูงสุดที่ใช้ในเครือข่ายระบบสายส่งเชื่อมต่อถึงกันส่วนการตั้งค่า burst-time เท่ากับ 7.5 วินาทีเนื่องจากผู้วิจัยทำการทดสอบภายใต้ข้อมูลที่เก็บมาเป็นค่าสูงสุดและใช้ในรูปแบบเอ็กซ์โพเนียนเชียล ซึ่งสมมุติฐานว่าภายในเครือข่ายจุฬา 24 ชั่วโมงมีช่วงที่มีปริมาณแทรฟฟิคสูงสุดประมาณ 3 ชั่วโมงดังนั้นภายใต้การทดลอง 60 วินาทีจะมีช่วงปริมาณแทรฟฟิคขึ้นสูงสุดนานประมาณ 7.5 วินาที

การแนบ ตัวสร้างข้อมูล กับ ยูดีพี

```
$chulasrc_exp_in($i) attach-agent $chulasrc_udp_in($i)
$chulasrc_exp_out($i) attach-agent $chulasrc_udp_out($i)
$chulasrc_exp_out(1) set rate_ 224.3 M
$chulasrc_exp_in(1) set rate_ 149.9 M
$chulasrc_exp_out(2) set rate_ 149.55 M
$chulasrc_exp_in(2) set rate_ 27.2 M
```

3.4.3 การสร้างเอ็มเพ็ก4 แทรฟฟิค [15]

เริ่มแรกเราสร้างโหนดเพื่อส่งข้อมูลแบบ เอ็มเพ็ก4 ชื่อ mpeg_server และโหนดที่รับข้อมูลของเอ็มเพ็ก4 ชื่อ mpeg_client

```
#Create Video Client
set mpeg_client [$ns node]
set mpeg_client_null [new Agent/Null]
$ns attach-agent $mpeg_client $mpeg_client_null
#Create Video Server
set mpeg_server [$ns node]
set mpeg_server_udp [new Agent/UDP]
```

3.4.4 สร้างตัวสร้างข้อมูลเอ็มเพ็ก4 [15]

ส่งกลุ่มข้อมูลด้วยอัตรา 30 เฟรมต่อวินาที

```
#MPEG4 Traffic
set mpeg_server_mpeg4 [new Application/Traffic/MPEG4]
$mpeg_server_mpeg4 attach-agent $mpeg_server_udp
$mpeg_server_mpeg4 set initialSeed_ 0.4
$mpeg_server_mpeg4 set rate_Factor_ 5
$ns connect $mpeg_server_udp $mpeg_client_null
```

initialSeed_ กำหนดการสุ่มค่าการส่งเฟรมกลุ่มข้อมูลโดยกำหนดให้ค่าเท่ากับ 0.4 เป็นค่าเริ่มต้น

rate_factor_ ในคำนวณขนาดของเฟรมมีหน่วยเป็นไบต์จากสมการ

3.4.5 การจำลองโพรโทคอลเอ็มพีแอลเอส [11][12]

การทดสอบโพรโทคอลเอ็มพีแอลเอสนั้นผู้วิจัยได้ใช้ส่วนต่อขยายของแพ็ทช์โพรโตคอลเอ็มพีแอลเอส รุ่น2.27 มาเป็นพื้นฐานในการทำทดสอบแบบจำลองบนโพรโตคอลเอ็มพีแอลเอสต่อไป

การกำหนดความสำคัญของกลุ่มข้อมูลเอ็มเพ็ก4 ทำได้โดยการกำหนดความสำคัญของการไหล (Flow) โดยแบ่งความสำคัญออกเป็น 2 ระดับคือ

ระดับปกติ สำหรับการไหลของข้อมูลจำลองที่มีอยู่ในเครือข่ายปัจจุบันทั้งหมด

ระดับสูงกว่าปกติ สำหรับการไหลของข้อมูลเอ็มเพ็ก4 ซึ่งมีความสำคัญมากกว่า

ในการจำลองเหตุการณ์ โหนดที่เป็นอุปกรณ์จะถูกสร้างเพื่อให้รองรับการทำงานของ Label Switching และบันทึกโมดูลโพรโทคอลเอ็มพีแอลเอสของโหนดใส่อาเรย์ชื่อ infra_LSR

```
for {set i 0} {$i < 8 } {incr i} {
        set infra($i) [$ns mpls-node]
        set infra_LSR($i) [eval $infra($i) get-module "MPLS"]
}
```

การเชื่อมต่อระหว่างโหนดอุปกรณ์จะต้องใช้ CBQ เพื่อโหนดจะสามารถจัดลำดับกลุ่มข้อมูลตามลำดับความสำคัญได้ ส่วนการเชื่อมต่อระหว่างโหนดอุปกรณ์และโหนดสร้างข้อมูลยังคงใช้ DropTail เช่นเดิม

