

การพัฒนาโปรแกรมเพื่อให้โปรแกรมค้นผ่านเว็บไซต์เว็บแคชร่วมกัน



นาย ยุทธกานต์ ประจักษ์เกียรติ

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2543

ISBN 974-130-863-9

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DEVELOPMENT OF A PROGRAM FOR WEB BROWSERS
TO USE A SHARED WEB CACHE

MR.YOOTTHAKARN PRUNGKIAT



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in Computer Science

Department of Computer Engineering

Chulalongkorn University

Academic Year 2000

ISBN 974-130-863-9

หัวข้อวิทยานิพนธ์	การพัฒนาโปรแกรมเพื่อให้อีโปรแกรมค้นผ่านเว็บไซต์เว็บแคชร่วมกัน
โดย	นาย ยุทธกานต์ ปรงเกียรติ
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ บุญชัย ไสวรรณวงษ์กุล
อาจารย์ที่ปรึกษาร่วม	อาจารย์ ชัชวาล วงศ์ศิริประเสริฐ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัย
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร.ยรรยง เต็งอำนวย)

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ บุญชัย ไสวรรณวงษ์กุล)

..... อาจารย์ที่ปรึกษาร่วม
(อาจารย์ ชัชวาล วงศ์ศิริประเสริฐ)

..... กรรมการ
(อาจารย์ ดร.ณัฐวดี หนูไพโรจน์)

ยุทธกานต์ ประจักษ์เกียรติ : การพัฒนาโปรแกรมเพื่อให้โปรแกรมค้นหาผ่านเว็บใช้เว็บแคช
ร่วมกัน (DEVELOPMENT OF A PROGRAM FOR WEB BROWSERS TO USE A
SHARED WEB CACHE) อ. ที่ปรึกษา : ผศ.บุญชัย ไสววรรณวิชกุล, อ. ที่ปรึกษาร่วม:
อ. ชัชวาล วงศ์ศิริประเสริฐ, 66 หน้า. ISBN 974-130-863-9.

การทำเว็บแคช เป็นแนวทางหนึ่งในการแก้ปัญหาภาวะของตัวบริการเว็บ และลดความ
คับคั่งของเครือข่ายอินเทอร์เน็ต อันเป็นผลสืบเนื่องมาจากการใช้งานเว็บบราวเซอร์ วิทยานิพนธ์
ฉบับนี้นำเสนอการออกแบบ และพัฒนาโปรแกรมเพื่อให้โปรแกรมค้นหาผ่านเว็บใช้งานเว็บแคชร่วม
กันในข่ายงานบริเวณเฉพาะที่

โปรแกรมเพื่อให้โปรแกรมค้นหาผ่านเว็บใช้งานเว็บแคชร่วมกันนี้ จะแบ่งออกเป็น 3 ส่วน
คือ (1) โปรแกรมส่วนของการดักยูอาร์แอล ทำหน้าที่ดักยูอาร์แอลจากโปรแกรมเน็ตสเคป (2)
โปรแกรมส่วนการจัดการแคชในส่วนของไคลเอนต์ ทำหน้าที่ในการค้นหาเว็บแคชที่อยู่ในไคลเอนต์
อื่นในข่ายงานบริเวณเฉพาะที่ และบริหารฐานข้อมูลแคชของเน็ตสเคป (3) โปรแกรมสำหรับการ
จัดการแคชระยะไกล ทำหน้าที่ในการรองรับคำร้องขอ และทำการโอนย้ายเพิ่มข้อมูลไปยังผู้ร้องขอ

ขั้นตอนวิธีการทดสอบความทนทานของโปรแกรมเพื่อให้โปรแกรมค้นหาผ่านเว็บใช้งานเว็บ
แคชร่วมกันต่อการใช้งานของผู้ใช้ ที่นำเสนอในงานวิจัยนี้ ได้ทำการทดสอบการจำลองการนำเข้า
ยูอาร์แอลกับโปรแกรมเน็ตสเคป รุ่นที่ 4.76 โดยมีคอมพิวเตอร์ทดสอบที่ต่อเป็นข่ายงานบริเวณ
เฉพาะที่ จำนวน 5 เครื่อง ทำการทดสอบ 2 กรณี คือ (1) การจำลองการนำเข้ายูอาร์แอลให้กับ
โปรแกรมเน็ตสเคป จำนวน 180 ยูอาร์แอล ที่อัตรา 1 ยูอาร์แอลต่อ 1 นาที และ (2) การจำลองการ
นำเข้ายูอาร์แอลจำนวน 4,000 ยูอาร์แอล ที่อัตรา 16.67 ยูอาร์แอลต่อ 1 นาที ซึ่งผลการทดสอบ
พบว่า ในกรณีที่ (1) โปรแกรมทำงานได้อย่างมีประสิทธิภาพ ส่วนในกรณีที่ (2) พบว่าโปรแกรม
เกิดข้อผิดพลาด และพบว่าโปรแกรมสามารถทนทานการนำเข้ายูอาร์แอลได้สูงสุดประมาณ 2,000
ยูอาร์แอล

ภาควิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต.....
สาขาวิชา.....วิทยาศาสตร์คอมพิวเตอร์..... ลายมือชื่ออาจารย์ที่ปรึกษา.....
ปีการศึกษา.....2543..... ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....

4070386921 : MAJOR COMPUTER SCIENCE

KEYWORD: WEB CACHE/ WEB CACHE SHARING/ WEB BROWSER/ PROXY SERVER

YOOTTHAKARN PRUNGKIAT : DEVELOPMENT OF A PROGRAM FOR
WEB BROWSERS TO USE A SHARED WEB CACHE. THESIS ADVISOR :
ASSIST. PROF. BOONCHAI SOWANWANICHAKUL, THESIS COADVISOR:
CHATCHAWAN WONGSIRIPRASERT, 66 pp. ISBN 974-130-863-9.

Web Caching is one of the techniques to reduce bottlenecks in the Internet, and also loosen workload at the web servers. This thesis presents a design and development of a program for web browsers to use a shared web cache among client browsers on LAN.

The developed program is divided into 3 parts. (1) URL Interceptor Program or UIC which trapping URLs. (2) Cache Client Manager program or CCM that can find documents in other client browser caches on LAN based on broadcasting techniques, and also manage Netscape cache database. (3) Remote Cache Manager program (RCM) which awaits requests from CCM and transfer the requested files to CCM.

The test of the robustness of the program is based on URL-input process simulation. There are 2 simulations performed on 5 computers which running Netscape V.4.76 on LAN. The first simulation test was accomplished by sending 180 URLs to Netscape at rate 1 URL/min. This case took 3 Hours and the developed program can operate normally. The second test was arranged by sending 4,000 URLs to Netscape at rate 16.67 URLs/min. In this experiment, developed program failed and terminated after sending approximately 2,000 URLs to Netscape. This second experiment showed that the program could provide up to approximately 2,000 URLs at this rate of request.

Department	Computer Engineering	Student's signature
Field of study	Computer Science	Advisor's signature
Academic year	2000	Co-advisor's signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยความกรุณาจาก ผศ.บุญชัย โสวรรณวณิชกุล และ อาจารย์ ชัชวาล วงศ์ศิริประเสริฐ ซึ่งได้ให้ความช่วยเหลือติดตามผลการวิจัย อีกทั้งได้สละเวลาในการให้คำปรึกษาและแนะนำต่าง ๆ มากมายอันเป็นประโยชน์อย่างยิ่งแก่ผู้วิจัยมาโดยตลอด

ขอขอบคุณอาจารย์ ดร.ยรรยง เต็งอำนวย และอาจารย์ ดร.ณัฐววุฒิ หนูไพโรจน์ ที่ได้สละเวลาในการสอบวิทยานิพนธ์

ขอขอบคุณเพื่อน ๆ พี่ ๆ และ อาจารย์ชัชวาล วงศ์ศิริประเสริฐ ที่ได้ให้ความช่วยเหลือและคำแนะนำต่าง ๆ ในการเขียนโปรแกรม

ขอขอบคุณ คุณผาณิต ลิ้มเกียรติเชิดชู ที่ได้ให้ความช่วยเหลือในการอ่าน และพิมพ์รายงานวิทยานิพนธ์ฉบับนี้

ท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่ได้ให้การสนับสนุนช่วยเหลือและให้กำลังใจมาโดยตลอด

ยุทธกานต์ ปรงเกียรติ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญภาพ.....	ฌ
สารบัญตาราง.....	ฎ
บทที่	
1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของวิทยานิพนธ์.....	2
1.3 ขอบเขตของวิทยานิพนธ์.....	2
1.4 ขั้นตอนการดำเนินการ.....	3
2 ทฤษฎีที่เกี่ยวข้องและแนวความคิดการวิจัย.....	4
2.1 เว็บแคช.....	4
2.1.1 เว็บแคชของโปรแกรมคั่นผ่านเว็บ.....	4
2.1.2 เว็บแคชของตัวบริการแทน (Proxy Server).....	5
2.1.3 การใช้งานเว็บแคชร่วมกันของตัวบริการแทน [1].....	6
2.2 ดีดีอี (DDE).....	7
2.3 โปรแกรมบริการดีดีอีของเน็ตสเคป (Netscape DDE Server).....	9
2.4 เว็บแคชของเน็ตสเคป.....	9
2.5 โปรแกรมช่วยงาน.....	12
2.5.1 โปรแกรม cachedump.exe.....	12
2.5.2 โปรแกรมตัวบริการแทนของเกณฑ์วิธีการส่งข้อความหลายมิติแบบเฉพาที่ (Local Proxy Server Program).....	13
3 การออกแบบโปรแกรม.....	15
3.1 แนวคิดในการพัฒนาโปรแกรม.....	15
3.2 การออกแบบโปรแกรม.....	18
3.2.1 โปรแกรมที่ใช้ทำการดักยูอาร์แอลที่โปรแกรมเน็ตสเคป ต้องการโหลด หรือ UIC (URL Interceptor Program).....	19

สารบัญ (ต่อ)

หน้า

บทที่	
3.2.2 โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ หรือ CCM (Cache Client Manager Program)	21
3.2.3 โปรแกรมสำหรับการจัดการแคชระยะไกล (Remote Cache Manager Program).	24
4 รายละเอียดการพัฒนาโปรแกรม	26
4.1 โปรแกรมที่ใช้ทำการดักยูอาร์แอลที่โปรแกรมเน็ตสเคป ต้องการโหลด หรือ UIC (URL Interceptor Program).....	26
4.1.1 คลาส CDDEOject	28
4.1.2 คลาส CDDEConversation	33
4.1.3 ฟังก์ชันสำหรับการเรียกย้อนกลับ NstestDdeCallBack.....	39
4.2 โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ หรือ UCM (Cache Client Manager Program)	41
4.2.1 ส่วนของการแพร่กระจายข้อความไปในข่ายงานบริเวณเฉพาะที่.....	41
4.2.2 ส่วนของการสร้างการเชื่อมต่อและโอนย้ายเพิ่มข้อมูล	43
4.2.3 ส่วนของโปรแกรมในการค้นหาข้อมูลในฐานข้อมูลแคช และปรับปรุงฐานข้อมูลแคช	47
4.3 โปรแกรมสำหรับการจัดการแคชระยะไกล หรือ RCM (Remote Cache Manager Program)	49
4.3.1 โปรแกรมสำหรับการรองรับการแพร่กระจายข้อความ	50
4.3.2 ส่วนของการสร้างการเชื่อมต่อและโอนย้ายเพิ่มข้อมูล	52
5 การทดสอบเพื่อประเมินผลโปรแกรม	56
6 สรุปผลการวิจัยและข้อเสนอแนะ	59
6.1 สรุปผลการวิจัย.....	59
6.2 ข้อจำกัด.....	59
6.3 ข้อเสนอแนะ	60
ภาคผนวก.....	62
ภาคผนวก ก.....	63
ภาคผนวก ข.....	65
ประวัติผู้เขียนวิทยานิพนธ์	66

สารบัญภาพ

	หน้า
รูปที่ 2.1 แสดงการใช้งานตัวบริการแทน	5
รูปที่ 2.2 แสดงการทำงานของดีดีอี.....	8
รูปที่ 2.3 แสดงการตั้งค่าคำสั่ง About:cache และผลลัพธ์	12
รูปที่ 2.4 แสดงตัวอย่างการใช้งานโปรแกรม cachedump.exe และผลลัพธ์	13
รูปที่ 2.5 แสดงการใช้งานโปรแกรมตัวบริการแทน.....	14
รูปที่ 2.6 แสดงการแก้ไขฟังก์ชัน ProxyToServer() ซึ่งเป็นฟังก์ชันการทำงานของโปรแกรม Proxy.exe.....	14
รูปที่ 3.1 แสดงขั้นตอนการทำงานของโปรแกรมเน็ตสเคป	16
รูปที่ 3.2 แสดงการสอดแทรกขั้นตอนการหาเว็บแคชที่ต้องการจากโคลเอนต์อื่นมาใส่ในแคช เฉพาะที่	17
รูปที่ 3.3 แสดงความสัมพันธ์ขององค์ประกอบของโปรแกรม	18
รูปที่ 3.4 แสดงการทำงานของส่วนดักยูอาร์แอล	19
รูปที่ 3.5 แสดงขั้นตอนการทำงานหลังจากโปรแกรมบริการดีดีอีที่สร้างขึ้นได้รับยูอาร์แอลเข้าจาก โปรแกรมบริการดีดีอีของเน็ตสเคป.....	20
รูปที่ 3.6 แสดงอัลกอริทึมของโปรแกรมสำหรับจัดการแคชในส่วนของโคลเอนต์.....	23
รูปที่ 3.7 แสดงอัลกอริทึมของโปรแกรมสำหรับการจัดการแคชระยะไกล.....	25
รูปที่ 4.1 แสดงความสัมพันธ์ระหว่างคลาส CDDEObject และคลาส CDDEConversation	26
รูปที่ 4.2 แสดงแผนผังแบบลำดับการทำงานสำหรับโปรแกรมที่ใช้ทำการดักยูอาร์แอลที่โปรแกรม เน็ตสเคป ต้องการโหลด.....	27
รูปที่ 4.3 แสดงเมธอด WWW_RegisterProtocol ของคลาส CDDEObject.....	32
รูปที่ 4.4 แสดงเมธอด WWW_UnRegisterProtocol ของคลาส CDDEObject	32
รูปที่ 4.5 แสดงเมธอด WWW_Alert ของคลาส CDDEConversation.....	35
รูปที่ 4.6 แสดงเมธอด WWW_BeginProgress ของคลาส CDDEConversation	35
รูปที่ 4.8 แสดงเมธอด WWW_MakingProgress ของคลาส CDDEConversation	36
รูปที่ 4.9 แสดงเมธอด WWW_SetProgressRange ของคลาส CDDEConversation	37
รูปที่ 4.10 แสดงเมธอด WWW_OpenURL ของคลาส CDDEConversation	38
รูปที่ 4.11 แสดงการลงทะเบียนฟังก์ชันเรียกย้อนกลับโดยใช้ฟังก์ชัน DdeInitialize()	39
รูปที่ 4.13 แสดงส่วนของโปรแกรมการกำหนดค่าเพื่อการแพร่กระจายข้อความ โดยการใช้ ฟังก์ชัน setsockopt().....	42

สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 4.14 แสดงส่วนของโปรแกรมการกำหนดค่าเลขที่อยู่ของการแพร่กระจาย.....	42
รูปที่ 4.15 แสดงรูปแบบของข้อความที่จะทำการแพร่กระจายไปในข่ายงานบริเวณเฉพาะที่	42
รูปที่ 4.16 แสดงการทำงานของโปรแกรมสำหรับสร้างการเชื่อมต่อระหว่างโปรแกรมสำหรับการจัดการ การแคชในส่วนขอโคลเอนต์ และโปรแกรมสำหรับการจัดการแคชระยะไกล	44
รูปที่ 4.17 แสดงโปรแกรมสำหรับการรองรับการเชื่อมต่อของโปรแกรมสำหรับการจัดการแคช ระยะไกล	45
รูปที่ 4.18 แสดงโปรแกรมการทำงานของฟังก์ชันรับเพิ่มข้อมูลที่ส่งจากโปรแกรมสำหรับการจัดการ การแคชระยะไกลโดยการใช้เทคนิค I/O Overlapped.....	46
รูปที่ 4.19 แสดงโปรแกรมการทำงานของฟังก์ชัน RecvComplete ซึ่งเป็นฟังก์ชันสำหรับการ เรียกย้อนกลับ	47
รูปที่ 4.20 แสดงโปรแกรมของฟังก์ชัน DB * net_OpenExtCacheFatDB()	48
รูปที่ 4.21 แสดงโปรแกรมของฟังก์ชัน NET_WriteCacheFAT()	48
รูปที่ 4.22 แสดงฟังก์ชัน NET_IsUrlInDiskCache()	49
รูปที่ 4.24 แสดงฟังก์ชันสำหรับรองรับการแพร่กระจายข้อความ	51
รูปที่ 4.25 แสดงขั้นตอนการทำงานของโปรแกรมในส่วนการสร้างการเชื่อมต่อและโอนย้าย เพิ่มข้อมูล.....	52
รูปที่ 4.26 แสดงการทำงานของโปรแกรมโอนย้ายเพิ่มข้อมูล	53
รูปที่ 4.27 แสดงฟังก์ชัน SendBuffer()	55
รูปที่ 5.1 แสดงโครงสร้างของเครือข่ายและการเชื่อมต่อสู่ภายนอก	56

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงหัวข้อบริการ ที่จำเป็นต้องมีในโปรแกรมบริการที่ดีี่สร้างขึ้น	9
ตารางที่ 2.2 แสดงรายละเอียดของเขตข้อมูลในแฟ้มข้อมูล FAT.DB.....	11
ตารางที่ 4.1 แสดงตัวแปรภายในคลาส CDDEOject	28
ตารางที่ 4.2 แสดงเมธอดของคลาส CDDEOject.....	30
ตารางที่ 4.3 แสดงตัวแปรภายในคลาส CDDEConversation	33
ตารางที่ 4.4 แสดงเมธอดของคลาส CDDEConversation	34
ตารางที่ ก.1 แสดงรายละเอียดหัวข้อบริการต่าง ๆ ของโปรแกรมบริการที่ดีี่ของเน็ตสเค๊ป.....	63



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องด้วยการเจริญเติบโตของเครือข่ายอินเทอร์เน็ต (Internet) ทำให้มีผู้ใช้งานเครือข่ายดังกล่าวเป็นจำนวนมาก และหนึ่งในการใช้งานเครือข่ายอินเทอร์เน็ต คือการใช้บริการเวปไซด์เว็บบ (World Wide Web) ที่มีอย่างแพร่หลาย และเนื่องจากจำนวนผู้ใช้ที่มากในปัจจุบัน และมีแนวโน้มที่จะเพิ่มขึ้นอีกในอนาคต ทำให้เกิดปัญหาภาระของเครือข่ายอินเทอร์เน็ต และภาระของตัวบริการเว็บบ (Web Server) เพิ่มขึ้น จนทำให้ผู้ใช้เกิดความไม่สะดวกในการใช้บริการดังกล่าว เนื่องด้วยเหตุผลสำคัญในการนำเว็บบเพจ (Web page) จากตัวบริการเว็บบส่งไปยังผู้ใช้เกิดความล่าช้า เพราะว่าการระยะทางจากตัวบริการเว็บบอาจอยู่ห่างไกลหลายพันกิโลเมตร

จึงเกิดแนวทางแก้ไข โดยอาศัยแนวคิดที่จะทำเว็บบแคช (Web Cache) ขึ้นโดยมีหลักการง่าย ๆ คือควรมีการเก็บเว็บบเพจไว้ให้ใกล้กับผู้ใช้มากที่สุด ซึ่งสื่อที่ใช้เก็บเว็บบแคชอาจจะเป็นหน่วยความจำ หรือดิสก์ของผู้ใช้งานเอง เรียกว่าเป็นการทำเว็บบแคชเฉพาะที่ (Local Web Cache) บนตัวไคลเอนต์ (Client) เอง ซึ่งเป็นการแก้ปัญหาได้ส่วนหนึ่ง และอาจจะลดภาระของเครือข่ายอินเทอร์เน็ตได้บ้าง แต่ในขณะเดียวกันกรณีที่ไคลเอนต์ดังกล่าวต้องการใช้งานเว็บบเพจที่ไม่มีอยู่ในเว็บบแคชเฉพาะที่ของตนเอง ก็ยังคงต้องร้องขอไปยังตัวบริการเว็บบ ซึ่งทำให้ตัวบริการเว็บบยังคงต้องรับภาระในการประมวลผลคำร้องขอซึ่งมีจำนวนมาก จึงมีแนวความคิดในการทำเว็บบแคชในระดับที่สูงขึ้น เรียกว่าการทำตัวบริการแทน (Proxy Server) [1,2] ซึ่งตัวบริการแทนจะทำหน้าที่เก็บเว็บบเพจที่ไคลเอนต์ซึ่งติดต่อกับตัวบริการแทนดังกล่าวเคยไปนำมา ซึ่งจะทำให้ในกรณีที่ไคลเอนต์ที่ต้องการใช้เว็บบเพจที่ไม่มีในเว็บบแคชเฉพาะที่ จะทำการร้องขอเว็บบเพจไปยังตัวบริการแทนก่อน ถ้าเว็บบเพจดังกล่าวไม่ได้มีการเก็บอยู่บนตัวบริการแทน ตัวบริการแทนก็จะร้องขอไปยังตัวบริการเว็บบในเครือข่ายอินเทอร์เน็ตที่เก็บเว็บบเพจดังกล่าว จากรายงานการวิจัย [3] พบว่าสามารถปรับปรุงภาระของตัวบริการเว็บบ และภาระของเครือข่ายอินเทอร์เน็ตได้ รวมทั้งแก้ไขปัญหาความล่าช้าในการนำเว็บบเพจมาสู่ผู้ใช้ได้ เนื่องจากตัวบริการแทนจะอยู่ใกล้ชิดกับไคลเอนต์มากกว่าตัวบริการเว็บบ ซึ่งจากแนวความคิดในการสร้างตัวบริการแทนดังกล่าว ทำให้มีผู้คิดที่จะนำตัวบริการแทนหลาย ๆ ตัวที่ติดต่อกันอยู่ในเครือข่ายอินเทอร์เน็ต มาทำการใช้งานเว็บบเพจที่เก็บอยู่ในแต่ละตัวบริการแทนร่วมกันโดยมี ICP (Internet Cache Protocol) [3] เป็นตัวจัดการระหว่างตัวบริการแทน ซึ่งทำให้ปริมาณของพื้นที่ที่ใช้เก็บเว็บบเพจมีมากขึ้น แต่พบว่าตัวโพรโทคอล (Protocol) ดังกล่าวมีข้อจำกัด และไม่สามารถขยายให้ใช้งานได้กับตัวบริการแทนจำนวนมากได้ สำหรับการใช้งานเว็บบแคชร่วมกันโดย ICP มีหลักการคือ เมื่อตัวบริการแทนใดต้องการใช้งานเว็บบ

เพจที่ไม่มีอยู่ในเว็บแคชเฉพาะที่ของตนเอง ก็จะไปยังตัวบริการแทนอื่น ๆ ที่ทำการใช้งาน แคชร่วมกันอยู่ ถ้าพบก็จะนำเว็บเพจดังกล่าวเข้ามาและทำเป็นเว็บแคชเฉพาะที่ของตนเอง โดยที่ ตัวบริการแทนแต่ละตัวไม่มีการทำงานร่วมกันในเรื่องของการแทนที่ (Replacement) [1,4,5] เว็บ แคชเฉพาะที่ ซึ่งการใช้งานแคชร่วมกันแบบนี้เรียกว่า การใช้งานเว็บแคชร่วมกันแบบง่าย (Simple Cache Sharing) [3,6]

สำหรับในกรณีขององค์กรขนาดเล็กที่มีเครื่องคอมพิวเตอร์ใช้งานจำนวนไม่มาก ประมาณ 5 ถึง 10 เครื่อง และไม่มียงบประมาณที่จะจัดตั้งตัวบริการแทน ดังนั้นเมื่อไม่มีการใช้ตัวบริการแทน มาช่วยในการทำเว็บแคช เพื่อเป็นการเพิ่มประสิทธิภาพการให้บริการเว็บบอร์ดเว็บของผู้ใช้ใน องค์กรดังกล่าว จึงควรมีวิธีที่จะช่วยในการทำเว็บแคชโดยที่ผู้ใช้ไม่จำเป็นต้องเสียค่าใช้จ่ายเพิ่มเติมอีก นั่นคือการใช้ทรัพยากรที่มีอยู่ของผู้ใช้ให้เป็นประโยชน์สูงสุด สำหรับวิธีที่น่าเสนอข้างต้น คือการจัดตั้งตัวบริการแทน วิธีนี้จะทำให้มีเนื้อที่ในการทำเว็บแคชเพิ่มขึ้น ในกรณีนี้ประสิทธิภาพ จะขึ้นอยู่กับปริมาณเว็บเพจที่สามารถเก็บได้ ซึ่งขึ้นอยู่กับความจุของดิสก์ของตัวบริการแทน ดังนั้น ถ้าเราสามารถมีความจุดิสก์ที่เพิ่มขึ้นในการทำเว็บแคช ประสิทธิภาพในการให้บริการเว็บบอร์ดเว็บ ก็จะดีขึ้น แต่เนื่องจากเนื้อที่ที่ใช้ในการทำเว็บแคชเฉพาะที่ของโปรแกรมค้นผ่านเว็บมีจำกัด และ เพื่อเพิ่มเนื้อที่ในการทำเว็บแคช นั่นคือควรมีการนำเว็บแคชของโปรแกรมค้นผ่านเว็บของแต่ละ เครื่องไคลเอนต์มาทำการใช้งานร่วมกันนั่นเอง

1.2 วัตถุประสงค์ของวิทยานิพนธ์

พัฒนาโปรแกรมเพื่อสามารถทำให้โปรแกรมค้นผ่านเว็บสามารถทำการใช้งานเว็บแคช เฉพาะที่ของตนเอง และไคลเอนต์อื่น ๆ ที่อยู่ในข่ายงานบริเวณเฉพาะที่ (LAN) ร่วมกันได้

1.3 ขอบเขตของวิทยานิพนธ์

1. พัฒนาโปรแกรมเพื่อสามารถทำให้โปรแกรมเน็ตสเคป (Netscape) รุ่น 4.7x บน Window 95 สามารถทำการใช้งานเว็บแคชเฉพาะที่ของตนเอง และไคลเอนต์อื่น ๆ ที่ อยู่ในข่ายงานบริเวณเฉพาะที่ร่วมกันได้
2. การทดสอบโปรแกรมจะใช้เครื่องคอมพิวเตอร์ในการทดสอบ 5 เครื่อง และมีขั้นตอน การทดสอบดังนี้
 1. ทำการเคลียร์เว็บแคชเฉพาะที่ในแต่ละเครื่องทดสอบ
 2. จำลองการร้องขอยูอาร์แอลทุก 1 นาทีโดยกำหนดยูอาร์แอลในการทดสอบ จำนวน 180 ยูอาร์แอลต่อ 1 เครื่อง

1.4 ขั้นตอนการดำเนินการ

1. ศึกษาโพรโทคอลที่ใช้ในการใช้งานเว็บแคมร่วมกันระหว่างไคลเอนต์ และศึกษาโพรโทคอลที่ใช้ในการสื่อสารระหว่างไคลเอนต์ในข่ายงานบริเวณเฉพาะที่
2. ออกแบบและพัฒนาโปรแกรม
3. ทดสอบโปรแกรม
4. สรุปผลการวิจัยและข้อเสนอแนะ
5. จัดทำรายงานวิทยานิพนธ์



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีที่เกี่ยวข้องและแนวคิดการวิจัย

ในการศึกษาการวิจัย หัวข้อวิทยานิพนธ์เรื่อง “การพัฒนาโปรแกรมเพื่อให้โปรแกรมค้นผ่านเว็บใช้เว็บแคชร่วมกัน” ได้ทำการศึกษาผลงานวิจัยและงานเขียนอื่น ๆ ที่เกี่ยวข้อง เพื่อเป็นการศึกษาประเด็นสำคัญ ๆ สำหรับการอ้างอิงในบางส่วนของงานวิจัย ได้แก่

- เว็บแคช
- ดีดีอี (DDE)
- โปรแกรมบริการดีดีอีของเน็ตสเคป (Netscape DDE Server)
- เว็บแคชของเน็ตสเคป
- โปรแกรมช่วยงาน

2.1 เว็บแคช

เว็บแคชเป็นเทคนิคสำหรับการลดการคับคั่งของการสื่อสารในระบบเครือข่าย และเพื่อลดความล่าช้าในการตอบสนองในการส่งข้อมูลในระบบเครือข่ายกับผู้ใช้บริการเวปไซด์เว็บ โดยมีหลักการทำงานของแคชคือ การนำส่วนของข้อมูลจากข้อมูลทั้งหมดมาไว้ให้ใกล้กับสถานที่ที่จะทำการประมวลผลข้อมูลดังกล่าว [2] เช่น การที่ซีพียูมีแคชเพื่อนำข้อมูลเข้าและออกจากหน่วยความจำเพื่อนำไปประมวลผล หรือการที่ในระบบปฏิบัติการปัจจุบันมีการทำแคช เพื่อนำข้อมูลเข้าหรือออกจากดิสก์ ซึ่งจะทำให้สามารถลดคอขวดในการสื่อสารของอุปกรณ์ที่มีความเร็วไม่เท่ากัน อีกทั้งยังเป็นการเพิ่มประสิทธิภาพของการประมวลผลให้ดียิ่งขึ้น

สำหรับการทำเว็บแคช คือการจัดเก็บเว็บเพจที่ผู้ใช้เคยไปหรือเคยเรียกใช้เมื่อก่อนหน้านี้ไว้ โดยคาดหมายว่าผู้ใช้จะมีการใช้งานเว็บเพจดังกล่าวซ้ำอีกในเวลาต่อมา เพื่อให้ประโยชน์ในการใช้งานของผู้ใช้ โดยการที่ผู้ใช้เรียกใช้งานเว็บเพจดังกล่าว ไม่จำเป็นต้องไปนำเว็บเพจมาจากตัวบริการเว็บ ซึ่งอาจอยู่ไกลหลายพันกิโลเมตร ซึ่งจะช้ากว่าการนำเว็บเพจที่จัดเก็บอยู่ในหน่วยความจำหรือดิสก์ของผู้ใช้เองขึ้นมาใช้งาน และยังสามารถลดความคับคั่งของการสื่อสารบนระบบเครือข่ายในกรณีที่สามารถพบเว็บเพจที่ต้องการในเว็บแคชที่ผู้ใช้งานเก็บไว้ สำหรับการทำเว็บแคชได้มีการจัดทำในหลายระดับ ดังนี้ การทำเว็บแคชของโปรแกรมค้นผ่านเว็บ การทำเว็บแคชของตัวบริการแทน, การใช้งานเว็บแคชร่วมกันของตัวบริการแทน

2.1.1 เว็บแคชของโปรแกรมค้นผ่านเว็บ

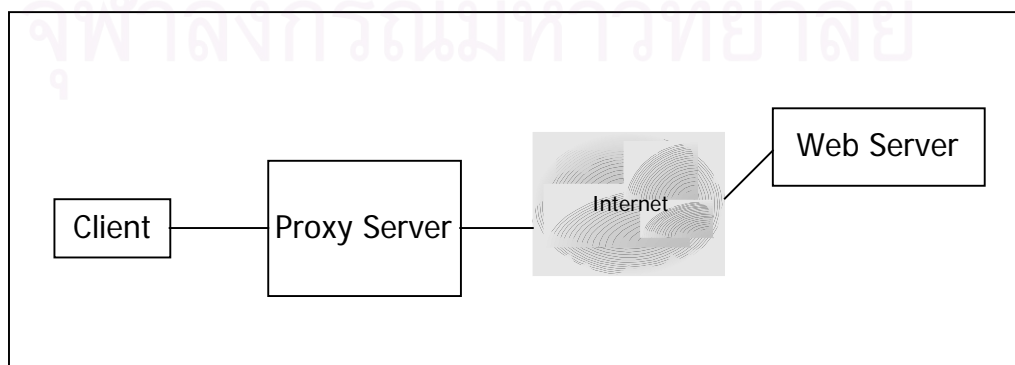
การทำเว็บแคชในระดับของโปรแกรมค้นผ่านเว็บนั้น โปรแกรมค้นผ่านเว็บจะทำการแคชโดยมีการใช้ทั้งหน่วยความจำและดิสก์ในการทำเว็บแคช ซึ่งผู้ใช้สามารถกำหนดขนาดของ

ทรัพยากรที่จะใช้ในการทำเว็บแคช เช่น สามารถกำหนดขนาดดิสก์ที่ใช้เพื่อทำเว็บแคช โดยทั่วไปจะอยู่ที่ประมาณ 5 ถึง 50 เมกกะไบต์ (Megabyte) เพื่อรองรับการทำงานของโปรแกรมในการจัดเก็บข้อมูลเกี่ยวกับการใช้งานบริการเว็ลด์ไวต์เว็บของผู้ใช้

2.1.2 เว็บแคชของตัวบริการแทน (Proxy Server)

สำหรับการทำเว็บแคชของตัวบริการแทน จะแตกต่างกับการทำเว็บแคชของ โปรแกรมค้นผ่านเว็บ โดยเว็บแคชของตัวบริการแทนสามารถถูกใช้งานร่วมกันโดยผู้ใช้งานหลาย ๆ คนในเวลาเดียวกัน ซึ่งตัวบริการแทนจะเป็นโปรแกรมที่ทำงานอยู่บนเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูง ซึ่งมีหน่วยความจำหลักตั้งแต่ 64 ถึง 512 เมกกะไบต์ และมีขนาดดิสก์ตั้งแต่ 5 ถึง 50 กิกะไบต์ (Gigabyte) สำหรับการทำเว็บแคช ตัวบริการแทนจะทำหน้าที่เป็นตัวกลางระหว่างไคลเอนต์และตัวบริการเว็บที่ให้บริการเว็ลด์ไวต์เว็บในเครือข่ายอินเทอร์เน็ต การใช้งานตัวบริการแทนต้องมีการระบุภายในโปรแกรมค้นผ่านเว็บ โดยการระบุยูอาร์แอล (URL) ของเครื่องที่โปรแกรมตัวบริการแทนทำงานอยู่ สำหรับการใช้งานโปรแกรมตัวบริการแทน มักมีการใช้งานร่วมกับโปรแกรมป้องกันการบุกรุก (Firewall) ซึ่งมีจุดประสงค์เพื่อป้องกันการบุกรุกจากภายนอก โดยติดตามและควบคุมไคลเอนต์ในการติดต่อกับเซิร์ฟเวอร์ภายนอก ซึ่งทำให้ไคลเอนต์ที่ต้องการติดต่อไปยังภายนอกถูกบังคับให้ทำการติดต่อไปยังโปรแกรมตัวบริการแทนที่ทำงานอยู่บนเครื่องเดียวกับที่โปรแกรมป้องกันการบุกรุกทำงานอยู่

สำหรับการทำงานของตัวบริการแทน จะเริ่มจากการที่ไคลเอนต์ต้องการใช้งานเว็บเพจจึงร้องขอไปยังตัวบริการแทน ตัวบริการแทนจะตรวจดูว่ามีเว็บเพจที่ต้องการเก็บอยู่หรือไม่ ถ้ามีเว็บเพจที่ร้องขอมาในหน่วยความจำหรือดิสก์ ตัวบริการแทนจะทำการส่งข้อมูลเว็บเพจดังกล่าวกลับไปยังผู้ใช้งานทันที แต่ถ้าไม่มีเว็บเพจที่ร้องขอมา ตัวบริการแทนก็จะทำการร้องขอไปยังตัวบริการเว็บ ในเครือข่ายอินเทอร์เน็ตที่ให้บริการเว็บเพจดังกล่าว เมื่อตัวบริการเว็บส่งเว็บเพจกลับมา ตัวบริการแทนก็จะทำหน้าที่รับเว็บเพจและส่งกลับไปยังไคลเอนต์ที่ร้องขอมา ซึ่งสามารถแสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 แสดงการใช้งานตัวบริการแทน

2.1.3 การใช้งานเว็บแคชร่วมกันของตัวบริการแทน [1]

เนื่องจากใช้งานเว็บแคชจะมีประสิทธิภาพมากขึ้นก็ต่อเมื่อมีเนื้อที่ในการทำเว็บแคชมากขึ้น แต่เนื่องจากเนื้อที่ในการทำเว็บแคชของแต่ละตัวบริการแทนมีจำนวนจำกัด จึงมีแนวคิดในการที่จะให้มีการใช้งานเว็บแคชร่วมกันในกลุ่มของตัวบริการแทน สำหรับการใช้งานเว็บแคชร่วมกันมีเทคนิคดังต่อไปนี้ [3]

- ไม่มีการใช้งานเว็บแคชร่วมกัน (No Cache Sharing) เมื่อตัวบริการแทนไม่พบ เว็บเพจที่ถูกร้องขอมา จะไม่มีการร้องขอไปยังตัวบริการแทนอื่น ๆ
- การใช้งานเว็บแคชร่วมกันแบบง่าย (Simple Cache Sharing) เมื่อตัวบริการแทนไม่พบเว็บเพจที่ถูกร้องขอมา ก็จจะร้องขอไปยังตัวบริการแทนอื่น ๆ ที่อยู่ในกลุ่มของตัวบริการแทน และเมื่อในกลุ่มของตัวบริการแทนพบเว็บเพจก็จะส่งกลับไปยังตัวบริการแทนที่ร้องขอเว็บเพจดังกล่าว และตัวบริการแทนที่ร้องขอเว็บเพจดังกล่าวก็จะทำการเก็บเว็บเพจดังกล่าวไว้ด้วย
- การใช้งานเว็บแคชร่วมกันแบบมีเพียงหนึ่งสำเนา (Single-Copy Cache Sharing) เมื่อตัวบริการแทนไม่พบเว็บเพจที่ถูกร้องขอมา ก็จจะร้องขอไปยังตัวบริการแทนอื่น ๆ ที่อยู่ในกลุ่มของตัวบริการแทน และเมื่อในกลุ่มของตัวบริการแทนพบเว็บเพจ ก็จจะส่งกลับไปยังตัวบริการแทนที่ร้องขอเว็บเพจดังกล่าว แต่ตัวบริการแทนที่ร้องขอเว็บเพจดังกล่าวจะไม่ทำการเก็บเว็บเพจที่ร้องขอไป สำหรับตัวบริการแทนที่ทำการส่งเว็บเพจดังกล่าวให้กับตัวบริการแทนที่ร้องขอเว็บเพจดังกล่าวจะทำการเพิ่มลำดับความสำคัญของการเก็บเว็บเพจดังกล่าว
- การใช้งานเว็บแคชร่วมกันแบบเป็นหนึ่งเดียวกัน (Global Cache) กลุ่มของตัวบริการแทนจะทำงานร่วมกันในการทำเว็บแคช โดยเสมือนว่ามีเพียง 1 ตัวบริการแทน และตัวบริการแทนจะทำงานร่วมกันในการทำการแทนที่เว็บเพจด้วย

สำหรับการใช้งานเว็บแคชร่วมกันในกลุ่มของตัวบริการแทนที่มีอยู่ในปัจจุบันมีการใช้งานเว็บแคชร่วมกันแบบง่าย โดยมีโพรโทคอลที่ใช้ในการจัดการคือ ICP

จากแนวคิดในการทำเว็บแคชเฉพาะที่ข้างต้น และการใช้งานเว็บแคชร่วมกันในระดับตัวบริการแทน สามารถนำมาประยุกต์ใช้ในระดับไคลเอนต์ได้ โดยใช้ทั้ง 2 แนวความคิดร่วมกัน โดยสมมติว่าถ้าไม่มีการทำเว็บแคชในระดับตัวบริการแทนแล้ว โปรแกรมค้นผ่านเว็บของไคลเอนต์ที่ติดต่อกันอยู่ในข่ายงานบริเวณเฉพาะที่ ก็น่าจะสมารถทำการใช้งานเว็บแคชเฉพาะที่ร่วมกันได้โดยอาศัยหลักการเดียวกับที่ทำในระดับตัวบริการแทนได้ โดยที่เมื่อไคลเอนต์ต้องการใช้งานเว็บเพจแล้วไม่พบเว็บเพจดังกล่าวในเว็บแคชเฉพาะที่ของตนเอง แทนที่จะร้องขอไปยังตัวบริการเว็บใน

เครือข่าย อินเทอร์เน็ตโดยตรง ควรจะร้องขอไปยังไคลเอนต์อื่น ๆ ที่อยู่ในข่ายงานบริเวณเฉพาะที่ก่อน ถ้ามีไคลเอนต์ที่มีเว็บเพจดังกล่าวอยู่เว็บเพจก็จะถูกส่งกลับไปยังไคลเอนต์ที่ทำการร้องขอมา ถ้าใน ทุก ๆ ไคลเอนต์ไม่มีการเก็บเว็บเพจดังกล่าวไว้ ตัวไคลเอนต์จึงค่อยร้องขอไปยังตัวบริการเว็บในเครือข่ายอินเทอร์เน็ตอีกครั้ง เรียกว่าเป็นการใช้งานเว็บแคชร่วมกันของไคลเอนต์

2.2 ดีดีอี (DDE)

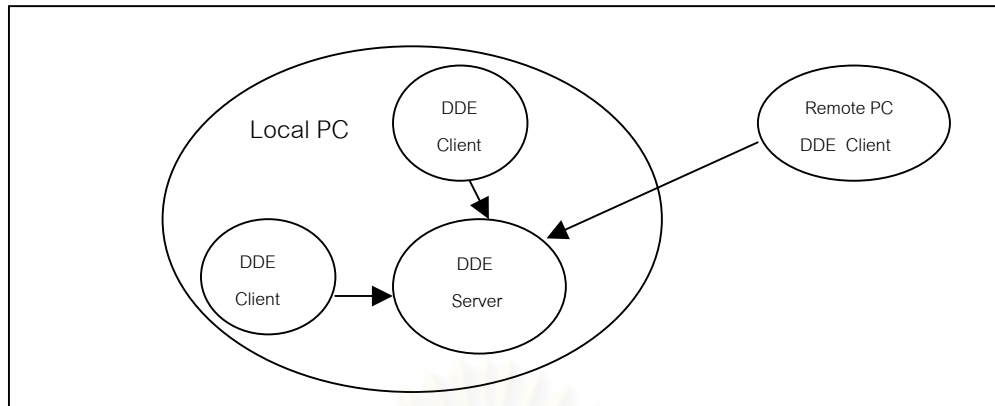
ดีดีอี เป็นวิธีการดั้งเดิมที่ระบบปฏิบัติการวินโดวส์สามารถทำให้โปรแกรมประยุกต์สามารถแลกเปลี่ยนข้อมูลกันได้ โดยวิธีการส่งข้อความ (Message) ระหว่างกันโดยมีข้อกำหนดตามโพรโทคอลที่ได้ตกลงกัน การทำงานของดีดีอีนั้นเป็นแบบเดียวกับไคลเอนต์เซิร์ฟเวอร์ คือ มีส่วนที่เป็นเซิร์ฟเวอร์ที่ทำหน้าที่ในการให้บริการข้อมูล และมีไคลเอนต์เป็นผู้ร้องขอข้อมูล ซึ่งตัวผู้ร้องขอนี้สามารถทำงานเป็นกระบวนการไคลเอนต์ (Client Process) อยู่บนเครื่องคอมพิวเตอร์เดียวกันกับกระบวนการเซิร์ฟเวอร์ (Server Process) หรือว่าทำงานอยู่บนเครื่องคอมพิวเตอร์เครื่องอื่นได้ และสำหรับโปรแกรมประยุกต์บางตัวสามารถทำหน้าที่เป็นได้ทั้งไคลเอนต์และเซิร์ฟเวอร์ เช่น โปรแกรมประยุกต์ EXCEL

สำหรับการใช้งานดีดีอี มีหลักการดังนี้คือ โปรแกรมประยุกต์ที่ทำหน้าที่เป็นไคลเอนต์ทำการสร้างการเชื่อมต่อไปยังโปรแกรมประยุกต์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ โดยการระบุพารามิเตอร์ 3 ตัว คือ

1. ชื่อของโปรแกรมบริการดีดีอี (DDE Server Name)
2. ชื่อหัวข้อบริการ (Topic Name) หรือฟังก์ชันที่ต้องการเรียกใช้
3. ชื่อข้อมูล (Item Name) ที่โปรแกรมประยุกต์ต้องการให้ส่งกลับ หรือชื่อข้อมูลที่ต้องการส่งให้

ตัวอย่างเช่น กรณีต้องการติดต่อโปรแกรมประยุกต์ EXCEL ชื่อของโปรแกรมบริการดีดีอีของโปรแกรมประยุกต์ก็คือ EXCEL ชื่อหัวข้อบริการที่ต้องการก็คือชื่อของเอกสารที่บรรจุข้อมูลอยู่ และชื่อข้อมูลก็คือแถวลำดับของข้อมูลที่อยู่ในเอกสารดังกล่าว

สำหรับบางกรณีไคลเอนต์สามารถที่จะสร้างการเชื่อมต่อโดยไม่ระบุ ชื่อหัวข้อบริการและชื่อข้อมูลที่ต้องการ ซึ่งการทำแบบนี้เรียกว่า WILDCONNECT ซึ่งจะมีผลทำให้การเชื่อมต่อสามารถใช้งานได้กับทุก ๆ ชื่อหัวข้อบริการของโปรแกรมบริการดีดีอี (DDE Server)



รูปที่ 2.2 แสดงการทำงานของดีดีอี

เมื่อไคลเอนต์ได้ทำการสร้างการเชื่อมต่อไปยังโปรแกรมบริการดีดีอีแล้ว ก็จะมีการกำหนดค่าเริ่มต้นของทุกรายการ (Transaction) ด้วย ซึ่งรายการสามารถแบ่งออกเป็นประเภทดังนี้

- รายการร้องขอ (REQUEST Transaction)

เกิดขึ้นเมื่อไคลเอนต์ทำการร้องขอข้อมูลมายังโปรแกรมบริการดีดีอี

- รายการแจ้งเตือน (ADVISE or NOTIFY Transaction)

เป็นการร้องขอเพื่อให้โปรแกรมบริการดีดีอีแจ้งให้ทราบกรณีมีการเปลี่ยนแปลงของข้อมูล

- รายการประมวลผล (EXECUTE Transaction)

เป็นการส่งให้กับโปรแกรมบริการดีดีอีประมวลผล

- รายการโปก (POKE Transaction)

เป็นการส่งข้อมูลที่ไม่ได้ร้องขอ ให้กับโปรแกรมบริการดีดีอี

ประโยชน์การใช้งานโปรแกรมบริการดีดีอีส่วนใหญ่ มักถูกใช้งานเป็น โปรแกรมเฝ้าสังเกต (Programming Monitor) เพื่อเฝ้าดูการทำงานของอีกโปรแกรมหนึ่ง เพราะว่าโปรแกรมทั้งสองทำงานแยกจากกันจึงไม่เกิดการรบกวนระหว่างกัน จะต่างกับการที่กระบวนการไคลเอนต์ ถูกนำไปรวมอยู่ด้วยกันกับกระบวนการเซิร์ฟเวอร์ในลักษณะสายโยงใย (Thread) ซึ่งในการทำงานในลักษณะนี้ กระบวนการเซิร์ฟเวอร์จะต้องรอการทำงานของกระบวนการไคลเอนต์ ดังนั้นประสิทธิภาพจะขึ้นอยู่กับความสามารถของไคลเอนต์ในการประมวลผลข้อมูล และถ้ามีไคลเอนต์ใดเกิดหยุดการทำงานขึ้นมา อาจเกิดการรบกวนกระบวนการเซิร์ฟเวอร์ และอาจทำให้กระบวนการเซิร์ฟเวอร์เกิดหยุดการทำงานด้วย

2.3 โปรแกรมบริการดีดีของเน็ตสเคป (Netscape DDE Server)

สำหรับโปรแกรมบริการดีดีของเน็ตสเคป และโปรแกรมบริการดีดีของโปรแกรมค้นผ่านเว็บอื่น ๆ เช่น IE (Internet Explorer) ถูกสร้างขึ้นเพื่อให้โปรแกรมประยุกต์ภายนอกสามารถทำการติดต่อ แลกเปลี่ยนข้อมูลและสามารถควบคุมการทำงานของโปรแกรมค้นผ่านเว็บได้

โปรแกรมบริการดีดีของเน็ตสเคปนี้ มีชื่อเรียกโปรแกรมบริการดีดีว่า "NETSCAPE" และมีชื่อหัวข้อมบริการ ที่ใช้ในการควบคุมเน็ตสเคป ดังแสดงในภาคผนวก ก

สำหรับโปรแกรมบริการดีดีที่จะทำการพัฒนาขึ้นเพื่อควบคุมการทำงานของเน็ตสเคปนี้ จะมีหัวข้อมบริการที่จะต้องทำการพัฒนาดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 แสดงหัวข้อมบริการ ที่จำเป็นต้องมีในโปรแกรมบริการดีดีที่สร้างขึ้น

Topic name	ความหมาย
WWW_Alert	แสดงกล่องข้อความแสดงความผิดพลาด (alert) ของเน็ตสเคป
WWW_BeginProgress	แสดงสถานะเริ่มต้นการโหลดเว็บเพจ
WWW_EndProgress	แสดงสถานะการโหลดเว็บเพจสิ้นสุดลงแล้ว
WWW_MakingProgress	แสดงสถานะปัจจุบันของการโหลดเว็บเพจ
WWW_OpenURL	โหลดเว็บเพจ
WWW_RegisterProtocol	ทำการลงทะเบียน (Register) โพรโทคอลกับโปรแกรมบริการดีดีของเน็ตสเคป
WWW_SetProgressRange	กำหนดสถานะความก้าวหน้าของการโหลดเว็บเพจ สำหรับข้อความแสดงความก้าวหน้า (Progress message)
WWW_UnRegisgerProtocol	ทำการยกเลิกการลงทะเบียน (Unregister) โพรโทคอลกับโปรแกรมบริการดีดีของเน็ตสเคป

2.4 เว็บแคชของเน็ตสเคป

เมื่อโปรแกรมเน็ตสเคปทำการร้องขอเว็บเพจจากตัวบริการเว็บ ตัวบริการเว็บก็จะทำการส่งสิ่งที่ร้องขอกลับมาให้เน็ตสเคป สิ่งที่เน็ตสเคปได้รับกลับมาจากการร้องขออย่างครบถ้วนสมบูรณ์ และมีประเภทไม้ม (MIME Type) เป็นตัวอักษร (text/plain) รูปภาพ (image/jpeg หรือ image/gif หรือ image/cgm) ดรรชนีของเกณฑ์วิธีการส่งข้อความหลายมิติ (Indexes of HTTP) (text/html) และจาวาคลาส (JAVA Class) เพิ่มข้อมูลที่มีประเภทไม้มที่กล่าวข้างต้นจะถูก

เน็ตสเคปทำการแคชไว้ในดิสก์ทั้งสิ้น ยกเว้นเอกสารที่ต้องการความปลอดภัย (Secure Document) เอกสารที่มีการกำกับไว้ในเฮดเดอร์ตอบกลับ (เฮดเดอร์ Cache-control:) mail (จดหมาย) และ news(ข่าว)

สำหรับการทำเว็บแคชของเน็ตสเคปนั้น มี 2 ประเภทคือ

- ดิสก์แคช

ดิสก์แคชจะถูกใช้ในการแคชเอกสาร ที่มีประเภทไม่มุกชนิดที่กล่าวข้างต้น ยกเว้นเพียงแต่เอกสารที่ต้องการความปลอดภัย เน็ตสเคปจะทำการแคชเอกสารต่าง ๆ ไว้ในฐานข้อมูลแคช (Cache Database) ซึ่งเป็นแฟ้มข้อมูลที่มีชื่อว่า FAT.DB และแฟ้มข้อมูลนี้จะถูกเก็บอยู่ในแคชไดเรกทอรีของเน็ตสเคป (c:\progra~1\netscape\users\default\cache) ซึ่งเป็นไดเรกทอรีที่ถูกกำหนดโดยเน็ตสเคปตอนติดตั้งโปรแกรมครั้งแรก โดยแฟ้มข้อมูล FAT.DB มีโครงสร้างการจัดเรียงตำแหน่งไบต์แคชตามที่กำหนดไว้ในแฟ้มข้อมูล Extcache.h ที่สามารถพบได้ภายใต้ ไดเรกทอรีของโปรแกรมต้นฉบับของ mozilla (1998/04/29) (c:\mozilla\lib\libnet\extcache.h) ซึ่งโครงสร้างการจัดเรียงตำแหน่งไบต์แคชที่นำมาแสดงในตารางที่ 2.2 เป็นการจัดเรียงลำดับไบต์ของแคชเวอร์ชัน 5

การนำเอาแฟ้มข้อมูลแคชออกจากดิสก์แคช เน็ตสเคปอาศัยการพิจารณาจาก "เขตข้อมูล Expires" ¹ ของในแต่ละแฟ้มข้อมูลแคชนั้น ๆ โดยอาศัยหลักการ LRU (least-recently-used) โดยพิจารณาจากเวลาการเข้าถึงครั้งสุดท้าย (Last Accessed Time) และเวลาการแก้ไขครั้งสุดท้าย (Last Modified Time) แต่แฟ้มข้อมูลแคชที่จะถูกนำออกไปก่อน จะเป็นแฟ้มข้อมูลแคชที่มีเวลาการเข้าถึงครั้งสุดท้าย และเวลาการแก้ไขครั้งสุดท้ายไม่สอดคล้องกัน หรือเป็นแฟ้มข้อมูลแคชที่ไม่ได้มาจาก ยูอาร์แอลของเกณฑ์วิธีการส่งข้อความหลายมิติ (HTTP URL)

- หน่วยความจำแคช

หน่วยความจำแคชจะใช้เพื่อทำการแคชเอกสารที่ต้องการความปลอดภัย เอกสารประเภทจดหมาย และข่าว โดยทั่วไปเอกสารต่าง ๆ จะถูกเก็บในดิสก์แคช ไม่ใช่ในหน่วยความจำแคช แต่หน่วยความจำแคชอาจถูกใช้ได้ในกรณีที่ดิสก์แคชไม่สามารถใช้งานได้

¹ ตารางที่ 2.2

ตารางที่ 2.2 แสดงรายละเอียดของเขตข้อมูลในแฟ้มข้อมูล FAT.DB

ชนิดข้อมูล	รายละเอียด
int32	Size of the entire structure ⁺
int32	Version of the structure format ⁺ (CACHE_FORMAT_VERSION)
time_t	Last_modified
time_t	Last_accessed
time_t	Expires
uint32	Content_length ⁺
Bool	is_netsite ⁻
time_t	Lock_date
Char	Filename*
int32	Filename_len ⁺
int32	Security_on ⁺
unsigned char	sec_info*
int32	Method ⁺
Char	address*
uint32	Post_data_size
Char	post_data*
Char	post_headers*
Char	content_type*
Char	content_encoding*
Char	charset*
Bool	Incomplete_file ⁻
uint32	Total_content_length ⁺
Char	page_services_url *