ส่วนการกำหนดโพรพาเกชั่นดีเลย์นั้นตั้งค่าเท่ากับ 1 ms. ทั้งรูปแบบ CBQ และ DropTail รูปแบบการบริการทั้ง CBQ และ DropTail จะต้องใช้เป็นแบบ ยูดีพี เท่านั้นในการรันแทรฟฟิครบกวนในเครือข่ายจำลองตลอดเวลาทั้งหมด 8 โหนด

```
$ns duplex-link $infra(0) $infra(1) 1Gb 1ms CBQ
```

```
$ns duplex-link $infra(0) $chulasrc(0) 1Gb 1ms DropTail
$ns duplex-link $infra(1) $chulasrc(1) 1Gb 1ms DropTail
```

การกำหนดคิวบัฟเฟอร์ไว้ที่แต่ละโหนดในการเก็บคิวบัฟเฟอร์ได้มีขนาดเท่ากับ 28621 ไบต์ ซึ่งเป็นค่าที่นำมาจากข้อมูลอุปกรณ์สวิตช์ของซิสโก้

```
[[$ns link $infra(7) $chulasrc(7)] queue] set limit_ 28621
```

การกำหนดให้โหนดอุปกรณ์ทุกโหนดใช้โพรโทคอล LDP

```
$ns configure-ldp-on-all-mpls-nodes
```

การกำหนด LDP คือการกำหนด Label Switching Message กันมันจะทำการแลกเปลี่ยนป้ายกันระหว่างโหนดต่อโหนดไปเรื่อยตามเส้นทางที่ถูกกำหนดไว้โดยอัตโนมัติ
กำหนดคุณสมบัติรูปแบบการให้บริการ ในที่นี้คือ Simple Best Effort Service , SBTS และกำหนดโหนดให้บริการกลุ่มข้อมูลของโพรโทคอล LDP ในรูปแบบ SBTS

```
$ns cfg-cbq-for-SBTS 10 DropTail 500000.000000 0.000000 auto 0.000000
$ns bind-ldp-to-SBTS
$ns enable-control-driven
```

ในการสร้างข้อมูลต้องทำการกำหนดหมายเลขการไหล และกำหนดหมายเลขการไหลว่าจะมีรูปแบบการให้บริการแบบ SBTS โดยยึดเหนี่ยวหมายเลขการไหลเข้ากับ SBTS ซึ่งการไหลของข้อมูลในเครือข่ายจะมีหมายเลข 10-17 ในทิศทางไหลเข้าเครือข่าย(IN) และมีหมายเลข 20-27ในทิศทางไหลออกไปยังอินเตอร์เน็ต(OUT) ส่วนการไหลของ เอ็มเพ็ก4 กำหนดให้มีหมายเลขตั้งแต่ 9000 ขึ้นไปโดยขึ้นอยู่กับจำนวนผู้รับชมซึ่งในการทดลองจะใช้จำนวน 1, 10 และ100 ไคลเอ็นท์ในการดึงข้อมูลเอ็มเพ็ก4

```
set chulasrc_udp_in($i) [new Agent/UDP]
set chulasrc_udp_out($i) [new Agent/UDP]
```

```
$ns attach-agent $chulasrc($i) $chulasrc_udp_in($i)
$ns attach-agent $chulasrc($i) $chulasrc_udp_out($i)

$chulasrc_udp_in($i) set fid_ 1$i
$chulasrc_udp_out($i) set fid_ 2$i

$ns bind-flowid-to-SBTS 1$i
$ns bind-flowid-to-SBTS 2$i
```

การกำหนดระดับของกลุ่มข้อมูลนั้นต้องกำหนดที่โมดูลโพรโทคอลเอ็มพีแอลเอสของแต่ละโหนดโดยกำหนดให้การไหลของข้อมูลในเครือข่ายมีระดับคือระดับ 1 ที่ต่ำกว่า การไหลของข้อมูลเอ็มเพ็ก 4 มีระดับสูงกว่าคือระดับ 7 [10]

```
for {set j 0} { $j < 8 } { incr j} {
        $infra_LSR($j) set-flow-prio 1 40 1
        $infra_LSR($j) set-flow-prio 1 50 1
        $infra_LSR($j) set-flow-prio 1 [expr 10+$j] 1
        $infra_LSR($j) set-flow-prio 1 [expr 20+$j] 1

        $infra_LSR($j) set-flow-prio 1 9000 7
}
```

3.4.6 การจำลอง โพรโทคอลอาร์เอสวีพี [12][16]

เนื่องจากมีการใช้แพ็ทช์โพรโทคอลอาร์เอสวีพีทีอี รุ่น 2.27 [28] ทำงานควบคู่กับโพรโทคอลเอ็มพีแอลเอสดังนั้นการสร้างโหนดอุปกรณ์จึงเหมือนกัน แต่แทนที่จะกำหนดให้ทุกโหนดอุปกรณ์ใช้โพรโทคอล LDP จึงเปลี่ยนมาใช้โพรโทคอลอาร์เอสวีพีทีอีแทนซึ่งยังคงใช้ LDP ในการเปลี่ยนป้ายชื่อเช่นเดิม ในงานวิจัยนี้ได้ใช้ส่วนต่อขยายของโพรโตคอลอาร์เอสวีพีทีอีรุ่น 2.27 [17] จะมีคุณลักษณะฟังก์ชันการทำงานเหมือนกับโพรโตคอลอาร์เอสวีพีมากในการทำการจองแบนด์วิดท์ของกลุ่มข้อมูลไปตามแต่ละโหนดการเชื่อมต่อ ดังนั้นงานวิจัยนี้จะอ้างอิงพื้นฐานการใช้ส่วนขยายของโพรโตคอลอาร์เอสวีพีทีอีรุ่น 2.27 เป็นแบบจำลองทดสอบโพรโทคอลอาร์เอสวีพีต่อไป

```
$ns configure-rsvpte-on-all-mpls-nodes
```

การกำหนดระดับความสำคัญตามประเภทของข้อมูลก็ต่างไปจากโพรโทคอลเอ็มพีแอลเอสกล่าวคือ โหนดมิได้กำหนดความสำคัญของการไหลตามหมายเลขการไหล แต่กำหนดตามรูปแบบการให้บริการ ซึ่งในที่นี้ ให้ข้อมูลปัจจุบันในเครือข่ายมีรูปแบบ SBTS และข้อมูล เอ็มเพ็ก4 มีรูปแบบ HBTS ดังนั้นจึงต้องกำหนดคุณสมบัติรูปแบบการให้บริการทั้ง SBTS และ HBTS และให้ข้อความควบคุมของโพรโทคอลอาร์เอสวีพีเป็นแบบ SBTS

```
$ns cfg-cbq-for-SBTS 10 DropTail 500000.000000 0.000000 auto 0.000000
$ns cfg-cbq-for-HBTS 10 DropTail 500000.000000 0.000000 auto 0.000000
$ns bind-rsvpte-to-SBTS
```

สร้างช่วงการสื่อสาร (session) ที่อุปกรณ์ต้นทาง ไปยังอุปกรณ์ปลายทาง

```
set ses(1) [$infra_LSR(7) session $infra(1) 1]
```

ให้โมดูลโพรโทคอลอาร์เอสวีพีของอุปกรณ์ต้นทาง ส่ง Path message ไปตามทางโหนด 6_5_4_3_2_1 ซึ่งกำหนดหมายเลขการส่งกลุ่มข้อมูลด้วยป้ายชื่อ (label swithching path id)เท่ากับ 2000 เพื่อสำรองความกว้างของช่องสัญญาณจำนวน 50 Mbpsคือผู้วิจัยกำหนดให้ทำการจองทรัพยากรเครือข่ายเป็นครึ่งหนึ่งขนาดแบนวิดท์ที่ต่ำสุดในโมเดลที่1คือความเร็ว100Mbps.พร้อมระบุระดับความสำคัญของรูปแบบการให้บริการ SBTS เป็นระดับ 1 และ HBTS เป็นระดับ 7

```
$ns at 1.000000 "$infra_LSR(7) PATH-resv-er $ses(1) 50Mb 50 50 $infra(1) 2000 1 7 6_5_4_3_2_1"
```

กำหนดให้หมายเลขการไหลของข้อมูล เอ็มเพ็ก4 ยึดเหนี่ยวเข้ากับหมายเลขการส่งกลุ่มข้อมูลด้วยป้ายชื่อ เพื่อให้การไหลของข้อมูล เอ็มเพ็ก4 ไหลไปตามเส้นทางที่สำรองไว้ ซึ่งต้องทำการกำหนดไปยังทุกอุปกรณ์

```
for {set j 0} { $j < 8 } { incr j} {
        $ns at 1.2 "$infra_LSR($j) bind-flow-erlsp 1 9000 2000"
}
```

3.4.7 การคำนวณอัตราการสูญหาย, ช่องสัญญาณที่ส่งได้ และ ค่าเวลาหน่วง [14]

คำนวณจากการวิเคราะห์ข้อมูลดิบเทรสไฟล์(Trace File) ซึ่งได้มาจากจากการประมวลผลเอ็นเอสทู ที่ได้จากการจำลองเหตุการณ์ในแต่ละรูปแบบ และการวิเคราะห์ใช้ gawk script ซึ่งมีความสามารถในการอ่านไฟล์, แบ่งคอลัมภ์ และประมวลผลแต่ละคอลัมภ์ ทำให้ง่ายต่อการเขียนโปรแกรมวิเคราะห์ค่าที่สนใจในการดึงค่ามาจัดเก็บแล้วทำการคำนวณตามฟังก์ชันการทำงานของสคริปต์นั้นซึ่งผู้วิจัยสนใจการหาค่าอัตราการสูญหาย ช่องสัญญาณที่ส่งได้ และค่าเวลาหน่วง ซึ่งนำใช้วิจัยในการวัดคุณภาพระบบวีดีโอบนเครือข่ายจุฬาฯต่อไปมีรายละเอียดดังนี้