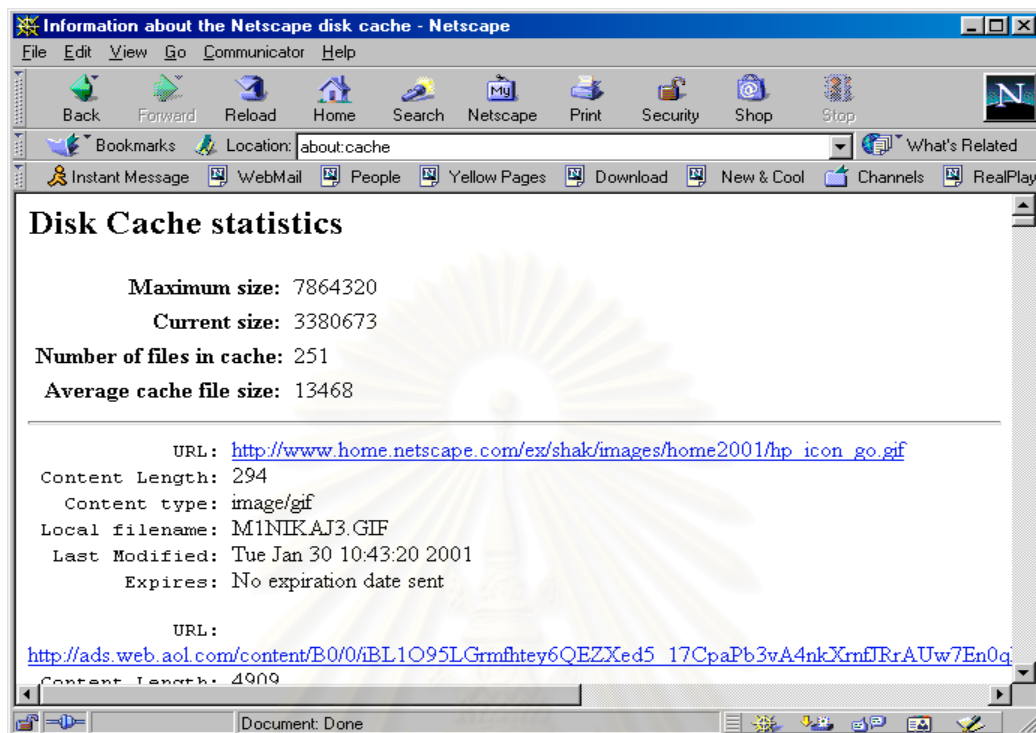
* string lengths all include null terminators

⁺ all integer constants are stored as 4 bytes

⁻ all booleans are stored as one byte

เน็ตสเคปได้เตรียมส่วนของการตรวจสอบว่ามีแฟ้มข้อมูลใดบ้างถูกแคชอยู่ที่ใดโดยผ่าน
 โพรโทคอล About: ซึ่งมีรูปแบบการทำงานดังนี้ คำสั่ง About:cache จะแสดงสิ่งที่เก็บอยู่ในดิสก์

แคช และคำสั่ง About:cache?long จะแสดงสิ่งที่เก็บอยู่ดิสก์แคชแบบมีรายละเอียดมากขึ้น ซึ่งในรูปที่ 2.3 จะแสดงผลการทำงานของคำสั่ง About:cache



รูปที่ 2.3 แสดงการทำงานของคำสั่ง About:cache และผลลัพธ์

2.5 โปรแกรมช่วยงาน

โปรแกรมช่วยงานในการทำวิทยานิพนธ์มี 2 โปรแกรมดังนี้

- โปรแกรม cachedump.exe
- โปรแกรมตัวบริการแทนของเกณฑ์วิธีการส่งข้อความหลายมิติแบบเฉพาะที่ (Local Proxy Server Program)

2.5.1 โปรแกรม cachedump.exe

โปรแกรม cachedump.exe เป็นโปรแกรมที่ได้จากการพัฒนาจากโปรแกรมต้นฉบับที่ได้จากการโหลดโปรแกรมต้นฉบับ mozilla โดยโปรแกรมต้นฉบับของโปรแกรม cachedump.exe สามารถพบได้ภายใต้ไดเรกทอรีชื่อของโปรแกรมต้นฉบับ mozilla (c:\mozilla\lib\net\cachedump.c) ซึ่งนำมาทำการคอมไพล์ใหม่จะได้เป็น cachedump.exe เป็นโปรแกรมที่ทำงานบนดอสโหมด ผู้ใช้สามารถสำเนาโปรแกรมดังกล่าวลงไปในแคชไดเรกทอรีของเน็ตสเคป จากนั้นทำการเปิดหน้าต่างดอส ทำการเรียกชื่อโปรแกรม ตามด้วยชื่อแฟ้มของฐานข้อมูลแคช (ตัวอย่างการใช้งาน cachedump fat.db) โปรแกรมจะทำการเทข้อมูล (dump) แคชแต่ละแฟ้มข้อมูลที่ถูกรวบกันที่ข้อมูลอยู่ในฐานข้อมูลแคชนั้นออกมาทีละตัว

โปรแกรม `cachedump.exe` นี้มีประโยชน์ต่อหัวข้อการวิจัยนี้เป็นอย่างมาก โดยผู้วิจัยใช้เป็นตัวอย่างสำหรับการใช้งานฟังก์ชันการจัดการกับฐานข้อมูลแคชของโปรแกรมเน็ตสเคป สำหรับตัวอย่างการใช้งาน และผลลัพธ์ที่ได้แสดงในรูปแบบที่ 2.4

```

C:\Program Files\Netscape\Users\default\Cache>cachedump fat.db
URL: http://www.samarts.com/images/b2_customerserv.gif
file: M0PE73U6.GIF
is_relative_path: TRUE
content_type: image/gif
content_length: 709
last_modified: Sat Mar 17 22:21:48 2001
-----

```

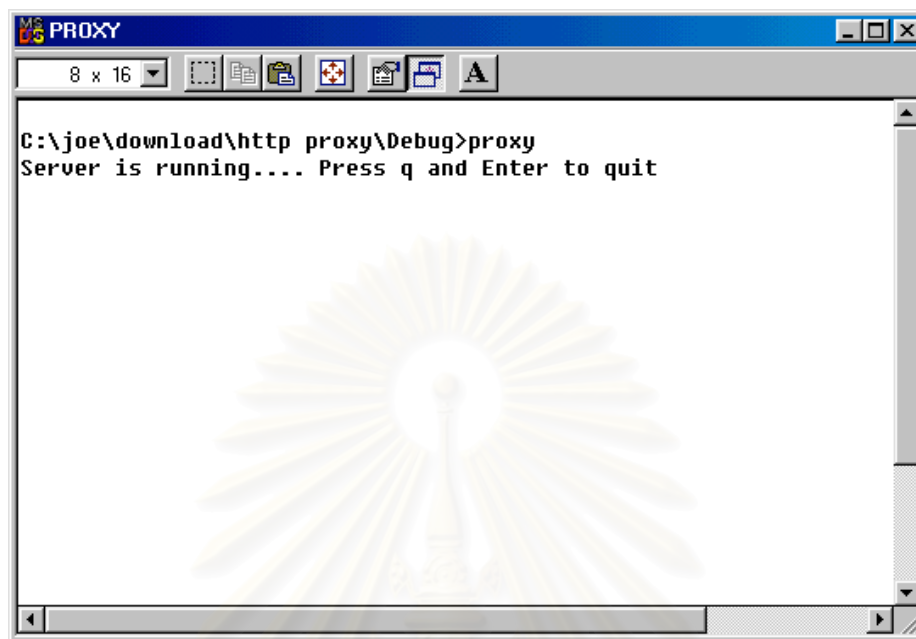
รูปที่ 2.4 แสดงตัวอย่างการใช้งานโปรแกรม `cachedump.exe` และผลลัพธ์

2.5.2 โปรแกรมตัวบริการแทนของเกณฑ์วิธีการส่งข้อความหลายมิติแบบเฉพาะที่ (Local HTTP Proxy Server Program)

โปรแกรมตัวบริการแทนของเกณฑ์วิธีการส่งข้อความหลายมิติแบบเฉพาะที่ เป็นโปรแกรมที่ทำงานอยู่ในแบบดอสโหมดเช่นเดียวกัน ทำหน้าที่เป็นตัวบริการแทนให้กับโปรแกรมค้นผ่านเว็บ โดยที่โปรแกรมตัวบริการแทนนี้จะรอรับคำสั่ง (Listen) อยู่ที่หมายเลขช่องทาง (Port Number) ที่ 5000

การใช้งานก็เพียงแค่เปิดหน้าต่างดอส จากนั้นก็ทำการเรียกโปรแกรม `Proxy.exe` ขึ้นมาทำงาน ในขณะเดียวกันถ้าเราต้องการให้โปรแกรมค้นผ่านเว็บทำงานผ่านโปรแกรมตัวบริการแทนที่เราดำเนินงานไว้ จะต้องเข้าไปแก้ไขโครงแบบ (configuration) ที่เกี่ยวกับการใช้งานโปรแกรมค้นผ่านเว็บ สำหรับโปรแกรมเน็ตสเคปให้เข้าไปแก้ไขโครงแบบที่เมนู `edit` หัวข้อ `preference` ให้เพิ่มหมายเลขไอพีของเครื่องคอมพิวเตอร์ ที่โปรแกรมตัวบริการแทนดำเนินงานอยู่เข้าไป ในที่นี้ใช้ `127.0.0.1` กรณีที่โปรแกรมตัวบริการแทนทำงานอยู่บนเครื่องเดียวกับโปรแกรมค้นผ่านเว็บ หรือหมายเลขไอพีของเครื่อง ที่ทำการดำเนินงานโปรแกรมตัวบริการแทน และกำหนดหมายเลขช่องทางเป็น 5000

การออกจากโปรแกรมตัวบริการแทนนี้ ทำได้โดยการกดแป้นอักษร q แล้วตามด้วย Enter
 ดังแสดงการใช้งานโปรแกรม Proxy.exe ในรูปที่ 2.5



รูปที่ 2.5 แสดงการใช้งานโปรแกรมตัวบริการแทน

ประโยชน์ของการใช้โปรแกรมนี้ในการทำวิทยานิพนธ์ คือการใช้โปรแกรม Proxy.exe ที่ทำการแก้ไขบางส่วนของโปรแกรมให้สามารถทำการบันทึกเวลาสิ้นสุดสำหรับการโหลดเว็บเพจได้ โดยการเพิ่มคำสั่งในการจับเวลาและคำสั่งเขียนเพิ่มข้อมูลลงในฟังก์ชัน ProxyToServer() ดังรูปที่ 2.6

```
// Read data from remote and send data to local
UINT ProxyToServer(LPVOID pParam)
{
    .....
    // cook up a string to send
    while(!pPar->pPair->IsProxy_ServerClosed &&!pPar->pPair->IsUser_ProxyClosed)
    {
        retval = recv(conn_socket,Buffer,sizeof (Buffer),0 );
        if (retval == SOCKET_ERROR ) {
            fprintf(stderr,"recv() failed: error %d\n",WSAGetLastError());
            closesocket(conn_socket);
            pPar->pPair->IsProxy_ServerClosed=TRUE;
            break;
        }
        Len=retval;
        if (retval == 0) {
            // record time
            gwTick = GetTickCount();
            fprintf(stream1,"%ldEND\n",gwTick);
            // when server close connection record time
            printf("Server closed connection\n");
            .....
            return 1;
        }
    }
}
```

รูปที่ 2.6 แสดงการแก้ไขฟังก์ชัน ProxyToServer() ซึ่งเป็นฟังก์ชันการทำงานของโปรแกรม Proxy.exe

บทที่ 3 การออกแบบโปรแกรม

เทคนิคต่าง ๆ ที่ใช้ในการเขียนโปรแกรมสำหรับงานวิจัยเพื่อควบคุมการทำงานของเน็ตสเคป การออกแบบ และการแบ่งส่วนของโปรแกรมเพื่อให้โปรแกรมค้นผ่านเว็บโดยใช้เว็บแคชร่วมกัน ซึ่งมีรายละเอียดในหัวข้อต่าง ๆ ดังนี้

- แนวคิดในการพัฒนาโปรแกรม
- การออกแบบโปรแกรม

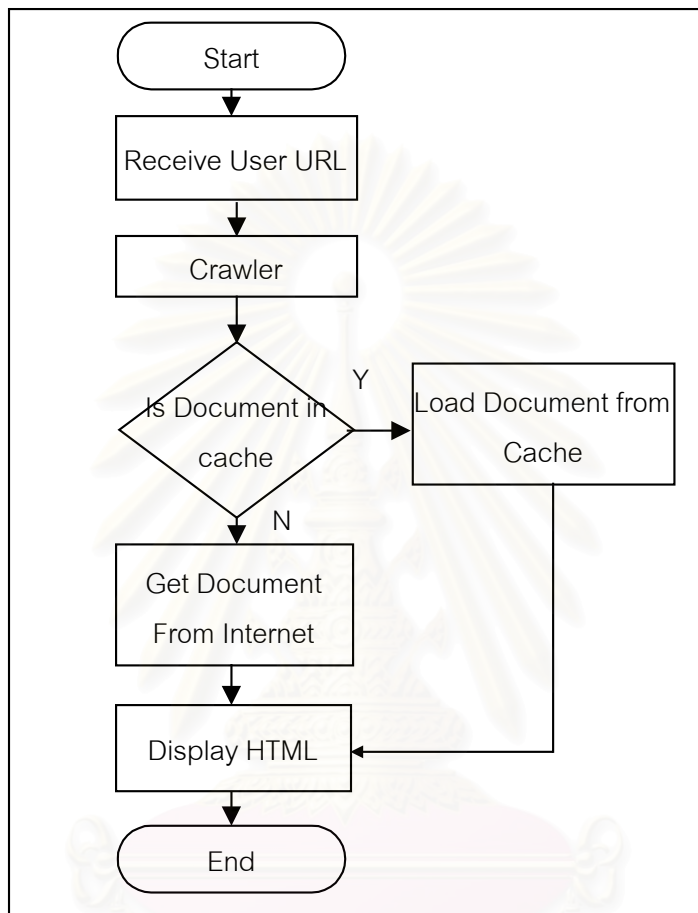
3.1 แนวคิดในการพัฒนาโปรแกรม

จากการศึกษากระบวนการทำงานในการใช้งานโปรแกรมเน็ตสเคป เพื่อทำการโหลดเว็บเพจจากตัวบริการเว็บ พบว่ามีขั้นตอนการทำงานดังนี้ เริ่มจากการที่ผู้ใช้ระบุยูอาร์แอล ที่ต้องการให้โปรแกรมเน็ตสเคปไปนำข้อมูลมาแสดง โปรแกรมเน็ตสเคปจะต้องทำการหาว่าจะต้องทำการโหลดเพิ่มข้อมูลใดบ้าง (Crawling) และเมื่อทราบว่าจะต้องทำการโหลดเพิ่มข้อมูลใดบ้างแล้ว โปรแกรมเน็ตสเคปก็จะทำการสำรวจเว็บแคชเฉพาะที่ของตนเอง ว่ามีเพิ่มข้อมูลดังกล่าวอยู่ในเว็บแคชเฉพาะที่หรือไม่ กรณีที่ไม่พบเพิ่มข้อมูลดังกล่าวในเว็บแคชเฉพาะที่ของตนเอง โปรแกรมเน็ตสเคปจะทำการออกคำสั่ง GET² เพื่อร้องขอไปยังตัวบริการเว็บในเครือข่ายอินเทอร์เน็ตเพื่อนำเพิ่มข้อมูลดังกล่าวมาสู่ผู้ใช้ ส่วนสำหรับกรณีที่พบเพิ่มข้อมูลในเว็บแคชเฉพาะที่ของตนเอง โปรแกรมเน็ตสเคปก็ไม่จำเป็นต้องออกคำสั่ง GET เพื่อร้องขอไปยังตัวบริการเว็บเพื่อนำเพิ่มข้อมูลมา แต่จะใช้เพียงคำสั่งตรวจสอบว่าเพิ่มข้อมูลดังกล่าวยังสามารถใช้งานได้หรือไม่ ซึ่งการทำงานของโปรแกรมเน็ตสเคปสามารถแสดงขั้นตอนการทำงานดังกล่าว ได้ดังรูปที่ 3.1

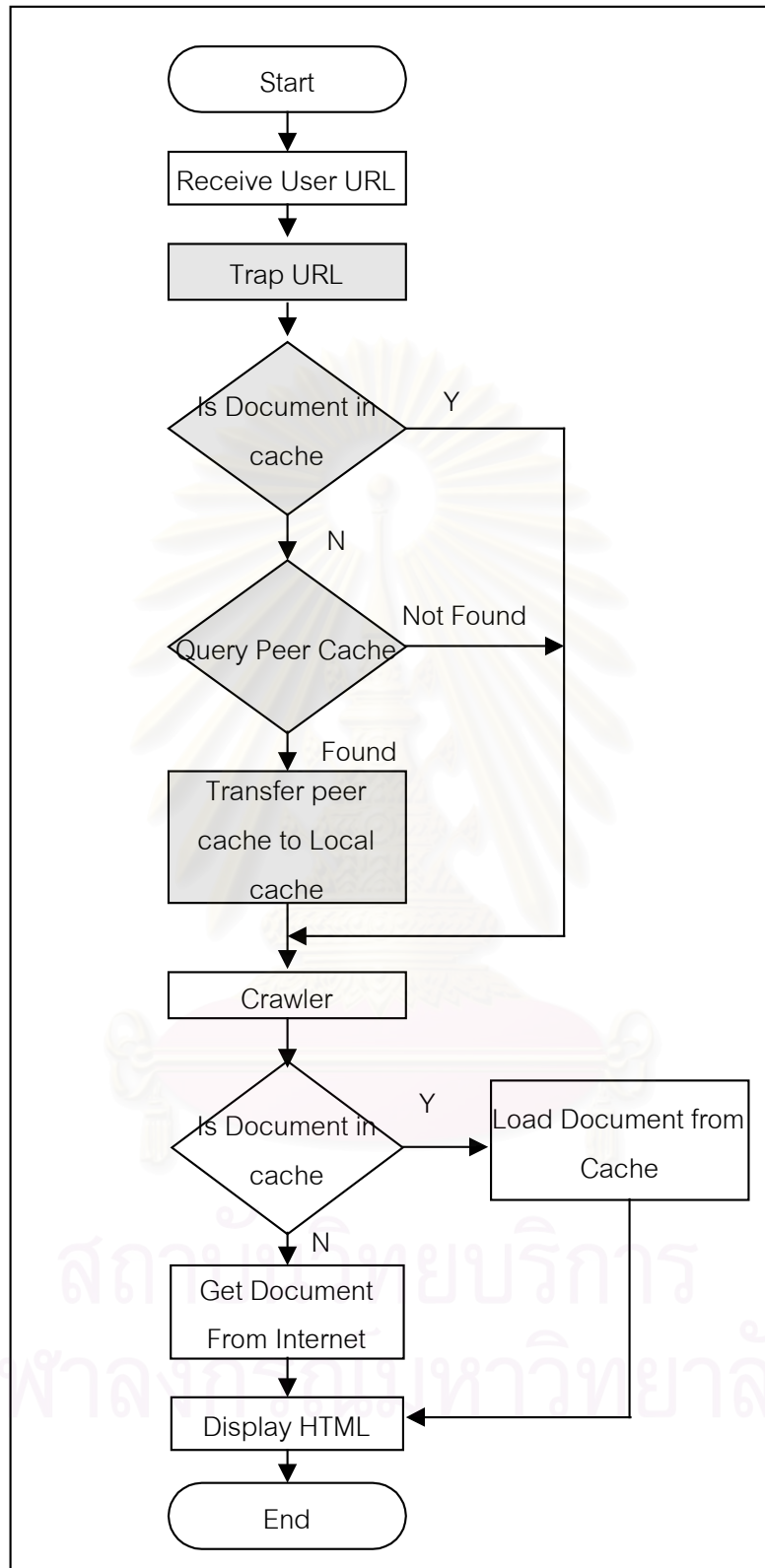
ซึ่งจากขั้นตอนการทำงานของโปรแกรมเน็ตสเคปนี้ จะเห็นว่าก่อนการทำการร้องขอเพิ่มข้อมูลใด ๆ ไปยังตัวบริการเว็บ โปรแกรมเน็ตสเคปต้องทำการสำรวจเพิ่มข้อมูลดังกล่าวในเว็บแคชเฉพาะที่ของตนเองก่อน ซึ่งในขั้นตอนของการสำรวจเพิ่มข้อมูลนี้ ถ้าสามารถดำเนินการทำงานของโปรแกรมเน็ตสเคป ก่อนขั้นตอนที่โปรแกรมจะทำการสำรวจเว็บแคชเฉพาะที่ของตนเอง โดยทำการสอดแทรกขั้นตอนการทำงาน ในการไปนำเว็บแคชที่โปรแกรมเน็ตสเคปต้องการที่อยู่ใน

² RFC 2068 สำหรับเกณฑ์วิธีการส่งข้อความหลายมิติรุ่น 1.1 และ RFC 1945 สำหรับเกณฑ์วิธีการส่งข้อความหลายมิติรุ่น 1.0

ไคลเอนต์อื่นในข่ายงานเฉพาะที่เดียวกัน มาทำการใส่เข้าไปในเว็บแคชเฉพาะที่ของเครื่องคอมพิวเตอร์ที่โปรแกรมเน็ตสเคปนั้นทำงานอยู่ แล้วจึงค่อยให้โปรแกรมเน็ตสเคป เข้าไปสำรวจเว็บแคชเฉพาะที่ของตน ดังแสดงในรูปที่ 3.2 ส่วนที่แรกคือส่วนของการทำงานที่เพิ่มเข้าไป



รูปที่ 3.1 แสดงขั้นตอนการทำงานของโปรแกรมเน็ตสเคป



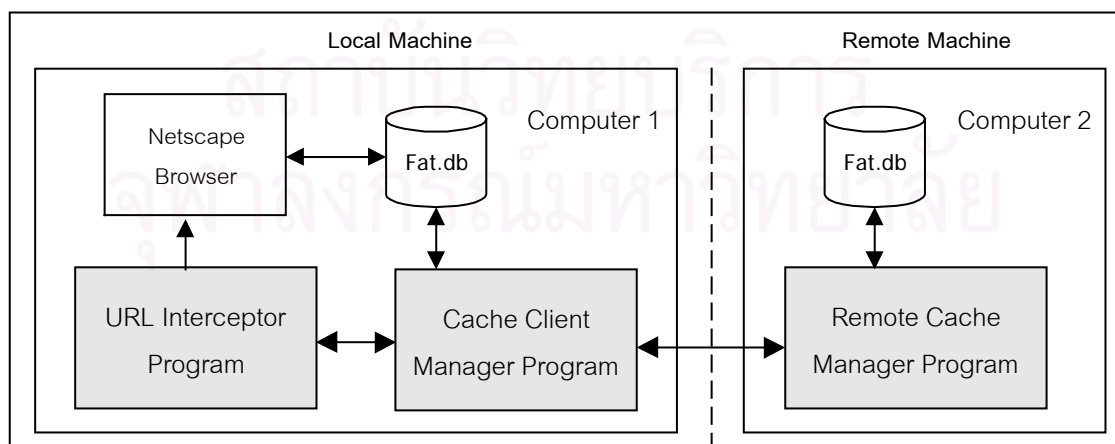
รูปที่ 3.2 แสดงการสอดแทรกขั้นตอนการหาเว็บแคชที่ต้องการจากไคลเอนต์อื่นมาได้ในแคชเฉพาะที่

3.2 การออกแบบโปรแกรม

เมื่อพิจารณาถึงหลักการทำงานที่นำเสนอไปข้างต้น สามารถนำมาออกแบบโปรแกรม “การพัฒนาโปรแกรมเพื่อให้โปรแกรมค้นผ่านเว็บใช้งานเว็บแคชร่วมกัน” ร่วมกับการศึกษาการใช้งานเว็บแคชร่วมกันอย่างง่าย ดังที่กล่าวมาแล้วในหัวข้อ 2.1.3 ทำให้การออกแบบโปรแกรมนี้สามารถแบ่งออกได้เป็น 3 ส่วน ดังนี้

- โปรแกรมที่ใช้ทำการดักยูอาร์แอลที่โปรแกรมเน็ตสเค็ปต้องการโหลด หรือ UIC (URL Interceptor Program) เป็นโปรแกรมส่วนของการดักยูอาร์แอล ที่โปรแกรมเน็ตสเค็ปต้องการโหลด
- โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ หรือ CCM (Cache Client Manager Program) เป็นโปรแกรมส่วนที่รับการทำงานต่อจากในส่วนของโปรแกรมที่ใช้ทำการดักยูอาร์แอลที่โปรแกรมเน็ตสเค็ปต้องการโหลด เพื่อทำการค้นหาเว็บแคชที่อยู่ในไคลเอนต์อื่นในข่ายงานบริเวณเฉพาะที่ โดยอาศัยการแพร่กระจายข้อความ (Broadcast Message) ไปในข่ายงานบริเวณเฉพาะที่ และจัดการในส่วนของการบริหารฐานข้อมูลแคช ในส่วนของการรับเข้าสู่ฐานข้อมูลแคช
- โปรแกรมสำหรับการจัดการแคชระยะไกล หรือ RCM (Remote Cache Manager Program) ทำหน้าที่ในส่วนของการรับการแพร่กระจายข้อความของไคลเอนต์ ทำการจัดการข้อความที่ร้องขอเข้ามาของไคลเอนต์อื่น ทำการบริหารฐานข้อมูลแคชในส่วนของ การค้นหา ทำการสร้างการเชื่อมต่อและโอนย้ายเพิ่มข้อมูลไปยังไคลเอนต์ที่ร้องขอเข้ามา

จากการอธิบายข้างต้น ในการแยกองค์ประกอบของโปรแกรมออกเป็น 3 ส่วนหลักนั้น สามารถแสดงให้เห็นถึงโครงสร้างของโปรแกรม และความสัมพันธ์ขององค์ประกอบทั้งสามของโปรแกรม ดังรูปที่ 3.3

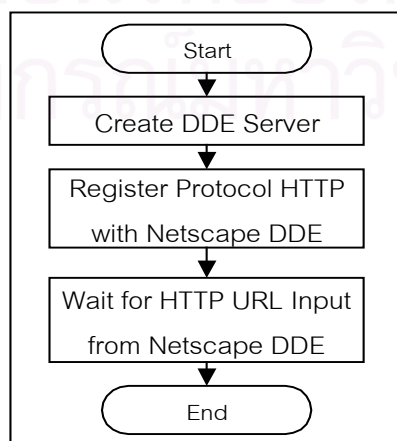


รูปที่ 3.3 แสดงความสัมพันธ์ขององค์ประกอบของโปรแกรม

3.2.1 โปรแกรมที่ใช้ทำการดักยูอาร์แอลที่โปรแกรมเน็ตสเค็ป ต้องการโหลด หรือ UIC (URL Interceptor Program)

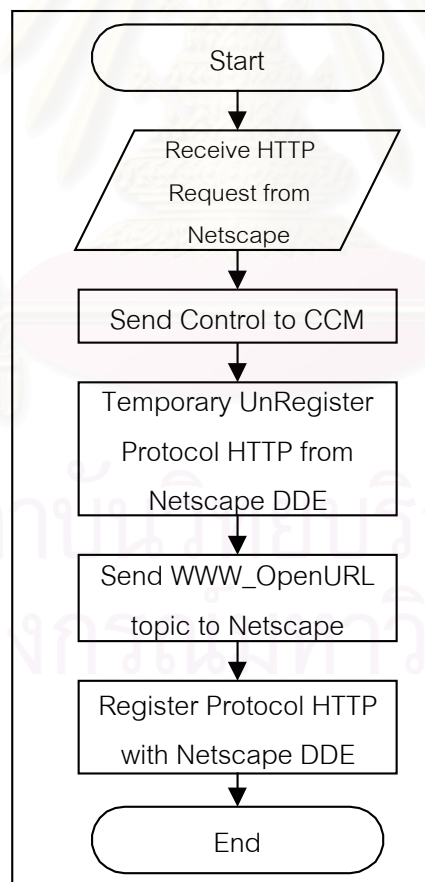
สำหรับโปรแกรมส่วนของการดักยูอาร์แอลที่โปรแกรมเน็ตสเค็ปต้องการโหลดนี้ สามารถใช้ลักษณะการทำงานแบบดีดีอีซึ่งโปรแกรมเน็ตสเค็ปสนับสนุนการทำงานดังกล่าว สำหรับงานวิจัยนี้ได้ทำการพัฒนาโปรแกรมในส่วนนี้ในลักษณะการทำงานภายใต้ดีดีอี ซึ่งใช้เป็นลักษณะในการสื่อสารระหว่างโปรแกรมโดยการส่งข้อความระหว่างกัน ในรูปแบบของตัวอักษร ในช่วงของการดำเนินงาน ซึ่งโปรแกรมเน็ตสเค็ปจะมีโปรแกรมบริการดีดีอีเป็นมาตรฐาน มีลักษณะในการทำงานคือ ต้องมีการระบุชื่อของโปรแกรมบริการดีดีอีและหัวข้อบริการ ซึ่งชื่อโปรแกรมบริการดีดีอีในที่นี้มีชื่อมาตรฐานสำหรับโปรแกรมเน็ตสเค็ป รุ่นที่ 4.x ว่า “NETSCAPE” และมีหัวข้อบริการต่าง ๆ เช่น หัวข้อบริการ WWW_OpenURL ใช้สำหรับร้องขอเพื่อให้โปรแกรมเน็ตสเค็ปทำการเปิดยูอาร์แอลในหน้าต่างของเน็ตสเค็ป หรือหัวข้อบริการ WWW_RegisterProtocol ใช้เพื่อลงทะเบียนโพรโทคอลกับโปรแกรมบริการดีดีอีของเน็ตสเค็ป เพื่อที่จะเป็นตัวบอกโปรแกรมเน็ตสเค็ปว่าถ้ามีการนำเข้ายูอาร์แอลที่มีการลงทะเบียนโพรโทคอลดังกล่าวไว้ ให้โปรแกรมเน็ตสเค็ปทำการส่งการควบคุมให้กับโปรแกรม หรือโปรแกรมบริการดีดีอีที่ระบุขอเป็นตัวจัดการโพรโทคอลดังกล่าว เป็นต้น

โปรแกรมที่ทำการพัฒนาขึ้นในส่วนนี้ จะต้องทำการสร้างโปรแกรมบริการดีดีอี เซิร์ฟเวอร์/ไคลเอนต์ขึ้นมา เพื่อแลกเปลี่ยนข้อมูลกับโปรแกรมบริการดีดีอีของเน็ตสเค็ป เพื่อทำการดักไม่ให้โปรแกรมเน็ตสเค็ปไปทำการโหลดเว็บเพจ โดยการทำการลงทะเบียนเทคนิควิธีการส่งข้อความหลายมิติกับโปรแกรมบริการดีดีอีของเน็ตสเค็ป ผ่านทางหัวข้อบริการของโปรแกรมบริการดีดีอีของเน็ตสเค็ป โดยใช้หัวข้อบริการ WWW_RegisterProtocol เมื่อทำการลงทะเบียนเทคนิควิธีการส่งข้อความหลายมิติกับโปรแกรมบริการดีดีอีของเน็ตสเค็ปแล้ว ทุกยูอาร์แอลที่ถูกระบุว่าใช้เทคนิควิธีการส่งข้อความหลายมิติในการสื่อสาร จะถูกส่งมายังโปรแกรมบริการดีดีอีที่สร้างขึ้น ดังแสดงการทำงานในรูปที่ 3.4



รูปที่ 3.4 แสดงการทำงานของส่วนดักยูอาร์แอล

เมื่อโปรแกรมบริการดีดีอีที่สร้างขึ้นได้รับยูอาร์แอลจากโปรแกรมบริการดีดีอีของเน็ตสเค็ป ก็จะไปเรียกการทำงานในโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ เพื่อทำการค้นหา แคชในข่ายงานบริเวณเฉพาะที่ และหลังจากจบการทำงานในส่วนของโปรแกรมสำหรับการจัดการ แคชในส่วนของไคลเอนต์ โปรแกรมบริการดีดีอีที่สร้างขึ้นจะทำการร้องขอไปยัง โปรแกรมบริการ ของเน็ตสเค็ป เพื่อขอยกเลิกการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติเป็นการชั่วคราว เพื่อให้โปรแกรมบริการดีดีอีของเน็ตสเค็ป ทำการจัดการกับเกณฑ์วิธีการส่งข้อความหลายมิติด้วยตัว ของโปรแกรมบริการดีดีอีของเน็ตสเค็ปเอง โดยการใช้หัวข้อบริการ WWW_UnRegisterProtocol เมื่อได้ทำการยกเลิกการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติแล้ว จากนั้นจึงเรียกใช้หัวข้อ บริการ WWW_OpenURL ของโปรแกรมบริการดีดีอีของเน็ตสเค็ป เพื่อทำการร้องขอให้โปรแกรม เน็ตสเค็ปทำการโหลดเว็บเพจดังกล่าว หลังจากการร้องขอสิ้นสุดลง โปรแกรมบริการดีดีอีที่สร้าง ขึ้นจะทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติ กับโปรแกรมบริการดีดีอีของเน็ตสเค็ป ผ่านหัวข้อบริการ WWW_RegisterProtocol อีกครั้งหนึ่ง ซึ่งจะทำอย่างนี้ทุกครั้งที่โปรแกรม เน็ตสเค็ป ต้องการโหลดเว็บเพจ ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 แสดงขั้นตอนการทำงานหลังจากโปรแกรมบริการดีดีอีที่สร้างขึ้นได้รับยูอาร์แอลเข้าจากโปรแกรม บริการดีดีอีของเน็ตสเค็ป

3.2.2 โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ หรือ CCM (Cache Client Manager Program)

โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ เป็นส่วนหนึ่งของโปรแกรมที่ทำหน้าที่จัดการกับยูอาร์แอลของเกณฑ์วิธีการส่งข้อความหลายมิติ ที่ได้รับจากส่วนของโปรแกรมที่ใช้ทำการดึงยูอาร์แอลที่โปรแกรมเน็ตสเคปต้องการโหลด เมื่อได้รับยูอาร์แอลแล้ว โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ จะทำการตรวจสอบว่ามีแฟ้มข้อมูลแคชที่สัมพันธ์กับยูอาร์แอลดังกล่าว อยู่ในฐานข้อมูลแคชหรือไม่ ซึ่งอาจใช้วิธีการเขียนโปรแกรมเพื่ออ่านแฟ้มข้อมูล FAT.DB โดยตรงหรือใช้ “ฟังก์ชันเอพีไอ (API) สาธารณะสำหรับแคชของโปรแกรมเน็ตสเคป”³ ซึ่งมีชื่อฟังก์ชันว่า NET_IsUrlInDiskCache() เพื่อทำการตรวจสอบยูอาร์แอล ว่ามีการเก็บอยู่ในฐานข้อมูลแคชหรือไม่ แต่การใช้งานฟังก์ชันดังกล่าวมีข้อจำกัดคือ จะตรวจสอบได้เฉพาะในลักษณะที่ต้องทราบยูอาร์แอลแบบเต็ม (Full URL) โดยการระบุยูอาร์แอลแบบเต็ม คือการระบุชื่อเซิร์ฟเวอร์พาธของแฟ้มข้อมูล และชื่อแฟ้มข้อมูลของเอกสารดังกล่าว เช่น “http://www.chula.ac.th/testpage/frame_menu_b_1.gif” แต่ในลักษณะการใช้งานจริง ผู้ใช้จะระบุยูอาร์แอลที่ต้องการจะโหลดเฉพาะในส่วนชื่อเซิร์ฟเวอร์เท่านั้น เช่น “http://www.chula.ac.th” ทำให้การพัฒนาโปรแกรมในส่วนนี้ต้องทำการสร้างลิสต์สำหรับการค้นหาขึ้น โดยทำการสร้างลิสต์ในตอนเริ่มต้นของโปรแกรม และทำการปรับปรุงลิสต์เมื่อโปรแกรมเน็ตสเคปทำการโหลดยูอาร์แอลที่ผู้ใช้ทำการป้อนให้เรียบร้อยแล้ว

สำหรับการค้นหา จะค้นหาจากในลิสต์ที่ทำการสร้างขึ้นจากข้อมูลในดิสก์แคช โดยใช้การค้นหาแบบเรียงลำดับ (Sequential Search) ซึ่งทำให้การพัฒนาโปรแกรมทำได้ง่าย และการค้นหาจากลิสต์ไม่ใช่คอขวดสำหรับโปรแกรม เนื่องจากลิสต์ที่สร้างขึ้นจากข้อมูลในดิสก์แคชนี้ มีข้อมูลน้อยกว่า 300 ± 20 ยูอาร์แอล (สำหรับตัวเลข 300 ± 20 ได้จากการนับจำนวนยูอาร์แอลในลิสต์ที่สร้างขึ้น โดยทดลองให้โปรแกรมเน็ตสเคป ทำการโหลดยูอาร์แอลจำนวน 2000 ยูอาร์แอล จากนั้นเขียนโปรแกรมเพื่อทำการนับ ยูอาร์แอลที่เก็บอยู่ในดิสก์แคช พบว่ามียูอาร์แอลถูกแคชอยู่ประมาณ 300 ± 20 ยูอาร์แอล)

- กรณีที่พบยูอาร์แอลดังกล่าวในฐานข้อมูลแคช โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์จะทำการร้องขอไปยังโปรแกรมบริการดีอีซีของเน็ตสเคป ผ่านหัวข้อ

³ ฟังก์ชันเอพีไอสาธารณะสำหรับแคชของ Netscape เป็นฟังก์ชันเอพีไอ ที่เกี่ยวกับการจัดการแคชเพิ่มของโปรแกรม Netscape สามารถทราบรายละเอียดเพิ่มเติมเกี่ยวกับชื่อฟังก์ชันและวิธีการใช้งานได้ที่ <http://www.mozilla.org/projects/netlib/> [07/96]

บริการ WWW_OpenURL เพื่อร้องขอให้โปรแกรมเน็ตสเคป ทำการเปิดยูอาร์แอลที่ผู้ใช้ทำการป้อนเข้ามา จากนั้นโปรแกรมจะคืนการทำงานไปยังโปรแกรมที่ใช้ทำการดึงยูอาร์แอลที่โปรแกรมเน็ตสเคปต้องการโหลด เพื่อทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติอีกครั้งหนึ่ง

- กรณีที่ทำการตรวจสอบในฐานข้อมูลแคช แล้วไม่พบแฟ้มข้อมูลแคชที่สัมพันธ์กับยูอาร์แอลดังกล่าว โปรแกรมในส่วนนี้จะทำการเตรียมข้อความสำหรับการแพร่กระจาย (Broadcast ของโพรโตคอล IP) ซึ่งสำหรับข้อความดังกล่าว จะเป็นตัวอักษรประกอบด้วยข้อมูล 2 ส่วนคือ หมายเลขไอพีของไคลเอนต์ที่ทำการแพร่กระจายข้อความ และยูอาร์แอลที่ต้องการจะโหลด เมื่อทำการเตรียมในส่วนนี้ของข้อความเรียบร้อยแล้ว ก็ทำการแพร่กระจายข้อความออกไปในข่ายงานบริเวณเฉพาะที่ และรอโปรแกรมสำหรับการจัดการแคชระยะไกลที่ดำเนินงานอยู่บนเครื่องอื่นตอบกลับมา ซึ่งการรอจะให้เวลาในการรอรอยาวนานที่สุดเท่ากับ 2 วินาที (เวลาโดยประมาณสำหรับเวลาตั้งแต่การแพร่กระจายข้อความ จนโปรแกรมสำหรับการจัดการแคชระยะไกลทำการตอบกลับ และทำการโอนย้ายแฟ้มข้อมูลแคช เข้าสู่แคชไดเร็กทอรีของเครื่องที่โปรแกรมสำหรับการจัดการแคชในส่วนไคลเอนต์ดำเนินงานอยู่ และเวลาในการทำการปรับปรุงฐานข้อมูลแคช)

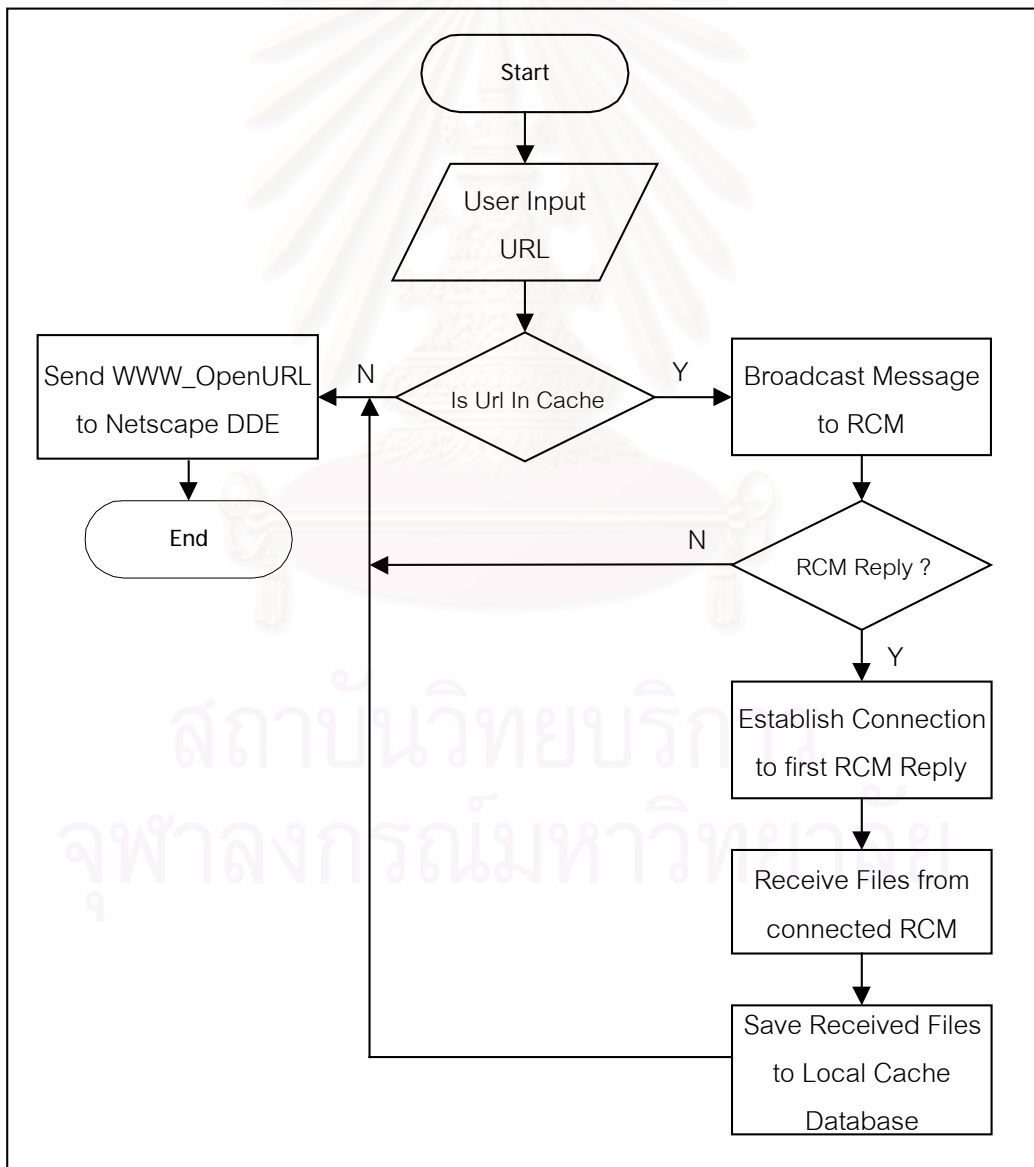
กรณีที่ไม่มีคำตอบกลับภายใน 2 วินาทีจะถือว่าไม่มีเว็บเพจอยู่ในไคลเอนต์อื่นที่อยู่ในข่ายงานบริเวณเฉพาะที่นั้น โปรแกรมสำหรับการจัดการแคชในส่วนนี้ของไคลเอนต์จะทำงานต่อ โดยโปรแกรมสำหรับการจัดการแคชในส่วนนี้ของไคลเอนต์จะทำการร้องขอไปยังโปรแกรมบริการดีเอสไอของเน็ตสเคป ผ่านหัวข้อบริการ WWW_OpenURL เพื่อร้องขอให้โปรแกรมเน็ตสเคป ทำการเปิดยูอาร์แอลที่ผู้ใช้ทำการป้อนเข้ามา จากนั้นโปรแกรมจะคืนการทำงานไปยังโปรแกรมที่ใช้ทำการดึงยูอาร์แอลที่โปรแกรมเน็ตสเคปต้องการโหลด เพื่อทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติอีกครั้งหนึ่ง

กรณีมีการตอบกลับ โปรแกรมสำหรับการจัดการแคชในส่วนนี้ของไคลเอนต์จะทำการสร้างการเชื่อมต่อไปยังโปรแกรมสำหรับการจัดการแคชระยะไกลตัวแรก ที่ทำการตอบกลับว่ามียูอาร์แอลที่ต้องการ ถ้ากรณีมีโปรแกรมสำหรับการจัดการแคชระยะไกลต้องการตอบกลับมากกว่า 1 ตัวโปรแกรมสำหรับการจัดการแคชในส่วนนี้ของไคลเอนต์จะเลือกโปรแกรมสำหรับการจัดการแคชระยะไกลตัวที่ตอบกลับเร็วที่สุดสำหรับโปรแกรมสำหรับการจัดการแคชระยะไกลจะได้กล่าวในหัวข้อถัดไป หลังจากการสร้างการเชื่อมต่อแล้วจะทำการโอนย้ายแฟ้มข้อมูลเข้าสู่แคชไดเร็กทอรี และทำ

การปรับปรุงฐานข้อมูลแคช โดยการใช้ฟังก์ชันเอพีไอสาธารณะสำหรับแคชของเน็ตสเคป ชื่อฟังก์ชัน NET_WriteCacheFAT() เพื่อทำการปรับปรุงฐานข้อมูลแคช

เมื่อทำการโอนย้ายเพิ่มข้อมูลแคชและปรับปรุงฐานข้อมูลแคชเรียบร้อยแล้ว โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์จะทำการร้องขอไปยังโปรแกรมบริการดีอีของเน็ตสเคป ผ่านหัวข้อบริการ WWW_OpenURL เพื่อร้องขอให้โปรแกรมเน็ตสเคป ทำการเปิดยูอาร์แอลที่ผู้ใช้ทำการป้อนเข้ามา จากนั้นโปรแกรมจะคืนการทำงานไปยังโปรแกรมที่ใช้ทำการดึงยูอาร์แอลที่โปรแกรมเน็ตสเคปต้องการไหลด เพื่อทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติอีกครั้งหนึ่ง

สำหรับการอัลกอริทึมของโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ ถูกแสดงอยู่ในรูปที่ 3.6



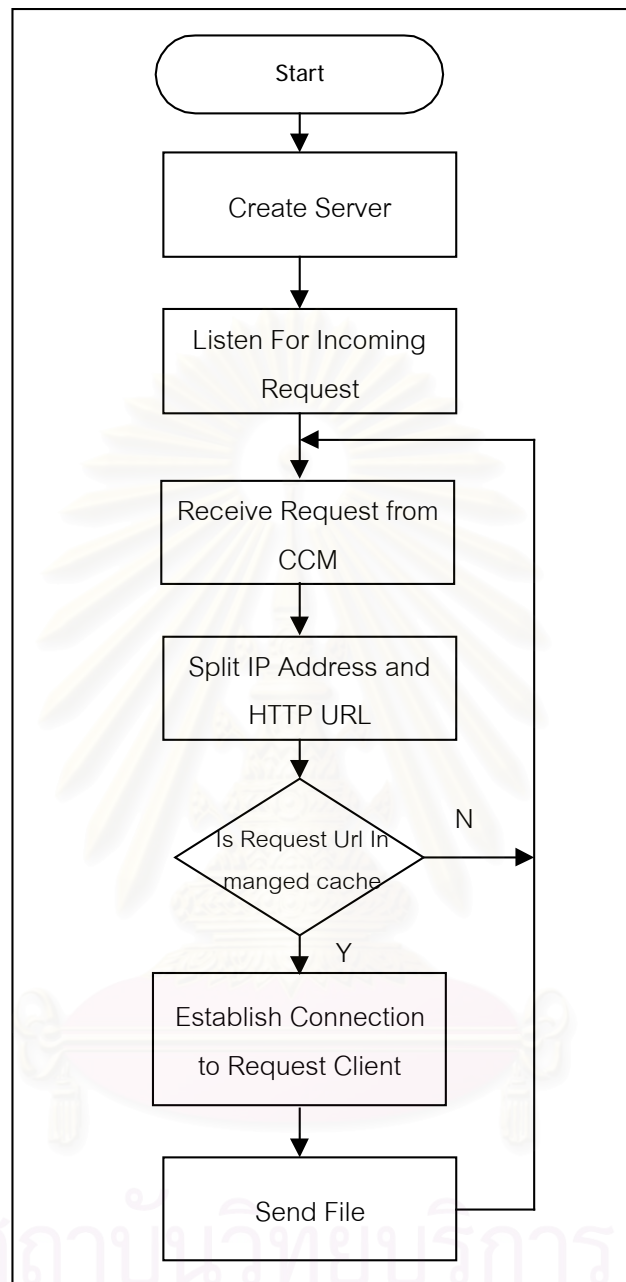
รูปที่ 3.6 แสดงอัลกอริทึมของโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์

3.2.3 โปรแกรมสำหรับการจัดการแคชระยะไกล (Remote Cache Manager Program)

โปรแกรมสำหรับการจัดการแคชระยะไกล ทำหน้าที่ในการรองรับข้อความร้องขอจากโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ กรณีที่โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ไม่พบเพิ่มข้อมูลแคชที่เกี่ยวข้องกับยูอาร์แอลที่ผู้ใช้ร้องขอ เมื่อโปรแกรมสำหรับการจัดการแคชระยะไกล ได้รับคำร้องขอเข้ามาแล้วโปรแกรมสำหรับการจัดการแคชระยะไกลจะทำการตีความหมายของคำร้องขอ โดยแยกรายละเอียดของข้อความ เป็นหมายเลขไอพีของผู้ร้องขอ และยูอาร์แอลที่ผู้ร้องขอต้องการ จากนั้นทำการค้นหายูอาร์แอลดังกล่าวในเพิ่มข้อมูลแคชของเครื่องที่โปรแกรมสำหรับการจัดการแคชระยะไกลทำงานอยู่ โดยการค้นหาจากลิสต์ที่ถูกสร้างขึ้นจากฐานข้อมูลแคช โดยใช้การค้นหาแบบเรียงลำดับ

- กรณีที่ตรวจพบยูอาร์แอลในฐานข้อมูลแคช โปรแกรมสำหรับการจัดการแคชระยะไกล จะทำการสร้างการเชื่อมต่อไปยังโปรแกรมการจัดการแคชในส่วนของไคลเอนต์ และทำการโอนย้ายเพิ่มข้อมูลที่เกี่ยวข้องกับยูอาร์แอลดังกล่าว ส่งไปให้กับโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ เพื่อทำการนำไปปรับปรุงข้อมูลในส่วนของฐานข้อมูลแคช สำหรับกรณีที่มีหลาย ๆ โปรแกรมสำหรับการจัดการแคชระยะไกลพยายามสร้างการเชื่อมต่อไปยังไคลเอนต์ที่ร้องขอยูอาร์แอลเข้ามา โปรแกรมสำหรับการจัดการแคชระยะไกลตัวแรกที่ทำการตอบกลับ จะสามารถสร้างการเชื่อมต่อกับโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ และโอนย้ายเพิ่มข้อมูลได้สำเร็จ ส่วนโปรแกรมสำหรับการจัดการแคชระยะไกลที่ตอบกลับช้ากว่า จะทำการปิดการเชื่อมต่อ และวนซ้ำ (Loop) กลับไปรอรับการแพร่กระจายข้อความอีกครั้ง
- กรณีที่ไม่สามารถตรวจพบยูอาร์แอลที่ร้องขอเข้ามาในฐานข้อมูลแคช ของเครื่องที่โปรแกรมสำหรับการจัดการแคชระยะไกลทำงานอยู่ โปรแกรมจะทำการวนซ้ำเพื่อไปรอรับการแพร่กระจายข้อความอีกครั้ง

สำหรับการทำงานของโปรแกรมสำหรับการจัดการแคชระยะไกลแสดงในรูปแบบที่ 3.7



รูปที่ 3.7 แสดงอัลกอริทึมของโปรแกรมสำหรับการจัดการแคชระยะใกล้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4 รายละเอียดการพัฒนาโปรแกรม

จากการออกแบบโปรแกรมสำหรับการใช้งานเว็บเบราว์เซอร์ร่วมกันของโปรแกรมคั่นผ่านเว็บนั้น ประกอบด้วย 3 ส่วน ดังนี้

- โปรแกรมที่ใช้ทำการดักยูอาร์แอลที่โปรแกรมเน็ตสเคป ต้องการโหลด หรือ UIC (URL Interceptor Program)
- โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ หรือ CCM (Cache Client Manager Program)
- โปรแกรมสำหรับการจัดการแคชระยะไกล หรือ RCM (Remote Cache Manager Program)

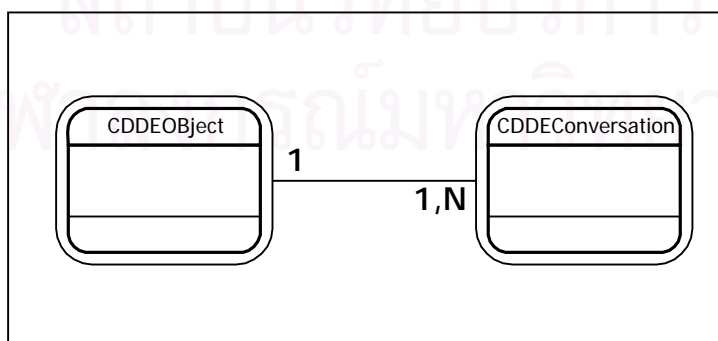
สามารถอธิบายถึงรายละเอียดในการทำงานในแต่ละโปรแกรม ได้ดังต่อไปนี้

4.1 โปรแกรมที่ใช้ทำการดักยูอาร์แอลที่โปรแกรมเน็ตสเคป ต้องการโหลด หรือ UIC (URL Interceptor Program)

โปรแกรมที่ใช้ทำการดักยูอาร์แอลที่โปรแกรมเน็ตสเคป ต้องการโหลดนี้ ทำหน้าที่เป็นโปรแกรมบริการที่ดีซึ่งประกอบด้วยคลาสและฟังก์ชันในการพัฒนาโปรแกรม โดยประกอบด้วย 2 คลาสสำคัญ และฟังก์ชันสำหรับการเรียกย้อนกลับ (CALLBACK Function) ดังนี้