3.4.7.1 อัตราการสูญหาย

เป็นการคำนวณกลุ่มข้อมูลของ Flow หนึ่งๆที่ถูกละทิ้งในลักษณะปลายทางถึงปลายทาง กล่าวคือเป็นจำนวนกลุ่มข้อมูลเอ็มเพ็ก4 ที่ถูกละทิ้งตามอุปกรณ์ต่างๆตลอดเส้นทางรวมกัน และนับเป็นต่อวินาที

วิธีการคำนวณอัตราการสูญหายแบบจุดต่อจุด

1. กำหนดตัวแปรและให้ค่าเริ่มต้นในบรรทัดที่ 2-6 ซึ่งตัวแปรแต่ละตัวมีความหมายดังนี้

- fsDrops คือการนับเฟรมของเอ็มเพ็ก4 ทั้งหมดที่ถูกส่งออกมาตามหมายเลขการไหลตั้งแต่หมายเลข 9000 ว่ามีการสูญหายไปทั้งหมดเท่าไรภายใต้เวลา 60 วินาที
- droprate คือการนับเฟรมของเอ็มเพ็ก4 ที่ถูกส่งออกมาตามหมายเลขการไหลตั้งแต่หมายเลข 9000 ว่ามีการสูญหายไปทั้งหมดเท่าไรภายในหนึ่งวินาที
- numFs คือการนับจำนวนกลุ่มข้อมูลของเอ็มเพ็ก4 การส่งจากโหนด 19 ไปยังโหนด7 ภายใต้เวลาทั้งหมดคือ 60 วินาที
- sentrate คือการนับจำนวนกลุ่มข้อมูลของเอ็มเพ็ก4การส่งจากโหนด 19 ไปยังโหนด7 ภายใต้ต่อหนึ่งวินาที
- action = "+" คือการเข้าคิว(Enqueue)แถวคอย
- action = "-" คือการออกจากคิว(Dequeue)แถวคอย
- action = "d" คือพารามิเตอร์ที่เป็นตัวบ่งชี้ว่ามีกลุ่มข้อมูลสูญหายไปในแต่ละกลุ่มข้อมูลไอดีที่ส่งออกมาในข้อมูลดิบเทรสไฟล์(Trace File)
- stepคือเป็นการตั้งช่วงเวลาในการเก็บข้อมูลจากข้อมูลดิบเทรสไฟล์(TraceFile)ให้มีความสัมพันธ์กับเวลาในการคำนวณเป็นวินาทีซึ่งช่วงเวลาในข้อมูลดิบเทรสไฟล์ละเอียดมากเป็นมิลลิวินาที จึงตั้งช่วงในการนับ Step เริ่มแรกให้เป็น 1 วินาทีก่อนจากนั้นให้ทำการเก็บค่าค่าตัวแปร numFs กับ sentrate และfsDrops กับ droprate ซึ่งแบ่งเป็นคนละลูปการทำงาน จนครบหนึ่งวินาทีแรกแล้วไปทำการคำนวณสูตรอัตราการสูญหายต่อไป จากนั้นแล้วทำการเพิ่มเวลาขึ้นไปทีละหนึ่งวินาทีเรื่อยๆในช่วงเวลา 60 วินาที แล้ววนกลับมาเช็คเวลาในลูป step ต่อไปของการเช็คช่วงเวลานั้นก็จะทำการเก็บค่าตัวแปร numFs กับ sentrate และfsDrops กับ droprate อีกแบบนี้ไปเรื่อยๆจนครบ 60 วินาที

2. บรรทัดที่ 9-19 เป็นการแยกคอลัมภ์ของข้อมูลดิบเทรสไฟล์และบันทึกค่าลงตัวแปร
3. บันทึกปริมาณกลุ่มข้อมูลของเอ็มเพ็ก4 ที่ถูกส่งออกมาจากโหนด 19 ไปยังโหนด 7 ทั้งแบบโดยรวมและต่อวินาที บรรทัดที่ 20-24
4. นับปริมาณกลุ่มข้อมูลของ เอ็มเพ็ก4 ที่ถูกละทิ้ง (หมายเลขการไหล 9000)ทั้งแบบโดยรวมและต่อวินาที บรรทัดที่ 26-31
5. เมื่อครบทุกๆ 1 วินาทีให้ทำการคำนวณอัตราการสูญหายจากสูตร droprate*100/sentrate ซึ่งจะมีหน่วยเป็นเปอร์เซ็นต์และนับเวลารวมทั้งหมด 60 วินาทีคำนวณอัตรการสูญหายทั้งหมดจากสูตร fsDrops*100/numFs ซึ่งมีหน่วยเป็นแพ็คเก็ตต่อวินาที แสดงผล บรรทัดที่ 32-38

```
1   BEGIN {
2           fsDrops = 0;
3           droprate = 0;
4           numFs = 0;
5           sentrate = 0;
6           step = 1;
7   }
8   {
9           action = $1;
10          time = $2;
11          from = $3;
12          to = $4;
13          type = $5;
14          pktsize = $6;
15          flow_id = $8;
16          src = $9;
17          dst = $10;
18          seq_no = $11;
19          packet_id = $12;
20      if (from==19 && to==7 && action == "+") {
21                  numFs++;
```

```
22                      if (time <= step) {
23                              sentrate++;
24                      }
25      }
26              if (flow_id >=9000 && action == "d") {
27                      fsDrops++;
28                      if (time <= step) {
29                              droprate++;
30                      }
31              }
32              if (time > step) {
33      printf("time %f Sent rate %d pkt/s (all %d) Drop rate %d pkt/s (all %d) Percent %f /sec  %f all\n",
34       step-1,sentrate,numFs, droprate,fsDrops,droprate*100/sentrate,fsDrops*100/numFs);
35                              droprate = 0;
36                              sentrate = 0;
37                              ++step;
38              }
39      }
40       END {
41              printf("number of packets sent:%d lost:%d\n", numFs, fsDrops);
42              }
```