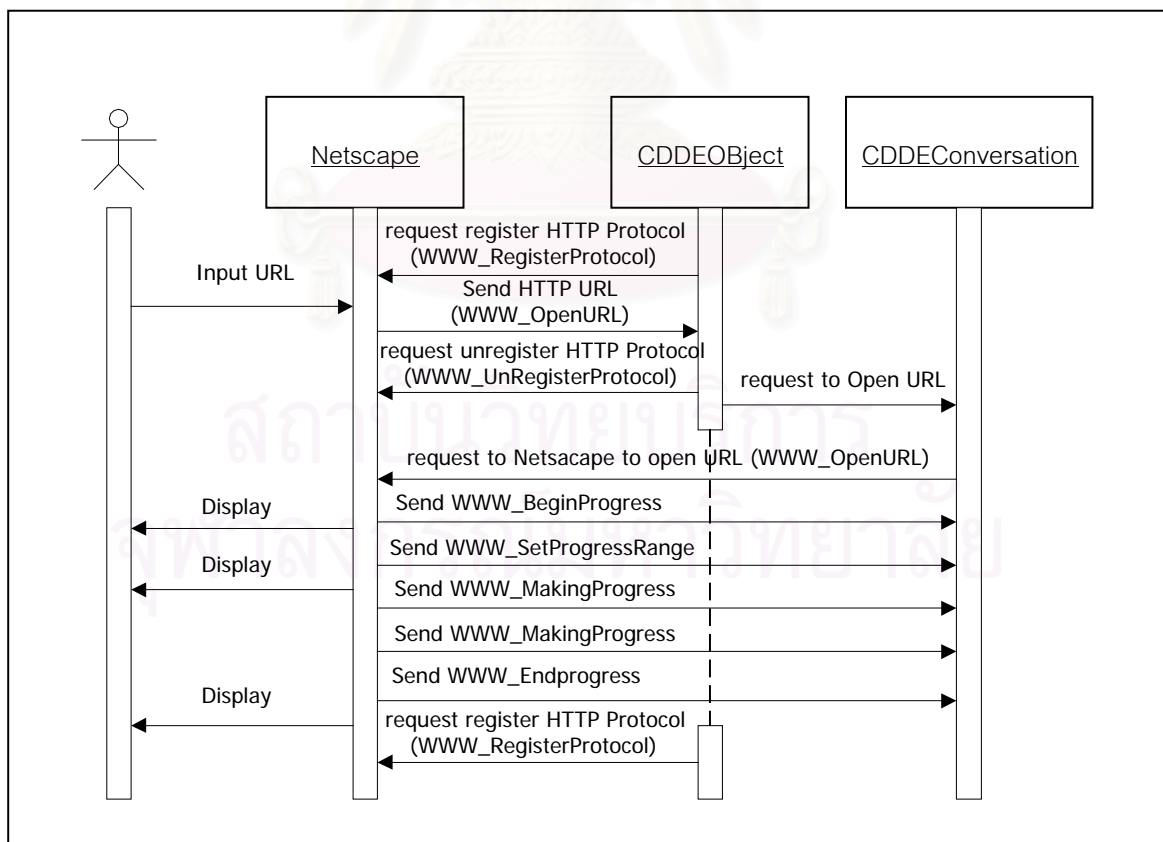
- คลาส CDDEObject
- คลาส CDDEConversation
- ฟังก์ชันสำหรับการเรียกย้อนกลับ NstestDdeCallBack

ซึ่งสามารถแสดงความสัมพันธ์ของทั้งสองคลาสได้ดังรูปที่ 4.1



รูปที่ 4.1 แสดงความสัมพันธ์ระหว่างคลาส CDDEObject และคลาส CDDEConversation

โปรแกรมที่ใช้ทำการดักยูอาร์แอลที่โปรแกรมเน็ตสเคปต้องการโหลดนี้ มีขั้นตอนการทำงานดังนี้คือ เริ่มต้นจากคลาส CDDEObject ทำการเรียกใช้หัวข้อบริการของโปรแกรมบริการดีดีอีของเน็ตสเคป เพื่อทำการลงทะเบียนบนทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติ หลังจากทำการลงทะเบียนเรียบร้อยแล้ว เมื่อผู้ใช้ทำการป้อนยูอาร์แอลของเกณฑ์วิธีการส่งข้อความหลายมิติให้กับโปรแกรมเน็ตสเคป โปรแกรมบริการดีดีอีของเน็ตสเคปจะทำการส่งยูอาร์แอลที่ป้อนเข้ามาให้กับคลาส CDDEObject จากนั้นคลาส CDDEObject ทำการยกเลิกการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติกับโปรแกรมบริการดีดีอีของเน็ตสเคป เมื่อได้ทำการยกเลิกการลงทะเบียนเรียบร้อยแล้ว จึงร้องขอไปยังคลาส CDDEConversation เพื่อทำการเปิดยูอาร์แอล คลาส CDDEObject จะทำการร้องขอกลับไปยังโปรแกรมบริการดีดีอีของเน็ตสเคป เพื่อให้โปรแกรมเน็ตสเคปทำการเปิดยูอาร์แอลดังกล่าว โปรแกรมเน็ตสเคปก็จะเริ่มทำการโหลดเว็บเพจ และคอยส่งข้อความแสดงความก้าวหน้าของการโหลดเว็บเพจที่ผู้ใช้ต้องการ ให้กับคลาส CDDEConversation และจะส่งข้อความสำหรับแสดงว่าการร้องขอได้สิ้นสุดลงแล้วให้กับคลาส CDDEConversation เมื่อคลาส CDDEConversation ได้รับข้อความสำหรับแสดงว่าการร้องขอสิ้นสุดลงแล้ว คลาส CDDEObject ก็จะทำกรร้องขอลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติกับโปรแกรมบริการดีดีอีของเน็ตสเคปอีกครั้ง ดังแสดงในแผนผังลำดับการทำงานในรูปที่ 4.2



รูปที่ 4.2 แสดงแผนผังแบบลำดับการทำงานสำหรับโปรแกรมที่ใช้ทำการดักยูอาร์แอลที่โปรแกรมเน็ตสเคปต้องการโหลด

4.1.1 คลาส CDDEObject

คลาส CDDEObject เป็นคลาสที่เกี่ยวข้องกับการจัดการกับ โปรแกรมบริการดีดีอีที่พัฒนาขึ้นเอง และสนับสนุนหัวข้อบริการบางส่วน เหมือนกันกับโปรแกรมบริการดีดีอีของเน็ตสเคป โดยจะทำการพัฒนาเฉพาะหัวข้อบริการที่ได้กล่าวไว้แล้วในตารางที่ 2.1

ดังนั้นในการกำหนดค่าตัวแปรในคลาสนี้ จะต้องมีการกำหนดค่าแอตทริบิวต์ต่างๆ โดยทำการระบุชนิดของข้อมูลและรายละเอียดของตัวแปรนั้นๆ แสดงในตารางที่ 4.1

ตารางที่ 4.1 แสดงตัวแปรภายในคลาส CDDEObject

แอตทริบิวต์	ชนิดข้อมูล	รายละเอียด
static m_dwIdInst	DWORD	The DDEML instance identifier
Topic	DWORD	Possible Topic
static m_hszBrowser	HSZ	Client Browser DDE Name
static m_aTopics[m_MaxTopics]	HSZ	Topics Array
static m_colRunning	CobList	List or currently object running
m_rIndex	POSITION	Position of object in List
static m_iInstanceCounter	Int	Use to Generate Unique DDEML (Object Running Number)
m_csServiceName	Cstring	Use to Generate Unique DDEML (Object Name)
m_hszServiceName	HSZ	Used to generate unique DDE services on a per instance basis
m_colConversations	CobList	The list of DDE object that this service is handling

ในส่วนของเมธอด (Method) ของคลาส CDDEObject ที่ได้นำมาใช้ในการพัฒนาโปรแกรมสำหรับการใช้งานเว็บแคชร่วมกันของโปรแกรมค้นผ่านเว็บ เพื่อให้โปรแกรมสามารถทำงานได้ตามกระบวนการที่ได้ออกแบบไว้ สามารถแสดงได้ในตารางที่ 4.2

สำหรับการทำงานของโปรแกรมในส่วนนี้ จะทำการสร้างอ็อบเจกต์ของคลาส CDDEObject ขึ้นมาเพียงหนึ่งตัวโดยให้ชื่อของอ็อบเจกต์ดังกล่าวว่า "NSTEST1" เพื่อใช้เป็นชื่อสำหรับเรียกโปรแกรมบริการดีดีอีที่ทำการพัฒนาขึ้นในงานวิจัย พร้อมกับทำการลงทะเบียนฟังก์ชันสำหรับการเรียกย้อนกลับ ซึ่งจะได้กล่าวถัดไปในข้อ 4.1.3 เพื่อรองรับข้อความ ที่ได้จากการร้องขอ และทำการนำไปประมวลผลต่อไป

จากขั้นตอนข้างต้น โปรแกรมจะทำการเรียกใช้เมธอด WWW_RegisterProtocol เพื่อที่จะทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติกับโปรแกรมบริการดีดีไอของเน็ตสเคป การลงทะเบียนมีการทำงานในลักษณะเวลาดำเนินงาน คือจะมีการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติกับโปรแกรมบริการดีดีไอของเน็ตสเคป เมื่อขณะที่ดำเนินงานโปรแกรม หลังจากทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติ กับโปรแกรมบริการดีดีไอของเน็ตสเคปเรียบร้อยแล้ว ทุกยูอาร์แอลที่ผู้ใช้ร้องขอผ่านโปรแกรมเน็ตสเคป ที่มีการระบุว่าจะใช้เกณฑ์วิธีการส่งข้อความหลายมิติ จะถูกส่งกลับมายังโปรแกรมบริการดีดีไอที่พัฒนาขึ้น จากนั้นเมื่อโปรแกรมบริการดีดีไอที่พัฒนาขึ้นได้รับยูอาร์แอลที่ส่งมาจากโปรแกรมบริการดีดีไอของเน็ตสเคปแล้ว โปรแกรมบริการดีดีไอที่พัฒนาขึ้น จึงส่งยูอาร์แอลดังกล่าวให้กับโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ เพื่อทำงานในขั้นตอนถัดไป และหลังจากการทำงานในส่วนของจัดการแคชในส่วนของไคลเอนต์ โปรแกรมบริการดีดีไอที่พัฒนาขึ้นจะทำการเรียกใช้เมธอด WWW_UnRegisterProtocol ในคลาส CDDEObject เพื่อทำการยกเลิกการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติ กับโปรแกรมบริการดีดีไอของเน็ตสเคป การยกเลิกการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติกับโปรแกรมบริการดีดีไอของเน็ตสเคปนี้ เพื่อที่โปรแกรมบริการดีดีไอที่สร้างขึ้น จะสามารถเรียกใช้เมธอด WWW_OpenURL ของคลาส CDDEConversation ซึ่งจะได้กล่าวในหัวข้อถัดไป เพื่อทำการร้องขอไปยังโปรแกรมบริการดีดีไอของเน็ตสเคปผ่านหัวข้อบริการ WWW_OpenURL เพื่อทำการเปิดยูอาร์แอลที่ต้องการ ในลักษณะการทำงานที่กล่าวข้างต้นโปรแกรมบริการดีดีไอของเน็ตสเคปจะทำหน้าที่เป็นเซิร์ฟเวอร์ และโปรแกรมบริการที่สร้างขึ้นจะทำหน้าที่เป็นไคลเอนต์

หลังจากโปรแกรมบริการดีดีไอที่พัฒนาขึ้นทำการร้องขอหัวข้อบริการ WWW_OpenURL ไปยังโปรแกรมบริการดีดีไอของเน็ตสเคปเรียบร้อยแล้ว โปรแกรมบริการดีดีไอของเน็ตสเคปก็จะเริ่มทำการเปิดยูอาร์แอลที่ต้องการ พร้อมกับส่งข้อความความก้าวหน้ากลับมาให้กับโปรแกรมบริการที่สร้างขึ้นผ่านหัวข้อบริการซึ่งเป็นเมธอดภายในคลาส CDDEConversation เมื่อโปรแกรมบริการดีดีไอของเน็ตสเคปทำการเปิดยูอาร์แอลที่ต้องการให้กับโปรแกรมบริการที่สร้างขึ้นเรียบร้อยแล้ว โปรแกรมบริการดีดีไอของเน็ตสเคปก็จะร้องขอมายังโปรแกรมบริการที่สร้างขึ้น ผ่านหัวข้อบริการ WWW_EndProgress ซึ่งเป็นตัวบอกให้กับโปรแกรมบริการที่สร้างขึ้นทราบว่าการร้องขอยูอาร์แอลดังกล่าวได้เสร็จสิ้นลง จากนั้นโปรแกรมบริการที่สร้างขึ้นจะได้ทำการร้องขอหัวข้อบริการ WWW_RegisterProtocol เพื่อทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติกับโปรแกรมบริการดีดีไอของเน็ตสเคปอีกครั้ง

ตารางที่ 4.2 แสดงเมธอดของคลาส CDDEObject

เมธอด	การทำงาน
static CDDEObject *ResolveService(const char *pServiceName)	Get pointer of CDDEObject use Pointer to ServiceName
static CDDEObject *ResolveService(HSZ hszServiceName)	Get pointer of CDDEObject use Pointer to hszServiceName
static CDDEObject *ResolveConversation(CDDEConversation *pConv)	Resove conversation of given pConv
HDDEDATA ServerConnect(HSZ hszTopic)	Connect to Server with hszTopic
void ServerConnectConfirm(HCONV hConv)	Already get connect
static void DDEDisconnect(CDDEConversation *pConv)	Disconnect CDDEConversation from CDDEObject
CString GetServiceName() { return m_csServiceName; }	Return m_csServiceName
HSZ GetServiceHSZ() { return m_hszServiceName; }	Return m_hszServiceName
static int EnumTopic(HSZ hszTopic)	Enumerate Topic
void LoadTopics();	Load Our Topic to m_aTopics[m_MaxTopics]
void FlushTopics()	Flush All Topic
CDDEConversation *ClientConnect(int iTopic)	Return CDDEConversation for client topic
void ServerReturned(HDDEDATA hData, const char *pFormat, ...)	Get what return from server
HSZ ClientArguments(const char *pFormat, ...)	Create HSZ data for Client parameter
DWORD WWW_Activate(DWORD dwWindowID, DWORD dwFlags = 0ul)	Function for WWW_Activate topic
void WWW_CancelProgress(DWORD dwTransactionID)	Function for WWW_CancelProgress topic
void WWW_Exit()	Function for WWW_Exit topic
void WWW_ListWindows()	Function for WWW_ListWindows topic
void WWW_Version()	Function for WWW_Version topic
void WWW_GetWindowInfo(DWORD dwWindowID);	Function for WWW_GetWindowInfo topic

ตารางที่ 4.2 (ต่อ) แสดงเมธอดของคลาส CDDEObject

เมธอด	การทำงาน
void WWW_OpenURL(CString csURL, CString csSaveAs, DWORD dwWindowID, DWORD dwFlags, CString csPostFormData, CString csPostMIMEType, CString csProgressServer)	Function for WWW_ OpenURL topic
void WWW_RegisterProtocol(CString csServer, CString csProtocol)	Function for WWW_ RegisterProtocol topic
void WWW_RegisterURLEcho(CString csServer)	Function for WWW_ RegisterURLEcho topic
void WWW_RegisterViewer(CString csServer, CString csMIMEType, DWORD dwFlags)	Function for WWW_ RegisterViewer topic
void WWW_ShowFile(CString csFileName, CString csMimeType, DWORD dwWindowID, CString csURL)	Function for WWW_ ShowFile topic
void WWW_UnRegisterProtocol(CString csServer, CString csProtocol)	Function for WWW_ UnRegisterProtocol topic
void WWW_UnRegisterURLEcho(CString csServer)	Function for WWW_ UnRegisterURLEcho topic
void WWW_UnRegisterViewer(CString csServer , CString csMIMEType)	Function for WWW_ UnRegisterViewer topic
void CommonConstruction()	Common Construction Code

สำหรับฟังก์ชันสำคัญหรือเมธอดภายในคลาสนี้ ที่พัฒนาขึ้นเพื่อรองรับการทำงานสำหรับประเภทรายการที่เกิดขึ้น ในกรณีที่โปรแกรมบริการดีดีอ็อกซ์ที่สร้างขึ้นทำหน้าที่เป็นเซิร์ฟเวอร์ได้แก่

- WWW_RegisterProtocol เป็นฟังก์ชันที่พัฒนาขึ้นเพื่อทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติกับโปรแกรมบริการดีดีอ็อกซ์ของเน็ตสเคป เป็นการบอกให้โปรแกรมบริการดีดีอ็อกซ์ของเน็ตสเคปทราบว่า ถ้ามียูอาร์แอลที่ระบุโพรโทคอลในการสื่อสาร เป็นยูอาร์แอลของเกณฑ์วิธีการส่งข้อความหลายมิติ ให้ทำการส่งการควบคุมมายังโปรแกรมบริการดีดีอ็อกซ์ที่สร้างขึ้น พร้อมทั้งส่งข้อมูลสำหรับประเภทรายการดังกล่าวมาด้วย ซึ่งก็คือยูอาร์แอลที่ผู้ใช้งานร้องขอ สำหรับการลงทะเบียนโพรโทคอล สามารถทำได้โดยการส่งชื่อของโปรแกรมบริการดีดีอ็อกซ์ที่สร้างขึ้น และต้องการให้เป็นตัวจัดการกับโพรโทคอลที่จะทำการลงทะเบียน ซึ่งในงานวิจัยนี้โปรแกรมบริการดีดีอ็อกซ์ที่สร้างขึ้นมีชื่อ

ว่า “NSTEST1” และโพรโทคอลที่ต้องการลงทะเบียนคือ “HTTP” ดังแสดงในโปรแกรมรูปที่ 4.3

```
//WWW_RegisterProtocol
void CDDEObject::WWW_RegisterProtocol(CString csServer, CString csProtocol) {
    CDDEConversation *pConv = ClientConnect(m_RegisterProtocol);
    TwoByteBool bRegistered = FALSE;

    if(pConv != NULL) {
        HSZ hszItem = ClientArguments("QCS,QCS", &csServer, &csProtocol);
        HDDEDATA hRegistered = pConv->ClientRequest(hszItem);

        if(hRegistered != NULL){
            ServerReturned(hRegistered, "BL", &bRegistered);
        }
        pConv->DoDisconnect();
    }
}
```

รูปที่ 4.3 แสดงเมธอด WWW_RegisterProtocol ของคลาส CDDEObject

- WWW_UnRegisterProtocol เป็นเมธอดของโปรแกรมบริการดีดีซี เพื่อทำการยกเลิกการลงทะเบียนโพรโทคอล ที่ไม่ต้องการจากโปรแกรมบริการดีดีซีของเน็ตสเค๊ป ซึ่งในงานวิจัยนี้ใช้ทำการยกเลิกการลงทะเบียนที่วิธีการส่งข้อความหลายมิติจากโปรแกรมบริการดีดีซีของเน็ตสเค๊ป การยกเลิกการลงทะเบียนโพรโทคอล ทำโดยการระบุชื่อของโปรแกรมบริการดีดีซีที่พัฒนาขึ้นซึ่งในงานวิจัยนี้คือ “NSTEST1” และโพรโทคอลที่ต้องการยกเลิกการลงทะเบียนคือ “HTTP” ดังแสดงในโปรแกรมรูปที่ 4.4

```
//WWW_UnRegisterProtocol
void CDDEObject::WWW_UnRegisterProtocol(CString csServer, CString csProtocol) {
    CDDEConversation *pConv = ClientConnect(m_UnRegisterProtocol);
    TwoByteBool bUnRegistered = FALSE;

    if(pConv != NULL) {
        HSZ hszItem = ClientArguments("QCS,QCS", &csServer, &csProtocol);
        HDDEDATA hUnRegistered = pConv->ClientRequest(hszItem);

        if(hUnRegistered != NULL) {
            ServerReturned(hUnRegistered, "BL", &bUnRegistered);
        }
        pConv->DoDisconnect();
    }
}
```

รูปที่ 4.4 แสดงเมธอด WWW_UnRegisterProtocol ของคลาส CDDEObject

4.1.2 คลาส CDDEConversation

คลาส CDDEConversation เป็นคลาสที่ทำการจัดการเกี่ยวกับการสนทนา (Conversation) ระหว่างโปรแกรมบริการดีดีอี คือโปรแกรมบริการดีดีอีที่พัฒนาขึ้น และโปรแกรมบริการดีดีอีของเน็ตสเค๊ป

ดังนั้นในการกำหนดค่าตัวแปรในคลาสนี้ จะต้องมีการกำหนดค่าแอตทริบิวต์ต่างๆ โดยทำการระบุชนิดของข้อมูลและรายละเอียดของตัวแปรนั้นๆ แสดงในตารางที่ 4.3

ตารางที่ 4.3 แสดงตัวแปรภายในคลาส CDDEConversation

แอตทริบิวต์	ชนิดข้อมูล	รายละเอียด
m_hConv	HCONV	Conversation that we're handling
static m_colConversations	CobList	List of all conversations going on (for informational lookup), and instance
m_rIndex	POSITION	Position of Object in list

ในส่วนของเมธอดของคลาส CDDEConversation ที่ได้นำมาใช้ในการพัฒนาโปรแกรมสำหรับการใช้งานเว็บแคชร่วมกันของโปรแกรมค้นผ่านเว็บ เพื่อให้โปรแกรมสามารถทำงานได้ตามกระบวนการที่ได้ออกแบบไว้ สามารถแสดงได้ในตารางที่ 4.4

อ็อบเจกต์ในคลาสนี้จะถูกสร้างขึ้นก็ต่อเมื่อมีการสร้างการสนทนาเกิดขึ้น เพื่อทำการแลกเปลี่ยนข้อมูล ระหว่างโปรแกรมบริการดีดีอีที่ได้พัฒนาขึ้นและโปรแกรมบริการดีดีอีของเน็ตสเค๊ป โดยมีการจัดการในรูปแบบลิงคิลิสต์ เมื่อการสนทนาจบลงอ็อบเจกต์ดังกล่าวที่เป็นตัวแทนของการสนทนาดังกล่าวก็จะถูกทำลาย และนำออกไปจากลิสต์

สำหรับฟังก์ชันสำคัญหรือเมธอดภายในคลาสนี้ที่พัฒนาขึ้นเพื่อรองรับการทำงานสำหรับประเภทรายการที่เกิดขึ้นได้แก่

- WWW_Alert เพื่อทำการแสดงกล่องข้อความผิดพลาดของเน็ตสเค๊ป ทำการปรับปรุงข้อความของหน้าต่างโปรแกรมประยุกต์ที่สร้างขึ้น และทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติกับโปรแกรมบริการดีดีอีของเน็ตสเค๊ปอีกครั้ง เนื่องจากสาเหตุการเกิดข้อความผิดพลาดส่วนใหญ่มาจากการที่โปรแกรมเน็ตสเค๊ป ไม่สามารถโหลดเว็บเพจที่ผู้ใช้ต้องการได้ และทำให้การทำงานยังไม่เสร็จสิ้นถึงจุดที่จะทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติอีกครั้ง เนื่องจากการร้องขอให้โปรแกรมเน็ตสเค๊ปทำการโหลดเว็บเพจ จำเป็นต้องทำการยกเลิกการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติจากโปรแกรมบริการดีดีอีที่เราสร้างขึ้นก่อน

ตารางที่ 4.4 แสดงเมธอดของคลาส CDDEConversation

เมธอด	การทำงาน
CDDEConversation(HCONV hConv)	Construction with overload
static CDDEConversation *ResolveConversation (HCONV hConv)	Resolve conversation
void DoDisconnect()	Disconnect conversation
HDDEDATA ServerPoke(HSZ hszTopic, HSZ hszItem, HDDEDATA hDataPoke)	Poke Server
HDDEDATA ServerRequest(HSZ hszTopic, HSZ hszItem)	Request From Our Server to Netscape
HDDEDATA ClientRequest(HSZ hszItem)	Send along the item to the server, as an XTYP_REQUEST
HDDEDATA ClientPoke(HSZ hszItem)	Send along the item to the server, as an XTYP_POKE
void ClientPassed(HSZ hszItem, const char *pFormat, ...)	Extract Client Argument
HDDEDATA ServerReturns(const char *pFormat, ...)	Extract what Server return and return in function parameter
HDDEDATA WWW_Alert(HSZ hszItem);	(Netscape is server) Server Request WWW_Alert
HDDEDATA WWW_EndProgress(HSZ hszItem)	Server Poke WWW_EndProgress
HDDEDATA WWW_SetProgressRange(HSZ hszItem)	Server Poke WWW_SetProgressRange
HDDEDATA WWW_URLEcho(HSZ hszItem)	Server Poke WWW_URLEcho
HDDEDATA WWW_ViewDocFile(HSZ hszItem)	Server Poke WWW_ViewDocFile
HDDEDATA WWW_WindowChange(HSZ hszItem)	Server Poke WWW_WindowChange
HDDEDATA WWW_BeginProgress(HSZ hszItem);	(Netscape is server) Server Request WWW_BeginProcess
HDDEDATA WWW_MakingProgress(HSZ hszItem)	(Netscape is server) Server Request WWW_MakingProgress
HDDEDATA WWW_OpenURL(HSZ hszItem)	(Netscape is server) Server Request WWW_OpenURL
HDDEDATA WWW_QueryViewer(HSZ hszItem)	(Netscape is server) Server Request WWW_QueryViewer
void CommonConstruction()	Common Construction Code

จึงจำเป็นที่จะต้องมีการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติในฟังก์ชันนี้ เพื่อให้โปรแกรมบริการดีดีซีที่พัฒนาขึ้นสามารถทำงานต่อได้ ดังแสดงในรูปที่ 4.5

```

HDDEDATA CDDEConversation::WWW_Alert(HSZ hszItem) {
    CString csMessage;
    DWORD dwType;
    DWORD dwButtons;
    DWORD dwAnswer = 0;

    ClientPassed(hszItem, "QCS,DW,DW", &csMessage, &dwType, &dwButtons);

    char aBuf[512];
    sprintf(aBuf, "%lu WWW_Alert(%s,%lu,%lu);", dwAnswer, (const char *)csMessage,
        dwType, dwButtons);
    AfxGetApp()->GetMainWnd()->UpdateData(FALSE);
    AfxGetApp()->GetMainWnd()->SetDlgItemText(IDC_URL,"alert");

    CString ServerName = "NSTEST1";
    CDDEObject *pDDE1 = CDDEObject::ResolveService(ServerName);

    if (pDDE1== NULL) {
        ::MessageBox(NULL,"Can't find DDE object","MemberFxn",MB_OK);
    }

    pDDE1->WWW_RegisterProtocol("NSTEST1","http");

    return(ServerReturns("DW", &dwAnswer));
}

```

รูปที่ 4.5 แสดงเมธอด WWW_Alert ของคลาส CDDEConversation

- WWW_BeginProgress เป็นฟังก์ชันสำหรับการส่งข้อความ ให้กับโปรแกรมบริการดีดีซีที่ชื่อของโปรแกรมบริการดีดีซีดังกล่าว เป็นพารามิเตอร์สำหรับการเรียกใช้ฟังก์ชัน WWW_OpenURL ให้ทำการรับรายงานความก้าวหน้า และทำการปรับปรุงข้อความของหน้าต่างโปรแกรมประยุกต์ที่สร้างขึ้น ดังแสดงในรูปที่ 4.6

```

HDDEDATA CDDEConversation::WWW_BeginProgress(HSZ hszItem) {
    DWORD dwWindowID;
    CString csInitialMessage="Loading";
    DWORD dwTransactionID = (DWORD)time(NULL);

    ClientPassed(hszItem, "DW,QCS", &dwWindowID, &csInitialMessage);

    AfxGetApp()->GetMainWnd()->UpdateData(FALSE);
    AfxGetApp()->GetMainWnd()->SetDlgItemText(IDC_URL,"Loading");
    return(ServerReturns("DW", &dwTransactionID));
}

```

รูปที่ 4.6 แสดงเมธอด WWW_BeginProgress ของคลาส CDDEConversation

- WWW_EndProgress เป็นข้อความที่ส่งให้กับโปรแกรมบริการดีดีซีที่พัฒนาขึ้น เพื่อบอกว่าการปฏิบัติการในการร้องขอเพื่อโหลดเว็บเพจเสร็จสมบูรณ์แล้ว ฟังก์ชันนี้ทำหน้าที่ปรับปรุงข้อความของหน้าต่างโปรแกรมประยุกต์ที่สร้างขึ้น และทำการลง

ทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติกับโปรแกรมบริการดีดีซีของเน็ตสเคปอีกครั้งหนึ่ง ดังแสดงในรูปที่ 4.7

```

HDDEDATA CDDEConversation::WWW_EndProgress(HSZ hszItem)    {
    DWORD dwTransactionID;

    ClientPassed(hszItem, "DW", &dwTransactionID);

    AfxGetApp()->GetMainWnd()->UpdateData(FALSE);
    AfxGetApp()->GetMainWnd()->SetDlgItemText(IDC_URL, "End Progress");

    CString ServerName = "NSTEST1";
    CDDEObject *pDDE1 = CDDEObject::ResolveService(ServerName);

    pDDE1->WWW_RegisterProtocol("NSTEST1", "http");
    return(DDEFACK);
}

```

รูปที่ 4.7 แสดงเมธอด WWW_EndProgress ของคลาส CDDEConversation

- WWW_MakingProgress เป็นฟังก์ชันที่ทำหน้าที่ส่งรายงานความก้าวหน้าของสถานะปัจจุบันของการโหลดเว็บเพจให้กับโปรแกรมบริการดีดีซีที่พัฒนาขึ้น และทำการปรับปรุง ข้อความของหน้าต่างโปรแกรมประยุกต์ที่สร้างขึ้น ดังแสดงในรูปที่ 4.8

```

HDDEDATA CDDEConversation::WWW_MakingProgress(HSZ hszItem) {
    DWORD dwTransactionID;
    CString csMessage;
    DWORD dwProgress;
    TwoByteBool bStop = FALSE;

    ClientPassed(hszItem, "DW,QCS,DW", &dwTransactionID, &csMessage, &dwProgress);

    char aBuf[512];
    sprintf(aBuf, "%s WWW_MakingProgress(%lu,%s,%lu);", (const char *)ExplainBool(bStop),
        dwTransactionID, (const char *)csMessage, dwProgress);
    AfxGetApp()->GetMainWnd()->UpdateData(FALSE);
    AfxGetApp()->GetMainWnd()->SetDlgItemText(IDC_URL, aBuf);

    return(ServerReturns("BL", &bStop));
}

```

รูปที่ 4.8 แสดงเมธอด WWW_MakingProgress ของคลาส CDDEConversation

- WWW_SetProgressRange เป็นฟังก์ชันที่ทำหน้าที่ส่งสถานะปัจจุบันของการโหลดเว็บเพจของโปรแกรมเน็ตสเคป ให้กับโปรแกรมบริการดีดีซีที่พัฒนาขึ้น แต่ฟังก์ชันนี้จะถูกเรียกใช้งานหลังจาก WWW_BeginProgress ถูกจัดการโดยโปรแกรมบริการดีดีซีที่พัฒนาขึ้นแล้ว และฟังก์ชันนี้จะทำการปรับปรุงข้อความสำหรับหน้าต่างของโปรแกรมประยุกต์ที่สร้างขึ้น ดังแสดงในรูปที่ 4.9

```

HDDEDATA CDDEConversation::WWW_SetProgressRange(HSZ hszItem)
    DWORD dwTransactionID;
    DWORD dwMaximum;

    ClientPassed(hszItem, "DW,DW", &dwTransactionID, &dwMaximum);

    char aBuf[512];
    sprintf(aBuf, "%s WWW_SetProgressRange(%lu,%lu);", (const char *)ExplainPoke(DDEFACK
        dwTransactionID, dwMaximum);

    AfxGetApp()->GetMainWnd()->UpdateData(FALSE);
    AfxGetApp()->GetMainWnd()->SetDlgItemText(IDC_URL,"Set ProgressRange");

    return(DDEFACK);
}

```

รูปที่ 4.9 แสดงเมธอด WWW_SetProgressRange ของคลาส CDDEConversation

- WWW_OpenURL เป็นฟังก์ชันสำหรับปรับปรุงข้อความสำหรับหน้าต่างของโปรแกรมประยุกต์ที่สร้างขึ้น และทำการร้องขอให้โปรแกรมเน็ตสเคป ทำการโหลดเว็บเพจที่ต้องการ โดยมีโปรแกรมบริการดีดีไอของเน็ตสเคปทำหน้าที่เป็นเซิร์ฟเวอร์ และโปรแกรมบริการดีดีไอที่สร้างขึ้นทำหน้าที่เป็นไคลเอนต์ผู้ร้องขอ ดังที่กล่าวไปแล้วในบทที่ 3 ว่าการที่จะทำการดักยูอาร์แอลจากโปรแกรมเน็ตสเคป จำเป็นต้องทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติกับโปรแกรมบริการดีดีไอของเน็ตสเคป ซึ่งเมื่อเน็ตสเคปได้รับยูอาร์แอลของเกณฑ์วิธีการส่งข้อความหลายมิติก็จะส่งให้กับโปรแกรมบริการดีดีไอที่เราพัฒนาขึ้นเป็นผู้จัดการ แต่เมื่อใดก็ตามที่เราต้องการให้โปรแกรมเน็ตสเคป ทำการจัดการกับเกณฑ์วิธีการส่งข้อความหลายมิติด้วยตัวเอง ก็เพียงแต่ทำการยกเลิกการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติ ออกจากโปรแกรมบริการดีดีไอที่ทำหน้าที่จัดการกับเกณฑ์วิธีการส่งข้อความหลายมิติอยู่ ซึ่งในกรณีนี้การที่จะให้โปรแกรมเน็ตสเคป ทำการโหลดเว็บเพจให้กับผู้ร้องขอ จึงจำเป็นต้องทำการยกเลิกการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติเป็นการชั่วคราวก่อน จากนั้นเมื่อโปรแกรมเน็ตสเคปทำการโหลดเว็บเพจเรียบร้อยแล้ว จึงทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติกับ โปรแกรมบริการดีดีไอที่สร้างขึ้นอีกครั้งหนึ่ง ซึ่งในการพัฒนานี้กำหนดให้มีการยกเลิกการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติภายในฟังก์ชัน WWW_OpenURL และทำการลงทะเบียนเกณฑ์วิธีการส่งข้อความหลายมิติ กับโปรแกรมบริการดีดีไออีกครั้งหนึ่งภายในฟังก์ชัน WWW_Alert กรณีที่เกิดความผิดพลาดโปรแกรมเน็ตสเคป ไม่สามารถโหลดยูอาร์แอลดังกล่าว หรือภายในฟังก์ชัน WWW_EndProgress เมื่อยูอาร์แอลดังกล่าวถูกจัดการโดยโปรแกรมเน็ตสเคปเรียบร้อยแล้ว ดังแสดงในรูปที่ 4.10


```

HDDEDATA CDDEConversation::WWW_OpenURL(HSZ hszItem)      {
    CString csURL;
    CString tmp;
    CString csSaveAs;
    DWORD dwWindowID;
    DWORD dwFlags;
    CString csPostFormData;
    CString csPostMIMEMType;
    CString csProgressServer;
    DWORD dwServicingID = (DWORD)time(NULL);

    ClientPassed(hszItem, "QCS,QCS,DW,DW,QCS,QCS,CS", &csURL, &csSaveAs,
        &dwWindowID, &dwFlags, &csPostFormData, &csPostMIMEMType,
        &csProgressServer);

    //update dlg current URL
    AfxGetApp()->GetMainWnd()->UpdateData(FALSE);
    AfxGetApp()->GetMainWnd()->SetDlgItemText(IDC_CURRENT,csURL);
    CString ip;
    currentURL = csURL;

    AfxGetApp()->GetMainWnd()->UpdateData(TRUE);
    AfxGetApp()->GetMainWnd()->GetDlgItemText(IDC_LOCALIP,ip);

    UrlList *Node;
    if (!Node->FindUrlInList(csURL) ) { //&& csURL.Find("chula.ac.th",0)== -1)
        // Broadcast
        BroadcastSocket (ip,csURL);
        // request for process
        ReqList *node = new ReqList();
        node->csURL = csURL;
        node->csSaveAs = csSaveAs ;
        node->dwWindowID = dwWindowID;
        node->dwFlags = dwFlags;
        node->csPostFormData = csPostFormData;
        node->csPostMIMEMType = csPostMIMEMType;
        node->csProgressServer = "NSTEST1";
        node->age = GetTickCount();
        return (ServerReturns("DW", &dwServicingID));
    }
    else {

        CString ServerName = "NSTEST1";
        CDDEObject *pDDE1 = CDDEObject::ResolveService(ServerName);

        if (pDDE1== NULL) {
            ::MessageBox(NULL,"Can't find DDE object","MemberFxn",MB_OK);
        }
        pDDE1->WWW_UnRegisterProtocol("NSTEST1","http");
        pDDE1->WWW_OpenURL(csURL, csSaveAs, dwWindowID,
            dwFlags,csPostFormData,csPostMIMEMType,"NSTEST1");
        // pDDE1->WWW_RegisterProtocol("NSTEST1","http");
    }

    return (ServerReturns("DW", &dwServicingID));
}

```

รูปที่ 4.10 แสดงเมธอด WWW_OpenURL ของคลาส CDDEConversation

4.1.3 ฟังก์ชันสำหรับการเรียกย้อนกลับ NstestDdeCallBack

ฟังก์ชัน NstestDdeCallBack เป็นฟังก์ชันที่ถูกลงทะเบียนเข้าไปกับโปรแกรมบริการดีดีซีที่สร้างขึ้นเพื่อรองรับเหตุการณ์ที่เกิดขึ้น ฟังก์ชันสำหรับการเรียกย้อนกลับนี้ จะถูกลงทะเบียนโดยการเรียกฟังก์ชัน DdeInitialize() ซึ่งเป็นฟังก์ชันมาตรฐานของโปรแกรมพัฒนาซอฟต์แวร์วิซวลซีพลัสพลัส (visual c++) โดยทำการกำหนดค่าเริ่มต้นและชื่อของฟังก์ชันสำหรับการเรียกย้อนกลับ ดังรูปที่ 4.11 โดยการลงทะเบียนฟังก์ชันสำหรับการเรียกย้อนกลับนี้ จะกระทำในตอนเริ่มต้นโปรแกรมซึ่งจากส่วนของโปรแกรมได้ทำการลงทะเบียนฟังก์ชัน NstestCallBack สำหรับการเรียกย้อนกลับในฟังก์ชัน InitInstance() ซึ่งฟังก์ชัน InitInstance() นี้จะถูกเรียกใช้งานเป็นฟังก์ชันแรกเมื่อเริ่มต้นโปรแกรม

```

BOOL CDde3App::InitInstance()
{
...
//Initialize dde
// initialize ddeml
if (DdeInitialize(&CDDEObject::m_dwIdInst, NstestDdeCallBack, APPCLASS_STANDARD, 0ul))
return false;
...
}

```

รูปที่ 4.11 แสดงการลงทะเบียนฟังก์ชันเรียกย้อนกลับโดยใช้ฟังก์ชัน DdeInitialize()

หลังจากได้ทำการลงทะเบียนฟังก์ชัน NstestDdeCallBack สำหรับการเรียกย้อนกลับเรียบร้อยแล้ว เมื่อใดก็ตามที่เกิดการติดต่อกันระหว่าง โปรแกรมบริการดีดีซีของเน็ตสเค๊ปกับโปรแกรมบริการดีดีซีที่พัฒนาขึ้น ทำให้สามารถทราบได้ว่าเกิดเหตุการณ์ประเภทใด และมีข้อมูลใดบ้างที่จะสามารถนำไปประมวลผลแยกตามประเภทรายการตามชนิดของรายการที่เกิดขึ้น ซึ่งจะไปทำการเรียกฟังก์ชันเพื่อการจัดการสำหรับทุกประเภทรายการที่เกิดขึ้น ดังแสดงในส่วนของโปรแกรมในรูปที่ 4.12 ซึ่งได้ทำการแสดงโปรแกรม NstestDdeCallBack ไว้

```

//      DDEML Callback function
#ifdef _WIN32
HDDEDATA CALLBACK NstestDdeCallBack(UINT type, UINT fmt,
    HCONV hconv, HSZ hsz1, HSZ hsz2, HDDEDATA hData, DWORD dwData1,
    DWORD dwData2)    {
#else
HDDEDATA CALLBACK _export NstestDdeCallBack(UINT type, UINT fmt,
    HCONV hconv, HSZ hsz1, HSZ hsz2, HDDEDATA hData, DWORD dwData1,
    DWORD dwData2)    {
#endif // _WIN32

    switch(type)        { // Return Type Receiver
    case XTYP_CONNECT:  { // XCLASS_BOOL server
        //      Find the object with the registered service name.
        CDDEObject *pObj = CDDEObject::ResolveService(hsz2);
        if (pObj != NULL)    {
            //Leave it up to the document to decide whether or not it wants the
            //conversation.
            return(pObj->ServerConnect(hsz1));
        }
        return(DDEFALSE);
    }
    case XTYP_CONNECT_CONFIRM:  { //XCLASS_NOTIFICATION server
        //      Find the object with the registered service name.
        CDDEObject *pObj = CDDEObject::ResolveService(hsz2);
        if(pObj != NULL)    {
            //The object has already accepted a conversation for the given
            //topic, so no need to check that. Let it know it has a real
            //conversation now, and give it the handle.
            pObj->ServerConnectConfirm(hconv);
        }
        return(NOTHING);
    }
    case XTYP_DISCONNECT:      { //XCLASS_NOTIFICATION client/server
        //      Find the object handling the conversation.
        CDDEConversation *pConv = CDDEConversation::ResolveConversation(hconv);
        if (pConv != NULL)    {
            //We also need to remove the conversation from the list of
            //conversations managed by some DDE service object.
            CDDEObject::DDEDisconnect(pConv);
            //It's over.
            delete pConv;
        }
        return(NOTHING);
    }
    case XTYP_POKE:           { //XCLASS_FLAGS server
        //Find the object handling the conversation.
        CDDEConversation *pConv = CDDEConversation::ResolveConversation(hconv);
        if (pConv != NULL)    {
            // Have the object handle the details.
            return(pConv->ServerPoke(hsz1, hsz2, hData));
        }
        return(DDE_FNOTPROCESSED);
    }
    case XTYP_REQUEST:        { //XCLASS_DATA server
        //      Find the object handling the conversation.
        CDDEConversation *pConv = CDDEConversation::ResolveConversation(hconv);
        if (pConv != NULL){
            return(pConv->ServerRequest(hsz1, hsz2));
        }
        return(NULL);
    }
    }
    //      end switch
//not handle
    return(NOTHING);
}

```

รูปที่ 4.12 โปรแกรม NstestCallBack ซึ่งเป็นฟังก์ชันสำหรับการเรียกย้อนกลับ

4.2 โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ หรือ UCM (Cache Client Manager Program)

โปรแกรมสำหรับการจัดการยูอาร์แอลในส่วนของไคลเอนต์นี้ ทำหน้าที่เป็นตัวจัดการกับยูอาร์แอลซึ่งได้รับจากในข้อ 4.2 เพื่อทำการนำไปค้นหาในฐานข้อมูลแคชของคอมพิวเตอร์อื่นในข่ายงานบริเวณเฉพาะที่ และทำการปรับปรุงฐานข้อมูลแคชดังกล่าวในบทที่ 3 ซึ่งฟังก์ชันในการพัฒนาโปรแกรมนี้สามารถแยกได้เป็นดังนี้

- ส่วนของการแพร่กระจายข้อความไปในข่ายงานบริเวณเฉพาะที่
- ส่วนของการสร้างการเชื่อมต่อและโอนย้ายเพิ่มข้อมูล
- ส่วนของโปรแกรมในการค้นหาข้อมูลในฐานข้อมูลแคช และปรับปรุงฐานข้อมูลแคช

4.2.1 ส่วนของการแพร่กระจายข้อความไปในข่ายงานบริเวณเฉพาะที่

โปรแกรมในส่วนของการแพร่กระจายข้อความไปในข่ายงานบริเวณเฉพาะที่ ทำหน้าที่ในการส่งข้อความออกไปในข่ายงานบริเวณเฉพาะที่ โดยมีฟังก์ชันของการแพร่กระจายข้อความดังมีรายละเอียดต่อไปนี้

การที่จะทำการแพร่กระจายข้อความออกไปในข่ายงานบริเวณเฉพาะที่นั้น จะต้องมีเลขที่อยู่ของการแพร่กระจาย (Broadcast address) ซึ่งเลขที่อยู่ของการแพร่กระจายก็คือ เลขที่อยู่ที่ใช้ส่งเดตาแกรมออกไปยังแม่ข่าย (Host) มากกว่า 1 ตัว ด้วยคำสั่ง `sendto()` [7] เพียงคำสั่งเดียว สำหรับเลขที่อยู่ของการแพร่กระจายมาตรฐาน มี 3 ระดับได้แก่

1. Local Broadcast มีเลขที่อยู่เท่ากับ “255.255.255.255” ออกแบบมาสำหรับการแพร่กระจายภายในข่ายงานบริเวณเฉพาะที่
2. Net-Directed Broadcast เป็นเลขที่อยู่ที่ใช้สำหรับการแพร่กระจายไปยังทุกเครื่องข่าย ที่อยู่ติดกันในระบบที่เราทำการแพร่กระจาย ซึ่งจะขึ้นอยู่กับคลาสเลขที่อยู่ (Address Class) ที่ใช้ ตัวอย่างเช่น “Net.255.255.255” “Net.B.255.255” และ “Net.B.C.255”
3. IP Multicast เป็นเลขที่อยู่พิเศษในคลาสเลขที่อยู่ D เลขที่อยู่นี้จะถูกใช้ได้เฉพาะกรณี Router สนับสนุน Internet Group Management Protocol (IGMP)

ในกรณีของโปรแกรมที่เราพัฒนาขึ้นจะต้องใช้ เลขที่อยู่ของการแพร่กระจายในข้อ 1 คือ “255.255.255.255” แต่การจะทำการแพร่กระจายข้อความออกไปได้นั้น ต้องมีการกำหนดเป็นพิเศษโดยการใช้ฟังก์ชัน `setsockopt()` [7] โดยกำหนดค่าตัวแปรต่าง ๆ ดังแสดงในรูปที่ 4.13

```
//enable broadcast using setsockopt
int nLevel = SOL_SOCKET;
int nOption = SO_BROADCAST; //data type for SO_BROADCAST
BOOL nOptionValue = TRUE; // set it to true
int nOptionValLen = sizeof(BOOL);
int nRet = setsockopt(MySocket,nLevel,nOption,(char *)&nOptionValue,nOptionValLen);
```

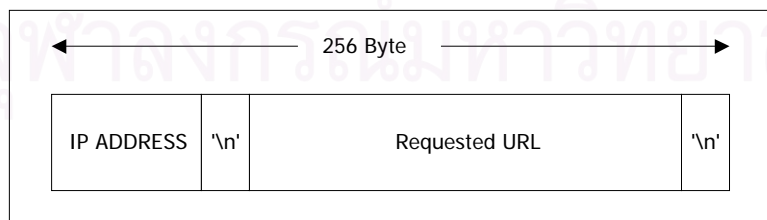
รูปที่ 4.13 แสดงส่วนของโปรแกรมการกำหนดค่าเพื่อการแพร่กระจายข้อความ โดยการใช้ ฟังก์ชัน setsockopt()

เมื่อได้มีการกำหนดค่าให้สามารถทำการแพร่กระจายข้อความโดยส่วนของโปรแกรมที่แสดงในรูปที่ 4.13 เรียบร้อยแล้วเราก็จะสามารถทำการแพร่กระจายข้อความ โดยการระบุเลขที่อยู่ของการแพร่กระจายดังแสดงในรูปที่ 4.14 และทำการแพร่กระจายข้อความ โดยการใช้ฟังก์ชัน sendto()

```
// Fill in the address structure for the server
//
SOCKADDR_IN saServer;
saServer.sin_family = AF_INET;
saServer.sin_addr.s_addr = htonl(INADDR_BROADCAST);
// local broad cast address = //255.255.255.255
saServer.sin_port = htons(nPort); //
//call sendto To Broadcast
sendto(MySocket,szBuf,strlen(szBuf),0,(LPSOCKADDR)&saServer,
sizeof(struct sockaddr));
```

รูปที่ 4.14 แสดงส่วนของโปรแกรมการกำหนดค่าเลขที่อยู่ของการแพร่กระจาย

สำหรับการแพร่กระจายข้อความออกไปในข่ายงานบริเวณเฉพาะที่นั้น ก่อนการแพร่กระจายข้อความจะต้องมีการทำการเตรียมข้อความสำหรับการแพร่กระจายก่อน ซึ่งข้อความที่โปรแกรมจะทำการแพร่กระจายนั้น กำหนดให้มีความยาวเท่ากับ 256 ไบต์ และบรรจุข้อมูลของผู้ส่งที่ต้องการคือ เลขที่อยู่ไอพีของผู้ที่ทำการแพร่กระจายข้อความตามด้วย '\n' จากนั้นต่อด้วยยูอาร์แอลที่โปรแกรมต้องการค้นหา และปิดด้วย '\n' ดังแสดงในรูปที่ 4.15



รูปที่ 4.15 แสดงรูปแบบของข้อความที่จะทำการแพร่กระจายไปในข่ายงานบริเวณเฉพาะที่

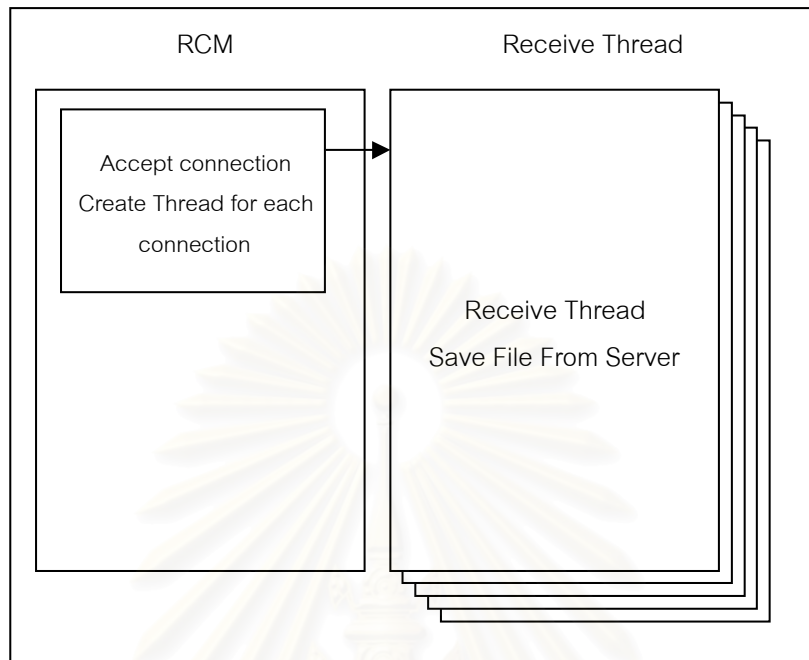
เนื่องจากการแพร่กระจายข้อความนั้น ทำโดยผ่านโพรโทคอล UDP เนื่องจากโพรโทคอล TCP ไม่สนับสนุนการแพร่กระจายข้อความ ดังนั้นโปรแกรมที่สร้างขึ้นจึงต้องสนับสนุนการทำงานในลักษณะเดทาแกรมและใช้โพรโทคอล UDP เป็นโพรโทคอลหลัก การแพร่กระจายข้อความทำโดยการส่งผ่านหมายเลขช่องทางที่กำหนด ซึ่งในวิทยานิพนธ์นี้กำหนดให้มีการรับการแพร่กระจายข้อความและแพร่กระจายข้อความ ผ่านหมายเลขช่องทางที่ 200 ซึ่งสำหรับการรับการแพร่กระจายข้อความจะอยู่ในส่วนของโปรแกรมสำหรับการจัดการแคชระยะไกล ซึ่งจะได้กล่าวถึงต่อไป

4.2.2 ส่วนของการสร้างการเชื่อมต่อและโอนย้ายเพิ่มข้อมูล

สำหรับโปรแกรมในส่วนนี้จะเกิดการดำเนินงาน หลังจากทำการแพร่กระจายข้อความไปในข่ายงานบริเวณเฉพาะที่แล้ว พบว่ามียูอาร์แอลดังกล่าวอยู่ในคอมพิวเตอร์ในข่ายงานบริเวณเฉพาะที่ และโปรแกรมสำหรับการจัดการแคชระยะไกลต้องการตอบกลับ โดยโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์จะเลือกติดต่อกับโปรแกรมสำหรับการจัดการแคชระยะไกลที่สร้างการเชื่อมต่อกลับมายังโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์เป็นครั้งแรก และทำการโอนย้ายเพิ่มข้อมูลเข้าสู่แคชไดเรกทอรีของเน็ตสเคป

สำหรับการสร้างการเชื่อมต่อระหว่างโปรแกรมสำหรับการจัดการแคชระยะไกล และโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์นี้ จะทำงานในลักษณะกระแสข้อมูล (Data Stream) คือต้องมีการเชื่อมต่อของทั้งสองฝ่ายเกิดขึ้นก่อน โดยใช้โพรโทคอล TCP เป็นโพรโทคอลหลักโดยการเชื่อมต่อจะทำการสื่อสารผ่านหมายเลขช่องทางที่ 5060 ซึ่งเป็นหมายเลขช่องทางที่ว่างและไม่ได้ถูกสำรองไว้ใช้กับโพรโทคอลใด ๆ

สำหรับขั้นตอนการทำงานของโปรแกรมนี เริ่มจากโปรแกรมหลักทำการสร้างสายโยงใยการรับข้อมูล (Receive Thread) ขึ้นมา หลังจากโปรแกรมหลักทำงานแล้วสายโยงใยการรับข้อมูล จะทำการบล็อก (Block) อยู่ที่คำสั่ง `accept()` [7] เพื่อทำการรอการเชื่อมต่อเข้ามา เมื่อมีการเชื่อมต่อเข้ามาสายโยงใยการรับข้อมูลจะทำการสร้างสายโยงใยการรับข้อมูลใหม่ขึ้นมารอการเชื่อมต่อเข้ามาอีก จากนั้นสายโยงใยการรับข้อมูลจะออกคำสั่ง `recv()` เพื่อรับข้อมูล ดังแสดงการทำงานของโปรแกรมในส่วนของการเชื่อมต่อได้ดังรูปที่ 4.16



รูปที่ 4.16 แสดงการทำงานของโปรแกรมสำหรับสร้างการเชื่อมต่อระหว่างโปรแกรมสำหรับการจัดการแคชในส่วน
ของไคลเอนต์ และโปรแกรมสำหรับการจัดการแคชระยะไกล