3.4.7.2 ปริมาณข้อมูลที่สามารถรองรับได้หรือช่องสัญญาณที่ส่งได้

คือปริมาณข้อมูลที่ผู้ชมได้รับจริงมีหน่วยเป็นเมกกะบิตต่อวินาที(Mbps) ทำได้โดยนับปริมาณข้อมูลที่ได้รับซึ่งมีหน่วยเป็นไบต์ แล้วนำมาคำนวณเป็นเมกกะบิตต่อวินาที อีกครั้ง

วิธีการคำนวณปริมาณข้อมูลที่สามารถรองรับได้

1. กำหนดตัวแปรและให้ค่าเริ่มต้นในบรรทัดที่ 3-5

○ Step คือเป็นการตั้งช่วงเวลาในการเก็บข้อมูลจากจากข้อมูลดิบเทรสไฟล์(Trace File)ให้มีความสัมพันธ์กับเวลาในการคำนวณเป็นวินาทีซึ่งช่วงเวลาในข้อมูลดิบเทรสไฟล์ละเอียดมากเป็นมิลลิวินาที จึงตั้งช่วงในการนับ Step เริ่มแรกให้เป็น 1 วินาทีก่อนจากนั้นให้ทำการเก็บค่าได้ที่ได้มาในหนึ่งวินาทีแรกมาคำนวณช่องสัญญาณที่ส่งได้ต่อไป จากนั้นแล้วทำการเพิ่ม

เวลาขึ้นไปทีละหนึ่งวินาทีเรื่อยๆ ในช่วงเวลา 60 วินาที แล้ววนกลับมาเช็คเวลาในลูป step ต่อไปของการเช็คช่วงเวลา

- action = "+" คือการเข้าคิว(Enqueue)แถวคอย
- action = "-" คือการออกจากคิว(Dequeue)แถวคอย

2. บรรทัดที่ 8-18 เป็นการแยกคอลัมภ์ของข้อมูลดิบเทรสไฟล์และบันทึกค่าลงตัวแปร
3. บันทึกขนาดของกลุ่มข้อมูลของ เอ็มเพ็ก4 ที่ถูกส่งออกมาจากโหนด 1 ไปยังโหนด 18 (หมายเลขการไหล 9000) ทั้งแบบโดยรวมและต่อวินาที บรรทัดที่ 25-27
4. เมื่อครบทุกๆ1วินาทีให้ทำการคำนวณปริมาณข้อมูลที่รองรับได้จากสูตร throughput = throughput +pktsize มีหน่วยเป็นเมกกะบิตต่อวินาทีและแสดงผล บรรทัดที่ 19-23

```
1  BEGIN {
2  # Initialization
3      highest_packet_id = 0;
4      step = 1;
5      throughput = 0;
6  }
7  {
8   action = $1;
9   time = $2;
10  from = $3;
11  to = $4;
12  type = $5;
13   pktsize = $6;
14  flow_id = $8;
15  src = $9;
16  dst = $10;
17  seq_no = $11;
18  packet_id = $12;
19  if (time > step) {
20          step = step+1;
21          throughput = throughput *8/1000000;
```

```
22          printf("%f %f Mbps\n", step-1, throughput);
23          throughput =0;
24  }else {
25          if (action == "+" && from =="1" && to == "18") {
26                  throughput = throughput +pktsize;
27          }
28  }
29  }
30  END {
31
32
```

3.4.7.3 เวลาหน่วงจากปลายทางถึงปลายทาง

คำนวณได้โดยการบันทึกเวลาของแพ็ตเก็ตที่มีหมายเลขการไหลที่เราจะวัดในที่นี้คือหมายเลขการไหลตั้งแต่เบอร์ 9000 ขึ้นไปคือเวลาที่กลุ่มข้อมูลถูกส่งออกมา และเวลาที่ปลายทางได้รับ แล้วนำมาหาผลต่าง

วิธีการคำนวณเวลาหน่วง

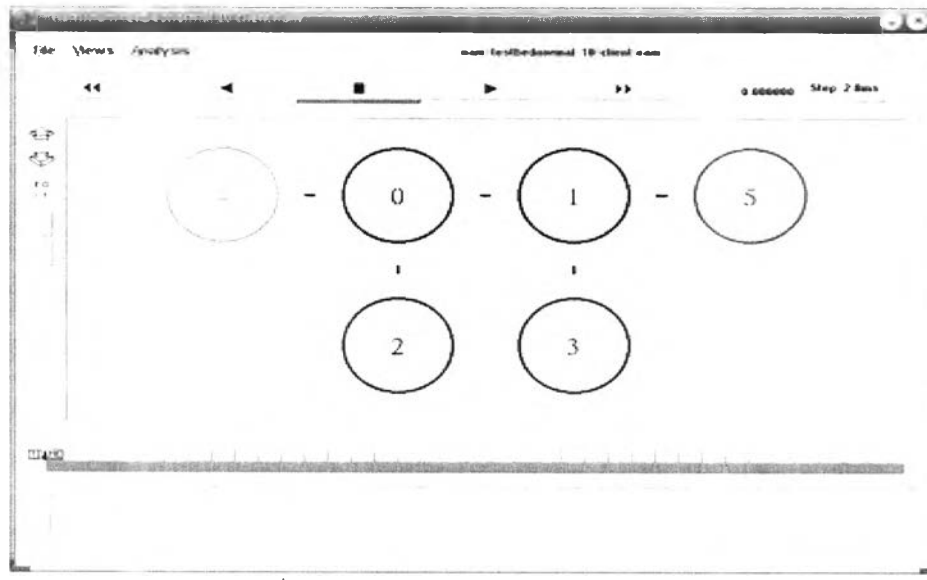
1. กำหนดตัวแปรและให้ค่าเริ่มต้นในบรรทัดที่ 2-4
2. บรรทัดที่ 7-17 เป็นการแยกคอลัมภ์ของข้อมูลดิบเทรสไฟล์และบันทึกค่าลงตัวแปร
3. บันทึกเวลาเริ่มต้นที่กลุ่มข้อมูลเอ็มเพ็ก4ถูกส่งออกมาจากโหนด19 โดยทำการบันทึกเวลาเริ่มของแพ็คเก็ตไอดีบนหมายเลขการไหลของไอดีนั้นๆ(flow_id,packet_id)ทั้งแบบโดยเวลารวมและต่อวินาที บรรทัดที่ 25-26
4. บันทึกเวลาที่กลุ่มข้อมูลเอ็มเพ็ก4ได้รับจากโหนด18 โดยทำการบันทึกเวลาสิ้นสุดการส่งของแพ็คเก็ตไอดีบนหมายเลขการไหลของไอดีนั้นๆ(flow_id,packet_id) (หมายเลขการไหล 9000) ทั้งแบบโดยเวลารวมและต่อวินาทีบรรทัดที่ 29-31
5. เมื่อสิ้นสุดการบันทึกทำการคำนวนเวลาหน่วงคือผลต่างของเวลาส่งและเวลารับใช้สูตร packet_duration = endtime – starttime และแสดงผล บรรทัดที่ 38-43

```
1   BEGIN {
2   highest_packet_id[100] = 0;
3   prevoius_delay = 0:
4   max_flowid = 0;
5   }
6   {
7     action = $1;
8    time = $2;
9    from = $3;
10   to = $4;
11   type = $5;
12   pktsize = $6;
13   flow_id = $8;
14   src = $9;
15   dst = $10;
```

```
 seq_no = $11;
packet_id = $12;

  if ( flow_id >= 9000 && action != "d" ) {
        if (action == "+" && from==19) {
              if ( packet_id > highest_packet_id[flow_id] )
                    highest_packet_id[flow_id] = packet_id;
              if ( flow_id > max_flowid )
                    max_flowid = flow_id;
              if ( start_time[flow_id,packet_id] == 0 )
        start_time[flow_id,packet_id] = time;
        }

if ( action == "r" && to==18 ) {
  end_time[flow_id,packet_id] = time;
            }

  } else {
     end_time[flow_id,packet_id] = -1;
  }
}
END {
for ( expectfid = 9000;  expectfid  <= max_flowid; expectfid++ ) {
   for ( packet_id = 0; packet_id <= highest_packet_id[expectfid]; packet_id++ ) {
      start = start_time[expectfid,packet_id];
     end = end_time[expectfid,packet_id];
     packet_duration = end - start;
     if ( start < end ) printf("time %.9f delay%f\n", start, packet_duration);
         }
}
         }
```

3.4.8 การพัฒนาคำสั่งบนการจำลองสถานการณ์ทดสอบ

มีวัตถุประสงค์เพื่อทดสอบโมดูลโพรโทคอลเอ็มพีแอลเอส และ โพรโทคอลอาร์เอสวีพี ที่ใช้เนื่องจากนักพัฒนาสามารถพัฒนาโมดูลของตนเองและรวมเข้ากับเอ็นเอสทู ตัวโปรแกรมอาจจะมีข้อผิดพลาด ผู้วิจัยจึงทำการสร้างสถานการณ์สมมติ โดยมีโครงสร้างของเครือข่ายดังรูปที่ 3-12เพื่อมาเป็นโมเดลอ้างอิงในการทดสอบว่ารูปแบบการทำงานของโมเดลที่ผู้วิจัยพัฒนาขึ้นมาคือการจำลองเหตุการณ์เครือข่ายจริงทั้งหมดมี 8 โหนดมีรูปแบบไปทางเดียวกัน และใกล้เคียงความจริงถูกต้องมากที่สุด



รูปที่ 3-11 รูปการทำงานของเครื่องมือแบบทสอบเครือข่ายเอ็นเอสทูในโมเดลการจำลองเหตุการณ์บนสถานการณ์ทดสอบ

การกำหนดตัวแปรโหนดตามตารางที่3-7

ตารางที่ 3-7 กำหนดชื่อตัวแปรโหนดต่างๆ ในการจำลองสถานการณ์ทดสอบ

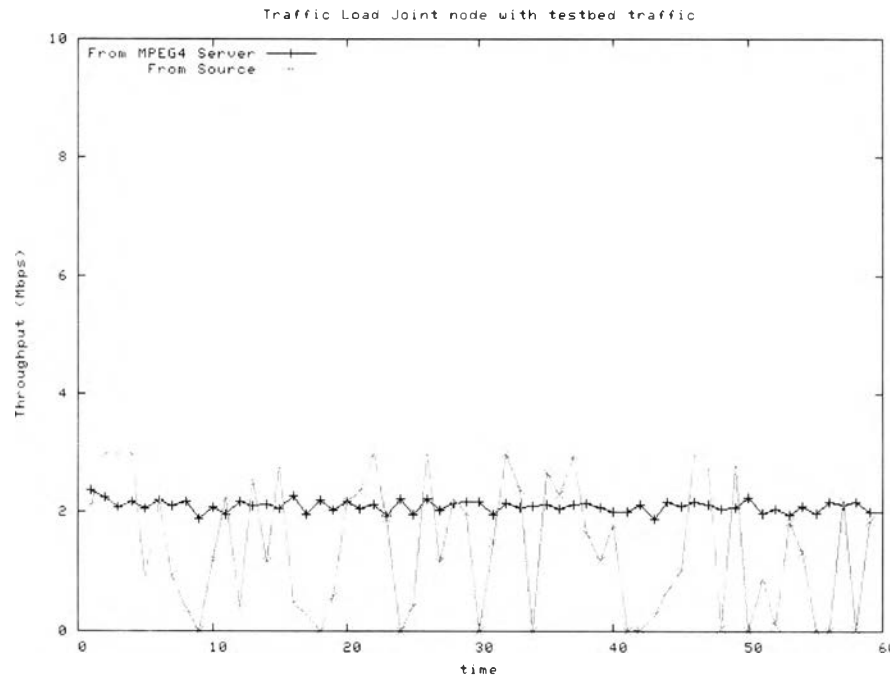
N5	เครื่องแม่ข่ายวีดีโอ
N1	โหนดอุปกรณ์หมายเลข1
N0	โหนดอุปกรณ์หมายเลข0
N3	โหนดสร้างข้อมูลของอุปกรณ์หมายเลข N1
N2	โหนดสร้างข้อมูลของอุปกรณ์หมายเลข N0
N4	เครื่องลูกข่ายวีดีโอ

การเชื่อมต่อระหว่างโหนดมีความเร็ว 4 Mbps เท่ากันหมด แต่ละโหนดมีขนาดของคิวบัฟเฟอร์ 250แพ็คเก็ต มีโพรพาเกชันดีเลย์ต่อโหนด 1 Mbps. การเชื่อต่อระหว่างโหนดอุปกรณ์จะ

ใช้ CBQในการเชื่อมต่อเพื่อสามารถจัดลำดับข้อมูลการส่งได้ ส่วนการเชื่อมต่อโหนดสร้างข้อมูลกับโหนดอุปกรณ์เป็นแบบ DropTail ไม่สามารถจัดลำดับการส่งได้

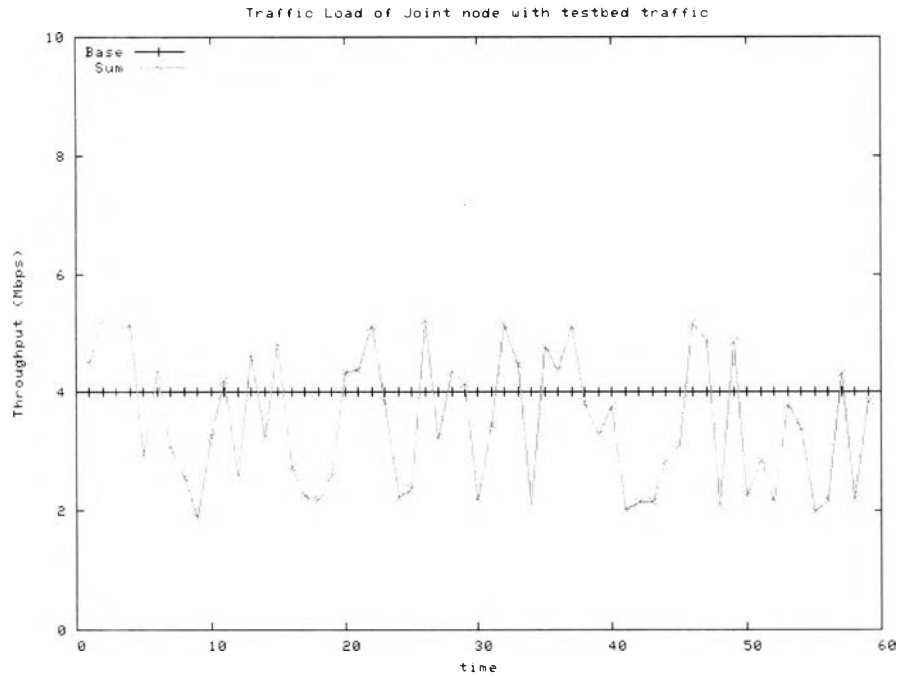
ตัวอย่างปริมาณข้อมูลกำหนดดังนี้

- โหนด 3 ทำการส่งข้อมูลแบบเอ๊กโปเนนเชียลไปให้โหนดที่ 2 ปริมาณสูงสุด 3 Mbps
- โหนด 5 คือเครื่องแม่ข่ายวีดีโอส่งข้อมูลเอ็มเพ็ก4 ไปยังผู้รับชมที่โหนด 4 ซึ่งมีผู้ชม 10 ราย ใช้แบนด์วิดท์ 2.877 Mbps.
- ปริมาณข้อมูลที่โหนด 1 (หรือเรียกว่า Joint . จุดเชื่อมต่อ) ได้รับจากโหนด 3 และโหนด 5 ซึ่งจะมีปริมาณการส่งแทรฟฟิคเข้าไปเครือข่ายแสดงดังกราฟที่3-1



กราฟที่ 3-1 แสดงอัตราการส่งแทรฟฟิคของเครื่องแม่ข่ายวีดีโอกับแทรฟฟิครบกวน 3 Mbps.

เมื่อพิจารณาความเร็วในการส่งของการเชื่อมต่อระหว่างโหนด 0 และโหนด 1 ซึ่งมีความเร็ว 4 Mbps แต่ปริมาณข้อมูลที่โหนด 1 ได้รับมีค่าเกิน 4 เมกกะบิตต่อวินาที มาจากการกำหนดปริมาณแทรฟฟิครบกวนจากโหนด3 ไปกลับยังโหนด2 ด้วยขนาด3 เมกกะบิตต่อวินาทีและแบนด์วิดท์ของเอ็มเพ็ก4 มีขนาด 2.877 เมกกะบิตต่อวินาที เป็นสาเหตุให้กลุ่มข้อมูลสูญหายที่โหนด 1 ดังรูปกราฟที่3-2



กราฟที่ 3-2 แสดงอัตราปริมาณแทรฟฟิค ช่องสัญญาณที่ส่งได้ ที่เกินแบนด์วิดท์ของโหนดแรก

ส่วนการจำลองแบบทดสอบโพรโทคอลเอ็มพีแอลเอสและโพรโทคอลอาร์เอสวีพีบนสถานการณ์ทดสอบคงใช้วิธีการทำคำสั่งสคริปต์แบบเดียวกับการจำลองเหตุการณ์เครือข่ายจุฬาฯ ในหัวข้อ3.4.5, 3.4.6 คือการกำหนดแบบจำลองของโพรโทคอลเอ็มพีแอลเอสยังคงกำหนดความสำคัญของหมายเลขการไหลของกลุ่มข้อมูลเอ็มเพ็ก4 ให้มีระดับสูงกว่าคือระดับ 7 ส่วนระดับการไหลของข้อมูลแทรฟิครบกวนมีระดับต่ำกว่าคือระดับ 1 ซึ่งรูปแบบการไหลเป็นแบบ SBTS คือจะใช้โพรโทคอล LDP ในการแลกเปลี่ยนป้ายกันระหว่างโหนดต่อโหนดไปเรื่อยตามเส้นทาง และการกำหนดแบบจำลองโพรโทคอลอาร์เอสวีพีกำหนดรูปแบบการให้บริการของแทรฟฟิครบกวนเป็นแบบ SBTS ส่วนกลุ่มข้อมูลเอ็มเพ็กเป็นแบบ HBTS จะทำการไหลข้อมูลไปตามเส้นทางที่สงวนมาจากทรัพยากรเส้นทางของเครือข่ายข้อมูลแทรฟฟิครบกวนไหลเดิมซึ่งกำหนดหมายเลขการไหลเท่ากับ 2000 ในการทำสำรองความกว้างของช่องสัญญาณจำนวน 1 Mbps. ซึ่งผู้วิจัยใช้ค่า 1 Mbps. เนื่องจากต้องการดึงแบนด์วิดท์จากหนึ่งในสี่ของความเร็ว 4 Mbps. เป็นการเริ่มต้นทดสอบก่อน