สำหรับการรองรับการเชื่อมต่อจากโปรแกรมสำหรับการจัดการแคชระยะไกล โปรแกรมสำหรับการจัดการแคชในส่วนไคลเอนต์จะทำการสร้างสายโยงใยการรับข้อมูลใหม่ในการรับการเชื่อมต่อแต่ไม่เกิน 5 สายโยงใยการรับข้อมูล ตามที่ระบุในคำสั่ง Listen() [7] ก่อนหน้านั้น ดังแสดงในรูปที่ 4.17 แสดงฟังก์ชันการทำงานเพื่อทำการรองรับการเชื่อมต่อของโปรแกรมสำหรับการจัดการแคชระยะไกล

ปัญหาที่พบได้ในกรณีที่ใช้ฟังก์ชัน recv() แบบธรรมดา นี้คือโปรแกรมได้รับค่าผิดพลาดกลับมาเป็น WSAEWOULDBLOCK [7] แสดงว่าเกิดการบล็อกข้อก่เกิดขึ้น เนื่องจากการรับยังไม่เสร็จสมบูรณ์ ทำให้สายโยงใยที่เราสร้างขึ้นไม่สามารถจบลง ดังนั้นเมื่อมีการสร้างสายโยงใยขึ้นจนครบ 5 สายโยงใยแล้ว จะไม่สามารถสร้างขึ้นใหม่ได้อีกทำให้ฝั่งส่งไม่สามารถทำการสร้างการเชื่อมต่อเข้ามาได้อีก ซึ่งปัญหาที่เกิดขึ้นนี้เกิดจากการที่โพรโทคอลสแต็ก (Protocol Stack) ของวินโดวส์ไม่สามารถจะทำงานให้ได้นั้น

วิธีการแก้ไขปัญหาดังกล่าว สามารถทำได้โดยการใช้เทคนิค I/O Overlapped สำหรับการเขียนโปรแกรมแบบซ็อกเก็ต โดยทำการลงทะเบียนฟังก์ชันสำหรับการเรียกย้อนกลับ เพื่อการรับข้อมูล และทำเปลี่ยนมาใช้ฟังก์ชันคำสั่ง WSAREcv() [7] ในการรับข้อมูลแทนคำสั่ง recv() ซึ่ง

หน้าที่ของโปรแกรมสำหรับการเรียกย้อนกลับนี้จะทำหน้าที่ในการรับข้อมูล และตรวจสอบสถานะของซ็อกเก็ต ซึ่งถ้าสถานะของซ็อกเก็ตเท่ากับ WSA_IO_PENDING [7] แสดงว่ายังมีข้อมูลที่ต้องทำการรับอยู่ ดังแสดงในโปรแกรมรูปที่ 4.18 แสดงการทำงานของฟังก์ชันที่ใช้เทคนิค I/O Overlapped และโปรแกรมในรูปที่ 4.19 แสดงฟังก์ชัน RecvComplete() ซึ่งเป็นฟังก์ชันสำหรับการเรียกย้อนกลับตามลำดับ

```

UINT StreamServer(void *pParam)
{
    WORD wVersionRequested = MAKEWORD(2,0);
    WSADATA wsaData;
    sockaddr_in local;
    SOCKET listen_socket;
    int nRet;

    nRet = WSASStartup(wVersionRequested, &wsaData);

    if (nRet){
        WSACleanup();
        return 0;
    }

    if (wsaData.wVersion != wVersionRequested)
    {
        return 0;// FALSE;
    }

    local.sin_family=AF_INET;
    local.sin_addr.s_addr=INADDR_ANY;
    local.sin_port=htons(SERVER_PORT);

    listen_socket = socket(AF_INET,SOCK_STREAM,IPPROTO_TCP);

    if (listen_socket == INVALID_SOCKET)
    {
        WSACleanup();
        return 0;
    }

    if (::bind(listen_socket,(sockaddr *)&local,sizeof(local))!=0)
    {
        ::MessageBox(NULL,"Error in Binding socket","Start Server",MB_OK);
        WSACleanup();
        return 0;
    }

    if (::listen(listen_socket,5)!=0)
    {
        WSACleanup();
        return 0;
    }
    gListen_Socket=listen_socket;
    AfxBeginThread(ReceiveThread,NULL); //Start accept function
    return 1;
}

```

รูปที่ 4.17 แสดงโปรแกรมสำหรับการรอรับการเชื่อมต่อของโปรแกรมสำหรับการจัดการแคชระยะไกล


```

// Setup Chanel and receive data from remote
UINT ReceiveThread(void *pParam)
{
    sockaddr_in from;
    SOCKET remote_socket; // remote socket
    int fromlen;//,retval;
    CString tmp;
    BOOL isopen = FALSE;
    BOOL isdatablock = FALSE;

    fromlen = sizeof(from);
    remote_socket = accept(gListen_Socket,(struct sockaddr*)&from,&fromlen);
    AfxBeginThread(ReceiveThread,pParam); //Start another thread to listen.
    if( remote_socket==INVALID_SOCKET)
    {
        return 0 ; //-5;
    }

//// iooverlapped
    BYTE aBuffer[BUFSIZE];
    IOREQUEST ioRequest;
    memset(&ioRequest.over,0,sizeof(WSAOVERLAPPED));

    ioRequest.socket = remote_socket;
    ioRequest.fFinished = FALSE;
    ioRequest.pBuffer = aBuffer;

    WSABUF wsabuf;
    wsabuf.len = BUFSIZE;
    wsabuf.buf = (char*)aBuffer;
    int nRet;
    DWORD dwRecv ;
    DWORD dwFlags = 0;
    nRet = WSAREcv(remote_socket,&wsabuf,
        1,
        &dwRecv,
        &dwFlags,
        (LPWSAOVERLAPPED)&ioRequest,
        RecvComplete);
    if (nRet ==SOCKET_ERROR) {
        if (WSAGetLastError() != WSA_IO_PENDING) {
            closesocket(remote_socket);
            return 0;
        }
    }
    // receive the file content
    while(1)
    {
        SleepEx(INFINITE,TRUE);
        if (ioRequest.fFinished)
            break;
    }

    closesocket(remote_socket); // don't forget close here
    AfxEndThread(0);
    return 1;
}

```

รูปที่ 4.18 แสดงโปรแกรมการทำงานของฟังก์ชันรับแฟ้มข้อมูลที่ส่งจากโปรแกรมสำหรับการจัดการแคชระยะไกลโดยใช้เทคนิค I/O Overlapped

```

void CALLBACK RecvComplete(DWORD dwError,
                          DWORD cbRecv,
                          LPWSAOVERLAPPED lpOver,
                          DWORD dwFlags)
{
    static FileHeader fheader;
    FILE *stream;
    int nRet;
    CString fullname;

    if (dwError){
        return;
    }
    LPIOREQUEST pReq = (LPIOREQUEST)lpOver;

    if (cbRecv == 0){
        pReq->fFinished = TRUE;
        return;
    }

    .....

    // IOREQUEST ioRequest;
    // memset(&ioRequest.over,0,sizeof(WSAOVERLAPPED));

    WSABUF wsabuf;
    wsabuf.len = BUFSIZE;
    wsabuf.buf = (char *)pReq->pBuffer;

    DWORD dwRecv;
    dwFlags = 0;
    // int nRet;
    nRet = WSAREcv(pReq->socket,
                  &wsabuf,
                  1,
                  &dwRecv,
                  &dwFlags,
                  lpOver,
                  RecvComplete);

    if (nRet == SOCKET_ERROR)
    {
        if (WSAGetLastError() != WSA_IO_PENDING)
        {
            pReq->fFinished = TRUE;
        }
    }
}

```

รูปที่ 4.19 แสดงโปรแกรมการทำงานของฟังก์ชัน RecvComplete ซึ่งเป็นฟังก์ชันสำหรับการเรียกย้อนกลับ

4.2.3 ส่วนของโปรแกรมในการค้นหาข้อมูลในฐานข้อมูลแคช และปรับปรุงฐานข้อมูลแคช

โปรแกรมในการค้นหาข้อมูลในฐานข้อมูลแคช และปรับปรุงฐานข้อมูลแคชนี้ เป็นการนำบางส่วนที่ได้จากการค้นคว้าและทดลองใช้ จากการทดลองดาวน์โหลดโปรแกรม Mozilla Browser (1998/04/29) และทำการพิจารณาตัดบางส่วนของโปรแกรมมาทำการรวมกันอยู่ในส่วนของ

โปรแกรมนี้ โดยโปรแกรมส่วนใหญ่ที่ทำหน้าที่จัดการแคชนั้นจะอยู่ใน NetLib⁴ เป็นส่วนใหญ่ สำหรับฟังก์ชันที่เราจะใช้จาก NetLib จะเป็นฟังก์ชันในการจัดการฐานข้อมูลแคชที่สำคัญได้แก่ ฟังก์ชันต่าง ๆ ดังนี้

- ฟังก์ชันที่ทำงานเกี่ยวกับการเปิดฐานข้อมูลแคช

ฟังก์ชัน DB * net_OpenExtCacheFatDB() ทำหน้าที่เปิดฐานข้อมูลแคช ดังแสดงในโปรแกรมรูปที่ 4.20

```
static DB * net_OpenExtCacheFatDB (char *filename)
{
    DB *rv;
    HASHINFO hash_info = {
        16*1024, /* bucket size */
        0,      /* fill factor */
        0,      /* number of elements */
        0,      /* bytes to cache */
        0,      /* hash function */
        0};     /* byte order */

    rv = dbopen(filename,O_RDWR | O_CREAT,0644,DB_HASH,&hash_info);
    if(!rv)
    {
        printf("Could not open cache database: %s\n", filename);
        exit(1);
    }
    return(rv);
}
```

รูปที่ 4.20 แสดงโปรแกรมของฟังก์ชัน DB * net_OpenExtCacheFatDB()

- ฟังก์ชันในการเขียนฐานข้อมูลแคช

ฟังก์ชัน NET_WriteCacheFAT() ทำหน้าที่ในการเขียนแคชอ็อบเจกต์ลงสู่ฐานข้อมูลแคช ดังแสดงในโปรแกรมรูปที่ 4.21

```
PUBLIC void
NET_WriteCacheFAT(char *filename, Bool final_call)
{
    net_StoreDiskCacheSize();
    if(!cache_database)
        return;

    if(-1 == (*cache_database->sync)(cache_database, 0))
    {
        TRACEMSG(("Error syncing cache database"));
    }
    /* close the database */
    (*cache_database->close)(cache_database);
    cache_database = 0;
}

return;
```

รูปที่ 4.21 แสดงโปรแกรมของฟังก์ชัน NET_WriteCacheFAT()

⁴ NetLib เป็นไลบรารีหนึ่งของการพัฒนาโปรแกรม Mozilla Browser ฟังก์ชันส่วนใหญ่ของไลบรารีทำหน้าที่สำหรับการจัดการเครือข่าย และจัดการแคช ข้อมูลเพิ่มเติมที่ <http://www.mozilla.org/netlib>

- ฟังก์ชันที่ทำการค้นหาข้อมูลในฐานข้อมูลแคช
ฟังก์ชัน NET_IsUrlInDiskCache() ทำหน้าที่ในการหายูอาร์แอลแบบเต็มที่ต้องการ
จากฐานข้อมูลแคช ดังแสดงในโปรแกรมรูปที่ 4.22

```

/* returns TRUE if the url is in the disk cache
*/
PUBLIC Bool
NET_IsURLInDiskCache(URL_Struct *URL_s)
{
    DBT *key;
    DBT data;
    int status;

    if(!cache_database)
    {
        TRACEMSG(("Cache database not open"));
        return(0);
    }
    key = net_GenCacheDBKey(URL_s->address,
                           URL_s->post_data,
                           URL_s->post_data_size);

    status = (*cache_database->get)(cache_database, key, &data, 0);
    net_FreeCacheDBTdata(key);
    if(status == 0)
        return(TRUE);
    else
        return(FALSE);
}

```

รูปที่ 4.22 แสดงฟังก์ชัน NET_IsUrlInDiskCache()

สำหรับฟังก์ชันอื่น ๆ นอกจากนี้ที่เกี่ยวข้องและถูกนำมาจาก NetLib เพื่อการพัฒนาโปรแกรมเพื่อให้โปรแกรมค้นผ่านเว็บไซต์เว็บแคชร่วมกันจะไม่กล่าวถึงแต่จะสามารถพบได้จากแฟ้ม
extcache.c h_bigkey.c h_func.c h_log2.c h_page.c hash.c memmove.c
mkcache.c mkextcah.c mktemp.c xp_file.c xp_list.c xp_str.c

4.3 โปรแกรมสำหรับการจัดการแคชระยะไกล หรือ RCM (Remote Cache Manager Program)

โปรแกรมสำหรับการจัดการแคชระยะไกลนี้ ทำหน้าที่รองรับการแพร่กระจายข้อความของไคลเอนต์ ทำการจัดการกับข้อความที่ร้องขอเข้ามาของไคลเอนต์อื่น ทำการบริหารฐานข้อมูลแคชในส่วนของการทำงานโดยใช้ฟังก์ชันที่ได้กล่าวข้างต้น ทำการสร้างการเชื่อมต่อและโอนย้ายแฟ้มข้อมูลไปยังไคลเอนต์ที่ร้องขอเข้ามา โดยโปรแกรมสำหรับการจัดการแคชระยะไกลนี้ จะสามารถแบ่งออกเป็นส่วยย่อยตามลักษณะการทำงานดังนี้

- โปรแกรมสำหรับการรองรับการแพร่กระจายข้อความ (Datagram Server)
- โปรแกรมสำหรับการเชื่อมต่อและส่งแฟ้มข้อมูลไปยังไคลเอนต์ที่ร้องขอ (Transfer program)

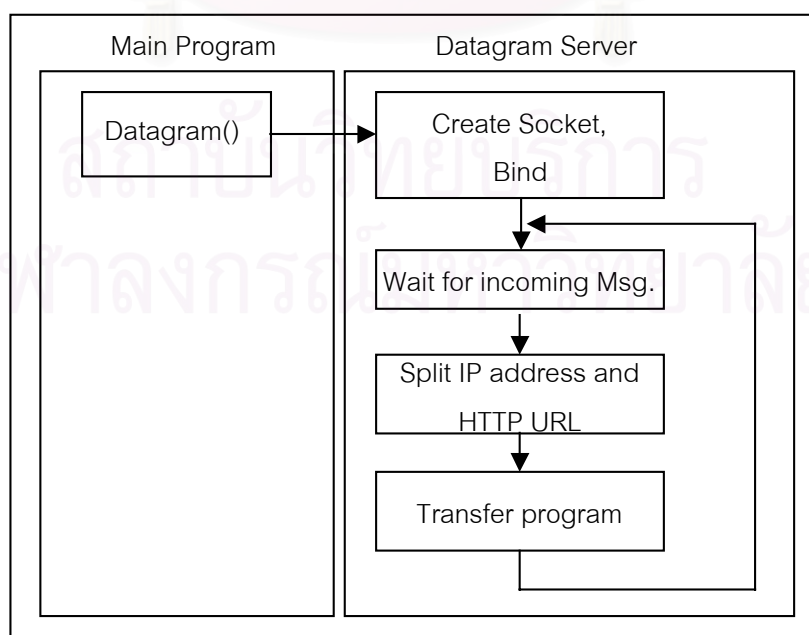
4.3.1 โปรแกรมสำหรับการรอรับการแพร่กระจายข้อความ

โปรแกรมสำหรับการรอรับการแพร่กระจายข้อความจากไคลเอนต์อื่นนี้ ทำหน้าที่ในการรับข้อความแบบเดตาแกรม ที่ถูกส่งจากโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ การทำงานของโปรแกรมสำหรับการรอรับการแพร่กระจายข้อความนี้จะทำงานโดยอาศัยโพรโทคอล UDP เป็นหลัก และทำการรับข้อมูลการแพร่กระจายบนหมายเลขช่องทางที่ 200

การทำงานของโปรแกรมสำหรับการรอรับการแพร่กระจายนี้ เริ่มต้นจากการที่โปรแกรมหลักทำการสร้างสายโยงใยสำหรับรอรับการแพร่กระจายข้อความ ซึ่งจะมีอยู่เพียงสายโยงใยเดียว โปรแกรมสายโยงใยที่ถูกสร้างขึ้นจะทำการสร้างซ็อกเก็ต และทำการเรียกคำสั่ง bind() [7] เพื่อทำการเชื่อมต่อ (bind) ซ็อกเก็ตกับเลขที่อยู่ไอพีและหมายเลขช่องทางของโปรแกรมสำหรับการจัดการแคชระยะไกล เพื่อให้โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์สามารถพบโปรแกรมสำหรับการจัดการแคชระยะไกลดังกล่าว และสามารถทำการแพร่กระจายข้อความได้

เมื่อโปรแกรมสำหรับรอรับการแพร่กระจายข้อความ ได้รับข้อความจากการแพร่กระจายข้อความแล้ว จะทำการส่งข้อความไปเพื่อทำการประมวลผลต่อ โดยการแตกข้อความดังกล่าวออกเป็นเลขที่อยู่ไอพีของผู้ร้องขอ และยูอาร์แอลที่ต้องการ จากนั้นทำการเรียกการทำงานในโปรแกรมสำหรับการเชื่อมต่อและส่งเพิ่มข้อมูลไปยังไคลเอนต์ที่ร้องขอ ซึ่งจะได้กล่าวถัดไป

สุดท้ายเมื่อโปรแกรมสำหรับการรอรับการแพร่กระจายข้อความ ทำการประมวลผลข้อความที่ได้รับเรียบร้อยแล้วจะทำการวนซ้ำกลับไปเพื่อทำการรอรับข้อความแพร่กระจายอีกครั้ง ซึ่งสามารถแสดงลำดับขั้นตอนการทำงานดังรูปที่ 4.23 และในโปรแกรมรูปที่ 4.24



รูปที่ 4.23 แสดงการทำงานของฟังก์ชันสำหรับรอรับการข้อความแพร่กระจาย

```

UINT DatagramServer(void *pParam)
{
    // Create a UDP/IP datagram socket
    short nPort = 200;
    ...
    Initialize Socket
    ...
    // SOCKET theSocket;
    ListenSocket = socket(AF_INET,           // Address family
                        SOCK_DGRAM, // Socket type
                        IPPROTO_UDP); // Protocol
    if (ListenSocket == INVALID_SOCKET)
    {
        return 0;
    }
    // Fill in the address structure for listen server
    SOCKADDR_IN saServer;
    saServer.sin_family = AF_INET;
    saServer.sin_addr.s_addr = INADDR_ANY; // Let WinSock assign address
    saServer.sin_port = htons(nPort); // Use port passed from user
    // bind the name to the socket
    nRet = bind(ListenSocket, // Socket descriptor
                (LPSOCKADDR)&saServer, // Address to bind to
                sizeof(struct sockaddr) // Size of address
                );
    if (nRet == SOCKET_ERROR)
    {
        closesocket(ListenSocket);
        return 0;
    }
    int nLen;
    nLen = sizeof(SOCKADDR);
    char szBuf[256];

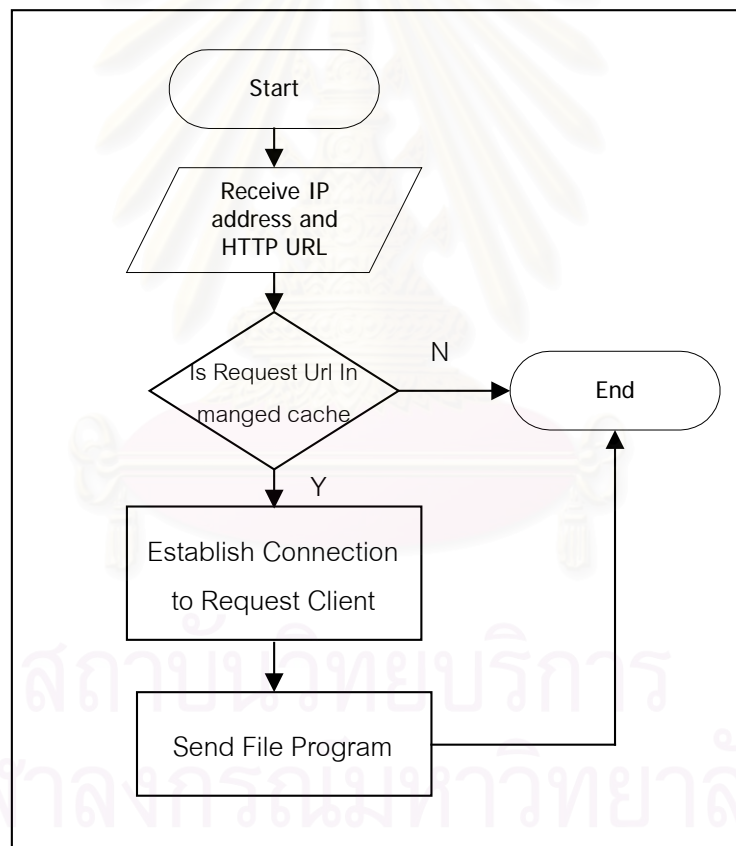
    nRet = gethostname(szBuf, sizeof(szBuf));
    if (nRet == SOCKET_ERROR)
    {
        closesocket(ListenSocket);
        return 0;
    }
    // Update Datagram status
    PostMessage(hDlg,UPDDLG,0,(LPARAM)"Datagram Server Start");
    // Wait for data from the client
    SOCKADDR_IN saClient;
    while (1){
        memset(szBuf, 0, sizeof(szBuf));
        nRet = recvfrom(ListenSocket, // Bound socket
                       szBuf, // Receive buffer
                       sizeof(szBuf), // Size of buffer in bytes
                       0, // Flags
                       (LPSOCKADDR)&saClient, // Buffer to receive client address
                       &nLen); // Length of client address buffer
        if (nRet == 0)
        // MessageBox(NULL,"No data receive","Datagram server",MB_OK);
        // Show that we've received some data
        //
        CString tmp,result,result1;
        tmp = szBuf;
        ProcessDatagramMsg(tmp) ;
    }
    closesocket(ListenSocket);
    return 1;
}

```

รูปที่ 4.24 แสดงฟังก์ชันสำหรับรอรับการแพร่กระจายข้อความ

4.3.2 ส่วนของการสร้างการเชื่อมต่อและโอนย้ายแฟ้มข้อมูล

การทำงานของโปรแกรมส่วนของการสร้างการเชื่อมต่อและโอนย้ายแฟ้มข้อมูลนี้ จะทำงานต่อจากในส่วนของโปรแกรมการรับข้อมูลจากการแพร่กระจาย เมื่อได้รับข้อมูลหมายเลขไอพีและยูอาร์แอลแล้ว โปรแกรมจะทำการตรวจสอบจากฐานข้อมูลแคช ว่าพบยูอาร์แอลที่ทางโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ต้องการอยู่หรือไม่ ถ้าพบก็จะทำการหาว่ามีแฟ้มข้อมูลใดบ้างเป็นของยูอาร์แอลที่ต้องการค้นหา (กรณีที่ไม่มีพบจะทำการจบโปรแกรม) จากนั้นจึงสร้างการเชื่อมต่อไปยังโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ และทำการโอนย้ายแฟ้มข้อมูลจากในแคชไดเรกทอรีของโปรแกรมเน็ตสเคป ของเครื่องที่โปรแกรมสำหรับการจัดการแคชระยะไกลดำเนินงานอยู่ เพื่อไปทำการบันทึกลงในแคชไดเรกทอรีของเครื่องที่โปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ทำงานอยู่ ดังแสดงในรูปที่ 4.25



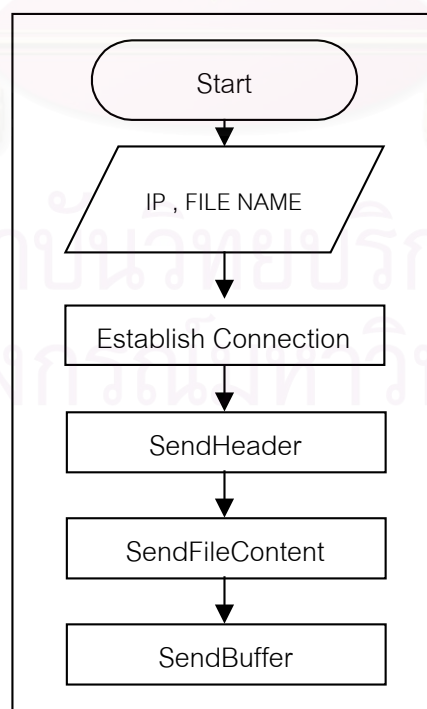
รูปที่ 4.25 แสดงขั้นตอนการทำงานของโปรแกรมในส่วนการสร้างการเชื่อมต่อและโอนย้ายแฟ้มข้อมูล

การทำงานของโปรแกรมในส่วนการสร้างการเชื่อมต่อและโอนย้ายแฟ้มข้อมูลนี้ จะทำการสื่อสารกับโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์ โดยอาศัยโพรโทคอล TCP ผ่านหมายเลขช่องทางที่ 5060 ซึ่งโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์จะมีโปรแกรมสำหรับการรวบรวมแฟ้มข้อมูลดำเนินงานอยู่ ดังที่ได้กล่าวไปแล้วข้างต้นในหัวข้อ 4.2.2

สำหรับการทำงานของโปรแกรมในส่วนการสร้างการเชื่อมต่อและโอนย้ายเพิ่มข้อมูล มีส่วนสำคัญที่จะได้กล่าวถึง คือส่วนของโปรแกรมสำหรับการโอนย้ายเพิ่มข้อมูล (Send File Program) ไปยังโปรแกรมสำหรับการจัดการแคชในส่วนไคลเอนต์ โดยโปรแกรมสำหรับการโอนย้ายเพิ่มข้อมูลนี้จะทำงานหลังจากที่โปรแกรมส่วนของการเชื่อมต่อและโอนย้ายเพิ่มข้อมูล ได้ทำการค้นหายูอาร์แอลในเว็บแคชเฉพาะที่แล้ว พบว่ามีเพิ่มข้อมูลที่สัมพันธ์กับยูอาร์แอลที่ต้องการค้นหาอยู่ ซึ่งจากตรงนี้จะทำให้ได้ชื่อของเพิ่มข้อมูลที่จะต้องส่งให้กับโปรแกรมสำหรับการจัดการแคชในส่วนของไคลเอนต์

เมื่อโปรแกรมสำหรับการโอนย้ายเพิ่มข้อมูลนี้ ได้เลขที่อยู่ไอพีของเครื่องที่โปรแกรมสำหรับการจัดการแคชในส่วนไคลเอนต์ทำงานอยู่แล้วและชื่อเพิ่มข้อมูลที่จะทำการส่งแล้ว โปรแกรมส่วนของการโอนย้ายเพิ่มข้อมูล ก็จะทำการสร้างการเชื่อมต่อไปยังเครื่องที่โปรแกรมสำหรับการจัดการแคชในส่วนไคลเอนต์ทำงานอยู่ จากนั้นทำการส่งไฟล์เฮดเดอร์ (File Header) ให้กับโปรแกรมสำหรับการจัดการแคชในส่วนไคลเอนต์ก่อน เพื่อให้โปรแกรมสำหรับการจัดการแคชในส่วนไคลเอนต์ทราบว่าเพิ่มข้อมูลอะไร มีขนาดเท่าใดและเป็นของยูอาร์แอลใด เพื่อที่จะใช้เป็นข้อมูลในการบันทึกข้อมูลลงไปในฐานะข้อมูลแคชในส่วนถัดไป

จากนั้นจึงเริ่มส่งตัวเพิ่มข้อมูลโดยการเรียกใช้งานฟังก์ชัน `SendFileContent()` เพื่อทำการเปิดเพิ่มข้อมูล อ่านเพิ่มข้อมูลใส่ในที่พักข้อมูลที่เตรียมไว้ จากนั้นจึงเรียกฟังก์ชัน `SendBuffer()` เพื่อทำการส่ง ดังแสดงการทำงานของโปรแกรมส่วนของการโอนย้ายเพิ่มข้อมูล ในรูปที่ 4.26



รูปที่ 4.26 แสดงการทำงานของโปรแกรมโอนย้ายเพิ่มข้อมูล

สำหรับฟังก์ชัน `SendBuffer()` จะใช้เทคนิคที่เรียก IO/Overlapped ซึ่งเป็นเทคนิคในการเขียนโปรแกรมด้วยวินโดวส์ซ็อกเก็ตอย่างหนึ่ง ซึ่งในฟังก์ชัน `SendBuffer()` นี้เอง จะมีการสร้างตัวแปรเหตุการณ์ขึ้นมา เพื่อทำการตรวจสอบว่าสามารถส่งข้อมูลได้ครบถ้วนหรือไม่ เพราะปัญหาที่พบจากการใช้การส่งแบบธรรมดา คือ การเปิดเพิ่มข้อมูล อ่านเพิ่มข้อมูลแล้วทำการส่งอย่างต่อเนื่อง มีปัญหาไม่สามารถส่งเพิ่มข้อมูลที่มีขนาดเกิน 1K ได้ เนื่องจากโพรโทคอลสแต็กของวินโดวส์ทำการส่งให้ได้ไม่ทัน

ดังนั้นการที่จะทำการส่งเพิ่มข้อมูลซึ่งมีขนาดเกิน 1K ได้ จำเป็นต้องใช้เทคนิคดังกล่าวเข้ามาช่วยเนื่องจากเพิ่มข้อมูลที่โปรแกรมต้องการจะส่งอาจมีขนาดมากกว่า 1K สำหรับเพิ่มข้อมูลประเภทรูปภาพที่เน็ตสเคปทำการแคชไว้ ดังแสดงในโปรแกรมรูปที่ 4.27 แสดงการทำงานของฟังก์ชัน `SendBuffer()`



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```

BOOL SendBuffer(SendFileParam param,LPBYTE pbuf,DWORD dwBufSize)
{
    WSABUF wsabuf;
    WSAOVERLAPPED over;
    DWORD dwRecv;
    DWORD dwFlags;
    DWORD dwRet;
    HANDLE hEvents;
    BOOL fPending;
    int nRet;

    //setup the WSABUF and WSAOVERLAPPED structures
    wsabuf.buf = (char *)pbuf;
    wsabuf.len = dwBufSize;
    over.hEvent = WSACreateEvent();
    fPending = FALSE;

    nRet = WSASend(param.socket, // socket
                  &wsabuf , // WSABUF
                  1, // Number of Buffer
                  &dwRecv , // Bytes receive
                  0, //Flags
                  &over , // WSAOVERLAPPED
                  NULL); //completion function
    if (nRet !=0)
    {
        if (WSAGetLastError()== WSA_IO_PENDING)
            fPending = TRUE;
        else {
            CloseHandle(over.hEvent);
            return FALSE ;
        }
    }

    // if the I/O Isn't finished
    if (fPending) {
        // wait for request to complete
        // or the exit event to be signaled
        hEvents = over.hEvent;
        dwRet = WaitForSingleObject(hEvents,INFINITE);

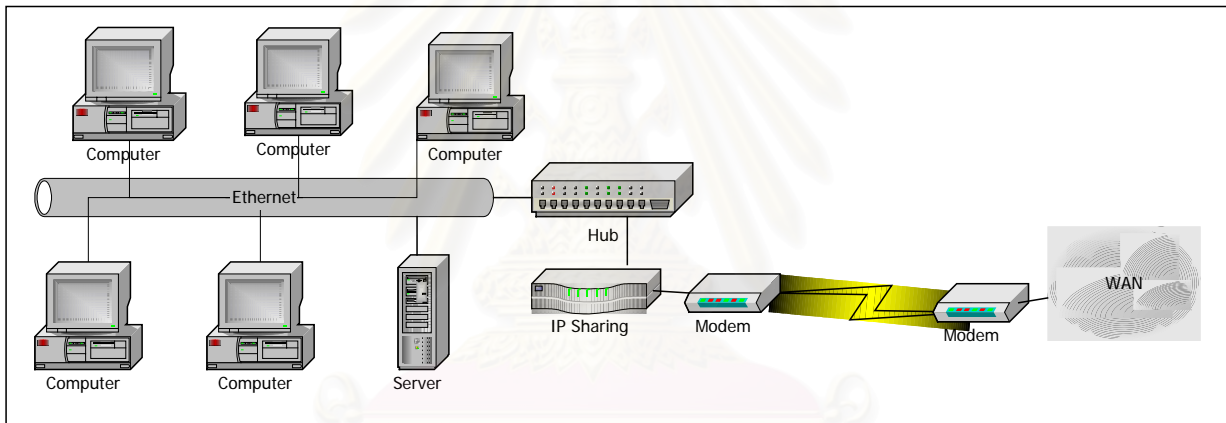
        //was the recv event signaled ?
        if (dwRet != 0)
        {
            CloseHandle(over.hEvent);
            return FALSE;
        }
        // Get i/o result
        if (!WSAGetOverlappedResult(param.socket,
                                    &over,
                                    &dwRecv,
                                    FALSE,
                                    &dwFlags))
        {
            CloseHandle(over.hEvent);
            return FALSE;
        }
    }
    CloseHandle(over.hEvent);
    return TRUE;
} // end function

```

รูปที่ 4.27 แสดงฟังก์ชัน SendBuffer()

บทที่ 5 การทดสอบเพื่อประเมินผลโปรแกรม

การทดสอบโปรแกรมเพื่อการใช้งานเว็บแควซรวมกันของโปรแกรมค้นผ่านเว็บ จะทำกับโปรแกรมเน็ตสเคป รุ่น 4.76 โดยทำการทดสอบกับคอมพิวเตอร์จำนวน 5 เครื่องที่ติดตั้งการ์ดแลน 10/100 ทำการเชื่อมต่อกันเป็นเครือข่าย มีอุปกรณ์เครือข่ายเป็นสวิตช์ 8 ช่องทางความเร็ว 10/100 แบนด์ (Brand) COMPEX รุ่น PS2208 จำนวน 1 เครื่อง เชื่อมต่ออยู่กับอุปกรณ์แชร์ไอพี (IP Sharing) แบนด์ Intel รุ่น Internet Station 56 K แบบ 10 เมกกะบิต ซึ่งเชื่อมต่ออยู่กับโมเด็ม 56 K แบบภายนอก แบนด์ D-Link และเชื่อมต่อสู่เครือข่ายอินเทอร์เน็ตผ่านทางเครือข่ายของบริษัทสามารถ ขณะทดสอบโมเด็มรายงานความเร็วในการขนถ่ายข้อมูลเท่ากับ 49333 bps ดังแสดงภาพโครงสร้างเครือข่ายข้างต้นในรูปที่ 5.1



รูปที่ 5.1 แสดงโครงสร้างของเครือข่ายและการเชื่อมต่อสู่ภายนอก

การทดสอบกระทำโดยการนำเอาแฟ้มปูม (Log File) ของการร้องขอยูอาร์แอลของผู้ใช้งานเว็ลด์ไวด์เว็บของจุฬาลงกรณ์มหาวิทยาลัย วันที่ 31 ตุลาคม 2542 มาทำการเลือกเอาเฉพาะยูอาร์แอลของเกณฑ์วิธีการส่งข้อความหลายมิติ และตัดเอาเฉพาะในส่วนขอชื่อแม่ข่ายที่ไม่ซ้ำกัน ซึ่งจะได้ยูอาร์แอลประมาณ 4,000 ยูอาร์แอลทำการบันทึกลงในแฟ้มข้อมูลชื่อ final.txt

การทดสอบแบ่งเป็น 2 กรณี

- กรณีที่ 1 ทดสอบ 180 ยูอาร์แอลที่ความเร็วในการจำลอง 1 ยูอาร์แอลต่อ 1 นาที การทดสอบจะใช้เวลาประมาณ 3 ชั่วโมง (เนื่องจากถ้าใช้ระยะเวลา นานกว่า 5 ชั่วโมงทางบริษัทสามารถอาจตัดการเชื่อมต่อจากทางบริษัทได้)
- กรณีที่ 2 ทดสอบ 4,000 ยูอาร์แอลที่ความเร็วในการจำลอง 16.67 ยูอาร์แอลต่อ 1 นาที จะใช้เวลาในการทดสอบ 4 ชั่วโมง

กรณีที่ 1

ทำการเลือกยูอาร์แอลมา 180 ยูอาร์แอลจากแฟ้มข้อมูล final.txt ทำการบันทึกลงแฟ้มข้อมูลชื่อ output.txt จากนั้นทำสำเนาแฟ้มข้อมูล output.txt ลงในไดเรกทอรี "c:\windows\desktop" ของเครื่องทดสอบทั้ง 5 เครื่อง จากนั้นทำการสลับตำแหน่งของยูอาร์แอลในแฟ้มข้อมูล output.txt ของในเครื่องทดสอบทั้ง 5 เครื่องเพื่อให้แตกต่างกัน

จากนั้นทำการทดสอบทำโดยทำการจำลองการร้องขอยูอาร์แอลของผู้ใช้ โดยทำการแก้ไขโปรแกรมให้โปรแกรมทำการอ่านแฟ้มข้อมูล output.txt เพื่อทำการร้องขอให้โปรแกรมเน็ตสเคปทำการเปิดยูอาร์แอลในแฟ้ม output.txt ทุก 1 ยูอาร์แอลต่อ 1 นาที การทดสอบเป็นไปได้ด้วยดี แต่บางครั้งอาจจำเป็นต้องทำการคลิกเพื่อปิดกล่องข้อความแสดงความผิดพลาด ซึ่งเกิดจากการที่โปรแกรมเน็ตสเคป ไม่สามารถทำการโหลดยูอาร์แอลดังกล่าวได้ อันสืบเนื่องมาจากปัญหาของเครือข่าย ซึ่งอาจเกิดได้หลายกรณีตั้งแต่ การที่โปรแกรมเน็ตสเคปไม่สามารถทำการแปลงชื่อยูอาร์แอลเป็นเลขที่อยู่ไอพีได้ ซึ่งอาจจะเกิดจากยูอาร์แอลดังกล่าวไม่มีอยู่จริง หรือเครื่องบริการชื่อโดเมน (DNS Server) ไม่สามารถให้บริการในขณะนั้น ปัญหาการหมดเวลารอ (timeout) ซึ่งอาจเกิดจากตัวบริการเว็บไม่สามารถที่จะทำการบริการได้ในขณะนั้น หรือเกิดจากปัญหาความคับคั่งในเครือข่ายอินเทอร์เน็ตของบริษัทสามารถในขณะนั้น

กรณีที่ 2

จากนั้นเมื่อทำการทดสอบที่ 1 ยูอาร์แอลต่อ 1 นาที เสร็จสิ้นลงก็ทำการดัดแปลงโปรแกรมให้ทำการจำลองการร้องขอยูอาร์แอลไปยังโปรแกรมเน็ตสเคป ทุก 16.67 ยูอาร์แอลต่อ 1 นาที เพื่อที่จะใช้ยูอาร์แอลทำการทดสอบในแต่ละเครื่องเท่ากับยูอาร์แอลทั้งหมดที่ได้มาจากการเลือกจากแฟ้มปุมซึ่งมีประมาณ 4,000 ยูอาร์แอล (ซึ่งจะทำให้ต้องใช้เวลาในการทดสอบ 4 ชั่วโมงซึ่งไม่เกินกำหนดเวลา 5 ชั่วโมงในการตัดการเชื่อมต่อ) ซึ่งเก็บอยู่ในแฟ้มข้อมูล final.txt ทำการสำเนาแฟ้มข้อมูล final.txt ไปยังไดเรกทอรี "c:\windows\desktop" ของเครื่องทดสอบทั้ง 5 เครื่อง จากนั้นทำการสลับตำแหน่งของยูอาร์แอลในแฟ้มข้อมูล final.txt ของในเครื่องทดสอบทั้ง 5 เครื่องเพื่อให้แตกต่างกัน ซึ่งถ้าการทดสอบเป็นไปอย่างปกติจะใช้เวลาประมาณ 4 ชั่วโมงในการดำเนินการโปรแกรม จากนั้นเริ่มการดำเนินการโปรแกรมที่ทำการแก้ไขแล้ว พบว่าขณะที่ทำการดำเนินการจำลองการร้องขอไปได้ประมาณ 2,000 ยูอาร์แอล โปรแกรมที่ทำการแก้ไขเกิดข้อผิดพลาด เกิดการจบโปรแกรม และยังทำให้โปรแกรมเน็ตสเคปหยุดทำงานด้วย เมื่อทำการทดสอบอีกครั้ง ก็พบปัญหาเช่นเดียวกัน ซึ่งปัญหาดังกล่าวอาจสืบเนื่องมาจากกรณีที่โปรแกรมเน็ตสเคป ไม่สามารถทำการเปิดยูอาร์แอลดังกล่าวให้กับผู้ใช้ ซึ่งเวลาที่โปรแกรมเน็ตสเคป ไม่สามารถทำการเปิดยูอาร์แอลได้โปรแกรมเน็ตสเคป จะทำการแสดงกล่องข้อความแสดงความผิดพลาดของการโหลด เพื่อให้ผู้ใช้

ทำการทราบและคลิกเพื่อทำการปิด จากนั้นจึงเริ่มทำการร้องขอยูอาร์แอลตัวใหม่ แต่ในที่นี้ โปรแกรมไม่สามารถทำการจำลองการทำงานเพื่อการคลิกปิดกล่องข้อความแสดงความผิดพลาด ดังนั้นเมื่อเกิดความผิดพลาดและกล่องข้อความแสดงความผิดพลาดยังคงอยู่จะไม่สามารถทำการเปิดยูอาร์แอลถัดไปในแฟ้มข้อมูลได้ ตรงส่วนนี้เองอาจทำให้โปรแกรมที่แก้ไขเพื่อจำลองการใช้งานของผู้ใช้เกิดความผิดพลาดได้ สำหรับสาเหตุของความผิดพลาดของการหยุดการทำงานอย่างผิดปกติตามที่ระบุไว้ในเอกสารของไมโครซอฟท์ [8] ได้แก่

1. ความผิดพลาดของสแต็กภายในระบบ โดยทั่วไปมีสาเหตุมาจากการพยายามอ้างอิงที่อยู่ในหน่วยความจำที่เกินช่วงของสแต็กเซกเมนต์
2. การทำงานคำสั่งที่ไม่ถูกต้อง โดยทั่วไปเกิดจากการพยายามไปเริ่มทำคำสั่งในช่วงที่เป็นข้อมูลหรือการเปลี่ยนแปลงคำสั่งในหน่วยความจำจนทำให้คำสั่งผิด
3. การคำนวณที่ได้ผลผิดพลาด โดยเฉพาะการหาร เช่นการหารด้วยศูนย์ เป็นต้น
4. ข้อผิดพลาดในการอ้างอิงหน่วยความจำ เช่นการอ้างอิงเกินขอบเขต (เนื่องมาจากการคำนวณตำแหน่งของหน่วยความจำผิดพลาด) หรือ การอ้างอิงหน่วยความจำตำแหน่ง 0 (NULL pointer) หรือการเขียนในหน่วยความจำที่อนุญาตให้อ่านอย่างเดียว เป็นต้น

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

การพัฒนาโปรแกรมเพื่อให้สามารถทำการใช้งานเว็บเบราว์เซอร์ร่วมกันของโปรแกรมค้นผ่านเว็บ
วิทยานิพนธ์นี้สามารถสรุปได้ดังนี้

6.1 สรุปผลการวิจัย

จากผลการทดลองในบทที่ 5 จะพบว่าเราสามารถทำการทดสอบโปรแกรมเพื่อทำการใช้
เว็บเบราว์เซอร์ร่วมกันได้เป็นอย่างดีสำหรับการจำลองแบบแรกที่ประมาณการร้องขอยูอาร์แอลทุก 1
นาทิต แต่ในกรณีที่สองสามารถทำการเปิดได้เพียงประมาณ 2,000 ยูอาร์แอลที่ 16.67 ยูอาร์แอลต่อ
1 นาทิต แล้วโปรแกรมเกิดการล้มเหลว แต่ถ้าเปลี่ยนเวลาในการจำลองในกรณีที่สองมาเท่ากับใน
กรณีแรกคือ 1 ยูอาร์แอลต่อ 1 นาทิต โปรแกรมนี้จะดำเนินงานได้นานถึง 11 ชั่วโมง แต่ในกรณีนี้
ไม่สามารถทดสอบได้ เนื่องจากทางบริษัทสามารถอาจมีการตัดการเชื่อมต่อหลังจาก 5 ชั่วโมง

6.2 ข้อจำกัด

ในการใช้งานโปรแกรมบริการดีดีของเน็ตสเคปเพื่อทำการโหลดเว็บเพจนั้น หลังจากที่เรา
ทำการลงทะเบียนโปรแกรมบริการดีดีที่ทำการพัฒนาขึ้นแล้วนั้น บางครั้งเว็บเพจที่สั่งให้โหลดไม่
สามารถแสดงผลได้ครบถ้วนสมบูรณ์ และในกรณีที่เว็บเพจมีจาวาสคริปต์ (Java Script) หรือเป็น
CGI หรือมีเอกสารที่ต้องการโปรแกรมแสดงภายนอก (External Viewer) การทำงานด้วยวิธี
ดังกล่าวจะไม่สามารถโหลดเว็บเพจได้สมบูรณ์ และบางครั้งยังทำให้โปรแกรมเน็ตสเคปเองหยุด
การทำงานอีกด้วย เนื่องด้วยจากลักษณะการทำงานของโปรแกรมบริการดีดีที่ต้องการส่งข้อ
ความระหว่างการดำเนินงาน บางครั้งจะเกิดอาการในลักษณะที่โปรแกรมบริการไม่สามารถให้
บริการได้ (Server Busy) และทำให้โปรแกรมที่ติดต่อกับโปรแกรมบริการดีดีนั้น หยุดการ
ทำงานไปด้วย ในเน็ตสเคป รุ่นที่ 6 ยังไม่มีข้อมูลมากพอว่าสนับสนุนหัวข้อบริการเดิมทั้งหมด ที่มี
อยู่ในโปรแกรมบริการดีดีของเน็ตสเคปรุ่นก่อนหน้าหรือไม่ ทราบแต่เพียงมีข้อมูลว่าชื่อของ
โปรแกรมบริการดีดีที่ใช้ในการติดต่อกับ เน็ตสเคป รุ่นที่ 6 นั้นคือ "NETSCAPE6"

ในอนาคตอันใกล้ต้องยอมรับว่าความนิยมในโปรแกรมเน็ตสเคป จะมีน้อยลงอีกทั้งผู้
พัฒนาโปรแกรมส่วนใหญ่ได้ให้ความสนใจในโปรแกรม Internet Explorer มากกว่าเนื่องจากมี
ข้อมูลสนับสนุนและมีความเข้ากันได้กับระบบปฏิบัติการวินโดวส์ที่ใช้ในปัจจุบัน และเป็นผลเนื่องมา
จากโปรแกรม Internet Explorer มีความสามารถที่มากกว่า เช่น สนับสนุน Active X เป็นต้น และ
ในส่วนของการทำงานโหลดเว็บเพจก็ไม่ได้ใช้ดีดีแต่ใช้ COM อ็อบเจกต์แทน ซึ่งมีการทำงานที่เร็ว
กว่า

6.3 ข้อเสนอแนะ

สามารถนำบางส่วนของงานนี้ไปพัฒนาต่อในส่วนการทำการกรองยูอาร์แอลที่ไม่ต้องให้ผู้ใช้ทำการเปิด หรือการทำโปรแกรมในการเฝ้าสังเกตยูอาร์แอลของผู้ใช้ได้ หรือพัฒนาต่อให้มีกลไกการใช้งานเว็บแคชร่วมกันในรูปแบบที่ซับซ้อนขึ้น



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

- [1] J.Yang,W. Wang and R.Muntz. Dynamic Web Caching. NDSU Technical Report 980042. University of California,Los Angeles,1998.
- [2] M. Baentsch, L. Baum, G. Molter, S. Rothkugel and P. Strum. World Wide Web Caching: The Application-Level View of the Internet. IEEE Communication Magazine,June 1997.
- [3] L.Fan,P.Cao,J.Almeida and A.Broder. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. Proc. Of ACM SIGCOMM Conference,1998.
- [4] I.Tatrinov, A.Rousskov and V. Soloviev. Static Caching In Web Server. NDSU Technical Report NDSU-CSOR-TR-97-04. North Dakota State University. Computer Science and Operations Research,April 1997.
- [5] M. Abrams, C. Standridge, G. Abdulla, S. Williams and E. Fox. CachingProxies: Limitations and Potentials. NDSU Technical Report TR-95-12. Virginia Polytechnic Inst. And State University,1995.
- [6] D.Wessels and K. Claffy. Internet Cache Protocol (ICP),version 2. RFC 2186,1997.
- [7] L.Napper.WinSock 2.0. IDG Books Worldwide,1998.
- [8] Microsoft.Cause of General Protection Faults, Microsoft Developers' Network : Window 3.x Knowledge Base, Version 3.0. Microsoft Inc.,April, 1995.
- [9] B.Lam.The Netscape Programmer's Guide.CAMBRIDGE UNIVERSITY PRESS,1998.



ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

ตารางแสดงรายละเอียดฟังก์ชันของโปรแกรมบริการดีดีอีของเน็ตสเคป

ภาคผนวก ก นี้ได้ทำการแสดงรายละเอียดฟังก์ชันต่าง ๆ ของโปรแกรมบริการดีดีอีของเน็ตสเคป ซึ่งโปรแกรมบริการดีดีอีของเน็ตสเคปใช้ชื่อโปรแกรมบริการดีดีอีในการเรียกใช้งานว่า "NETSCAPE" ตารางที่ ก.1 แสดงรายละเอียดหัวข้อบริการโดยทางด้านซ้ายของตารางก็คือหัวข้อบริการที่ใช้เพื่อควบคุมการทำงานของโปรแกรมเน็ตสเคป ส่วนทางตรงกลางเป็นคำอธิบายของแต่ละหัวข้อบริการ และทางด้านขวาอธิบายประเภทของรายการที่เกิดขึ้น

ตารางที่ ก.1 แสดงรายละเอียดหัวข้อบริการต่าง ๆ ของโปรแกรมบริการดีดีอีของเน็ตสเคป

<i>Topic</i>	<i>Description</i>	<i>Transaction Type</i>
WWW_Activate	Activate a Navigator window.	XTYPE_REQUEST
WWW_Alert	Display an alert box.	XTYPE_REQUEST
WWW_BeginProgress	Start sending progress topics.	XTYPE_REQUEST
WWW_CancelProgress	Notify Navigator to cancel download associated with the transaction.	XTYPE_POKE
WWW_EndProgress	Indicate that loading of the URL is complete.	XTYPE_POKE
WWW_Exit	Cause Navigator to attempt to exit.	XTYPE_POKE
WWW_GetWindowInfo	Get information about a Navigator window.	XTYPE_REQUEST
WWW_ListWindows	List all Navigator windows.	XTYPE_REQUEST
WWW_MakingProgress	Describe current progress with a range value.	XTYPE_REQUEST
WWW_OpenURL	Open a URL.	XTYPE_REQUEST
WWW_ParseAnchor	Resolve a relative URL.	XTYPE_REQUEST
WWW_QueryURLFile	Return the URL from which a file was loaded.	XTYPE_REQUEST
WWW_QueryViewer	Query a viewer with a filename.	XTYPE_REQUEST
WWW_RegisterProtocol	Register a custom protocol.	XTYPE_REQUEST
WWW_RegisterURLEcho	Register a notification for URL loading.	XTYPE_POKE

ตารางที่ ก.1 (ต่อ) แสดงรายละเอียดหัวข้อบริการต่าง ๆ ของโปรแกรมบริการดีดีของเน็ตสเค๊ป

<i>Topic</i>	<i>Description</i>	<i>Transaction Type</i>
WWW_RegisterViewer	Register a viewer with a MIME type.	XTYPE_REQUEST
WWW_RegisterWindowChange	Register a notification when a given Navigator window changes.	XTYPE_REQUEST
WWW_SetProgressRange	Set the progress range for progress messages.	XTYPE_POKE
WWW_ShowFile	Have Navigator load a file.	XTYPE_REQUEST
WWW_UnRegisterProtocol	Unregister the protocol registered with WWW_RegiesterProtocol.	XTYPE_REQUEST
WWW_UnRegisterURLEcho	Unregister echo registered with WWW_RegisterURLEcho.	XTYPE_POKE
WWW_UnRegisterViewer	Unregister viewer registered with WWW_RegisterViewer.	XTYPE_REQUEST
WWW_UnRegisterWindowChange	Unregister window changes registered with WWW_RegisterWindowChange.	XTYPE_REQUEST
WWW_URLEcho	Navigator loaded a URL.	XTYPE_POKE
WWW_Version	Check Navigator DDE version.	-
WWW_ViewDocFile	A file for a registered viewer.	XTYPE_POKE
WWW_WindowChange	A given Navigator window has changed.	XTYPE_POKE

ภาคผนวก ข

คู่มือการใช้งานโปรแกรมเพื่อช่วยการใช้งานเว็บแคชร่วมกันระหว่างโปรแกรมค้นผ่านเว็บ

ภาคผนวก ข นี้จะทำการแสดงส่วนของโปรแกรม และวิธีการใช้งาน โดยโปรแกรมสามารถรองรับการทำงานบน วินโดวส์ 9x และสามารถควบคุมได้เฉพาะเน็ตสเคป รุ่นที่ 4.x

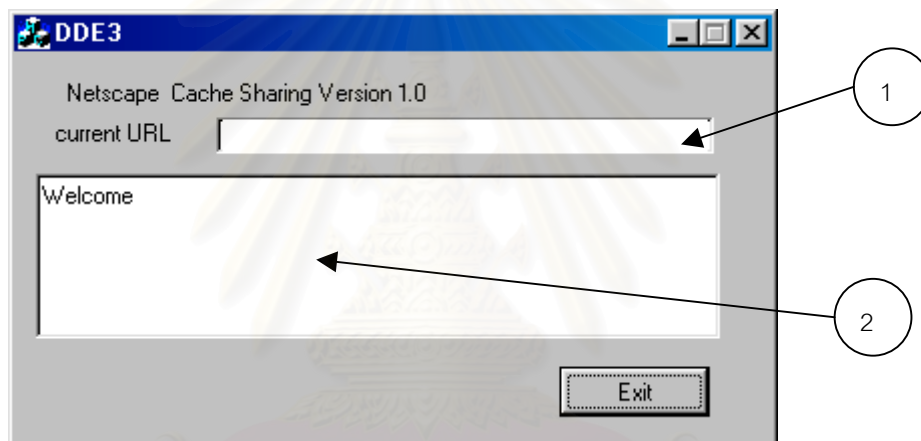
การเรียกใช้งานโปรแกรม

ทำการ Click ICON



dde3

จากนั้นเมื่อโปรแกรมทำงานก็จะแสดงวินโดว์



เมื่อเราทำการเปิดโปรแกรมเน็ตสเคป ยูอาร์แอลที่โปรแกรมเน็ตสเคป พยายามโหลดจะปรากฏในส่วนที่ 1 และสถานะการทำงานจะถูกแสดงในส่วนที่ 2 และเมื่อเราต้องการทำการปิดโปรแกรมก็สามารถทำได้โดยใช้ ปุ่ม Exit

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายยุทธกานต์ ปรงเกียรติ เกิดวันที่ 10 พฤศจิกายน พ.ศ. 2516 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรีวิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์ จากภาควิชาวิทยาศาสตร์คอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ เมื่อปี พ.ศ.2538 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย