สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง วิทยาเขตระยอง

ระบบปิดกั้นเว็บไซต์ที่ไม่เหมาะสม

WEB HOST BLOCKING SYSTEM

โดย

นางสาวกุสุมา สมอทอง

นายธีระศักดิ์ หงษ์น้อย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WEB HOST BLOCKING SYSTEM

BY

MISS. KUSUMA SAMORTHONG

MR. THEERASAK HONGNOI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบปิดกั้นเว็บไซต์ที่ไม่เหมาะสม Web Host Blocking System
ชื่อนักศึกษา	นางสาวกุสุมา สมอทอง รหัสนักศึกษา 44015638 นายธีระศักดิ์ หงษ์น้อย รหัสนักศึกษา 44015693
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ มยุรี เลิศเวชกุล
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2546

ปริญญานิพนธ์ฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเรียบร้อยแล้ว

.....

(ผู้ช่วยศาสตราจารย์ มยุรี เลิศเวชกุล)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบปิดกั้นเว็บไซต์ที่ไม่เหมาะสม
ชื่อนักศึกษา	นางสาวกุสุมา สมอทอง รหัสนักศึกษา 44015638
	นายธีระศักดิ์ หงษ์น้อย รหัสนักศึกษา 44015693
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์มยุรี เลิศเวชกุล
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต
	สาขาวิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2546

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของโครงงานวิจัย เพื่อสร้างโปรแกรมสำหรับกรองเนื้อหาในเว็บเพจที่เครื่องลูกข่ายร้องขอจากเว็บไซต์ต่าง ๆ บนเครือข่ายอินเตอร์เน็ต

โปรแกรมจะทำหน้าที่เปรียบเสมือนเป็นพร็อกซี่ โดยเป็นตัวเชื่อมระหว่างเครื่องไคลเอนต์และเว็บเซิร์ฟเวอร์ ทำให้ข้อมูลที่เครื่องทั้งสองสื่อสารกันนั้นจะต้องผ่านโปรแกรมปิดกั้นเว็บไซต์ไม่เหมาะสมนี้ก่อนเสมอ ซึ่งต้องแน่ใจได้ว่าข้อมูลที่ส่งจากเว็บเซิร์ฟเวอร์มาสู่ไคลเอนต์นั้นต้องมีเนื้อหาที่เหมาะสมเท่านั้น โดยการกรองข้อมูลที่ส่งผ่านโปรแกรมนั้นจะมี 2 แบบ คือ การกรอง URL จากการร้องขอการเข้าถึงเว็บไซต์ต่าง ๆ ของเครื่องไคลเอนต์ และการกรองเนื้อหาที่ส่งมาจากเว็บเซิร์ฟเวอร์ไปยังเครื่องไคลเอนต์จาก และข้อมูลที่ใช้ในการพิจารณาในการปิดกั้นเว็บไซต์ไม่เหมาะสมนั้นจะเก็บไว้ในไฟล์ ต่าง ๆ ซึ่งมี 3 ไฟล์ด้วยกันคือ Black List, White List และ Word List

ในส่วนของผลการดำเนินการนั้นเป็นไปตามขอบเขตที่กำหนดไว้ คือ สามารถปิดกั้นเว็บไซต์ที่ไม่เหมาะสมโดยผู้ใช้ไม่สามารถเข้าถึงข้อมูลนั้น ๆ ได้ และสามารถเพิ่มหรือแก้ไขข้อมูลที่ใช้ในการพิจารณาในการปิดกั้นเว็บไซต์ไม่เหมาะสมนั้นได้จากส่วนติดต่อผู้ใช้ (User Interface) และยังได้เพิ่มส่วนของการอัพเดท URL ไปยังไฟล์ Black List ได้โดยอัตโนมัติเมื่อพบว่าเว็บไซต์นั้นมีเนื้อหาที่ไม่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Web Host Blocking System	
Student	Miss. Kusuma Samorthong	ID. 44015638
	Mr. Theerasak Hongnoi	ID. 44015693
Advisor	Asst. Prof. Mayuree Lertvachakul	
Graduate Level	Bachelor Degree of Information Engineering	
Department	Information Engineering	
Academic Year	2003	

ABSTRACT

The objective of this project is to implement a web content filtering program for blocking inappropriate web site when client requests to access the internet.

The program verifies URL requests to make sure that the destination IP address and destination domain name is not in the Black List. For every response, the content will be filtered by proxy before sending to client. Black List, White List and Word List are the data that is used to consider which web site is inappropriate or not.

Experiment of using the program has shown that the request to the inappropriate web site has been blocked and also the inappropriate response contents. Moreover, the program automatically update inappropriate URL web site into Black List when it founds the inappropriate content on web page.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณผู้ช่วยศาสตราจารย์ มยุรี เลิศเวชกุล อาจารย์ที่ปรึกษาปริญญานิพนธ์ผู้ริเริ่มโครงงานนี้ขึ้นมา ที่คอยให้ความช่วยเหลือและให้คำปรึกษาตลอดระยะเวลาที่ทำปริญญานิพนธ์เป็นอย่างดี และขอขอบคุณคณาจารย์และเพื่อนๆ ที่ให้ความช่วยเหลือแนะนำ สุดท้ายนี้ขอกราบขอบพระคุณบิดาและมารดา ที่คอยให้กำลังใจและให้การสนับสนุนเสมอมา

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

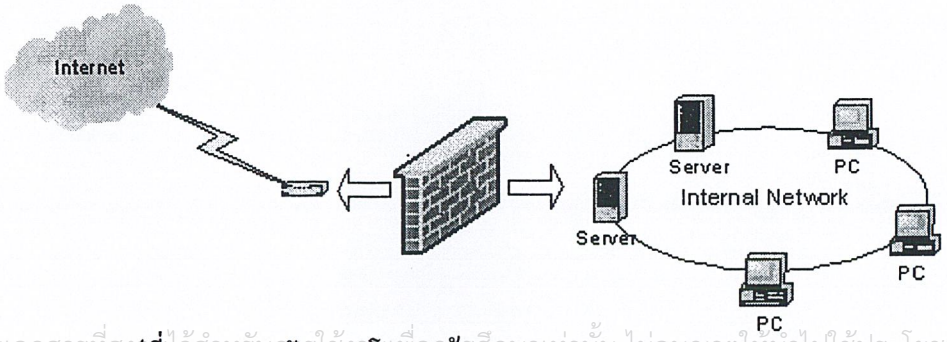
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 แนวคิดและที่มาของปัญหา

ปัจจุบันมีการใช้งานอินเตอร์เน็ตอย่างแพร่หลาย ดังจะเห็นได้จากการเพิ่มขึ้นของจำนวนผู้ใช้งานในช่วงเวลาที่ผ่านมา เนื่องจากมีการพัฒนาในด้านเทคโนโลยีคอมพิวเตอร์ให้สามารถใช้งานได้สะดวกมากขึ้น จึงทำให้บุคคลทั่วไปนั้นมีอิสระในการเข้าถึงข้อมูลในเครือข่ายอินเตอร์เน็ตมากขึ้นตามไปด้วย ส่วนใหญ่การใช้งานอินเตอร์เน็ตจะเป็นไปในเรื่องของการติดต่อสื่อสารผ่านไปรษณีย์อิเล็กทรอนิกส์ และการพูดคุย (ทั้งแบบพิมพ์ข้อความโต้ตอบและใช้เสียงพูด), การสืบค้นข้อมูลด้วยเว็บบราวเซอร์, การโอนย้ายแฟ้มข้อมูล เช่น โปรแกรมต่าง ๆ การทำธุรกรรมพาณิชย์อิเล็กทรอนิกส์รวมไปถึงการให้บริการด้านบันเทิง และอินเตอร์เน็ตยังเป็นแหล่งข้อมูลสำหรับการเรียนรู้ที่ไม่มีขีดจำกัดอีกด้วย นอกจากอินเตอร์เน็ตจะเป็นแหล่งรวบรวมข้อมูลต่าง ๆ ที่มีประโยชน์แล้ว อินเตอร์เน็ตก็ยังเป็นแหล่งรวบรวมข้อมูลที่ไม่เหมาะสมอีกมากมายด้วย และการที่บุคคลทั่วไปมีอิสระในการเข้าถึงข้อมูลต่าง ๆ ได้ ทำให้มีการเข้าถึงข้อมูลที่ไม่เหมาะสม ดังนั้นจึงควรมีการป้องกันการเข้าถึงข้อมูลที่ไม่เหมาะสมนี้

จากการที่บุคคลภายในเครือข่ายสามารถเข้าถึงข้อมูลที่ไม่เหมาะสมบนอินเตอร์เน็ตได้โดยง่าย จึงมีแนวคิดที่จะทำโครงงานเพื่อปิดกั้นการเข้าถึงเว็บไซต์ที่ไม่เหมาะสม โดยมีแนวทางในการออกแบบโปรแกรมให้หน้าที่คัดกรองการร้องขอที่จะส่งไปยังเว็บเซิร์ฟเวอร์ และกรองเนื้อหาบนเว็บเพจที่จะเข้าสู่เครือข่ายภายใน โดยโครงสร้างการทำงานแสดงดังรูปที่ 1.1 เมื่อมีข้อมูลเข้าออกจะต้องผ่านโครงงานนี้ซึ่งเป็นตัวคั่นกลางเสมอ จึงทำให้สามารถกรองข้อมูลที่เข้าออกอินเตอร์เน็ตได้ แนวทางดังกล่าวจะเป็นแนวทางในการทำโครงงานนี้ต่อไป



รูปที่ 1.1 แสดงลักษณะโครงสร้างของ Web Host Blocking System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์

1.2.1 เพื่อป้องกันการเข้าถึงเว็บไซต์ที่มีเนื้อหาที่ไม่เหมาะสมของบุคคลภายในเครือข่าย

1.2.2 เพื่อความสะดวกในการเพิ่มหรืออัพเดทเว็บไซต์ที่ไม่เหมาะสม

1.2.3 เพื่อศึกษาการทำงานของระบบที่จะนำมาใช้ในการปิดกั้นเว็บไซต์ที่ไม่เหมาะสม

1.2.4 เพื่อนำความรู้ที่ได้ศึกษามาประยุกต์ใช้ในการทำโครงงาน

1.3 ขอบเขตของโครงงาน

1.3.1 สามารถกรองเว็บไซต์ที่มีเนื้อหาที่ไม่เหมาะสมได้ โดยที่บุคคลในเครือข่ายไม่สามารถเข้า ถึงเว็บไซต์ที่มีเนื้อหาที่ไม่เหมาะสมได้ โดยคัดกรองจากข้อความที่เป็นภาษาอังกฤษที่อยู่บนเว็บเพจ

1.3.2 สามารถเพิ่มหรืออัพเดทเว็บไซต์สำหรับใช้ในการปิดกั้นเว็บไซต์ที่ไม่เหมาะสมได้ โดยการสร้างส่วนติดต่อกับผู้ใช้งาน (Graphic User Interface) ซึ่งจะทำให้ผู้ใช้สามารถเพิ่มหรืออัพเดทเว็บไซต์ได้อย่างสะดวก

1.4 ผลที่คาดว่าจะได้รับ

1.4.1 สามารถป้องกันไม่ให้บุคคลภายในเครือข่ายเข้าถึงเว็บไซต์ที่มีเนื้อหาที่ไม่เหมาะสมได้

1.4.2 สามารถเพิ่มความสะดวกในการเพิ่มหรืออัพเดทเว็บไซต์ที่ไม่เหมาะสมได้

1.4.3 สามารถศึกษาการทำงานของระบบที่จะนำมาใช้ในการปิดกั้นเว็บไซต์ที่ไม่เหมาะสมได้

1.4.4 สามารถนำความรู้ที่ได้ศึกษามาประยุกต์ใช้ในการทำโครงงานได้

1.5ขั้นตอนการดำเนินโครงงาน

1.5.1 ศึกษาหาข้อมูล

1.5.2 วิเคราะห์และออกแบบการทำงานของระบบ

1.5.3 สร้างระบบที่สามารถปิดกั้นเว็บไซต์ที่ไม่เหมาะสม

1.5.4 ทดสอบระบบและแก้ไข

1.5.5 สรุปผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

เนื่องจากการใช้งานอินเตอร์เน็ตในปัจจุบันมีการพัฒนาในด้านเทคโนโลยีเป็นอย่างมาก ทำให้มีอุปกรณ์และคอมโพเนนต์ (component) ในการให้บริการอินเตอร์เน็ตเกิดขึ้นมากมาย และมีรูปแบบในการให้บริการที่แตกต่างกันไปขึ้นอยู่กับความต้องการของผู้ใช้งานในแต่ละคน โดยคอมโพเนนต์ของระบบเครือข่ายอินเตอร์เน็ตนั้นมีหลากหลายหน้าที่ ซึ่งจะมีคุณสมบัติที่แตกต่างกันไปตามวัตถุประสงค์ ทั้งใช้เพื่อความสะดวกรวดเร็วในการติดต่อสื่อสาร, เพื่อความถูกต้องแม่นยำของข้อมูลที่รับส่ง หรือแม้กระทั่งใช้เพื่อความปลอดภัยของการส่งข้อมูล ซึ่งในคอมโพเนนต์ที่ใช้สำหรับรักษาความปลอดภัยของข้อมูลนั้นส่วนใหญ่แล้วจะมีคุณสมบัติในการคัดกรองข้อมูลที่ส่งผ่านคอมโพเนนต์นั้นสู่อินเตอร์เน็ตและเครือข่ายภายใน

ในการสร้างแอพพลิเคชั่น (Application) ให้มีความสามารถในการกรองเนื้อหาที่มีการรับส่งกันบนอินเตอร์เน็ตและการปิดกั้นการเข้าถึงข้อมูลบนอินเตอร์เน็ตของเครื่องลูกข่ายนั้น จะต้องมีส่วนที่ต้องพิจารณาร่วมกันหลายประการ เช่น พิจารณาว่าคอมโพเนนต์ของเครือข่ายใดที่เราจะนำคุณสมบัติมาทำการประยุกต์ หรือทำการดำเนินการ (Implement) ร่วมกันกับตัวโครงงานเพื่อให้ได้ผลที่ดีที่สุด และเป็นแนวทางในการปฏิบัติที่เหมาะสมที่สุด รวมถึงความเข้าใจ ในเรื่องโปรโตคอล (protocol) ที่ใช้, หลักการเขียนโปรแกรม และอัลกอริทึมที่ใช้สำหรับการกรองเนื้อหา ซึ่งได้รวบรวมเนื้อหาที่จำเป็น ต้องใช้งานดังนี้

2.1 การให้บริการทางอินเตอร์เน็ต

2.1.1 เว็บพร็อกซี่ (Web Proxy)

หลักการทำงานของพร็อกซี่ประเภทนี้ คือ การอนุญาตให้เครื่องไคลเอนต์ (Client) ภายในเครือข่ายและหลังไฟร์วอลล์ (firewall) ใช้งานอินเตอร์เน็ตผ่านตัวมันได้ โดยพร็อกซี่จะรอรับการร้องขอจากเครื่องไคลเอนต์ที่อยู่หลังไฟร์วอลล์ และจะส่งต่อการร้องขอนี้ไปยังเซิร์ฟเวอร์ (server) ปลายทางที่อยู่ด้านนอกไฟร์วอลล์ จากนั้นก็รอรับการตอบกลับมาแล้วส่งไปให้แก่เครื่องไคลเอนต์ต่อไป [1]

ปกติแล้วเครื่องไคลเอนต์ทุกเครื่องที่อยู่ในซับเน็ต (subnet) เดียวกันจะใช้งานเครื่องพร็อกซี่ตัวเดียวกัน ซึ่งจะทำให้เครื่องพร็อกซี่สามารถสำเนาข้อมูลชั่วคราวได้อย่างมีประสิทธิภาพ ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพจะเพิ่มขึ้นตามจำนวนของเครื่องไคลเอนต์ คนที่ใช้งานเครื่องพร็อกซี่จะรู้สึกราวกับว่าได้รับการตอบรับโดยตรงจากเครื่องเซิร์ฟเวอร์ที่ร้องขอไปแต่ว่ามีความรวดเร็วมากยิ่งขึ้น

เครื่องไคลเอนต์ที่ใช้งานไพรเวตไอพี (private IP) ก็สามารถที่จะใช้งานอินเตอร์เน็ตได้เช่นกัน เพียงแค่รู้ไอพีแอดเดรส (IP address) ของเครื่องพร็อกซี่เท่านั้น ในองค์กรที่ใช้ไอพีแอดเดรสในช่วงที่เป็นไพรเวตไอพี เช่นคลาส A ซึ่งเป็นลักษณะ 10.*.*.* ก็สามารถที่จะใช้งานอินเตอร์เน็ตได้ ถ้าเครื่องพร็อกซี่นั้นสามารถมองเห็นได้จากทั้งฝั่งที่เป็นไพรเวตไอพีภายในองค์กรและฝั่งอินเตอร์เน็ต

เครื่องพร็อกซี่ส่วนใหญ่จะถูกติดตั้งมาให้รองรับงานบริการเดียวเป็นหลัก ซึ่งพร็อกซี่เซิร์ฟเวอร์นั้นสามารถอนุญาตหรือจะปฏิเสธการร้องขอที่ถูกต้องตามที่กำหนดไว้ได้ ตัวอย่างเช่น เครื่องพร็อกซี่อนุญาตให้การร้องขอไปยังเครื่อง HTTP (HTTP) เซิร์ฟเวอร์ผ่านไปได้ในขณะที่ไม่อนุญาตให้ร้องขอไปยังเครื่องเอฟทีพี (FTP) เซิร์ฟเวอร์

2.1.1.1 การใช้งานเว็บพร็อกซี่

เราสามารถใช้งานพร็อกซี่เซิร์ฟเวอร์ได้หลายๆ อย่างโดยรวมทั้งความสามารถดังนี้

1.) อนุญาตและจำกัดการใช้งานอินเตอร์เน็ตของเครื่องไคลเอนต์ โดยวิธีการตรวจสอบหมายเลขไอพี
2.) สำเนาข้อมูลจากภายนอกเก็บไว้ เพื่อให้เครื่องภายในที่ต้องการข้อมูลเดียวกันเข้าถึงข้อมูลได้เร็วยิ่งขึ้น
3.) สามารถควบคุมการเข้าใช้งานอินเตอร์เน็ตและซับเน็ตที่ต้องการได้ โดยกำหนด URL (URL) เป็นหลัก
4.) ให้บริการการใช้งานอินเตอร์เน็ตแก่ระบบเครือข่ายที่ใช้ไอพีในช่วงไพรเวตไอพี
5.) แปลงข้อมูลให้อยู่ในรูปแบบของเอชทีเอ็มแอล (HTML) ซึ่งสามารถอ่านได้โดยบราวเซอร์

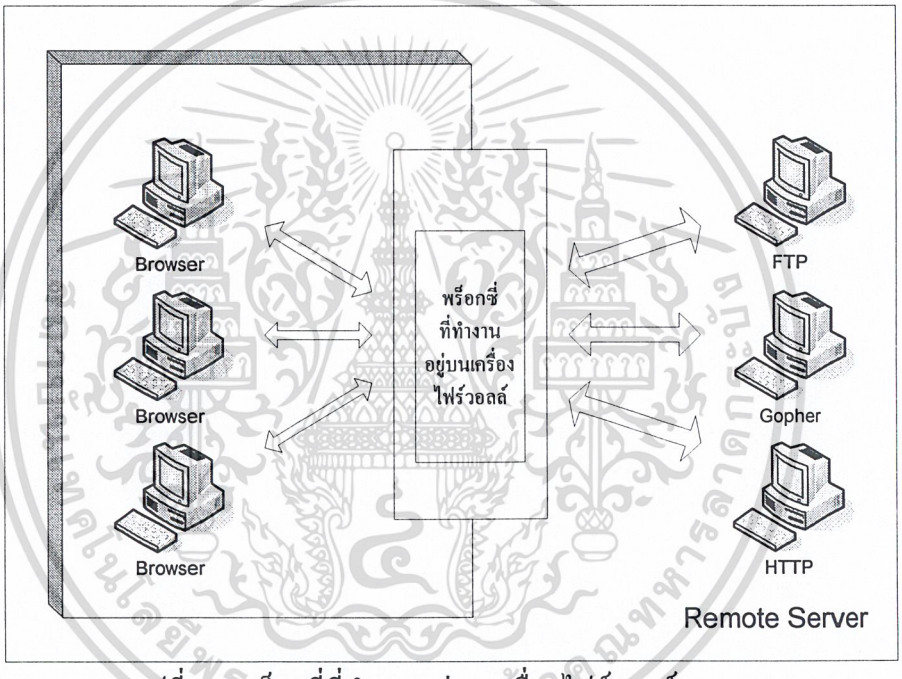
2.1.1.2 การเรียกใช้อินเตอร์เน็ตของบราวเซอร์

มีเครื่องบางเครื่องในเครือข่ายของเราที่ไม่สามารถเข้าใช้งานอินเตอร์เน็ตได้โดยตรง ตัวอย่างเช่น บราวเซอร์บางตัวไม่สามารถเข้าใช้งานอินเตอร์เน็ตได้โดยตรง เป็นเพราะว่าเครื่องที่ใช้งานอยู่นั้นอยู่ภายใต้การปกป้องโดยไฟร์วอลล์ ซึ่งในกรณีนี้พร็อกซี่เซิร์ฟเวอร์สามารถที่จะไปดึงข้อมูลที่ต้องการมาให้แก่บราวเซอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.1 พร็อกซี่เซิร์ฟเวอร์ทำงานอยู่บนเครื่องไฟร์วอลล์และได้มีการเชื่อมต่อไปยังอินเตอร์เน็ตโดยใช้ซอฟต์แวร์ของไฟร์วอลล์ ซึ่งเราสามารถที่จะตั้งเครื่องพร็อกซี่ไว้ที่เครื่องอื่นภายในเครือข่ายที่สามารถเรียกใช้อินเตอร์เน็ตโดยตรงได้ หรือจะทำการติดตั้งไว้ที่เครื่องที่อยู่ในไฟร์วอลล์ก็ได้

เมื่อเครื่องพร็อกซี่รับการร้องขอจากบราวเซอร์ในรูปแบบของ URL เครื่องพร็อกซี่ก็จะไปดึงข้อมูลจากตำแหน่งที่ให้ไว้ใน URL นั้น และแปลงเป็นรูปแบบของ HTML แล้วจึงส่งกลับมายังบราวเซอร์ที่อยู่ด้านหลังไฟร์วอลล์



รูปที่ 2.1 พร็อกซี่ที่ทำงานอยู่บนเครื่องไฟร์วอลล์

2.1.1.3 การสำเนาข้อมูล (cache)

โดยปกติแล้วเครื่องไคลเอนต์ภายในซับเน็ตเดียวก็กันจะใช้พร็อกซี่เซิร์ฟเวอร์ตัวเดียวกัน ซึ่งเครื่องพร็อกซี่บางตัวจะสำเนาข้อมูลสำหรับเครื่องไคลเอนต์ที่อยู่ในเครือข่ายเดียวกัน โดยการสำเนาข้อมูลหมายถึง การสำเนาข้อมูลจากอินเตอร์เน็ตมาเก็บไว้ยังเครื่องภายใน ดังนั้นเครื่องพร็อกซี่จึงไม่จำเป็นที่จะต้องไปร้องขอข้อมูลเหล่านี้มาซ้ำอีก เพียงแต่นำส่งข้อมูลเหล่านี้ไปให้เครื่องไคลเอนต์ที่ต้องการข้อมูลเดียวกันเท่านั้น

การสำเนานั้นจะเป็นประโยชน์แก่พร็อกซี่เซิร์ฟเวอร์เองมากกว่าตัวไคลเอนต์ เนื่องจากทำให้ประหยัดพื้นที่ในการจัดเก็บเพราะว่าจะจัดเก็บสำเนาเพียงชุดเดียว การสำเนาข้อมูลเก็บไว้จะเกิดประโยชน์มากถ้าข้อมูลที่สำเนาเอาไว้ถูกเรียกใช้จากเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

ไคลเอนต์หลาย ๆ เครื่อง ผู้ดูแลระบบสามารถที่จะคาดเดาได้ว่า ข้อมูลชนิดใดบ้างที่เป็นหรือไม่เป็นประโยชน์ถ้ามีการสำเนาไว้เป็นเวลานาน

การสำเนาทำให้สามารถเรียกเว็บได้ในขณะที่เครือข่ายอินเตอร์เน็ตไม่สามารถใช้งานได้ตราบเท่าที่เครื่องที่ร้องขอสามารถเชื่อมต่อกับเครื่องพร็อกซี่ได้ เช่นกรณีที่เครื่องเว็บเซิร์ฟเวอร์ไม่สามารถให้บริการได้ แต่เราก็สามารถที่จะดึงข้อมูลบางส่วนที่สำเนาไว้ได้

2.1.1.4 การควบคุมการใช้งานอินเตอร์เน็ต

เมื่อใช้พร็อกซี่เราสามารถที่จะกรองการใช้งานระดับโปรโตคอลของเครื่องไคลเอนต์ได้ พร็อกซี่สามารถที่จะควบคุมการใช้บริการในแต่ละโฮสต์และโดเมนได้ เช่น

1.) ตัดสินใจว่าการร้องขอรูปแบบใดบ้างที่ควรจะให้ใช้งาน

2.) กำหนด URL ที่จะใช้ในการกรองได้

3.) กำหนดว่าโปรโตคอลใดที่เครื่องไคลเอนต์สามารถเรียกใช้งานได้ โดยขึ้นอยู่กับหมายเลขไอพีของเครื่องไคลเอนต์นั้น ๆ

2.1.1.5 การปรับแต่งบราวเซอร์ให้ใช้งานพร็อกซี่

บราวเซอร์ (browser) ที่ต้องการใช้งานพร็อกซี่นั้น จะต้องบอกให้บราวเซอร์นั้นส่งการร้องขอผ่านไปยังเครื่องพร็อกซี่เซิร์ฟเวอร์ บราวเซอร์ส่วนใหญ่จะยอมให้เราตั้งค่านี้และมันจะ ส่งการร้องขอผ่านไปยังพร็อกซี่เซิร์ฟเวอร์ ในแต่ละบราวเซอร์ก็จะมีวิธีการกำหนดค่าได้หลายรูปแบบ ทั้งแบบโดเมนเนม (domain name) และหมายเลขไอพี อย่างไรก็ตามถ้าไม่มีการตั้ง ค่าให้แก่บราวเซอร์ ตัวบราวเซอร์ก็จะไม่ส่งค่าไปยังพร็อกซี่ การทำงานของบราวเซอร์ก็จะเชื่อมต่อกับเซิร์ฟเวอร์ตามปกติ

เมื่อเครื่องแต่ละเครื่องมีหมายเลขไอพีเป็นของตัวเอง และเชื่อมต่อกับเซิร์ฟเวอร์โดยตรงผ่านอินเตอร์เน็ต เมื่อบราวเซอร์ได้ทำการร้องขอแบบ HTTP ตัวเซิร์ฟเวอร์จะได้รับเส้นทาง (path) และคีย์เวิร์ด (keyword) ที่ใช้ร้องขอ ซึ่งตัวเส้นทางจะอ้างถึงเอกสารหรือโปรแกรมพวก CGI หรือทรัพยากรอื่น ๆ

เมื่อผู้ใช้ป้อน :

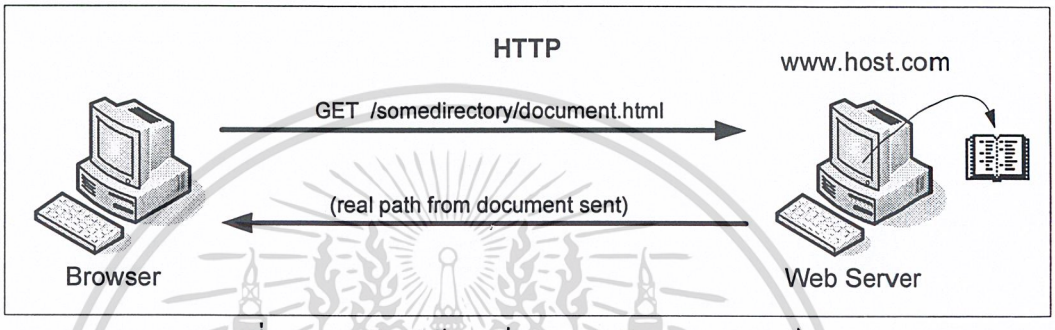
http://www.host.com/path/doc.html

บราวเซอร์จะเปลี่ยนเป็น :

GET /path/doc.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บราวเซอร์จะเชื่อมต่อกับเซิร์ฟเวอร์ จากนั้นจะส่งคำสั่งเข้าไปแล้วรอการตอบรับ ในตัวอย่างนี้บราวเซอร์ส่งการร้องขอไปยัง HTTP เซิร์ฟเวอร์ และบอกว่าต้องการทรัพยากรตัวไหนในเซิร์ฟเวอร์นั้นโดยไม่มีการบอกถึงโปรโตคอลหรือชื่อโฮสต์ใน URL และจากนั้นเครื่องเซิร์ฟเวอร์จะตอบกลับมาเป็นเอกสารที่ต้องการหรืออาจจะส่งข้อความว่าการร้องขอนั้นเป็นการร้องขอที่ผิดพลาด



รูปที่ 2.2 การเรียกใช้งานเว็บตามปกติของบราวเซอร์

2.1.1.6 การสื่อสารโดยผ่านพร็อกซี่เซิร์ฟเวอร์

พร็อกซี่นั้นจะทำตัวเป็นทั้งเซิร์ฟเวอร์และไคลเอนต์ โดยจะทำตัวเป็นเซิร์ฟเวอร์เมื่อรับการร้องขอ HTTP จากบราวเซอร์ และทำตัวเป็นไคลเอนต์เมื่อส่งการร้องขอไปยังเซิร์ฟเวอร์ภายนอกเพื่อดึงข้อมูลเหล่านี้มา พร็อกซี่นั้นจะทำข้อมูลในส่วนเฮดเดอร์ (header) ที่ได้รับมาส่งผ่านไปให้เซิร์ฟเวอร์โดยไม่มีการแก้ไขใด ๆ ซึ่งหมายความว่าบราวเซอร์นั้นได้ส่งข้อมูลไปให้เซิร์ฟเวอร์ได้อย่างครบถ้วนเมื่อผ่านพร็อกซี่

เมื่อบราวเซอร์ติดต่อผ่านพร็อกซี่ บราวเซอร์จะใช้โปรโตคอล HTTP ในการติดต่อกับพร็อกซี่ ถึงแม้ว่าผู้ใช้งานต้องการจะติดต่อไปยังเซิร์ฟเวอร์โดยใช้โปรโตคอลอื่นก็ตาม เช่น เอฟทีพี เป็นต้น

แทนที่บราวเซอร์นั้นจะส่งเฉพาะเส้นทางและคีย์เวิร์ดที่ต้องการไปยังพร็อกซี่เซิร์ฟเวอร์ บราวเซอร์จะส่ง URL แบบเต็มไปยังพร็อกซี่ หลังจากนั้นพร็อกซี่จะแปลงการร้องขอนั้นให้อยู่ในรูปแบบที่ใช้ในการติดต่อกับเครื่องเซิร์ฟเวอร์

ผู้ใช้ป้อน :

http://www.host.com/path/doc.html

บราวเซอร์ส่งไปให้พร็อกซี่ :

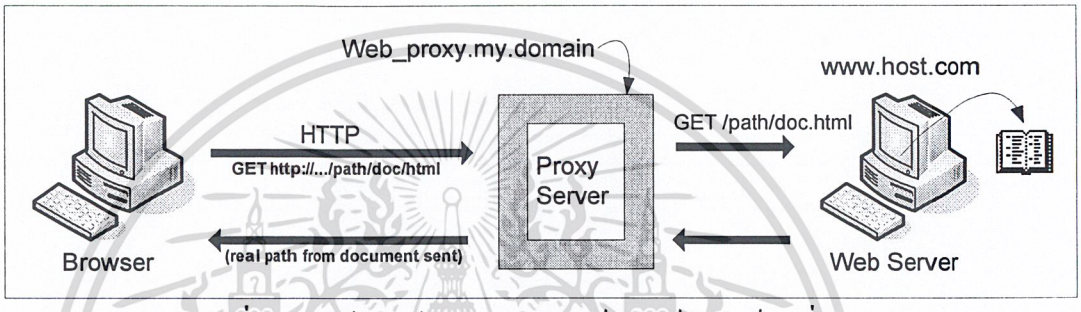
GET http://www.host.com/path/doc.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น การอ้างอิงถึงเอกสารนี้ต้องมีการอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พร็อกซี่ส่งไปยังเซิร์ฟเวอร์ :

GET /path/doc.html

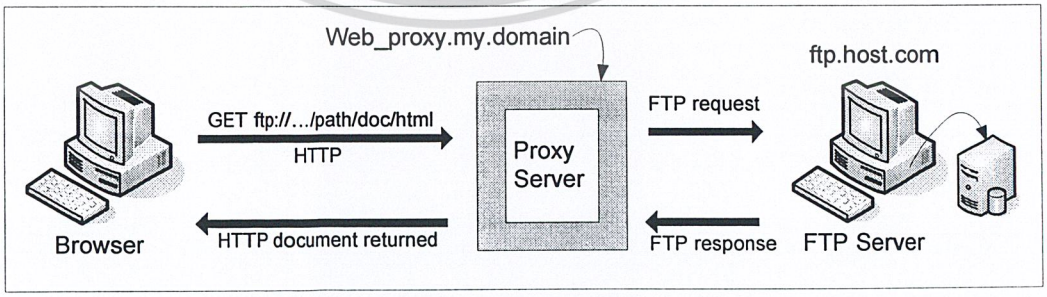
พร็อกซี่จะเชื่อมต่อกับเซิร์ฟเวอร์ จากนั้นจะส่งคำสั่งเข้าไปแล้วรอการตอบรับ ในตัวอย่างนี้ พร็อกซี่จะส่งการร้องขอไปยัง HTTP เซิร์ฟเวอร์ และบอกว่าต้องการทรัพยากรตัวไหนในเซิร์ฟเวอร์นั้น จากนั้นเซิร์ฟเวอร์จะตอบกลับมาเป็นเอกสารที่ต้องการ แล้วเครื่อง พร็อกซี่ก็จะส่งข้อมูลที่ตอบกลับมานี้ไปให้แก่บราวเซอร์อีกทีดังแสดงในรูปที่ 2.3



รูปที่ 2.3 การร้องขอไปยัง HTTP เซิร์ฟเวอร์ผ่านพร็อกซี่

2.1.1.7 บราวเซอร์ร้องขอ FTP โดยผ่านพร็อกซี่

รูปที่ 2.4 แสดงให้เห็นถึงการร้องขอผ่านพร็อกซี่โดยใช้ HTTP ถึงแม้ว่าข้อมูลที่ร้องขอจะอยู่บนเอฟทีพีเซิร์ฟเวอร์ พร็อกซี่สามารถรู้ได้ว่าเป็นการเชื่อมต่อแบบ FTP โดยดูจาก URL เมื่อพร็อกซี่จะสร้างการเชื่อมต่อ และดึงไฟล์มาจาก FTP เซิร์ฟเวอร์ก็จะส่งไฟล์กลับมายังบราวเซอร์โดยใช้ HTTP หรือในอีกกรณีหนึ่งคือ พร็อกซี่จะส่งค่าเป็นรายชื่อของไดเรกทอรี (directory) กลับมาเป็นรูปแบบของเอกสาร html



รูปที่ 2.4 การร้องขอไปยัง FTP เซิร์ฟเวอร์ผ่านพร็อกซี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

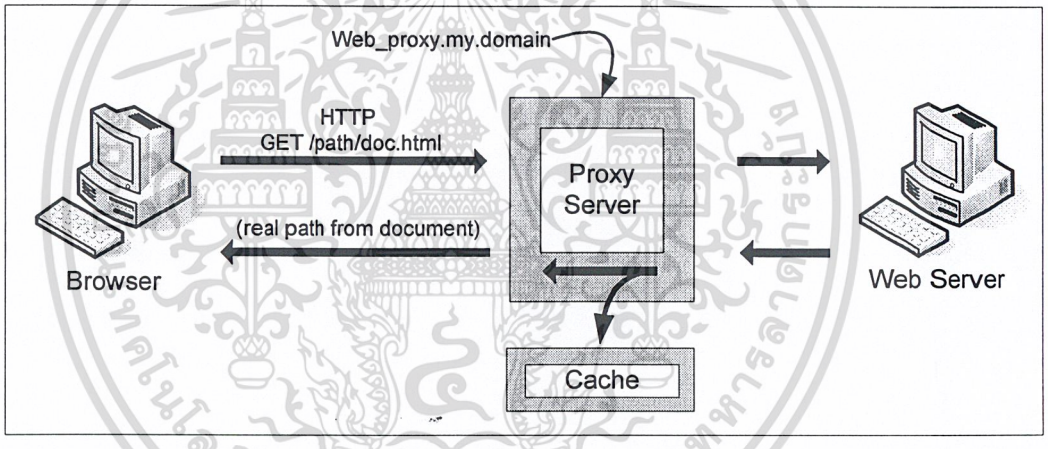
2.1.1.8 ข้อดีในการสำเนาข้อมูล

เมื่อบราวเซอร์ร้องขอไฟล์ พร็อกซี่จะตรวจสอบที่สำเนาของตัวเองว่าได้มีข้อมูลนี้อยู่หรือไม่ ถ้ามีไฟล์อยู่ในสำเนาพร็อกซี่ก็จะส่งไฟล์นี้ไปยังบราวเซอร์ ถ้าเราต้องการที่จะสำเนาข้อมูล เราต้องตัดสินใจว่า

1.) ข้อมูลอะไรบ้างที่ถูกเรียกใช้บ่อยเพียงพอที่จะเก็บเอาไว้ในสำเนา

2.) เราควรจะเก็บข้อมูลไว้นานแค่ไหนก่อนที่จะไปดึงข้อมูลที่ใหม่กว่านี้มาเก็บไว้

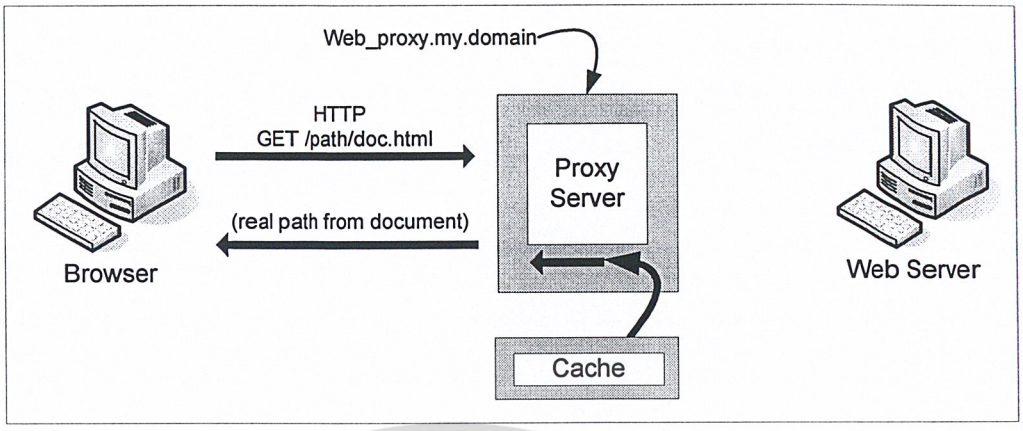
รูปที่ 2.5 จะแสดงให้เห็นวิธีที่พร็อกซี่สำเนาข้อมูลที่ดึงมาจากเซิร์ฟเวอร์แล้วเก็บเอาไว้ ซึ่งเครื่องไคลเอนต์สามารถร้องขอและดึงข้อมูลไปจากสำเนาได้ ถ้ามีการร้องขอข้อมูลอันเดิมในครั้งหลัง



รูปที่ 2.5 การสำเนาข้อมูลของพร็อกซี่

ถ้าพบข้อมูลในสำเนาว่าเป็นข้อมูลที่แก้ไข (update) แล้ว พร็อกซี่ก็ไม่จำเป็นต้องเชื่อมต่อไปยังเซิร์ฟเวอร์เพื่อร้องขอข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 การนำข้อมูลจากสำเนาส่งให้บราวเซอร์

การสำเนาข้อมูลสามารถช่วยลดเวลาในการรอคอยของผู้ใช้งาน เมื่อเขาร้องขอข้อมูลที่อยู่บนอินเตอร์เน็ต พร็อกซี่สามารถที่จะให้บริการข้อมูลเหล่านี้ได้รวดเร็วกว่าเซิร์ฟเวอร์ที่อยู่ห่างออกไป และเมื่อมีการสำเนาข้อมูลที่ผู้ใช้ส่วนใหญ่ต้องการก็จะสามารถช่วยลดการเชื่อมต่อและการใช้งานเครือข่ายภายนอกได้อีกด้วย

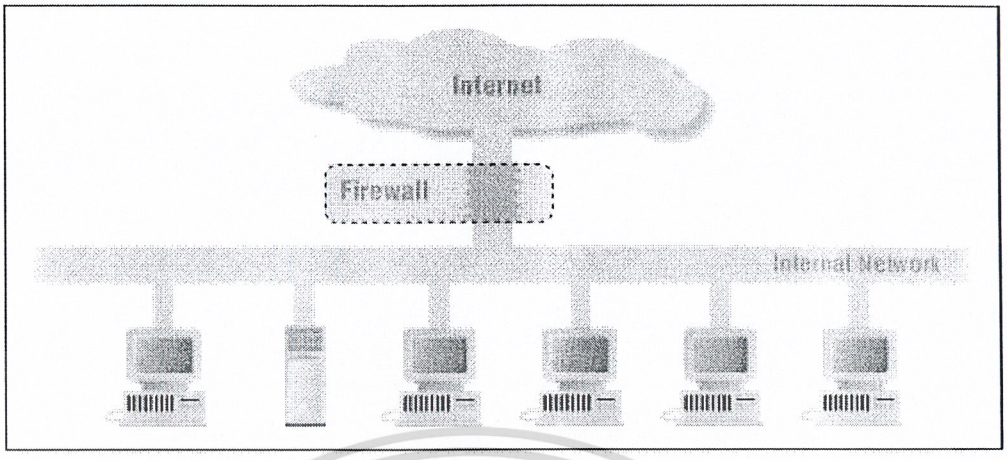
2.1.2 ระบบไฟร์วอลล์

ในความหมายทางด้านการก่อสร้างนั้น "ไฟร์วอลล์" จะหมายถึง กำแพงที่เอาไว้ป้องกันไฟไม่ให้ลุกลามไปยังส่วนอื่น ๆ ส่วนทางด้านคอมพิวเตอร์นั้นก็จะมีความหมายคล้าย ๆ กันก็คือ เป็นระบบที่เอาไว้ป้องกันอันตรายจากอินเตอร์เน็ตหรือเน็ตเวิร์กภายนอกนั่นเอง [2]

ไฟร์วอลล์เป็นคอมโพเนนต์ หรือกลุ่มของคอมโพเนนต์ที่ทำหน้าที่ให้บริการควบคุมการเข้าถึงระหว่างเน็ตเวิร์กภายนอกหรือเน็ตเวิร์กที่เราคิดว่าไม่ปลอดภัย เช่น อินเตอร์เน็ต กับเน็ตเวิร์กภายในหรือเน็ตเวิร์กที่เราต้องการจะป้องกันโดยที่คอมโพเนนต์นั้นอาจจะเป็นเราเตอร์,คอมพิวเตอร์หรือเน็ตเวิร์กประกอบกันก็ได้ ขึ้นอยู่กับวิธีการหรือสถาปัตยกรรม (Firewall Architecture) ที่ใช้

การควบคุมการเข้าถึงของไฟร์วอลล์นั้น สามารถทำได้ในหลายระดับและหลายรูปแบบขึ้นอยู่กับชนิดหรือเทคโนโลยีของไฟร์วอลล์ที่นำมาใช้ เช่น สามารถกำหนดได้ว่าจะให้มีการเข้ามาใช้เซอร์วิสอะไรได้บ้าง จากที่ไหน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 ไฟร์วอลล์กั้นระหว่างอินเตอร์เน็ตกับเน็ตเวิร์กภายใน

2.1.2.1 สิ่งที่ไฟร์วอลล์ช่วยได้

ไฟร์วอลล์สามารถช่วยเพิ่มความปลอดภัยให้กับระบบได้โดย

1.) บังคับใช้นโยบายด้านความปลอดภัย โดยการกำหนดกฎให้กับไฟร์วอลล์ว่าจะอนุญาตหรือไม่ให้ใช้เซอร์วิสชนิดใด

2.) ทำให้การพิจารณาดูแล และการตัดสินใจในด้านความปลอดภัยของระบบนั้นเป็นไปได้ง่ายขึ้น เนื่องจากการติดต่อทุกชนิดกับเน็ตเวิร์กภายนอกจะต้องผ่านไฟร์วอลล์ การดูแลที่จุดนี้เป็นการดูแลความปลอดภัยในระดับของเน็ตเวิร์ก (Network-based Security)

3.) บันทึกข้อมูล,กิจกรรมต่าง ๆ ที่ผ่านเข้าออกเน็ตเวิร์กอย่างมีประสิทธิภาพ

4.) ป้องกันเน็ตเวิร์กบางส่วนจากการเข้าถึงของเน็ตเวิร์กภายนอก เช่น ถ้าหากเรามีเน็ตเวิร์ก (network) บางส่วนที่เราต้องการที่จะให้ภายนอกเข้ามาใช้บริการ (เช่น ถ้ามีเว็บเซิร์ฟเวอร์) แต่ส่วนที่เหลือไม่ต้องการให้ภายนอกเข้ามา กรณีเช่นนี้เราสามารถใช้ไฟร์วอลล์ช่วยได้

5.) ไฟร์วอลล์บางชนิดสามารถป้องกันไวรัสได้ โดยจะทำการตรวจไฟล์ที่โอนย้ายผ่านทางโปรโตคอล HTTP , เอฟทีพี และเอสเอ็มทีพี (SMTP)

2.1.2.2 สิ่งที่ไฟร์วอลล์ช่วยไม่ได้

ถึงแม้ว่าไฟร์วอลล์จะสามารถช่วยเพิ่มความปลอดภัยให้กับเน็ตเวิร์กได้มาก โดยการตรวจดูข้อมูลที่ผ่านเข้าออก แต่อย่าลืมว่าสิ่งเหล่านี้ไม่สามารถป้องกันได้จากการใช้ไฟร์วอลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อันตรายที่เกิดจากเน็ตเวิร์กภายในไม่สามารถป้องกันได้ เนื่องจากอยู่ภายในเน็ตเวิร์กเองไม่ได้ผ่านไฟร์วอลล์เข้ามา

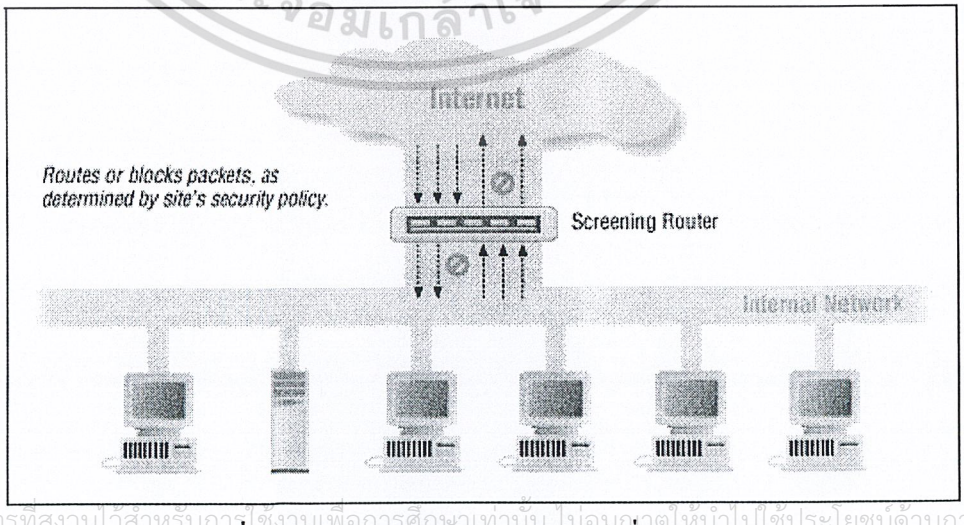
1.) อันตรายจากภายนอกที่ไม่ได้ผ่านเข้ามาทางไฟร์วอลล์ เช่นการไดอัล-อัพ (Dial-up) เข้ามายังเน็ตเวิร์กภายในโดยตรงโดยไม่ได้ผ่านไฟร์วอลล์
2.) อันตรายจากวิธีใหม่ ๆ ที่เกิดขึ้น ทุกวันนี้มีการพบช่องโหว่ใหม่ ๆ เกิดขึ้นทุกวัน เราไม่สามารถไว้ใจไฟร์วอลล์โดยการติดตั้งเพียงครั้งเดียว แล้วก็หวังให้มันมีความปลอดภัยตลอดไป เราต้องมีการดูแลรักษาอย่างต่อเนื่องสม่ำเสมอ
3.) ไวรัส ถึงแม้จะมีไฟร์วอลล์บางชนิดที่สามารถป้องกันไวรัสได้ แต่ก็ยังไม่มี ไฟร์วอลล์ชนิดใดที่จะสามารถตรวจสอบไวรัสได้จากในทุก ๆ โปรโตคอล

2.1.2.3 ชนิดของไฟร์วอลล์

ชนิดของไฟร์วอลล์แบ่งตามเทคโนโลยีที่ใช้ในการตรวจสอบและควบคุม แบ่งได้เป็น

1.) Packet Filtering

Packet Filter คือเราเตอร์ที่ทำการหาเส้นทาง (route) และส่งต่ออย่างมีเงื่อนไข ซึ่งจะพิจารณาจากข้อมูลที่อยู่ในเฮดเดอร์ของแพ็กเกจที่ผ่านเข้ามาเทียบกับกฎ (rules) ที่กำหนดไว้และตัดสินว่าควรจะทิ้ง (drop) แพ็กเกจนั้นไป หรือว่าจะยอม (accept) ให้แพ็กเกจนั้นผ่านไปได้



รูปที่ 2.8 ใช้ Screening Router ทำหน้าที่ Packet Filtering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการพิจารณาเฮดเดอร์การกรองแพ็คเกจ (Packet Filter) จะทำการตรวจสอบในระดับของอินเตอร์เน็ตเลเยอร์ (Internet Layer) และทรานสปอร์ตเลเยอร์ (Transport Layer) ในอินเตอร์เน็ตโมเดล (Internet model) ซึ่งในอินเตอร์เน็ตเลเยอร์จะมีคุณสมบัติ (attribute) ที่สำคัญต่อการกรองแพ็คเกจ ดังนี้

- ไอพีต้นทาง
- ปลายทาง
- ชนิดของโปรโตคอล (TCP, UDP และ ICMP)

และในระดับของทรานสปอร์ตเลเยอร์ มีแอตทริบิวต์ที่สำคัญคือ

- พอร์ตต้นทาง
- พอร์ตปลายทาง
- แฟล็ก (Flag ซึ่งจะมีเฉพาะในเฮดเดอร์ของแพ็กเกจ TCP)
- ชนิดของข้อความไอซีเอ็มพี (ICMP message) ในแพ็กเกจ ICMP

พอร์ตของทรานสปอร์ตเลเยอร์ทั้งใน TCP และ UDP นั้นจะเป็นสิ่งที่บอกถึงแอพพลิเคชันที่แพ็กเกจนั้นต้องการติดต่อด้วย เช่น พอร์ต 80 หมายถึง HTTP, พอร์ต 21 หมายถึง FTP เป็นต้น ดังนั้นเมื่อ การกรองแพ็คเกจ พิจารณาเฮดเดอร์จึงทำให้สามารถควบคุมแพ็กเกจที่มาจากที่ต่าง ๆ และมีลักษณะต่าง ๆ (ดูได้จากแฟลกของแพ็กเกจหรือชนิดของ ICMP ในแพ็กเกจ ICMP) ได้ เช่น ห้ามแพ็กเกจทุกชนิดจาก crack.cracker.net เข้ามายังเน็ตเวิร์ก 203.154.207.0/24 หรือ ห้ามแพ็กเกจที่มีไอพีต้นทางอยู่ในเน็ตเวิร์ก 203.154.207.0/24 ผ่านเราเตอร์เข้ามา (ในกรณีนี้เพื่อเป็นการป้องกัน ip spoofing)เป็นต้น

การกรองแพ็คเกจสามารถอิมพลีเมนต์ได้จาก 2 แพล็ตฟอร์ม คือ

1.) เราเตอร์ที่มีความสามารถในการทำการกรองแพ็คเกจ (ซึ่งมีในเราเตอร์ส่วนใหญ่)

2.) คอมพิวเตอร์ที่ทำหน้าที่เป็นเราเตอร์

ซึ่งจะมีข้อได้เปรียบเสียเปรียบกันดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 เปรียบเทียบข้อดีข้อเสียในการเลือกอุปกรณ์มาทำหน้าที่ส่วนกรองแพ็คเกจ

	ข้อดี	ข้อเสีย
เราเตอร์	ประสิทธิภาพสูงมีจำนวนอินเตอร์เฟสมาก	เพิ่มเติมฟังก์ชั่นการทำงานได้ยาก, อาจต้องการหน่วยความจำมาก
คอมพิวเตอร์ที่ทำหน้าที่เป็นเราเตอร์	เพิ่มฟังก์ชั่นการทำงานได้ไม่จำกัด	ประสิทธิภาพปานกลาง,จำนวนอินเตอร์เฟสน้อย,อาจมีความเสี่ยงจากระบบปฏิบัติการที่ใช้

ข้อดี-ข้อเสียของส่วนกรองแพ็คเกจ

ข้อดี

1.) ไม่ขึ้นกับแอพพลิเคชั่น
2.) มีความเร็วสูง
3.) รองรับการขยายตัวได้ดี

ข้อเสีย

บางโปรโตคอลไม่เหมาะสมกับการใช้กรองแพ็คเกจ เช่น FTP, ICQ

2.1.2.4 Stateful Inspection

โดยปกติแล้วส่วนกรองแพ็คเกจแบบธรรมดา (ที่เป็น Stateless แบบที่มีอยู่ในเราเตอร์ทั่วไป) นั้นจะควบคุมการเข้าออกของแพ็กเกจโดยพิจารณาข้อมูลจากเฮดเดอร์ของแต่ละแพ็กเกจนำมาเทียบกับกฎที่มีอยู่ ซึ่งกฎที่มีอยู่ก็จะเป็นกฎที่สร้างจากข้อมูลส่วนที่ อยู่ในเฮดเดอร์เท่านั้น ดังนั้นการกรองแพ็คเกจแบบธรรมดาจึงไม่สามารถทราบได้ว่าแพ็กเกจนี้อยู่ส่วนใดของการเชื่อมต่อ, เป็นแพ็กเกจที่เข้ามาติดต่อใหม่หรือเปล่า หรือว่าเป็นแพ็กเกจที่เป็นส่วนของการเชื่อมต่อที่เกิดขึ้นแล้ว เป็นต้น

Stateful Inspection เป็นเทคโนโลยีที่เพิ่มเข้าไปในส่วนกรองแพ็คเกจ โดยในการพิจารณาว่าจะยอมให้แพ็กเกจผ่านไปนั้น แทนที่จะดูข้อมูลจากเฮดเดอร์เพียงอย่างเดียว Stateful Inspection จะนำเอาส่วนข้อมูลของแพ็กเกจ (message content) และข้อมูลที่ได้จากแพ็กเกจก่อนหน้านี้ที่ได้ทำการบันทึกเอาไว้จำนำมาพิจารณาด้วย จึงทำให้สามารถระบุได้ว่าแพ็กเกจใดเป็นแพ็กเกจที่ติดต่อเข้ามาใหม่ หรือว่าเป็นส่วนหนึ่งของการเชื่อมต่อที่มีอยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างผลิตภัณฑ์ทางการค้าที่ใช้ Stateful Inspection Technology ได้แก่

1.) Check Point Firewall-1

2.) Cisco Secure Pix Firewall

3.) Sun Screen Secure Net

2.1.3 ระบบเนมเซิร์ฟเวอร์

การติดต่อส่งผ่านข้อมูลในเครือข่ายอินเตอร์เน็ตนั้น สิ่งที่จะช่วยทำให้การส่งผ่านข้อมูลประสบความสำเร็จคือ การอ้างอิงระบุตำแหน่งที่ชัดเจนถูกต้องโดยอาศัยหมายเลยไอพีเป็นจุดอ้างอิง แต่สำหรับ การส่งผ่านข้ามเครือข่ายไปยังอีกหลาย ๆ เครือข่าย เพื่อให้ถึงจุดหมายปลายทางก็ต้องอาศัยอุปกรณ์เราท์เตอร์เพื่อส่งข้อมูลต่อออกไปยังเครือข่ายที่ถูกต้อง ผ่านช่องทางเชื่อมต่อหรือวงจรเชื่อมโยงที่ดีที่สุด ในขบวนการนี้ก็ต้องอาศัยอัลกอริทึมในการหาเส้นทาง (routing algorithm) พิจารณาเส้นทางที่จะส่งข้อมูลไป และแม้ว่าเครือข่ายอินเตอร์เน็ตจะมีการขยายตัวเพิ่มขึ้นตลอดเวลาหรือมีบางครั้งที่บางเครือข่ายล่มไปไม่สามารถให้บริการได้ อุปกรณ์เราเตอร์ก็จะสามารถทราบสถานะต่าง ๆ ของเครือข่ายได้จากการแลกเปลี่ยนข้อมูลกัน, เรียนรู้เส้นทางในการส่งผ่านข้อมูลระหว่างกันโดยใช้โปรโตคอลสำหรับหาเส้นทาง (routing protocol) ซึ่งเป็นโปรโตคอลที่เราเตอร์ใช้ติดต่อระหว่างกัน

2.1.3.1 ระบบโดเมนเนม

ระบบโดเมนเนม หรือ DNS [3] เป็นระบบจัดการแปลงชื่อไปเป็นหมายเลขไอพี (name-to-IP address mapping) โดยมีโครงสร้างฐานข้อมูลแบบลำดับชั้นเพื่อใช้เก็บข้อมูลที่เรียกค้นได้อย่างรวดเร็ว กลไกหลักของระบบ DNS คือ ทำหน้าที่แปลงข้อมูลชื่อเป็นหมายเลขไอพี หรือทำกลับกันได้ นอกจากนี้ยังมีฟังก์ชั่นเพิ่มเติมอื่น ๆ อีก เช่น แจ้งชื่อของอีเมล์เซิร์ฟเวอร์ในโดเมนที่รับผิดชอบด้วย ในระบบ DNS จะมีการกำหนดname space ที่มีกฎเกณฑ์อย่างชัดเจน มีกลไกการเก็บข้อมูลเป็นฐานข้อมูลแบบกระจาย ทำงานในลักษณะของไคลเอนต์เซิร์ฟเวอร์ (Client/Server) โดยมี DNS เซิร์ฟเวอร์ ให้บริการเรียกค้นชื่อและแปลงข้อมูลให้ตามที่เครื่องลูกข่าย (DNS Client) เรียกเข้ามา เช่น เครื่องลูกข่ายที่ต้องการรับส่งอีเมล์ หรือโอนถ่ายไฟล์ข้อมูลให้กับเครื่องอื่นอาจจะทราบเพียงชื่อของเครื่องเซิร์ฟเวอร์ที่ให้บริการเท่านั้นแต่ไม่ทราบหมายเลขไอพี กลไก DNS จะทำหน้าที่แปลงข้อมูลและแจ้งให้เครื่องลูกข่ายทราบตามที่ได้ส่งคำสั่งขอข้อมูลมาและ กลไกการทำงานแบบไคลเอนต์เซิร์ฟเวอร์นี้เองที่ทำให้เครื่องซึ่งทำหน้าที่ DNS สามารถเป็นได้ทั้ง

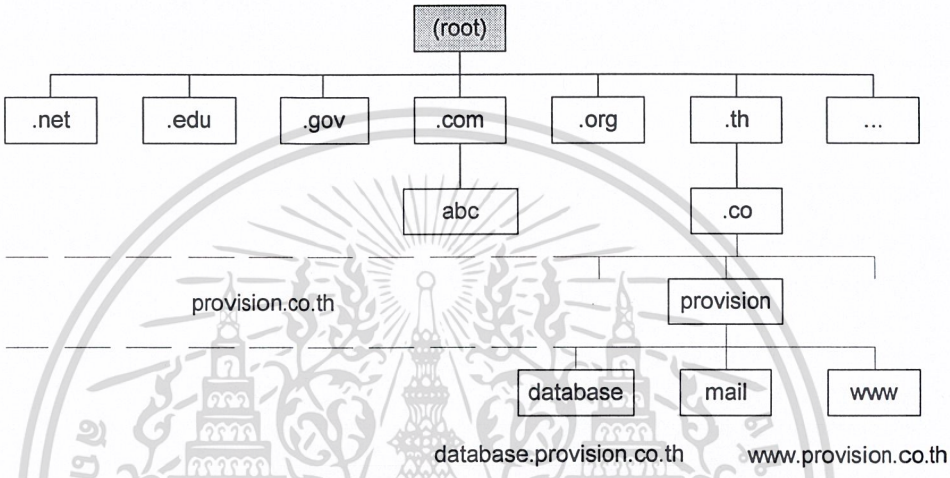
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์และไคลเอนต์ของ DNS ในเครื่องเดียวกัน ดังนั้นเครื่องที่ให้บริการในเครือข่ายอินเตอร์เน็ต 1 เครื่องจะมีการอ้างถึงได้หลายแบบดังนี้

1.) อ้างตามชื่อ domain เช่น mail.provision.co.th

2.) อ้างตาม IP address เช่น 204.183.255.30

3.) อ้างตามหมายเลขฮาร์ดแวร์หรือ MAC address เช่น 00:a0:7c:4d:f2



รูปที่ 2.9 แสดงโครงสร้างของการจัดการ DNS เป็นแบบลำดับชั้น

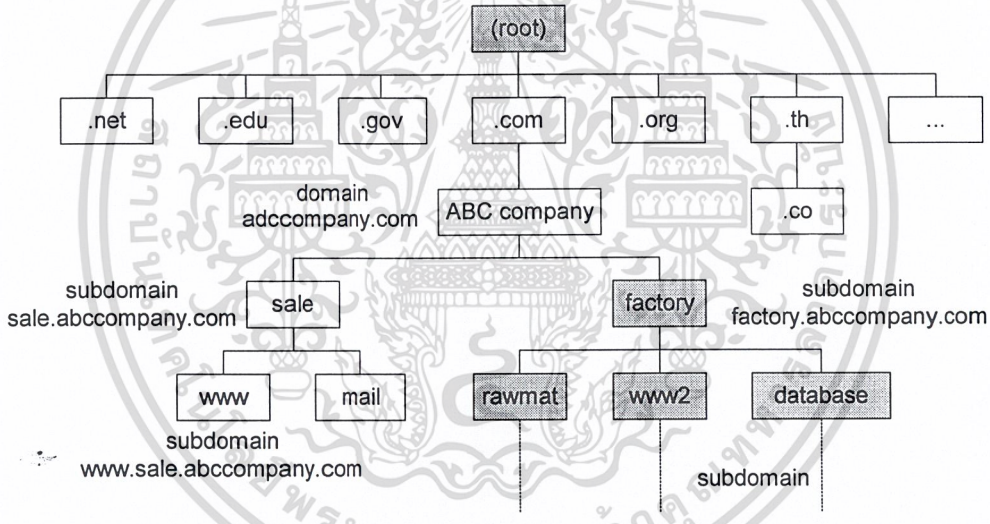
โครงสร้างของการจัดการ DNS เป็นแบบลำดับชั้น (hierarchical) โดยมองได้ในลักษณะของแผนภูมิต้นไม้ (tree) ตามตัวอย่างในรูปข้างต้นจะเห็นว่าลำดับชั้นบนสุดของแผนภูมิเรียกว่ารูท (root) และชั้นถัดลงมาแยกย่อยเป็นลำดับ ในแต่ละลำดับของ DNS เรียกว่าโหนด (node) ซึ่งแต่ละโหนดนั้นจะมีชื่อกำหนดไว้ หรือที่เราเรียกว่าโดเมนเนม ซึ่งอยู่ภายใต้ลำดับชั้นของ .provision และอยู่ภายใต้ .co (หมายถึง หน่วยงานที่เป็นบริษัท) และ .th (หมายถึงประเทศไทย) ตามลำดับว่า database.provision.co.th, mail.provision .co.th และ www.provision.co.th โดยมีข้อสังเกตคือ ในการอ่านชื่อโหนดข้อกำหนดที่สำคัญของ DNS คือ ชื่อลำดับชั้นที่สองที่ต่อจากรูทได้มีการกำหนดชื่อเฉพาะที่ระบุรายละเอียดของกลุ่มเอาไว้ชัดเจนแล้วดังนี้

- .mil แทนกลุ่มของหน่วยงานทางทหารของสหรัฐอเมริกา
- .gov แทนกลุ่มของหน่วยงานของรัฐบาล
- .com แทนกลุ่มขององค์กรหรือบริษัทเอกชน
- .net แทนองค์กรที่ทำหน้าที่เป็นผู้ให้บริการเครือข่าย
- .edu แทนสถาบันการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- .org แทนองค์กร หรือสมาคมต่าง ๆ
- .xx ตัวอักษร 2 ตัวแทนชื่อประเทศ เช่น .th สำหรับประเทศไทย, .ca สำหรับประเทศแคนาดา

จากที่กล่าวไปแล้วชื่อโหนดในโดเมนเนมแบบเต็มนั้นจะหมายถึงเครื่องเซิร์ฟเวอร์ที่ให้บริการในอินเตอร์เน็ตและเป็นชื่ออ้างอิงกับอีเมล์ได้ ในแต่ละโดเมนจะมีการกำหนดเครื่อง DNS เซิร์ฟเวอร์เพื่อคอยดูแลฐานข้อมูลให้ บางกรณีที่เป็นองค์กรขนาดใหญ่อาจจะแบ่งโดเมนย่อยลงเป็นสับโดเมน (subdomain) อีกก็ได้ และในแต่ละโดเมนย่อยก็จะมีเครื่อง DNS เซิร์ฟเวอร์ดูแลฐานข้อมูลเช่นกัน เรียกได้ว่าเครื่อง DNS เซิร์ฟเวอร์นั้นมีสิทธิ (authoritative) หรือได้รับอนุญาตในการดูแลโดเมนหรือสับโดเมนนั้น ๆ ตัวอย่างของโดเมนและสับโดเมนจะเป็นดังรูปที่ 2.10



รูปที่ 2.10 โครงสร้างเน็ตเวิร์กของบริษัท abccompany ที่มีการแบ่งเป็นโดเมนและโดเมนย่อย

จากรูปที่ 2.10 จะพบว่าบริษัท abccompany ได้จดทะเบียนชื่อโดเมนเนมเอาไว้เป็น abccompany.com ซึ่งในสับโดเมนนั้นได้แบ่งออกเป็น sale.abccompany.com และ factory.abccompany.com ในสับโดเมนทั้งสองอาจจะจัดให้มี DNS เซิร์ฟเวอร์สองเครื่องดูแลฐานข้อมูลโดยแยกกลุ่มกัน เช่น กลุ่มของ factory ที่แยกสับโดเมนย่อยลงไปอีกจัดให้มี DNS เซิร์ฟเวอร์เครื่องที่หนึ่งดูแล อีกด้านหนึ่งในกลุ่มของโดเมนย่อย sale จัดให้ DNS เซิร์ฟเวอร์เครื่องที่สองรับผิดชอบในการดูแลฐานข้อมูลของสับโดเมน sale.abccompany.com ซึ่งมี DNSเซิร์ฟเวอร์เครื่องที่สอง คือ ผู้มีสิทธิในการดูแลกลุ่มโดเมนย่อยที่ชื่อ sale.abccompany.com

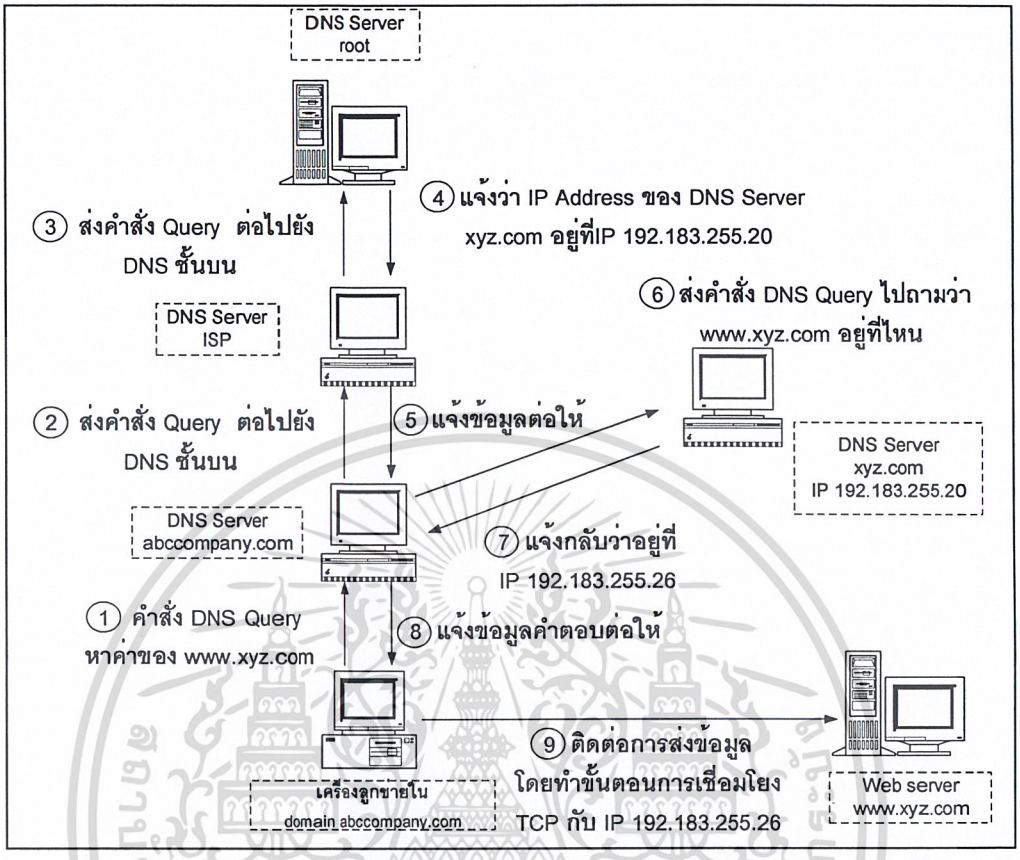
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.2 กลไกการทำงานของ DNS

กลไกการทำงานของ DNS มีขั้นตอนเบื้องต้นแสดงดังรูปที่ 2.11 โดยในที่นี้จะไม่ขอกล่าวถึงรายละเอียดปลีกย่อยอื่น ๆ ซึ่งมีอยู่มากและยังมีโปรโตคอลพิเศษคอยทำหน้าที่ต่าง ๆ อยู่เบื้องหลังด้วย เช่น โปรโตคอล ARP ช่วยแปลงค่าหมายเลขไอพีเป็นค่าฮาร์ดแวร์ เป็นต้น ตามรูปการทำงานของ DNS มีขั้นตอนที่สามารถอธิบายรายละเอียดได้ดังนี้

ขั้นตอนที่ 1 เครื่องคอมพิวเตอร์ไคลเอนต์ที่มีโดเมนเป็น abccompany.com ต้องการติดต่อกับเว็บไซต์ที่ชื่อ www.xyz.com ดังนั้นเครื่องไคลเอนต์นี้ จะส่งคำสั่งขอข้อมูลหมายเลข ไอพีด้วยกลไกรีโซล์ฟเวอร์ (resolver) ไปที่ DNS เซิร์ฟเวอร์ที่ดูแลโซน (zone) ของตนอยู่ คือ โดเมน abccompany.com ในกรณีนี้สมมุติว่าฐานข้อมูลที่มีใน DNS เซิร์ฟเวอร์ไม่มีข้อมูลหมายเลขไอพีของ www.xyz.com ทั้งนี้เพราะ DNS เซิร์ฟเวอร์ของโซน abccompany.com จะดูแลฐานข้อมูลเฉพาะเครื่องลูกข่ายตนเอง ดังนั้น DNS เซิร์ฟเวอร์นี้ก็จะทำการส่งคำสั่ง ขอข้อมูลต่อไปยัง DNS เซิร์ฟเวอร์ที่อยู่ในระดับบนกว่าซึ่งได้กำหนดเอาไว้ให้เป็นเครื่อง DNS เซิร์ฟเวอร์ของบริษัทผู้ให้บริการอินเตอร์เน็ต (ISP) นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 การทำงานของ DNS

ขั้นตอนที่ 2 เมื่อ DNS เซิร์ฟเวอร์ของ abc.company.com ส่งคำสั่งขอข้อมูลต่อไปยัง DNS เซิร์ฟเวอร์ ของบริษัทผู้ให้บริการอินเตอร์เน็ตแล้ว เครื่อง DNS เซิร์ฟเวอร์ของบริษัทผู้ให้บริการอินเตอร์เน็ตก็จะค้นหาข้อมูลจากฐานข้อมูลของตนเช่นเดียวกัน ในกรณีนี้สมมุติว่ายังไม่มีข้อมูลหมายเลขไอพีของ www.xyz.com อีกเหมือนกัน เครื่อง DNS เซิร์ฟเวอร์ของบริษัทผู้ให้บรอการอินเตอร์เน็ตจะส่วคำสั่งขอข้อมูลต่อออกไปยังเครื่อง DNS เซิร์ฟเวอร์ระดับบนขึ้นไปอีกซึ่งก็ได้มีการกำหนดไว้ว่าเป็นเครื่องรูทเซิร์ฟเวอร์ (root server)

ขั้นตอนที่ 3 เมื่อ DNS server ของ abc.company.com ส่งคำสั่งขอข้อมูลต่อไปยัง DNS server ของบริษัทผู้ให้บริการหรือ ISP แล้ว เครื่อง DNS เซิร์ฟเวอร์ ของ ISP ก็จะค้นหาข้อมูลจากฐานข้อมูลของตนเช่นเดียวกัน ในกรณีนี้สมมุติว่ายังไม่มีข้อมูลหมายเลขไอพีของ www.xyz.com อีกเหมือนกัน เครื่อง DNS เซิร์ฟเวอร์ของ ISP จะส่งคำสั่งขอข้อมูลต่อออกไปยังเครื่อง DNSเซิร์ฟเวอร์ในระดับบนขึ้นไปอีก ซึ่งก็ได้มีการกำหนดไว้ว่าเป็นรูทเซิร์ฟเวอร์(root server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 4 คำสั่งขอข้อมูลถูกส่งต่อไปยัง DNS server ของ root เพราะดูแลฐานข้อมูลของ domain name ในระดับสอง (.com)

ขั้นตอนที่ 5 ที่ DNS root server แม้ว่าจะไม่มีข้อมูลหมายเลข IP address ของ www.xyz.com ก็ตาม แต่มีข้อมูลที่ทราบว่า DNS server ที่ดูแล zone ของ domain xyz.com อยู่ที่ใด (มีหมายเลข IP address อะไร) DNS root server ก็จะส่งข้อมูลดังกล่าวไปให้ เพราะที่เครื่องSERVER ที่ดูแล domain xyz.com จะต้องมีข้อมูลของ IP address ของ www.xyz.com อยู่แน่นอน

ขั้นตอนที่ 6 DNS server ของ ISP จะรับข้อมูล IP address ของเครื่อง DNS server ที่ดูแล zone ของ domain xyz.com เป็น 192.183.255.20 และแจ้งต่อไปให้ DNS server ที่รับผิดชอบ domain xyz.com อีกทีหนึ่ง ในขั้นนี้เครื่อง DNS server ของบริษัท ISP จะเก็บค่าคำตอบเอาไว้ในหน่วยความจำสำเนาเพื่อใช้กรณีที่มีการเรียกข้อมูลซ้ำอีกในอนาคต จะได้ส่งคำตอบไปให้เลยโดยไม่ต้องไปขอข้อมูลซ้ำอีก ค่าที่เก็บเอาไว้จะมีระยะเวลาที่ต้องปรับปรุงข้อมูลใหม่ตามค่าในฟิลด์ TTL ที่กำหนดไว้ใน resource record

ขั้นตอนที่ 7 DNS server ของบริษัท abccompany.com จะรับข้อมูลหมายเลข IP address ของเครื่อง DNS server ที่ดูแล zone ของ domain xyz.com ตามที่เครื่อง DNS server ของ ISP ส่งมาให้ และเก็บลงหน่วยความจำสำเนาของตนเองเช่นกัน เผื่อมีการเรียกใช้อีกในอนาคต แล้วส่งคำสั่งไปถามข้อมูลว่าเครื่อง www.xyz.com อยู่ที่ไหน (มีหมายเลข IP address อะไร)

ขั้นตอนที่ 8 DNS server ของ domain xyz.com ตรวจสอบข้อมูลและแจ้งว่าเครื่อง www.xyz.com อยู่ที่ IP address 192.186.255.26 ข้อมูลถูกส่งกลับไปให้เครื่อง DNS server ของ abccompany.com

ขั้นตอนที่ 9 คำตอบที่ DNS server ของ abccompany.com ได้รับจะถูกส่งต่อให้กับเครื่องไคลเอนต์ที่ต้องการและก็จัดเก็บข้อมูลลงหน่วยความจำแคชเช่นกัน

ขั้นตอนที่ 10 เมื่อเครื่องลูกข่ายทราบว่า www.xyz.com มีหมายเลข IP address อยู่ที่ 192.183.255.26 ก็จะติดต่อกับเครื่อง www.xyz.com โดยถ้าใช้งานเว็บก็จะสร้างการเชื่อมต่อโดยโปรโตคอล HTTP และใช้งาน port 80 เพื่อเรียกดูข้อมูลในเว็บไซต์นั้นต่อไปตามกลไกของ TCP/IP

อาจมีผู้สงสัยว่าในการทำงานของโดเมนเนมเซิร์ฟเวอร์ในเครือข่ายอินเตอร์เน็ตนั้นมีวิธีการจัดการอย่างไร ซึ่งสามารถอธิบายได้ง่าย ๆ ตามขั้นตอนดังกล่าวที่ผ่านมา ซึ่งจะเห็นว่า DNS server จะถูกจัดลำดับในการดูแลฐานข้อมูลแยกกันตามกลุ่ม โดยแบ่งลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชั้นให้สอดคล้องกับการกำหนดชื่อ domain และในแต่ละลำดับของ DNS server นี้จะทราบว่าถ้าต้องการติดต่อขอข้อมูลจากลำดับบนขึ้นไปจะติดต่อได้จากหมายเลข IP address อะไร โดยในชั้นบนสุดเป็น root ที่จะดูแลข้อมูลของ domain ลำดับที่สองและย่อยลงไปตามชั้น และแต่ละเซิร์ฟเวอร์ที่ดูแล domain ของตนก็จะเรียกว่าเซิร์ฟเวอร์นั้นมีสิทธิ์ในการรับผิดชอบ zone ของตนเอง

2.1.4 โปรโตคอล HTTP (HTTP : HyperText Transfer Protocol)

โปรโตคอลนี้สร้างขึ้นสำหรับบริการที่เรียกว่าเวิล์ดไวด์เว็บ (World Wide Web) ในเครือข่ายอินเตอร์เน็ตโดยเฉพาะ โดยโปรโตคอลนี้จะเป็นตัวกำหนดวิธีในการส่งข้อมูลหรือไฟล์ระหว่างเครื่องลูกข่าย (Client) และเครื่องแม่ข่าย (Server) รวมถึงการกำหนดกฎระเบียบในการติดต่อด้วย โปรโตคอลนี้ถูกออกแบบมาให้มีความกะทัดรัด, สามารถทำงานได้รวดเร็ว, มีการระบวนการทำงานที่ไม่ซับซ้อน และมีคำสั่งที่ใช้งานไม่มากนัก แต่สามารถรองรับข้อมูลได้ทุกแบบ ไม่ว่าจะเป็นข้อมูลทั่วไปที่เข้ารหัสแบบ MIME หรือข้อมูลที่เป็นกราฟิก เช่น ไฟล์ที่เป็น GIF หรือ JPEG เป็นต้น [4]

2.1.4.1 วิธีการติดต่อของโปรโตคอล HTTP

โปรโตคอล HTTP นั้น อยู่บนพื้นฐานของระบบเครือข่ายแบบไคลเอนต์/เซิร์ฟเวอร์ (Client/Server) ที่ต้องมีการร้องขอรับบริการจากไคลเอนต์ (request) และการตอบสนองหรือการให้บริการของเซิร์ฟเวอร์ (response) จึงสามารถแบ่งการทำงานออกเป็น 2 ด้านคือ ด้านเว็บเซิร์ฟเวอร์ และด้านไคลเอนต์ โดยไคลเอนต์จะติดต่อเข้ามายังเซิร์ฟเวอร์และอ้างถึงแอดเดรสของเซิร์ฟเวอร์โดยใช้รูปแบบของ URL ส่วนด้านเซิร์ฟเวอร์จะส่งข้อมูลกลับมาในรูปแบบที่เป็นภาษา HTML (HyperText Markup Language) โดยที่โปรโตคอล HTTP ใช้วิธีการเข้ารหัสในแบบ MIME เป็นมาตรฐานของการทำงาน โปรโตคอลนี้ถูกออกแบบมาให้สามารถรับส่งข้อมูลผ่านพร็อกซี่หรือไฟร์วอลล์ต่าง ๆ ได้ โดยอาศัยการเชื่อมต่อผ่านทางโปรโตคอลทีซีพี/ไอพี (TCP/IP) อีกทีหนึ่ง โดยใช้พอร์ตหมายเลข 80 เป็นช่องทางมาตรฐานในการติดต่อ ในทางปฏิบัติจะใช้พอร์ตหมายเลขอื่นก็ได้ ในปัจจุบันเว็บบราวเซอร์ทั่วไปจะกำหนดค่ามาตรฐานไว้ที่พอร์ต 80 ดังนั้นหากมีการกำหนดไว้ที่พอร์ตอื่น จะทำให้เกิดความลำบากต่อผู้ใช้ที่ต้องระบุหมายเลขพอร์ตลงใน URL ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยเหตุที่การทำงานของโปรโตคอล HTTP เป็นแบบไคลเอนต์และเซิร์ฟเวอร์ ดังนั้นการติดต่อสื่อสารใด ๆ ผ่านโปรโตคอลนี้จำเป็นต้องมีเครื่องตัวแม่กับตัวลูก การสื่อสารนั้นจึงจะสมบูรณ์ได้ การติดต่อระหว่างไคลเอนต์ไปยังเซิร์ฟเวอร์ผ่านโปรโตคอล HTTP มีขั้นตอนดังนี้

ขั้นแรก : Open Socket

ไคลเอนต์สร้างการเชื่อมต่อ (Connection) กับเซิร์ฟเวอร์ผ่านซ็อกเก็ต (Socket)

ขั้นที่สอง : Request

ไคลเอนต์ส่งคำร้องขอข้อมูลไปยังเซิร์ฟเวอร์

ขั้นที่สาม : Information Transfer

เซิร์ฟเวอร์จะไปหาข้อมูลที่ไคลเอนต์ต้องการ

ขั้นที่สี่ : Response

เซิร์ฟเวอร์ส่งข้อมูลตอบสนอง (Response) กลับมายังไคลเอนต์เสมอ

ขั้นสุดท้าย : Close Socket

ปลดการเชื่อมต่อของซ็อกเก็ตของทั้งสองฝั่งออก

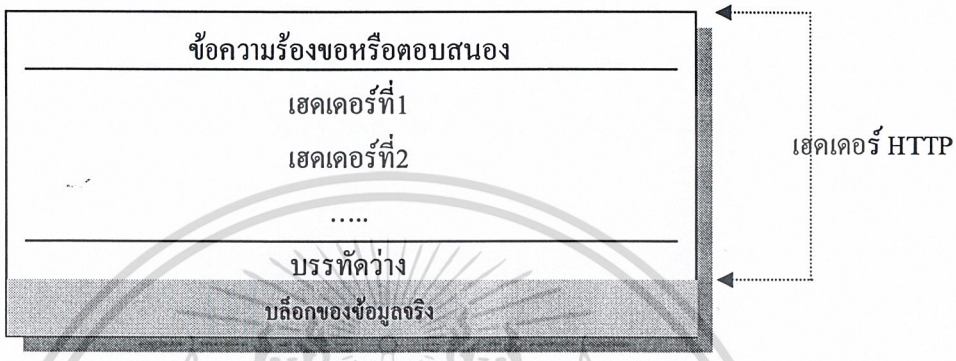
ด้วยการทำงานของโปรโตคอล HTTP ที่มีการเชื่อมต่อในระยะเวลาสั้น หรือที่เรียกว่าเป็นโปรโตคอลแบบ Connectionless ในลักษณะดังกล่าว ทำให้ในช่วงเวลาหนึ่ง ๆ เซิร์ฟเวอร์ที่ให้บริการเวิล์ดไวด์เว็บสามารถรองรับไคลเอนต์ได้จำนวนมากพร้อม ๆ กัน เพราะไม่มีใครได้ทำการเชื่อมต่อย่างถาวร

ในการร้องขอรับบริการจากเครื่องไคลเอนต์ และการตอบสนองหรือการให้บริการจากเซิร์ฟเวอร์นั้นย่อมต้องมีการรับส่งข้อมูลระหว่างกัน แต่ข้อมูลที่รับส่งกันในแต่ละครั้งนั้นไม่ได้มีเฉพาะข้อมูลเพียงอย่างเดียว แต่ละฝ่ายจะมีส่วนเฮดเดอร์ของ HTTP (HTTP header) เข้าไปในส่วนต้นของข้อมูลที่รับส่งกันด้วย ซึ่งเฮดเดอร์ของ HTTP จะเป็นตัวบอกว่าข้อมูลที่ส่งมานี้เป็นข้อมูลอะไร เป็นข้อมูลการร้องขอจากไคลเอนต์ หรือเป็นข้อมูลตอบสนองจากเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4.2 โครงสร้างของโปรโตคอล HTTP

โครงสร้างของข้อมูล HTTP จะแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ ส่วนเฮดเดอร์ หรือเรียกว่าเมตาดาต้า (metadata) จะเป็นส่วนเก็บข้อมูลที่จำเป็นต้องใช้ภายในโปรโตคอล ส่วนที่สองเป็นส่วนของข้อมูลจริงที่ต้องการรับส่งดังแสดงในรูปที่ 2.12



รูปที่ 2.12 โครงสร้างของข้อมูลที่ส่งผ่านโปรโตคอล HTTP

1.) เฮดเดอร์ HTTP (HTTP Header)

แสดงดังรูปที่ 2.17 เฮดเดอร์ HTTP ประกอบด้วย 2 ส่วน คือส่วนข้อมูลร้องขอหรือตอบสนอง และส่วนเฮดเดอร์ย่อย

1.1) ส่วนข้อมูลร้องขอหรือตอบสนอง เป็นส่วนในการแยกว่าเป็นข้อความตอบสนองจากเซิร์ฟเวอร์ หรือข้อความร้องขอจากไคลเอนต์ และรายละเอียดดังต่อไปนี้

1.1.1) ข้อความร้องขอ (request)

จากรูปที่ 2.17 ในส่วนข้อความการร้องขอ (บรรทัดแรก) จะประกอบด้วย 3 ส่วน คือ

- วิธีการร้องขอ หรือที่เรียกว่า “เมธอด”
- ไดเรกทอรีและชื่อไฟล์ที่ต้องการจากเซิร์ฟเวอร์
- เวอร์ชันของ HTTP ที่ไคลเอนต์ใช้อยู่

โดยที่แต่ละส่วนจะถูกแบ่งด้วยช่องว่าง (space) ซึ่งในบรรทัดแรกของเฮดเดอร์ HTTP มีรูปแบบได้ดังนี้

<Method> /path/file HTTP/x.x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การขอข้อมูลจากเซิร์ฟเวอร์จะระบุเฉพาะไดเรกทอรีและชื่อไฟล์ที่ต้องการ ไม่ต้องบอกชื่อโฮสต์และโดเมนเนม เพราะได้มีการสร้างการเชื่อมต่อกับโฮสต์ในโปรเซสก่อนหน้านี้ไปแล้ว การร้องขอข้อมูลจึงไม่ต้องระบุชื่อโฮสต์ซ้ำอีกครั้ง

นอกจากคำร้องขอที่อยู่ในบรรทัดแรกแล้ว ยังมีข้อมูลอื่น ๆ ที่จะส่งไปให้กับเซิร์ฟเวอร์ด้วย คือข้อมูลในส่วนที่สอง ซึ่งเรียกว่า "เฮดเดอร์" (header) ข้อมูลในเฮดเดอร์แต่ละเว็บบราวเซอร์แต่ละเวอร์ชันอาจจะไม่เหมือนกัน ซึ่งข้อมูลของเฮดเดอร์นี้จะบอกรายละเอียดของผู้ส่งว่ามีอะไรบ้าง เฮดเดอร์นี้จะบอกให้เซิร์ฟเวอร์ทราบได้ว่าข้อความร้องขอถูกส่งมาจากใคร และสามารถนำไปใช้เพื่อประโยชน์ในเรื่องอื่น ๆ ต่อไปได้

จากโครงสร้างข้อมูลที่รับส่งกันระหว่างไคลเอนต์กับเซิร์ฟเวอร์ในรูปที่ 2.12 หลังจากเฮดเดอร์รายการสุดท้ายแล้วจะมีบรรทัดว่าง หลังจากนั้นจะเป็นส่วนของบล็อกข้อมูล ทั้งนี้หากเป็นการร้องขอจากไคลเอนต์ก็จะขึ้นอยู่ว่าใช้เมธอดใดในการร้องขอ หากเป็นเมธอดเก็ต (GET) ก็ไม่จำเป็นต้องมีข้อมูลอะไรในส่วนนี้ เนื่องจากเซิร์ฟเวอร์จะไม่สนใจข้อมูลในส่วนนี้ ทั้งนี้เนื่องจากรูปแบบของโปรโตคอลนั่นเอง จะกล่าวโดยละเอียดต่อไป

เมื่อไคลเอนต์เชื่อมต่อกับเซิร์ฟเวอร์เรียบร้อยแล้ว ไคลเอนต์จะเป็นฝ่ายส่งข้อมูลการร้องขอไปยังเซิร์ฟเวอร์ ซึ่งการร้องขอไปยังเซิร์ฟเวอร์นี้สามารถทำได้หลายวิธี ทั้งนี้ทั้งนั้นก็ขึ้นอยู่กับเวอร์ชันของโปรโตคอล HTTP ใช้ หากเป็นเวอร์ชัน 1.0 จะมีวิธีการร้องขอมาตรฐาน 3 วิธี คือ GET, HEAD และ POST แต่หากเป็นโปรโตคอล HTTP เวอร์ชัน 1.1 จะมีวิธีการร้องขอเพิ่มจากเวอร์ชัน 1.0 อีกหลายวิธี เช่น OPTIONS, PUT, DELETE หรือ TRACE เป็นต้น ดังนั้นการที่เราจะเลือกว่าจะใช้โปรโตคอล HTTP เวอร์ชันไหน ต้องขึ้นอยู่กับเซิร์ฟเวอร์และไคลเอนต์ที่จะทำงานด้วย คือ หากว่าเซิร์ฟเวอร์สนับสนุนการทำงานของ HTTP 1.1 แล้ว วิธีการร้องขอก็สามารถใช้ของเวอร์ชัน 1.1 ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

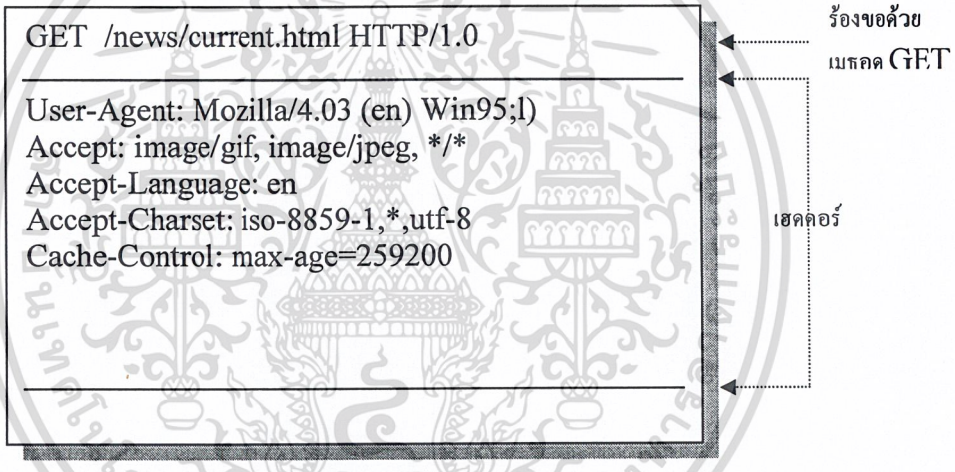
รูปแบบการร้องขอโดยโปรโตคอล HTTP เวอร์ชัน 1.0

เพื่อไม่ให้เกิดปัญหาในการทำงาน จึงควรใช้โปรโตคอล HTTP เวอร์ชัน 1.0 ซึ่งเว็บเซิร์ฟเวอร์ส่วนใหญ่ในอินเตอร์เน็ตจะสามารถรองรับการร้องขอจากเวอร์ชันนี้ โดยโปรโตคอล HTTP เวอร์ชัน 1.0 มีวิธีการร้องขออยู่ 3 วิธีด้วยกัน โดยมีรายละเอียดดังนี้

- การร้องขอด้วยเมธอด GET มีรูปแบบดังนี้

  GET /path/file HTTP/x.x

เป็นการร้องขอให้เซิร์ฟเวอร์ส่งไฟล์มาให้ หรือเป็นการร้องขอโดยมีการส่งข้อมูลจากทางไคลเอนต์ไปให้ด้วยก็ได้ ดูตัวอย่างประกอบในรูปที่ 2.13



รูปที่ 2.13 ตัวอย่างข้อความร้องขอด้วยเมธอด GET

นอกจากข้อความการร้องขอด้วยเมธอด GET ใช้สำหรับการร้องขอข้อมูลจากเซิร์ฟเวอร์แล้ว ข้อความร้องขอด้วยเมธอด GET นี้ยังสามารถใช้สำหรับส่งข้อมูลไปยังเครื่องเซิร์ฟเวอร์ได้อีกด้วย โดยข้อมูลที่ต้องการส่งให้กับเซิร์ฟเวอร์นั้นจะต่อท้าย URL โดยมีเครื่องหมาย ? กั้นระหว่าง URL กับข้อมูลนั้น ลักษณะข้อมูลจะประกอบด้วยตัวแปรและค่าของตัวแปรนั้น โดยเขียนต่อในลักษณะ <Variable Name>=<Variable Value> &<Variable Name>=<Variable Value>&..... แต่มีข้อจำกัดด้านขนาดของข้อมูลที่ส่งไปให้เซิร์ฟเวอร์ด้วยเมธอดนี้ เนื่องจากให้ส่งได้ครั้งละไม่เกิน 256 ตัวอักษร (นับจากที่เข้ารหัสแล้ว) ความยาวทั้งหมดเริ่มนับจากชื่อไดเรกทอรีเป็นต้นไป แต่ความยาวนี้ขึ้นอยู่กับระบบปฏิบัติการอีกทีหนึ่ง

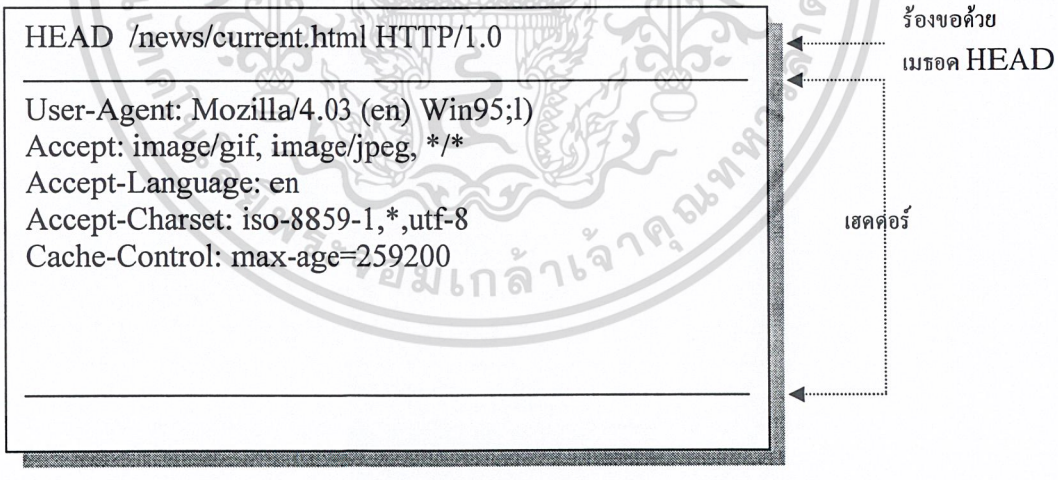
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การร้องขอด้วยเมธอด HEAD มีรูปแบบดังนี้

  HEAD /path/file HTTP/x.x

เป็นการร้องขอเพื่อถามเซิร์ฟเวอร์ว่ามีไฟล์ที่ต้องการอยู่ในเซิร์ฟเวอร์หรือไม่ (ถามเฉย ๆ ไม่ต้องการให้เซิร์ฟเวอร์ส่งไฟล์จริงมาให้) ซึ่งมีประโยชน์สำหรับการตรวจสอบว่ามีไฟล์ที่ต้องการอยู่ในเซิร์ฟเวอร์หรือไม่ หรือใช้ตรวจสอบความสมบูรณ์ของลิงค์ก็ได้ รหัสตอบสนองในบรรทัดสถานะจึงอาจเป็น 200 (มีไฟล์ที่ต้องการ) หรือ 404 (ไม่มีไฟล์ที่ต้องการ) และมีข้อมูลอื่นส่งเพิ่มเติมมาในเฮดเดอร์ด้วย เช่น วันที่ปรับปรุงแก้ไขไฟล์ครั้งสุดท้าย (Last Modified) เป็นต้น

การจะตรวจสอบว่ามีไฟล์นั้นอยู่ที่เซิร์ฟเวอร์หรือไม่ อาจใช้วิธีการร้องขอด้วยเมธอด GET แล้วใช้การเช็คผลการตอบสนอง แต่การตอบสนองจากการร้องขอด้วยเมธอด HEAD จะไม่มีการส่งเนื้อหาของไฟล์มาให้ (ไม่มีบล็อกข้อมูล) ดังนั้นใช้เมธอด HEAD จึงทำงานได้คำตอบเร็วกว่า มักจะใช้เมธอดนี้ในการตรวจสอบกับทางเซิร์ฟเวอร์ หากมีการปรับปรุงไฟล์นั้น จึงจะมีการดาวน์โหลดมาทับไฟล์เดิมในเครื่อง (ร้องขอไฟล์นั้นอีกครั้งด้วยเมธอด GET) ดูตัวอย่างประกอบในรูปที่ 2.14



รูปที่ 2.14 ตัวอย่างข้อความร้องขอด้วยเมธอด HEAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การร้องขอด้วยเมธอด POST มีรูปแบบดังนี้

  POST /path/file HTTP/x.x

เป็นการร้องขอให้เซิร์ฟเวอร์รับข้อมูลจากไคลเอนต์เพื่อนำไปประมวลผล หรือนำไปเก็บในฐานข้อมูลต่อไป โดยมีเงื่อนไขดังนี้

- ข้อมูลที่จะส่งไปให้เซิร์ฟเวอร์จะอยู่ภายในบล็อกข้อมูล ดังนั้นจึงต้องมีเฮดเดอร์เพื่อบอกรายละเอียดของข้อมูลในบล็อกข้อมูลแนบไปด้วย
- /path/file คือ ชื่อโปรแกรม CGI ในเซิร์ฟเวอร์ที่จะทำหน้าที่รับข้อมูลไปประมวลผล
- ข้อความตอบสนองที่เซิร์ฟเวอร์จะส่งกลับให้ไคลเอนต์ จะได้มาจากการทำงานของโปรแกรม CGI ในเซิร์ฟเวอร์ ดังนั้น CGI ที่รับข้อมูลไปจึงต้องทำหน้าที่ส่งข้อความตอบกลับให้ไคลเอนต์

โดยส่วนใหญ่การส่งข้อมูลจากฟอร์มในเว็บเพจไปประมวลผลด้วย CGI ในเซิร์ฟเวอร์ จะใช้เมธอดนี้มากที่สุด ความจริงแล้วการส่งข้อมูลไปยังเซิร์ฟเวอร์สามารถใช้เมธอด GET ก็ได้ แต่มีข้อจำกัดเรื่องความยาวของข้อมูลที่จะส่งไปให้เซิร์ฟเวอร์ ถ้าใช้เมธอด POST เพื่อร้องขอการส่งข้อมูลไปยังเซิร์ฟเวอร์ เฮดเดอร์ที่ชื่อ Content-Type จะถูกกำหนดให้เป็น application/x-www-form-urlencoded เพื่อบอกแก่เซิร์ฟเวอร์ว่าข้อมูลที่ส่งไปให้มีการเข้ารหัส และเฮดเดอร์ Content-Length จะใช้สำหรับบอกความยาวของข้อมูลที่เข้ารหัสแล้ว เพราะข้อมูลที่กรอกผ่านอินพุตในฟอร์มจะถูกเว็บบราวเซอร์เข้ารหัสก่อนส่งเสมอ การเข้ารหัสนี้เรียกว่า URL-encoded (ดูการเข้ารหัสในรูปที่ 2.15) ซึ่งมีรายละเอียดดังนี้

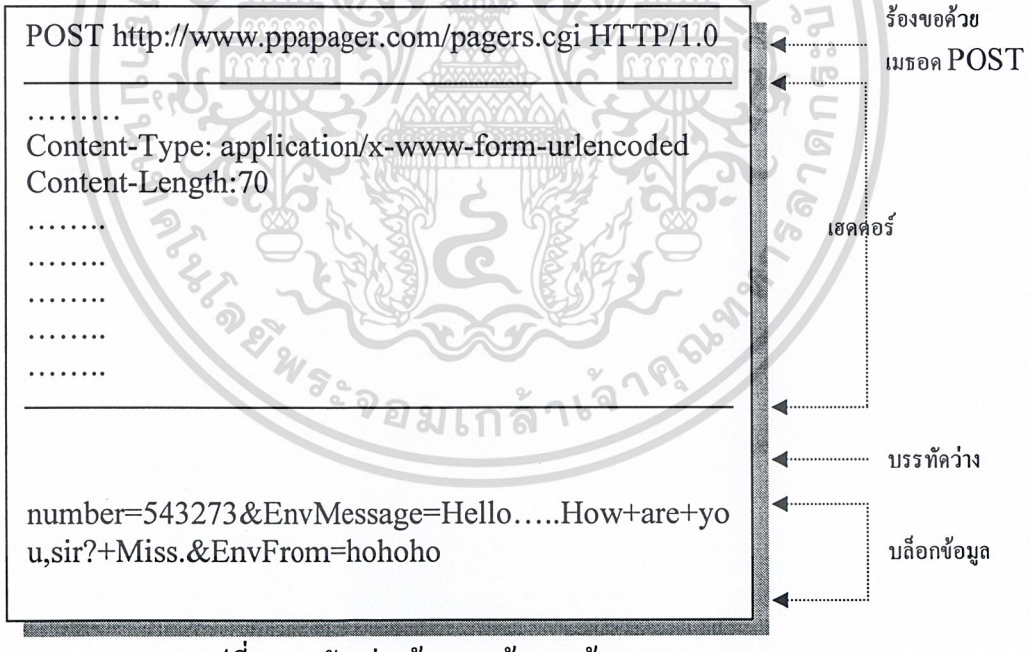
- แปลงตัวอักษรหรือเครื่องหมายบางตัวให้อยู่ในรูป %xx โดยที่ xx จะเป็นค่ารหัสแอสกีของตัวอักษรนั้น ตัวอักษรที่ต้องมีการแปลง เช่น =, &, % และ + เพราะเป็นเครื่องหมายที่ใช้เป็นตัวแบ่งแยกข้อมูลที่จะส่งไปให้เซิร์ฟเวอร์ดังรูปที่ 2.20
- เปลี่ยนช่องว่าง (Space) ทุกตัวเป็นเครื่องหมายบวก (+)
- รวมชื่อตัวแปรและค่าตัวแปรเข้าด้วยกัน โดยคั่นกลางด้วยเครื่องหมาย = และคั่นระหว่างตัวแปรด้วยเครื่องหมาย &

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักขระ	รหัส (%xx)
%	%25
&	%26
‘	%27
+	%2B
=	%3D
?	%3F

รูปที่ 2.15 URL-encoded

ตัวอย่างข้อความร้องขอด้วยเมธอด POST ที่เว็บบราวเซอร์สร้างขึ้น เพื่อส่งข้อมูลจากฟอร์มในเว็บเพจไปให้แก่ CGI ที่ชื่อ pagerkara.cgi ในไดเรกทอรี / ของเซิร์ฟเวอร์ http://www.ppapager.com รับไปทำงานดังรูปที่ 2.16



รูปที่ 2.16 ตัวอย่างข้อความร้องขอด้วยเมธอด POST

1.1.2) ข้อความตอบสนอง (response) และสถานะการทำงาน

โปรโตคอล HTTP นั้นได้กำหนดรหัสแสดงสถานะการทำงานของโปรโตคอลไว้ โดยแบ่งกลุ่มของรหัสสถานะออกไว้เป็น 5 กลุ่มดังแสดงในตารางที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่2.2 แสดงกลุ่มของรหัสสถานะการทำงานของโปรโตคอล HTTP

รหัสสถานะ	ประเภท	รายละเอียด
100-199	Informational	เป็นรหัสสถานะที่เปิดให้โปรแกรมประยุกต์ต่าง ๆ กำหนดใช้งานได้เอง
200-299	Successful	กลุ่มรหัสที่แสดงว่าการทำงานสำเร็จ
300-399	Redirection	กลุ่มรหัสนี้จะใช้ภายในโปรโตคอล HTTP เอง โดยเป็นการทำงานที่ต่อเนื่องมาจากโปรเซสก่อนหน้านี้ ซึ่งไคลเอนต์เป็นผู้สั่งงาน
400-499	Client Error	ใช้แสดงปัญหาที่เกิดขึ้นทางฝั่งไคลเอนต์
500-599	Server Error	ใช้แสดงปัญหาที่เกิดขึ้นทางฝั่งเซิร์ฟเวอร์

รหัสแสดงสถานะในแต่ละตัวจะนำหน้าด้วยตัวเลข 3 หลัก และตามด้วยตัวอักษร ซึ่งรหัสในกลุ่ม 100-199 จะเปิดกว้างให้ผู้ที่พัฒนาโปรแกรมประยุกต์สามารถกำหนดค่าขึ้นมาใช้งานได้เอง ส่วนรายละเอียดของรหัสในกลุ่มอื่น ๆ จะแสดงในตารางที่ 2.3

ตารางที่ 2.3 แสดงรหัสสถานะของ HTTP

รหัสสถานะ	ความหมาย
100 Continue (1.1)	ใช้ในกรณีที่บราวเซอร์อยู่ในระหว่างส่ง Request แต่ยังส่งไม่หมด แต่เซิร์ฟเวอร์ต้องการให้ทราบว่าได้รับ Request แล้วให้ส่งส่วนที่เหลือต่อไป
101 Switching Protocol (1.1)	ใช้ร่วมกับเฮดเดอร์ Upgrade กรณีที่ต้องการเปลี่ยนไปใช้โปรโตคอลอื่นที่ความสามารถสูงกว่า เช่น HTTP/2.0 ซึ่งอาจจะมีในอนาคต
200 OK	การทำงานสำเร็จเรียบร้อย
201 Created	คำสั่ง POST ทำงานเสร็จสมบูรณ์
202 Accepted	คำสั่ง POST ทำงานเสร็จสมบูรณ์
203 Non-Authoritative (1.1)	การร้องขอประสบผลสำเร็จ
204 No Content	ทำงานตามคำสั่งเรียบร้อย แต่ไม่ต้องการแสดงข้อความใด ๆ บนหน้าจอ
205 Reset Content (1.1)	เซิร์ฟเวอร์ได้รับข้อมูลเรียบร้อยแล้ว และบอกให้บราวเซอร์ลบข้อความที่กรอกในแบบฟอร์มเดิมออก เพื่อสะดวกในการกรอกข้อมูลถัดไป
206 Partial Content (1.1)	เซิร์ฟเวอร์ได้รับข้อมูลบางส่วนเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสสถานะ	ความหมาย
300 Multiple Choice	ถ้าค้นหาและพบแหล่งข้อมูลที่ต้องการหลายแห่ง เซิร์ฟเวอร์จะตอบกลับไปทั้งหมดเพื่อให้ไคลเอนต์สามารถเลือกแหล่งข้อมูลที่ต้องการเองได้
301 Moved Permanently	URL ที่ร้องขอได้ถูกย้ายไปที่อื่นแล้ว ดังนั้นการร้องขอใช้งาน กับ URL จะต้องเปลี่ยนเป็นแอดเดรสใหม่
302 Moved Temporarily	URL ที่ร้องขอมาได้ถูกย้ายไปที่อื่นชั่วคราว
303 See Other (1.1)	ใช้กรณีที่ต้องการบอกให้ทราบว่าสิ่งที่ต้องการอยู่ใน URI อื่น ซึ่งบราวเซอร์สามารถใช้ GET เพื่อเรียกดูเอกสารนั้น ๆ ได้
304 Not Modify	ใช้แสดงสถานะเมื่อใช้คำสั่ง GET ที่กำหนดเงื่อนไขเฉพาะเว็บไซต์ที่มีการเปลี่ยนแปลง ส่วนเว็บไซต์ที่ไม่มีการเปลี่ยนแปลงจะแสดงด้วยสถานะนี้
305 Use Proxy (1.1)	บอกให้บราวเซอร์ทราบว่าเอกสารที่ต้องการมีอยู่ใน Proxy ซึ่ง URL ของ Proxy จะกำหนดใน Location
400 Bad Request	คำสั่งจากไคลเอนต์ไม่ถูกต้อง
401 Unauthorized	ปฏิเสธการทำงานจากไคลเอนต์ที่ไม่ได้รับอนุญาต
403 Forbidden	เซิร์ฟเวอร์ไม่อนุญาตให้ใช้งาน หรือไคลเอนต์มีสิทธิ์ในการใช้งานเพียงพอ
404 Not Found	ไม่พบเว็บเซิร์ฟเวอร์ตาม URL ที่กำหนด
405 Method Not Allowed (1.1)	Method ที่ใช้ ไม่ได้รับอนุญาต กรณีนี้เซิร์ฟเวอร์จะระบุ Allow เพื่อบอกให้ทราบว่าอนุญาตให้ใช้ Method ใดบ้าง
406 Not Acceptable (1.1)	ข้อมูลที่ต้องการเป็นข้อมูลที่บราวเซอร์ไม่สามารถเข้าใจได้ เนื่องจากไม่อยู่ในรายการ Accept ที่ระบุใน Request Header
407 Proxy Authenticate - (Unauthorized) Request (1.1)	เหมือนกับ 401 แต่ต้องได้รับอนุญาตจาก Proxy จะระบุเฮดเดอร์ให้ทราบ ซึ่งบราวเซอร์สามารถส่ง Request ใหม่โดยระบุเฮดเดอร์ Proxy-Authorization ด้วย
408 Request Timeout (1.1)	บราวเซอร์ไม่ส่ง Request ตามเวลาที่เซิร์ฟเวอร์รอได้
409 Conflict (1.1)	บราวเซอร์ส่งข้อมูลที่มีความหมายขัดแย้งกันเอง
410 Gone (1.1)	เอกสารที่ต้องการไม่ได้อยู่บนเซิร์ฟเวอร์แล้ว
411 Length Required (1.1)	เซิร์ฟเวอร์ต้องการให้ระบุ Content-Length ด้วย
412 Precondition Failed (1.1)	เงื่อนไขบางอย่างที่กำหนดใน Request Header ตกไป
413 Request Entity Too Large (1.1)	ข้อมูลที่ส่งมามีขนาดใหญ่เกินกว่าที่เซิร์ฟเวอร์จะรองรับได้
	ค่า URL ที่ระบุ ยาวเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4.2.1 ส่วนเฮดเดอร์ย่อย

เฮดเดอร์ย่อยเป็นส่วนที่ใช้บอกรายละเอียดต่าง ๆ ของข้อมูล ทั้งการร้องขอและการตอบสนอง โดยมีลักษณะเป็นข้อความธรรมดาซึ่งมีรูปแบบการเขียนดังนี้

Header-name: Value

รายละเอียดปลีกย่อยของเฮดเดอร์ ได้แก่

- เฮดเดอร์อาจมีหลายประการ แต่ท้ายเฮดเดอร์แต่ละรายการต้องปิดด้วยรหัสลงบรรทัดใหม่
- Header-name หรือชื่อของเฮดเดอร์จะพิมพ์ตัวเล็กหรือใหญ่ก็ได้ ไม่มีผลต่อการตีความหมาย
- หลังเครื่องหมาย : ของเฮดเดอร์แต่ละรายการอาจเป็นช่องว่าง (Space) หรือแท็บ (Tab) ก็ได้
- เฮดเดอร์รายการใดที่ขึ้นต้นด้วยช่องว่างหรือแท็บ จะเสมือนว่าเป็นส่วนหนึ่งของเฮดเดอร์รายการก่อนหน้านี้ 1 บรรทัด

ใน HTTP เวอร์ชัน 1.0 กำหนดให้มีเฮดเดอร์ได้ถึง 16 รายการ แต่อาจจะไม่มีแม้แต่รายการเดียวเลยก็ได้ ส่วน HTTP เวอร์ชัน 1.1 กำหนดได้ 46 รายการ แต่ต้องมีเฮดเดอร์อย่างน้อย 1 รายการ คือ Host: เพื่อบอกชื่อโฮสต์และโดเมนเนม ในที่นี้จะขอกล่าวเพียงเฮดเดอร์ของ HTTP เวอร์ชัน 1.0 ซึ่งมีด้วยกันอยู่ 13 รายการ ดังแสดงในตารางที่ 2.4

ตารางที่ 2.4 รายละเอียดของเฮดเดอร์ย่อยของโปรโตคอล HTTP

เฮดเดอร์ย่อย	ความหมาย
1. Allow	กำหนดเมธอดที่สนับสนุนจุดประสงค์ของข้อมูลนี้เพื่อเจาะจงเมธอดการร้องขอจากไคลเอนต์ โดยจะไม่สนับสนุนเมธอด POST
2. Authorization	สำหรับผู้ที่ต้องการการรับรอง (authentication) ด้วยตัวเองจากเซิร์ฟเวอร์ (ซึ่งอาจไม่จำเป็น) หลังจากได้รับการตอบสนองด้วยรหัสสถานะ 401 (Unauthorized) ควรจะมีการเพิ่มเฮดเดอร์นี้เข้าไปด้วย
3. Content-Encoding	ใช้ระบุขนาดของข้อมูลในบล็อกข้อมูลมีหน่วยเป็นไบต์ (byte) เพื่อที่ผู้รับจะได้ทราบว่ามีข้อมูลส่งมาให้กี่ไบต์
4. Content-Length	ใช้ระบุขนาดของข้อมูลในบล็อกข้อมูลมีหน่วยเป็นไบต์ (byte) เพื่อที่ผู้รับจะได้ทราบว่ามีข้อมูลส่งมาให้กี่ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฮดเดอร์ย่อย	ความหมาย
5. Content-Type	ใช้ระบุชนิดของข้อมูลในบล็อกข้อมูลว่าเป็นข้อมูลประเภทไหน เช่น หากเป็นเอกสารแบบ HTML จะต้องระบุเป็น text/html ถ้าเป็นไฟล์รูปกราฟิกแบบ gif ก็ต้องระบุเป็น image/gif เป็นต้น แต่สำหรับเว็บบราวเซอร์รุ่นใหม่ ๆ แล้ว หากข้อมูลที่ได้รับไม่มีการระบุว่าเป็นประเภทไหนแล้ว เว็บบราวเซอร์จะถือว่าเป็นประเภท text/html เสมอ
6. From	เป็นเฮดเดอร์ในส่วนการร้องขอของไคลเอนต์ จะประกอบด้วย e-mail address ของผู้ใช้ที่ควบคุมการส่งข้อมูลการร้องขอ
7. Referer	เป็นเฮดเดอร์ในส่วนของการร้องขอของไคลเอนต์ เป็น URI ของแหล่งที่มาของการร้องขอ ข้อมูลส่วนนี้จะทำให้เซิร์ฟเวอร์สร้างรายการการย้อนหลัง (lists of back-links) การลือกอินเข้าระบบ การจัดการ Cache และอื่น ๆ มันยังส่งผลให้ข้อมูลในส่วน Referer นี้จะต้อง ไม่ถูกส่งมาถ้าไม่ได้มาจากแหล่งที่มี URI ของตัวเอง ตัวอย่างเช่น จาก คีย์บอร์ดของผู้ใช้เอง
8. Date	ข้อมูลส่วนนี้ใช้ระบุวันเวลาที่ข้อมูลการร้องขอถูกส่งมา
9. Expires	เป็นเฮดเดอร์ในส่วนการตอบสนองของเซิร์ฟเวอร์ใช้กำหนดวันที่หมดอายุของไฟล์ที่สงไปให้ไคลเอนต์ รายการนี้สามารถใช้ในทางเทคนิคเพื่อป้อนการเก็บไฟล์ไว้ในสำเนา (Cache) จากเว็บบราวเซอร์อย่าง Navigator ได้ โดยระบุวันที่ใน Expires: ให้ย้อนจากวันเวลาปัจจุบันนาน ๆ เมื่อเว็บบราวเซอร์รับไฟล์ไปก็จะเข้าใจว่าไฟล์นี้หมดอายุแล้ว ถึงแม้จะนำเนื้อหาไปแสดงในวินโดว์เว็บบราวเซอร์แต่จะไม่เก็บไว้ในสำเนา ทำให้ทางเซิร์ฟเวอร์มั่นใจได้ว่า ทุกครั้งที่ไคลเอนต์ร้องขอไฟล์จะต้องวิ่งมาขอจากเซิร์ฟเวอร์ใหม่ทุกครั้ง ถึงแม้จะเป็นการร้องขอไฟล์ เดิม ๆ และทางเซิร์ฟเวอร์ไม่มีการอัพเดท
10. If-Modified-Since	เป็นเฮดเดอร์ในส่วนการร้องขอของไคลเอนต์ ถูกใช้ร่วมกับเมธอด GET เพื่อใช้ในการกำหนดเงื่อนไขบอกแก่เซิร์ฟเวอร์ว่า ถ้าไฟล์ที่ร้องขอไปมีการแก้ไขหลังจากวันที่ได้ระบุในเฮดเดอร์นี้ เซิร์ฟเวอร์จึงค่อยส่งไฟล์นั้นมาให้ แต่ถ้ายังไม่ได้มีการแก้ไขหลังช่วงวันที่ระบุ เซิร์ฟเวอร์ไม่ต้องส่งไฟล์นั้นมาให้ เพียงแต่ส่งรหัสสถานะตอบสนองมาเป็น 304 (not Modified) แทน
11. Last-Modified	ข้อมูลนี้ใช้ระบุวันเวลาครั้งล่าสุดที่มีการ modify ข้อมูลนั้น ซึ่งจะต้องบอกวันที่และเวลาในรูปแบบของเวลามาตรฐาน GMT
12. Location	เป็นเฮดเดอร์ส่วนการตอบสนองของเซิร์ฟเวอร์ที่ใช้แจ้งแหล่งที่อยู่ของข้อมูลที่ไคลเอนต์ต้องการ สำหรับรหัสสถานะการตอบสนองที่ 3xx Location จะชี้ URL ที่ใช้สำหรับรีไดเรก (redirect)
13. Pragma	ข้อมูลในส่วน Pragma นี่ถูกใช้รวมกับ implementation-specific derictives ที่สามารถนำมาใช้กับผู้รับตลอดสายการร้องขอ/ตอบสนอง โดยข้อมูล prama

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4.2.2 ข้อมูลที่ต้องการรับส่ง

จากรูปที่ 2.17 ในส่วนสุดท้าย ซึ่งต่อจากส่วนของเฮดเดอร์ย่อยของเฮดเดอร์ HTTP จะเป็นส่วนของบล็อกข้อมูล ซึ่งเป็นส่วนของข้อมูลที่เราต้องการส่งจริง อาจจะเป็น HTML file หรือ Text file หรือข้อมูลชนิดอื่น ๆ

2.2 กลไกในการค้นหาคำ

2.2.1 ฟังก์ชั่นมาตรฐานเกี่ยวกับสตริง strstr ()

ฟังก์ชั่น strstr () ใช้สำหรับค้นหาตำแหน่งของสตริงที่พบครั้งแรกในสตริงอื่น ฟังก์ชั่นนี้กำหนดในไฟล์ string.h ฟังก์ชั่นนี้สามารถใช้ได้บนระบบปฏิบัติการ DOS, UNIX และ ANSI C [5]

2.2.1.1 การประกาศตัวแปร

```
char *strstr (const char *s1, const char *s2);
```

2.2.1.2 รูปแบบการเรียกใช้ฟังก์ชั่น

```
strstr (str1, str2)
```

2.2.1.3 การคืนค่า

ถ้าพบ str2 ใน str1 ฟังก์ชั่นจะคืนค่าตำแหน่งที่พบ str2 ครั้งแรกใน str1

ถ้าไม่พบ str2 ใน str1 ฟังก์ชั่นจะคืนค่าเป็น null

2.2.1.4 ตัวอย่าง

```
#include <stdio.h>
#include <string.h>
int main(void)
{
        char *str1 = "Borland International", *str2 = "nation", *ptr;
        ptr = strstr(str1, str2);
        printf("The substring is: %s\n", ptr);
        return 0;
}
```

ผลลัพธ์

```
The substring is:national
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 กฎการคำนวณแบบบอยเออร์มัวร์ (Boyer Moore Algorithm)

เป็นวิธีการสำหรับทำการเปรียบเทียบสตริง (string matching) ที่มีประสิทธิภาพมาก โดยจะมองเป็นสายข้อมูล (String) และคำที่เราต้องการค้นหา (Pattern) [6]

วิธีการคือตรวจสอบ (scan) ตัวอักษรในแพทเทอร์น (pattern) จากซ้ายไปขวาเพื่อเทียบกับสตริง (string) โดยดูว่าตัวอักษรสุดท้ายของแพทเทอร์นตรงกับอักษรใดในสตริงและต้องชิฟท์ (shift) ตัวแพทเทอร์นออกไปจำนวนกี่ครั้ง เมื่อนำค่าตัวอักษรในสตริงนั้นไปเทียบโดยฟังก์ชั่นที่ได้มีการคำนวณไว้ก่อนแล้ว เพื่อให้เลื่อนตัวอักษรในแพทเทอร์นกับสตริงนั้นให้ตรงกัน จากนั้นก็ให้ตรวจสอบตั้งแต่ตำแหน่งสุดท้ายของแพทเทอร์นว่ามีค่าเท่ากันตลอดทั้งแพทเทอร์นหรือไม่ ถ้าใช่แสดงว่ามีแพทเทอร์นนั้นอยู่ใน สตริงแต่ถ้าไม่ตรงก็ให้ชิฟท์ออกไปอีก เท่ากับความยาวของแพทเทอร์นนั้น แล้วกลับไปค้นหาตั้งแต่ต้นใหม่

ตัวอย่างการทำงานของ Boyer-Moore Algorithm

Pattern	E	X	A	M	P	L	E
ค่าที่ใช้ในการ Shift	6	5	4	3	2	1	0

String = HERE IS A SIMPLE EXAMPLE

```
E X A M P L E
H E R E   I S   A   S I M P L E   E X A M P L E
```

ขั้นตอนแรกจะพิจารณาตำแหน่งสุดท้ายของ pattern ก่อนว่าตรงกับตำแหน่งใดใน string ซึ่งในที่นี้คือ S เมื่อนำ S ไปตรวจสอบการ shift ในตารางที่ได้คำนวณไว้ก่อนหน้านี้แล้วก็พบว่าต้องเลื่อนไปเท่ากับความยาวของ pattern นั้นซึ่งในที่นี้คือ 7

```
        E X A M P L E
H E R E   I S   A   S I M P L E   E X A M P L E
```

เมื่อพิจารณาตำแหน่งท้ายสุดของ pattern ซึ่งก็คือ E ตรงกับอักษร P ใน string เราจึงนำค่าของ P ไปตรวจสอบในตารางว่าจะต้องเลื่อนไปจำนวนกี่ตำแหน่ง โดยค่าที่ได้คือ 2 จึงต้องเลื่อนไปอีกสองตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    E X A M P L E
H E R E   I S   A   S I M P L E   E X A M P L E
```

จากนั้นจึงเริ่ม scan ตั้งแต่ตัวท้ายของ pattern คือ E ไล่ไปทางด้านซ้ายว่าเท่ากันตลอดทั้งความยาวของ pattern หรือไม่

```
                  E X A M P L E
H E R E   I S   A   S I M P L E   E X A M P L E
```

จะเห็นได้ว่าจะตรงกันเพียง สี่ตัวเท่านั้นคือ MPLE แต่ว่าตัว I ใน string นั้นไม่ตรงกันกับ pattern จึงต้องเลื่อนไปอีกเท่ากับความยาวของ pattern

```
                        E X A M P L E
H E R E   I S   A   S I M P L E   E X A M P L E
```

ตำแหน่งของ string ที่ตรงกันกับตำแหน่งสุดท้ายของ pattern คือ L ซึ่งนำไปเทียบในตารางเพื่อตรวจดูว่าต้องเลื่อนจำนวนกี่ครั้ง ซึ่งในที่นี้ได้ค่าออกมาเท่ากับ 1

```
                      E X A M P L E
H E R E   I S   A   S I M P L E   E X A M P L E
```

จากนั้นก็เริ่มเปรียบเทียบตั้งตำแหน่งสุดท้ายของ pattern

```
                                  E X A M P L E
H E R E   I S   A   S I M P L E   E X A M P L E
```

เมื่อเปรียบเทียบจนถึงตำแหน่งแรกของ pattern ก็พบว่าเท่ากันจึงให้ค่าว่าพบ pattern นี้ใน string

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การเขียนโปรแกรมติดต่อ Socket

ในส่วนของการเขียนโปรแกรมสำหรับการรับส่งข้อมูลในเครือข่ายอินเตอร์เน็ตนั้น สิ่งที่สำคัญในการดำเนินงานอีกอย่างคือ การเขียนโปรแกรมเพื่อติดต่อซ็อกเก็ต (socket) [7] ซึ่งมีหลักการพิจารณาดังนี้

2.3.1 องค์ประกอบของ Socket

องค์ประกอบของซ็อกเก็ตประกอบด้วย 3 ส่วนด้วยกัน คือ รูปแบบการกำหนดแอดเดรสของคอมพิวเตอร์บนเน็ตเวิร์ค (Domain), ชนิดของ (socketType) และโปรโตคอลที่ใช้งาน

2.3.1.1 Socket Domain : เป็นตระกูลของโปรโตคอลที่จะใช้งาน ที่ใช้งานบ่อยก็คือ AF_INET หมายถึงโปรโตคอลที่ใช้งานกับ open network ใช้สำหรับโลคัลเน็ตเวิร์คของยูนิกส์ (TCP/IP)

2.3.1.2 Socket Types : แต่ละ Domains อาจมีหลาย ๆ Socket Type ซึ่งหมายถึงวิธีการส่งข้อมูลของแต่ละ Domains ไม่เพียงแต่ AF_UNIX ซึ่งเราสามารถติดตั้งได้ 2 ทางเท่านั้น โดยโดเมนของเน็ตเวิร์ค AF_INET จะมีวิธีการส่งข้อมูล 2 วิธีคือ streams (TCP) และ datagrames (UDP)

2.3.1.3 Socket Protocols : หลังจากที่ได้กำหนดตระกูลของเน็ตเวิร์คโปรโตคอลแล้ว ก็ต้องเลือกโปรโตคอลที่จะใช้งานจริง ซึ่งจะมีความจำเพาะต่อตระกูลของโปรโตคอลและวิธีการส่งข้อมูลที่ใช้งานตามตารางที่ 2.5

ตารางที่ 2.5 แสดงวิธีการส่งข้อมูลของ Socket

ตระกูลของโปรโตคอล	วิธีการส่งข้อมูล	ชื่อที่ใช้งานใน socket	โปรโตคอลที่ใช้งานจริง
AF_INET	SOCK_DGRAM	IPPROTO_UDP	UDP
AF_INET	SOCK_STREAM	IPPROTO_TCP	TCP
AF_INET	SOCK_RAW	IPPROTO_ICMP	ICMP
AF_INET	SOCK_RAW	IPPROTO_RAW	(raw)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การใช้งาน Socket

2.3.2.1 การสร้าง Socket

โดยใช้ฟังก์ชั่นระดับต่ำ socket, ฟังก์ชั่นจะส่งค่ากลับเป็นหมายเลข เพื่อใช้อ้างอิงถึง socket นั้น

```
# include   <sys/types.h>
# include   <sys/socket.h>

int  socket (int domain, int type, int protocol);
```

จะได้ออบเจ็คที่ใช้ติดต่อระหว่างโปรเซส ที่อยู่บนคอมพิวเตอร์เครื่องเดียวกัน หรือบนเน็ตเวิร์ค

domain หมายถึง ตระกูลของโปรโตคอลที่ใช้งาน ซึ่งอาจจะเป็นค่าต่อไปนี้

- AF_UNIX สำหรับติดต่อกันเองของโปรเซสที่อยู่บนคอมพิวเตอร์เดียวกัน (file system socket)
- AF_INET สำหรับระบบเน็ตเวิร์คของยูนิกซ์ (Unix network socket)
- AF_ISO สำหรับโปรโตคอลของ ISO
- AF_NS โปรโตคอล Xerox Network System

type จะกำหนดกรรมวิธีที่เกี่ยวข้องต่าง ๆ ในการรับส่งข้อมูลสำหรับ socket นั้น ๆ ดังต่อไปนี้

- SOCK_STREAM
- SOCK_DGRAM

protocol จะกำหนดโปรโตคอลที่จะใช้งานจริง ซึ่งจำเพาะต่อ 2 ออบเจ็คข้างต้น แต่โดยทั่วไปจะใช้ค่า 0 หมายถึงดีฟอล์ต (default) โปรโตคอล

ฟังก์ชั่นจะส่งค่ากลับเป็นหมายเลขแทน socket นั้น ซึ่งสามารถจะใช้ฟังก์ชั่นระบบ read / write เพื่ออ่าน/เขียนข้อมูลไปยัง socket และหลังจากใช้งานเสร็จจะต้องใช้ฟังก์ชั่นระบบ close เพื่อปิดการใช้งานด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.2 การกำหนดแอดเดรสให้กับ Socket

แต่ละ Domains จะมีรูปแบบการกำหนดแอดเดรสของ socket ต่างกันดังนี้

2.3.2.2.1 AF_UNIX จะกำหนดไว้ในโครงสร้างของข้อมูล sockaddr_un ซึ่งกำหนดไว้ใน sys/un.h ดังต่อไปนี้

```
struct   sockaddr_un {
     sa_family_t        sun_family;     /*      AF_UNIX     */
     char               sun_path[];     /*      pathname    */
};
```

sa_family จะกำหนดตระกูลของโปรโตคอลที่ใช้งาน

sun_path จะกำหนดแอดเดรสของ socket (ชื่อไฟล์)

2.3.2.2.2 AF_INET จะกำหนดแอดเดรสโดยโครงสร้างข้อมูล sockaddr_in ซึ่งกำหนดไว้ในไฟล์ netinet/in.h ดังต่อไปนี้

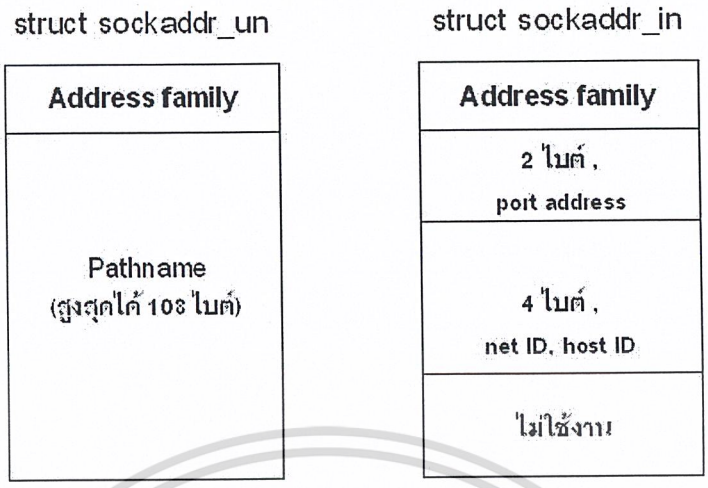
```
struct   sockaddr_in {
     short      int          sin_family;     /*      AF_INET        */
     unsigned   short  int   sin_port;       /*      Port  number   */
     struct     in_addr      sin_addr;       /*      IP Address     */
};
```

สำหรับโครงสร้างข้อมูล in_addr เป็นข้อมูล IP Address กำหนดโดย

```
struct   in_addr  {
        unsigned  long    ints_addr;
};
```

ไบต์ข้อมูล 4 ไบต์ ของ IP Address จะคิดเป็นค่าเลขจำนวนเต็มขนาด 32 บิต และจะเห็นได้ว่า Domains AF_INET จะอ้างถึง socket โดยใช้ Domains, IP Address และหมายเลขพอร์ต แต่ถ้ามองจากแอพพลิเคชั่นที่รันบนเน็ตเวิร์คแล้ว จะมองเห็น socket เป็นเพียงจำนวนเต็มค่าหนึ่ง (socket file descriptor)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 แสดงโครงสร้างของแอดเดรสที่ใช้งานใน AF_UNIX และ AF_INET

2.3.2.3 การกำหนด Socket ให้กับโปรเซส (Naming Socket)

การจะให้โปรเซสใช้งานได้จะต้องกำหนด socket นั้นให้แก่โปรเซส สำหรับ AF_UNIX จะกำหนดใช้ชื่อไฟล์ full path name แต่สำหรับ AF_INET จะต้องกำหนดหมายเลขพอร์ตที่จะใช้งาน โดยฟังก์ชั่น bind

```
# include        <sys/socket.h>
int bind (int socket, const struct sockaddr *address, size_t address_len);
```

socket — socket file descriptor

address — กำหนดแอดเดรสให้กับ socket

address_len — ความยาวโครงสร้างข้อมูลของแอดเดรส (ความยาวของแอดเดรสจะขึ้นอยู่กับ Domains ที่ใช้งาน)

2.3.2.4 การสร้าง Socket Queue

เพื่อรองรับข้อมูลการขอการเชื่อมต่อของไคลเอนต์ เซิร์ฟเวอร์จะต้องสร้าง queue โดยใช้ฟังก์ชั่นระดับต่ำ listen

```
# include        <sys/socket.h>
int        listen (int socket, int backlog);
```

ในระบบจะจำกัดจำนวนไคลเอนต์ในการติดต่อกับเซิร์ฟเวอร์บนคิว (queue) โดยกำหนดความจุ queue ไว้ใน backlog โดยถ้า queue ถูกใช้งานไปจนหมด การขอเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครั้งต่อไปจะทำไม่ได้ ซึ่งจะทำให้เซิร์ฟเวอร์ทำงานกับไคลเอนต์ในจำนวนที่เหมาะสม โดยปกติจะกำหนดค่า backlog ไว้ที่ 5 (ค่าสูงสุด)

2.3.2.5 การยอมรับการขอเชื่อมต่อ

หลังจากที่เซิร์ฟเวอร์โปรเซสรันฟังก์ชั่น listen แล้ว จะต้องกำหนดให้เซิร์ฟเวอร์โปรเซสรอรับการติดต่อจากไคลเอนต์ โดยใช้ฟังก์ชั่น accept มีรูปแบบการใช้งานดังต่อไปนี้

```
# include       <sys/socket.h>
int     accept (int socket, struct sockaddr *address, size_t *address_len);
```

socket — กำหนด server socket file descriptor ที่รอรับการติดต่อจากไคลเอนต์ จะส่งค่ากลับเมื่อ มีไคลเอนต์ติดต่อเข้ามา ซึ่งฟังก์ชั่นจะสร้าง socket ใหม่เพื่อติดต่อกับไคลเอนต์นั้นโดยเฉพาะ ซึ่งจะมีลักษณะเหมือนกับ socket ที่รอรับการติดต่อในตอนแรก (แต่มีข้อมูลบางอย่างเพิ่มเข้ามา) และโปรเซสต้นฉบับก็จะ fork สร้างโปรเซสลูกเพื่อทำงานกับไคลเอนต์โปรเซส และโปรเซสต้นฉบับเองก็จะรอการติดต่อจากไคลเอนต์อื่นต่อไป

address — แอดเดรสของไคลเอนต์ จะถูกนำไปเก็บไว้ในโครงสร้างของข้อมูล sockaddr ที่ชี้โดยพอยน์เตอร์ (ถ้าไม่ต้องการจะนำแอดเดรสของไคลเอนต์ไปใช้งานจะกำหนดเป็น null pointer ก็ได้)

Address_len — จะระบุความยาวของข้อมูลใน sockaddr ซึ่งในการส่งค่ากลับ address_len จะเก็บความยาวจริงของแอดเดรสไคลเอนต์ ซึ่งขึ้นอยู่กับ Domains เช่น AF_INET ก็จะเป็น 16 ไบต์ แต่ก็จะแตกต่างกันไปถ้าเป็น AF_UNIX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าไม่มีข้อมูลของไคลเอนต์ส่งเข้ามายัง queue ฟังก์ชั่น accept จะถูกบล็อกจนกระทั่งมีไคลเอนต์ติดต่อเข้ามา ซึ่งการใช้แฟลก O_NONBLOCK กับฟังก์ชั่น fcntl ก็จะเปลี่ยนการทำงานนี้ได้ดังต่อไปนี้

```
int     flags = fcntl (socket, F_GETFL, 0);
fcntl (socket, F_SETFL, O_NONBLOCK | flags);
```

ฟังก์ชั่น accept จะส่งค่ากลับเป็น 3 อย่างด้วยกัน คือ socket file descriptor ที่สร้างขึ้นมาใหม่เมื่อมีไคลเอนต์ติดต่อเข้ามา หรือ –1 ถ้าเกิด error โดย error ที่เป็นไปได้จะเหมือนกับ bind และ listen แต่มีการเพิ่ม EWOULDBLOCK เข้ามาในกรณีที่มีการใช้งาน O_NONBLOCK และไม่มีการติดต่อเข้าไป หรือ EINTR ถ้ามีการอินเตอร์รัพโปรเซส

นอกจากนี้ยังมีแอดเดรสของไคลเอนต์โปรเซส และความยาวของแอดเดรสที่รับได้จริง

socket ที่ส่งกลับจากฟังก์ชั่น accept (มีโปรเซสติดต่อเข้ามา) จะมีข้อมูลสำหรับอ้างอิงถึงองค์ประกอบการติดต่อ ครบถ้วน (5 tuple)

2.3.2.6 การติดต่อกับเซิร์ฟเวอร์โปรเซส

โปรแกรมไคลเอนต์ สามารถจะติดต่อไปยังเซิร์ฟเวอร์ โดยใช้งาน socket ที่ยังไม่ได้กำหนดให้กับโปรเซส) ส่งค่ากลับจากการใช้ฟังก์ชั่นระดับต่ำ socket (ติดต่อไปยัง server socket ด้วยฟังก์ชั่น connect

```
# include       <sys/socket.h>
int     connect  (int socket, const struct sockaddr *address, size_t address_len);
```

socket ที่เซิร์ฟเวอร์จะกำหนดด้วย address ที่มีความยาว address_len ถ้าทำงานสำเร็จจะส่งค่ากลับเป็น 0 หรือ 1 ถ้าเกิด error

2.3.2.7 การยกเลิกการใช้งาน Socket

โดยการใช้ฟังก์ชั่น close ร่วมกับ socket file descriptor และในการยกเลิกการใช้งาน socket ควรจะยกเลิกการทำงานทั้งที่ไคลเอนต์และเซิร์ฟเวอร์ ในบางกรณีฟังก์ชั่น close อาจจะถูกบล็อกถ้า socket มีข้อมูลที่ยังไม่ได้ส่งไป โดยเฉพาะถ้าเป็นโปรโตคอลที่มีความเชื่อถือในการส่งข้อมูลสูงเช่น TCP แต่มีการใช้ออพชั่น SOCK_LINGER กำหนดการจัดการกับข้อมูลที่ยังไม่ส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.18 แสดงการใช้งานฟังก์ชั่นในการสร้าง socket เพื่อติดต่อผ่านเน็ตเวิร์ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราะห์และการออกแบบ

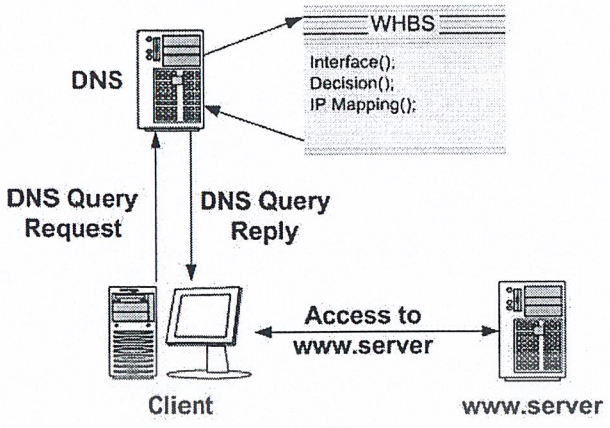
3.1 การวิเคราะห์

การปิดกั้นเว็บไซต์ที่ไม่เหมาะสมสามารถทำได้หลายวิธี ซึ่งอาจจะใช้ฮาร์ดแวร์, ซอฟท์แวร์ หรือใช้ทั้งฮาร์ดแวร์และซอฟท์แวร์ร่วมกันก็ได้ วิธีที่ใช้ทั้งฮาร์ดแวร์และซอฟท์แวร์ร่วมกันจะมี เน็ตเวิร์กอะแดปเตอร์ (Network adaptor) คอยรับข้อมูลที่เข้ามา และส่งข้อมูลที่ผ่านการตรวจสอบ แล้วออกไป ส่วนวิธีที่ใช้ซอฟท์แวร์นั้นก็สามารถทำได้หลายวิธี เช่น ใช้โดเมนเนมเซิร์ฟเวอร์, ไฟร์วอลล์หรือพร็อกซี่ ซึ่งแต่ละวิธีที่ได้กล่าวมานั้นอาจจะต้องทำการเพิ่มคุณสมบัติบางอย่างจึงจะปิดกั้นเว็บไซต์ที่ไม่เหมาะสมได้ ในที่นี้จะพิจารณาในส่วนของซอฟท์แวร์เพื่อนำมาวิเคราะห์ว่าควรจะดำเนินการที่โดเมนเนมเซิร์ฟเวอร์, ไฟร์วอลล์ หรือพร็อกซี่

3.1.1 การดำเนินการที่โดเมนเนมเซิร์ฟเวอร์ (DNS)

เราอาจจะทำ DNS ให้สามารถปิดกั้นเว็บไซต์ที่ไม่เหมาะสมได้ แต่ไม่สามารถทำให้ DNS กรองเนื้อหาในเว็บเพจได้ การทำให้ DNS สามารถปิดกั้นเว็บไซต์ที่ไม่เหมาะสมได้นั้นอาจจะทำได้โดยทำให้ DNS ไม่แปลงโดเมนเนมเป็นไอพีให้กับโดเมนเนมที่ไม่เหมาะสม โดยจะทำการเปรียบเทียบโดเมนเนมที่ผู้ใช้ป้อนกับโดเมนเนมที่มีอยู่ในโดเมนลิสต์ (โดเมนลิสต์ คือ รายชื่อโดเมนเนมที่ไม่เหมาะสม) ถ้ามีโดเมนเนมที่ผู้ใช้ป้อนอยู่ในโดเมนลิสต์ก็จะไม่ให้ DNS แปลงโดเมนเนมนั้นเป็นหมายเลขไอพี แต่ถ้าโดเมนเนมนั้นไม่อยู่ในโดเมนลิสต์ก็จะให้ DNS แปลงโดเมนเนมนั้นเป็นหมายเลขไอพีตามปกติ ซึ่งการกระทำดังกล่าวนั้นทำได้ค่อนข้างยาก เนื่องจาก DNS มีขนาดโปรแกรมที่ใหญ่มากยากต่อการนำมาแก้ไข และตามหลักการอีกประการหนึ่งคือ เมื่อเครื่องไคลเอนต์สร้างการเชื่อมต่อกับเว็บเซิร์ฟเวอร์แล้ว การส่งข้อมูลจะไม่ผ่านเครื่อง DNS อีก ทำให้ไม่สามารถกรองเนื้อหาของข้อมูลที่ส่งมาจากเว็บเซิร์ฟเวอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงการปิดกั้นการเข้าถึงเว็บไซต์ด้วย DNS

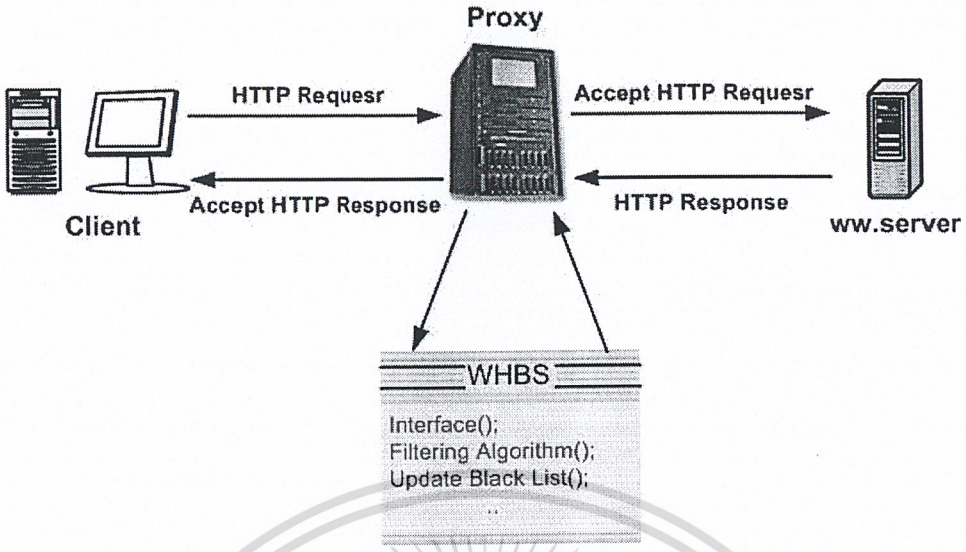
3.1.2 การดำเนินการที่ไฟร์วอลล์ (Firewall)

ไฟร์วอลล์นั้นจะมีคุณสมบัติที่แตกต่างกันขึ้นอยู่กับผู้ผลิตแต่ละราย ไฟร์วอลล์บางตัวสามารถกรองโดเมนเนมและเนื้อหาในเว็บเพจได้ ไฟร์วอลล์บางตัวสามารถกรองโดเมนเนมได้อย่างเดียว ดังนั้นถ้าต้องการดำเนินการที่ไฟร์วอลล์ก็สามารถทำได้โดย นำไฟร์วอลล์ที่มีคุณสมบัติในการกรองโดเมนเนม อย่างเดียวมาเพิ่มคุณสมบัติให้สามารถกรองเนื้อหาในเว็บเพจได้ ซึ่งการกระทำดังกล่าวนั้นมีขั้นตอนการดำเนินการค่อนข้างสลับซับซ้อน เนื่องจากการทำงานของไฟร์วอลล์จะอยู่ที่ลำดับชั้นของเคอร์เนล (kernel) ของระบบปฏิบัติการ

3.1.3 การดำเนินการที่พร็อกซี่ (Proxy)

พร็อกซี่แต่ละตัวนั้นจะมีคุณสมบัติที่แตกต่างกันขึ้นอยู่กับผู้พัฒนา พร็อกซี่บางตัวสามารถกรองโดเมนเนมได้ พร็อกซี่บางตัวสามารถกรองแบนเนอร์ได้ พร็อกซี่บางตัวก็มีความสามารถมากมาย เช่น squid เป็นต้น squid สามารถกรองโดเมนเนม, IP, URL ได้ และสามารถสำเนาข้อมูลเก็บไว้ได้ และ squid ยังมีคุณสมบัติอื่น ๆ อีกมากมาย โดยผู้ใช้จะต้องทำการกำหนดหน้าที่เพิ่มเอง แต่พร็อกซี่ทั่วไปไม่สามารถกรองเนื้อหาในเว็บเพจได้ ดังนั้นถ้าต้องการดำเนินการที่พร็อกซี่จะต้องทำการเพิ่มคุณสมบัติการกรองเนื้อหาในเว็บเพจเข้าไป เนื่องจากปกติแล้วเนื้อหาที่ต้องการกรองส่วนใหญ่นั้นจะอยู่ในระดับชั้นแอพพลิเคชั่น (Application Layer) ซึ่งพร็อกซี่ทำงานอยู่ในชั้นนี้เหมือนกัน ทำให้มีโอกาสในการดำเนินการได้มากกว่า 2 วิธีที่กล่าวไปแล้วข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงการปิดกั้นการเข้าถึงเว็บไซต์ด้วยพร็อกซี่และไฟร์วอลล์

ตารางที่ 3.1 แสดงการเปรียบเทียบคุณสมบัติทั่วไปของ DNS, ไฟร์วอลล์ และพร็อกซี่

DNS	Firewall	Proxy
1. เป็นระบบที่ง่ายและไม่สลับซับซ้อน	1. เป็นระบบที่สลับซับซ้อนมากที่สุด	1. เป็นระบบที่สลับซับซ้อนมาก
2. ตรวจสอบเฉพาะ DNS Message ที่ส่งมาถาม IP	2. ทำการตรวจสอบข้อมูลทั้งหมดในระหว่างการติดต่อสื่อสาร	2. ทำการตรวจสอบข้อมูลทั้งหมดในระหว่างการติดต่อสื่อสาร
3. ยากต่อการแก้ไขปรับปรุง	3. สามารถที่จะปรับปรุงเปลี่ยนแปลง แก้ไขการทำงานได้โดยง่าย	3. สามารถที่จะปรับปรุงเปลี่ยนแปลง แก้ไขการทำงานได้โดยง่าย
4. หลักการตรวจสอบจะตรวจสอบตามขั้นตอนการทำงานของ DNS	4. หลักการตรวจสอบจะขึ้นอยู่กับลักษณะการทำงานของข้อมูลทั้งหมด	4. หลักการตรวจสอบจะขึ้นอยู่กับการทำงานของ Proxy
5. ต้องแก้ไข DNS ไม่ให้แปลงโดเมนเนมที่ไม่เหมาะสม	5. Firewall ทั่วไปสามารถกรอง URL ได้	5. Proxy บางตัวสามารถกรอง URL ได้
6. ไม่สามารถแก้ไข DNS ให้สามารถกรองเนื้อหาได้ เพราะบราวเซอร์จะติดต่อมายัง DNS ครั้งเดียวเท่านั้นตอนถาม IP	6. Firewall บางตัวสามารถกรองเนื้อหาได้ บางตัวไม่สามารถกรองได้ขึ้นอยู่กับผู้ผลิตแต่ละราย ถ้าต้องการให้ Firewall ที่ไม่สามารถกรองเนื้อหาได้ให้กรองเนื้อหาได้ จะต้องเพิ่มฟังก์ชั่นในการกรองเข้าไป	6. Proxy ทั่วไปไม่สามารถกรองเนื้อหาได้ ต้องเพิ่มฟังก์ชั่นในการกรองเข้าไปจึงจะสามารถกรองเนื้อหาได้

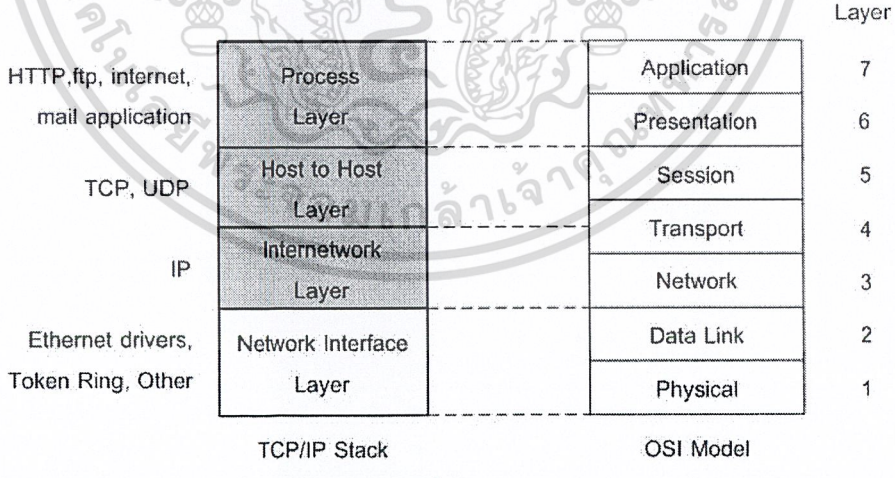
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 แสดงความแตกต่างของคุณสมบัติที่ต้องนำมาพิจารณา

	DNS	Firewall	Proxy
1. ความซับซ้อนของระบบ	น้อย	มากที่สุด	มาก
2. ความยากง่ายในการปรับปรุง	ยาก	ง่าย	ง่าย
3. กรอง URL	ไม่ได้	ได้	บางตัว
4. กรองเนื้อหา	ไม่ได้	บางตัว	ไม่ได้
5. ความยากง่ายในการนำมาดำเนินการให้สามารถกรอง URL และเนื้อหาได้	ยากมาก	ยาก	ปานกลาง

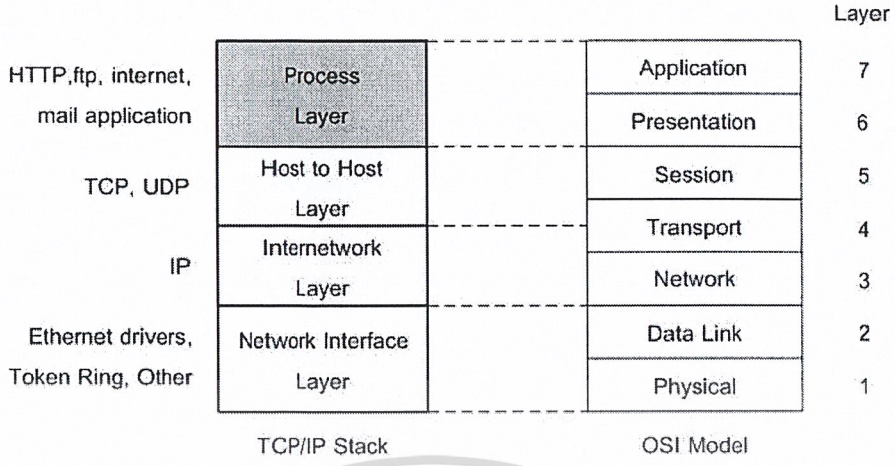
3.1.4 สรุปผลการวิเคราะห์

การดำเนินการที่เหมาะจะนำมาทำตัวกรองและปิดกั้นเว็บไซต์ที่ไม่เหมาะสมคือ การดำเนินการที่พร็อกซี่ เนื่องจาก DNS นั้นจะมีการทำงานที่ติดต่อกับการเข้าถึงเว็บไซต์ไม่มากนัก เพราะบราวเซอร์จะติดต่อกับ DNS ก็ต่อเมื่อต้องการทราบหมายเลขไอพีของเว็บเซิร์ฟเวอร์เท่านั้น ส่วนไฟร์วอลล์นั้นตามหน้าที่แล้วจะทำงานคล้ายกับพร็อกซี่ แต่ขอบเขตของการทำงานค่อนข้างลึกทำให้เกิดความยุ่งยากในการปรับแต่ง จึงต้องใช้ความรู้และความชำนาญอย่างสูงในการการดำเนินการ และเนื่องจากพร็อกซี่มีการทำงานส่วนใหญ่อยู่ในชั้นแอพพลิเคชั่น ซึ่งเป็นลำดับชั้นที่สามารถกรองข้อมูลที่ส่งผ่านได้อยู่แล้ว จึงเป็นการดำเนินการที่เหมาะสมที่สุด



รูปที่ 3.3 แสดงขอบเขตการทำงานซึ่งแบ่งตามเลเยอร์ของไฟร์วอลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงขอบเขตการทำงานซึ่งแบ่งตามเลเยอร์ของพร็อกซี่

3.2 การออกแบบ

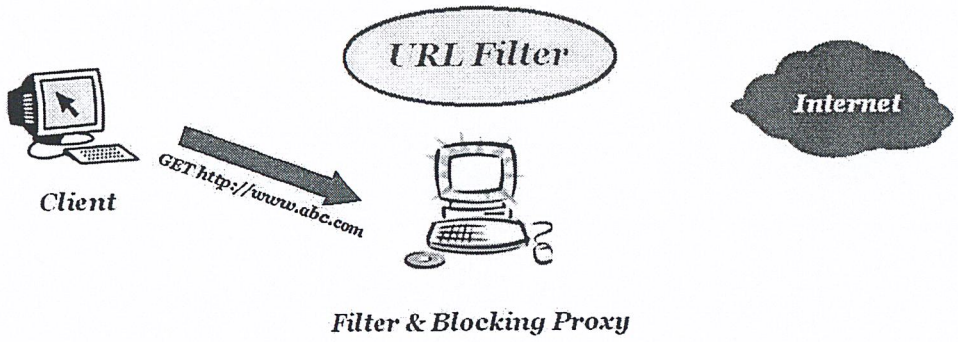
การออกแบบแบ่งออกเป็น 2 ส่วนดังนี้

3.2.1 การทำงานของระบบ

การทำงานของระบบนั้นจะมีการกรองข้อมูลที่รับส่งกันระหว่างไคลเอนต์และเซิร์ฟเวอร์ ซึ่งหากเป็นข้อมูลที่เครื่องไคลเอนต์ส่งไปร้องขอข้อมูลจากเว็บเซิร์ฟเวอร์นั้น จะเป็นข้อมูล URL ของเว็บเซิร์ฟเวอร์ ทำให้สามารถนำ URL มากรองเพื่อปิดกั้นการเข้าถึงเว็บไซต์ที่ไม่เหมาะสมได้ และหากเป็นข้อมูลที่เว็บเซิร์ฟเวอร์ส่งมาให้เครื่องไคลเอนต์นั้น จะเป็นเนื้อหาของเว็บเพจนั้น ทำให้สามารถกรองเนื้อหาของเว็บเพจก่อนที่จะส่งต่อไปยังเครื่องไคลเอนต์ได้ ซึ่งสามารถอธิบายขั้นตอนในการทำงานของระบบได้ดังนี้

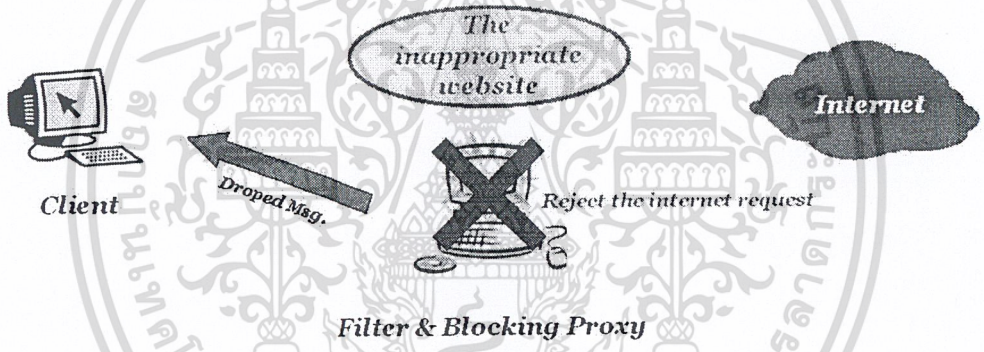
เมื่อเครื่องไคลเอนต์ร้องขอการเข้าถึงเว็บไซต์ (www.abc.com) และเมื่อการร้องขอนั้นถูกส่งมาที่ระบบ ระบบจะทำการกรอง URL ที่รับได้นั้น เพื่อพิจารณาว่าเป็นเว็บไซต์ที่เหมาะสมหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



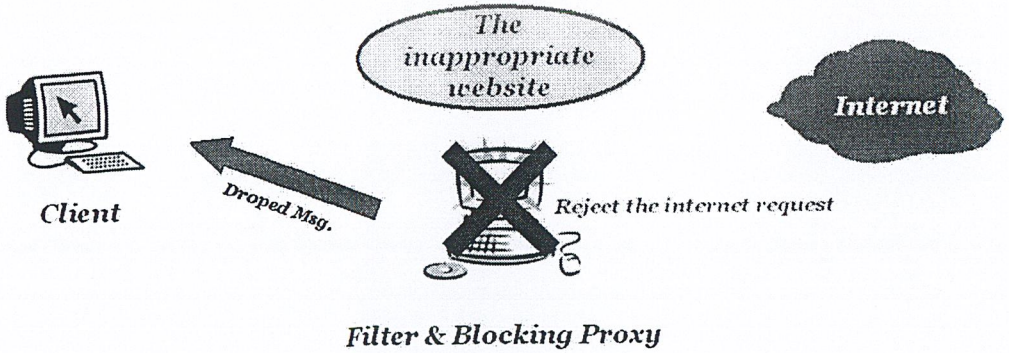
รูปที่ 3.5 แสดงการกรอง URL เมื่อได้รับการร้องขอจากเครื่องไคลเอนต์

และหากตรวจ URL แล้วพบว่าเป็นเว็บไซต์ที่ไม่เหมาะสม ระบบจะทำการปิดกั้นการเข้าถึงนั้นและส่ง Dropping Message กลับไปยังเครื่องไคลเอนต์ดังรูปที่ 3.6



รูปที่ 3.6 แสดงการปิดกั้นการเข้าถึงเว็บไซต์หากเป็นเว็บไซต์ที่ไม่เหมาะสม

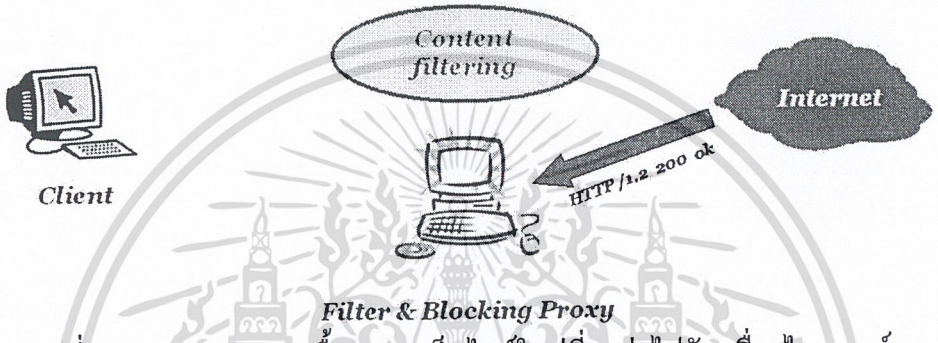
และหากพบว่าเป็นเว็บไซต์ที่เหมาะสม ระบบก็จะดำเนินการติดต่อไปยังเว็บไซต์ที่เครื่องไคลเอนต์ต้องการดังรูปที่ 3.7



รูปที่ 3.7 แสดงการส่งการร้องขอไปยังเซิร์ฟเวอร์ เมื่อเป็นเว็บไซต์ที่เหมาะสมหรือเว็บไซต์ใหม่

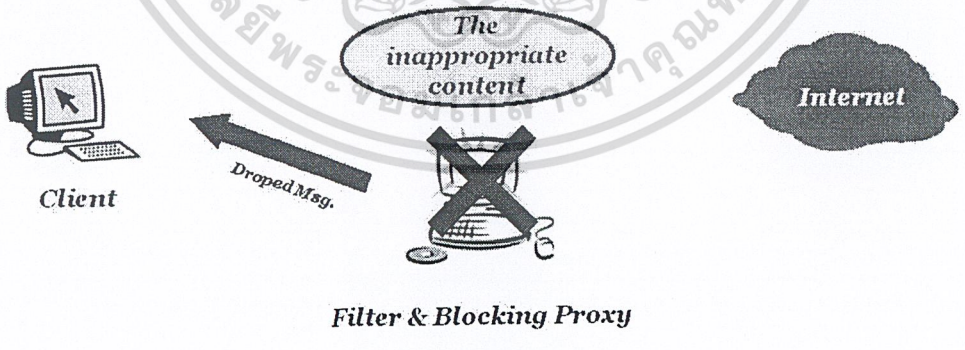
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผ่านพ้นขั้นตอนดังกล่าวแล้ว พร็อกซี่เซิร์ฟเวอร์จะส่งการร้องขอแทนเครื่องไคลเอนต์ และจะสร้างการเชื่อมต่อกับเว็บเซิร์ฟเวอร์เพื่อติดต่อกันด้วยโปรโตคอล HTTP เอง และเมื่อเว็บเซิร์ฟเวอร์ส่งข้อมูลที่ร้องขอกลับมาที่พร็อกซี่เซิร์ฟเวอร์ พร็อกซี่เซิร์ฟเวอร์ก็จะส่งข้อมูลต่อไปที่เครื่องไคลเอนต์อีกทอดหนึ่ง ซึ่งขั้นตอนนี้ทำให้ระบบสามารถกรองเนื้อหาที่ถูกส่งมาจากเว็บเซิร์ฟเวอร์ได้ ดังแสดงในรูปที่ 3.8 โดยการกลั่นกรองข้อมูลจะกรองที่ไฟล์ html ซึ่งจะกรองเฉพาะข้อมูลที่เป็นตัวหนังสือ (text) เท่านั้น



รูปที่ 3.8 แสดงการกรองเนื้อหาของเว็บไซต์ใหม่ที่จะส่งไปยังเครื่องไคลเอนต์

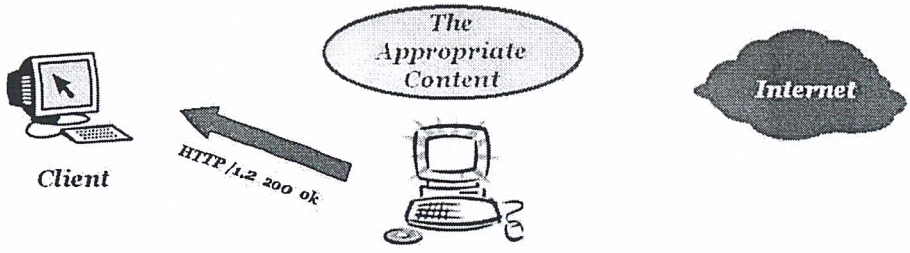
หากกรองเนื้อหาในเว็บเพจแล้วพิจารณาว่าเป็นเนื้อหาที่ไม่เหมาะสม ระบบจะทำการปิดกั้นข้อมูลที่จะส่งไปให้เครื่องไคลเอนต์ และจะส่ง Dropping Message กลับไปให้แทน และนำ URL นั้นไปใส่เพิ่มใน Black List ดังแสดงในรูปที่ 3.9



รูปที่ 3.9 แสดงการปิดกั้นข้อมูลที่จะส่งไปให้เครื่องไคลเอนต์
เมื่อพิจารณาว่าเป็นเนื้อหาที่ไม่เหมาะสม

หากกรองเนื้อหาในเว็บเพจแล้วพิจารณาว่าเป็นเนื้อหาที่เหมาะสม ระบบจะทำการส่งข้อมูลต่อไปยังเครื่องไคลเอนต์ ดังแสดงในรูปที่ 3.10

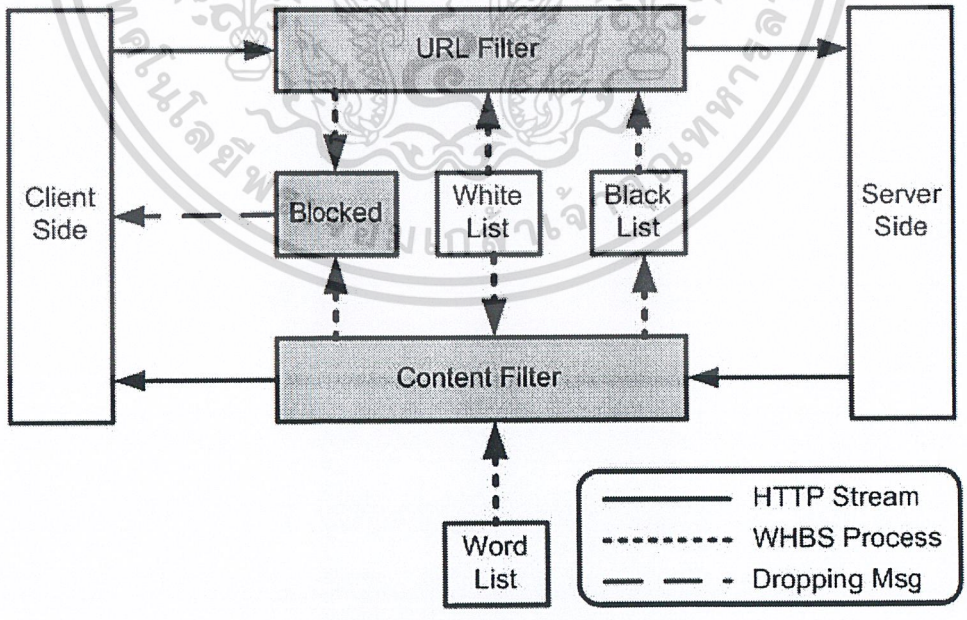
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงการส่งข้อมูลต่อไปยังเครื่องไคลเอนต์ เมื่อพิจารณาว่าเป็นเนื้อหาที่เหมาะสม

3.2.2 โครงสร้างของระบบโดยรวม

ระบบนี้จะแบ่งออกเป็น 2 ส่วน คือ ส่วนของการกรอง URL และส่วนของการกรองเนื้อหาในเว็บเพจ เมื่อมีการร้องขอจากไคลเอนต์เข้ามา ระบบจะนำ URL มากรอง ถ้าพบว่า URL นั้นไม่เหมาะสมก็จะปิดกั้น URL นั้น แต่ถ้าไม่ใช่ระบบก็จะส่งการร้องขอต่อไปยังเซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ส่งเว็บเพจกลับมา ระบบจะนำข้อมูลนั้นมากรอง ถ้าพบว่าเว็บเพจนั้นมีเนื้อหาที่ไม่เหมาะสมก็จะปิดกั้นเว็บเพจนั้นและทำการอัพเดท URL ลงในไฟล์ Black List แต่ถ้าไม่ใช่ก็จะส่งเว็บเพจนั้นกลับไปยังไคลเอนต์



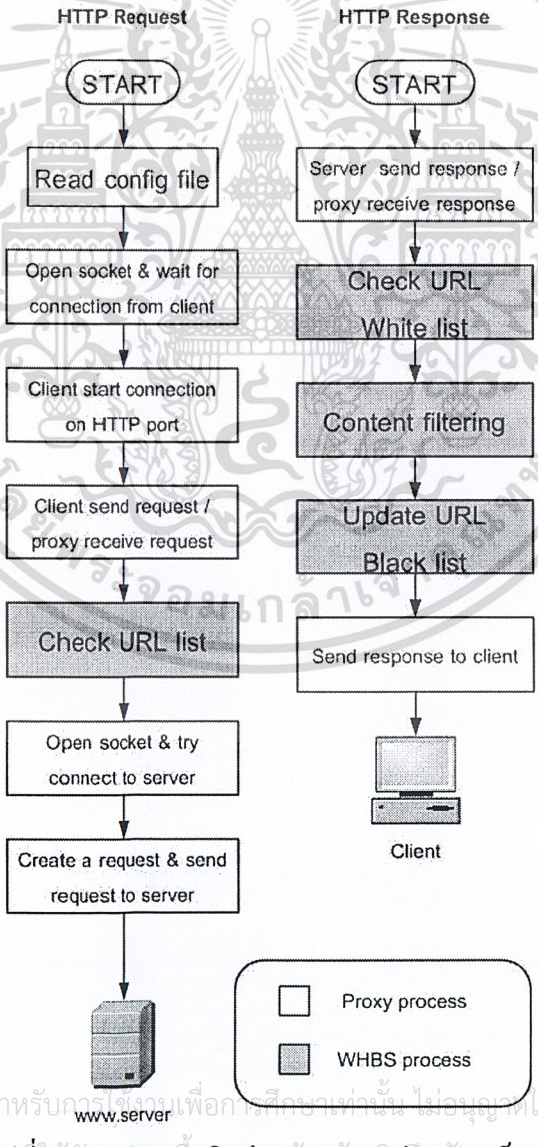
รูปที่ 3.11 แสดงโครงสร้างของระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การทำงานร่วมกับพร็อกซี่

จากการวิเคราะห์สรุปได้ว่าจะให้ระบบทำงานร่วมกับพร็อกซี่นั้น จึงได้เลือกพร็อกซี่ที่มีคุณสมบัติทั่วไปซึ่งไม่มีฟังก์ชั่นในการกรอง URL หรือเนื้อหา แล้วนำพร็อกซี่นั้นมาเพิ่มฟังก์ชั่นในการกรองเข้าไป

พร็อกซี่ที่จะนำมาปรับแต่งคือ BannerKiller ซึ่งได้ทำการดาวน์โหลดมาจากอินเตอร์เน็ต [8] BannerKiller เป็นพร็อกซี่ที่ไว้สำหรับกรองแบนเนอร์ แต่เราจะนำส่วนที่ทำหน้าที่ติดต่อกับไคลเอนต์และเซิร์ฟเวอร์มาใช้เท่านั้น ซึ่งส่วนดังกล่าวเป็นหน้าที่หลักของพร็อกซี่ ดังนั้นต่อไปเราจะเรียกส่วนนี้ว่าพร็อกซี่ จากนั้นเราก็ทำการออกแบบส่วนการทำงานที่จะต้องเพิ่มเข้าไปว่าจะติดต่อกับโปรเซสใดของ พร็อกซี่ ซึ่งได้แสดงดังรูปที่ 3.12



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.12 แสดงการติดต่อการทำงานร่วมกับพร็อกซี่

3.2.3.1 กระบวนการทำงานของพร็อกซี่

จากรูปที่ 3.10 สามารถอธิบายกระบวนการทำงานของพร็อกซี่ได้ดังนี้

1.) ส่วนของการรับ HTTP Request

ขั้นตอนที่ 1 อ่านค่าต่าง ๆ ที่กำหนดไว้ในไฟล์คอนฟิกมาเก็บไว้ เพื่อนำค่านี้มาใช้ในขั้นตอนต่อ ๆ ไป

ขั้นตอนที่ 2 สร้าง socket เพื่อรอรับการเชื่อมต่อจากไคลเอนต์ ตามขั้นตอนการใช้งาน socket ที่ได้กล่าวไปแล้วในหัวข้อ 2.3.2

ขั้นตอนที่ 3 ไคลเอนต์เริ่มการเชื่อมต่อที่พอร์ท HTTP จากนั้นพร็อกซี่จะทำการยอมรับการเชื่อมต่อ และกำหนดไคลเอนต์ใหม่

ขั้นตอนที่ 4 ไคลเอนต์ส่งการร้องขอมาที่พร็อกซี่ และพร็อกซี่รับการร้องขอนั้น

ขั้นตอนที่ 5 สร้าง socket เพื่อติดต่อกับเซิร์ฟเวอร์ และพยายามติดต่อไปยังเซิร์ฟเวอร์

ขั้นตอนที่ 6 สร้างการร้องขอและส่งการร้องขอนั้นไปยังเซิร์ฟเวอร์ โดยส่งการร้องขอไปที่ socket ของเซิร์ฟเวอร์

2.) ส่วนของการรับ HTTP Response

ขั้นตอนที่ 1 เซิร์ฟเวอร์ส่ง response มาที่พร็อกซี่

ขั้นตอนที่ 2 พร็อกซี่ส่งข้อมูลกลับไปให้ไคลเอนต์

3.2.3.2 กระบวนการทำงานของฟังก์ชั่นที่เพิ่มเข้าไปในพร็อกซี่

เนื่องจากพร็อกซี่นี้ไม่สามารถจะทำการกรอง URL และเนื้อหาของเว็บเพจได้ จึงจำเป็นที่จะต้องเพิ่มฟังก์ชั่นเข้าไปดังนี้

1.) ฟังก์ชั่นสำหรับกรอง URL

ฟังก์ชั่นนี้จะทำงานอยู่ระหว่างขั้นตอนที่ 4 และขั้นตอนที่ 5 ของพร็อกซี่ในส่วนของการรับ HTTP Request ขั้นตอนการทำงานของฟังก์ชั่นแสดงดังรูปที่ 3.13

จากรูปที่ 3.13 ฟังก์ชั่นจะรับค่ามาจากขั้นตอนที่ 4 เพื่อนำ URL ที่ไคลเอนต์ร้องขอมากรองโดยใช้ฟังก์ชั่น strstr ซึ่งได้อธิบายไปแล้วในหัวข้อที่ 2.2.1 ฟังก์ชั่นจะเช็คว่ามี URL นั้นอยู่ในไฟล์ “whiteList” หรือไม่

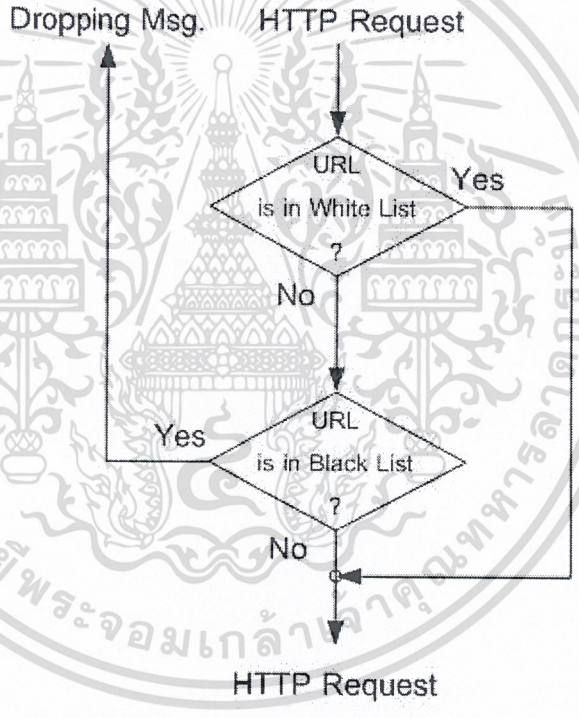
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ามี URL นั้นอยู่ในไฟล์ “whiteList” ก็จะส่ง URL นั้นไปทำงานต่อที่พร็อกซี่ในขั้นตอนที่ 5 ในส่วนของการรับ HTTP Request

ถ้าไม่มี URL นั้นอยู่ในไฟล์ “whiteList” ก็จะทำการเช็คว่า URL นั้นอยู่ในไฟล์ “blackList” หรือไม่

ถ้ามี URL นั้นอยู่ในไฟล์ “blackList” ก็จะส่งข้อความกลับไปยังไคลเอนต์ในรูปของ HTML ว่าไม่สามารถเข้าถึง URL นั้นได้

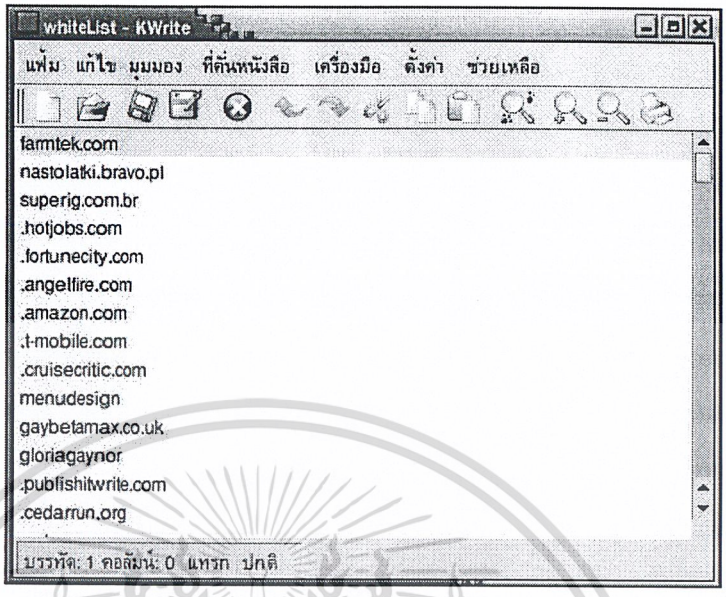
ถ้าไม่มี URL นั้นอยู่ในไฟล์ “blackList” ก็จะส่ง URLนั้นไปทำงานต่อที่พร็อกซี่ในขั้นตอนที่ 5 ในส่วนของการรับ HTTP Request



รูปที่ 3. 13 แสดงผังการทำงานเมื่อไคลเอนต์ส่ง HTTP Request

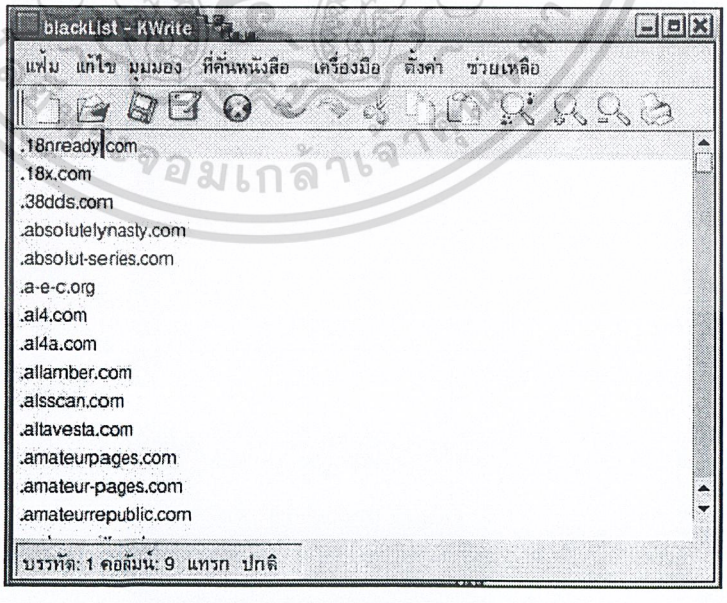
ไฟล์ “whiteList” จะเก็บ URL, domain หรือ ip ที่เหมาะสม ซึ่งดาวน์โหลดมาจาก http://web.onda.com.br/orso/sxcontrol.html ข้อมูลในไฟล์ “whiteList” แสดงดังรูปที่ 3.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงข้อมูลในไฟล์ “whiteList”

ไฟล์ “blackList” จะเก็บ URL, domain หรือ ip ที่ไม่เหมาะสม ซึ่งดาวน์โหลดมาจาก http://web.onda.com.br/orso/sxcontrol.html ข้อมูลในไฟล์ “blackList” แสดงดังรูปที่ 3.15



รูปที่ 3.15 แสดงข้อมูลในไฟล์ “blackList”

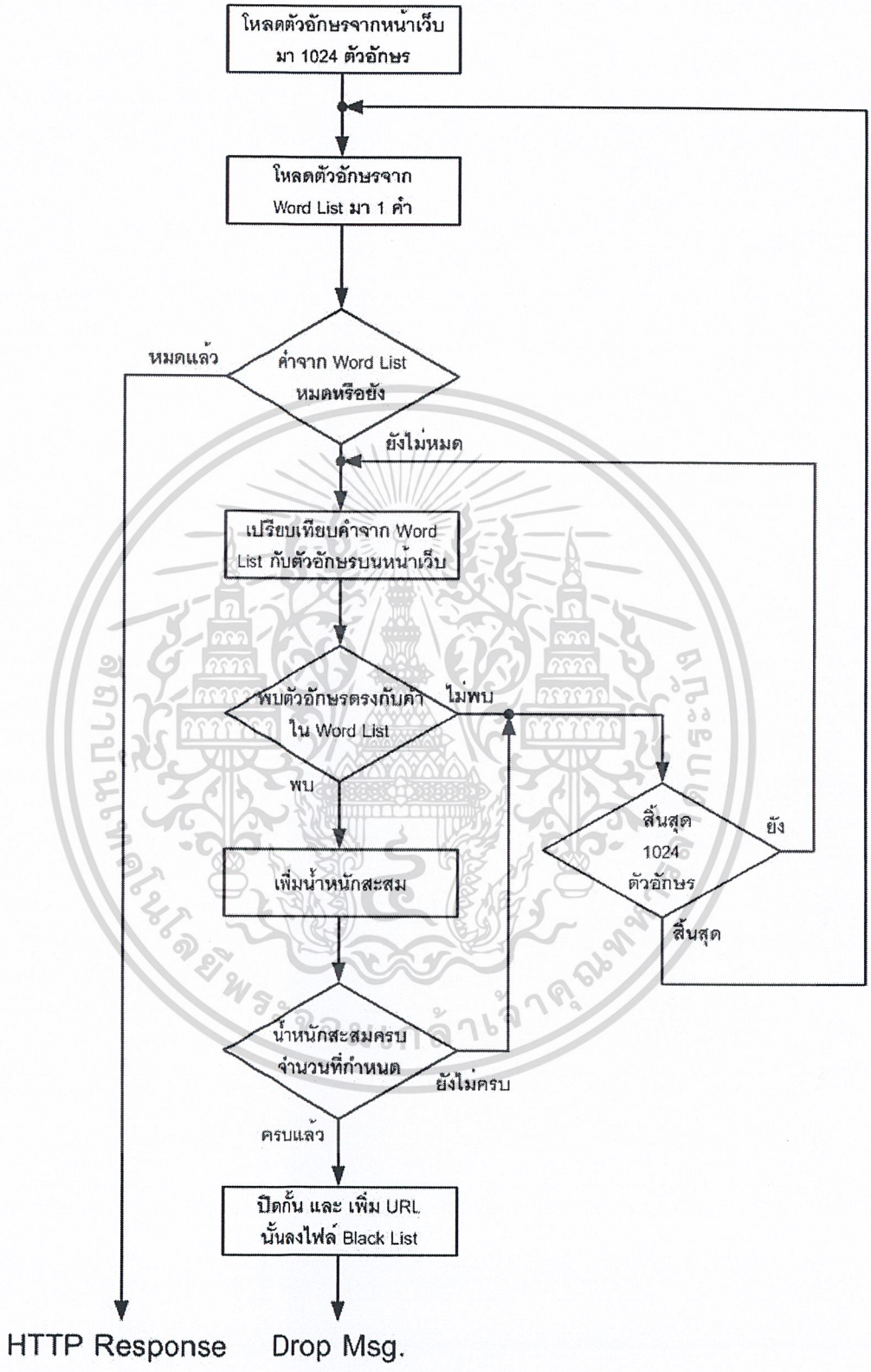
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) ฟังก์ชั่นสำหรับกรองเนื้อหาในเว็บเพจ

ฟังก์ชั่นนี้จะทำงานอยู่ระหว่างขั้นตอนที่ 1 และขั้นตอนที่ 2 ของพร็อกซี่ในส่วนของการรับ HTTP Response ขั้นตอนการทำงานของฟังก์ชั่นแสดงดังรูปที่ 3.16

จากรูปที่ 3.16 ฟังก์ชั่นจะรับค่ามาจากขั้นตอนที่ 1 เพื่อนำข้อมูลในเว็บเพจที่ส่งมาจากเซิร์ฟเวอร์มากรอง (โดยจะทำการกรองเพียงแค่ 1,024 ตัวอักษรเท่านั้น เนื่องจากการกรองข้อมูลทั้งเว็บเพจนั้นต้องใช้เวลาพอสมควร สำหรับเว็บเพจที่มีเนื้อหาเหมาะสม แต่จะใช้เวลาใกล้เคียงกันถ้าเว็บเพจนั้นมีเนื้อหาไม่เหมาะสม เนื่องจากถ้าพบกลุ่มคำที่ไม่เหมาะสมเกินค่าที่ตั้งไว้จะออกจากฟังก์ชั่นทันที จึงไม่จำเป็นที่จะต้องกรองข้อมูลทั้งเว็บเพจ) โดยฟังก์ชั่นจะเปรียบเทียบกลุ่มคำที่อยู่ในไฟล์ “wordList” กับกลุ่มคำที่อยู่ในเว็บเพจโดยใช้อัลกอริทึมของบอยเออร์มัวร์ ซึ่งได้อธิบายไปแล้วในหัวข้อที่ 2.2.2 ถ้าตรงกันจะบวกน้ำหนักของกลุ่มคำนั้นกับน้ำหนักเดิมก่อนหน้านี้ ถ้าน้ำหนักสะสมมากกว่าน้ำหนักที่กำหนดไว้ ฟังก์ชั่นจะส่งข้อความกลับไปให้ไคลเอนต์ในรูปของ HTML ว่าพบเนื้อหาที่ไม่เหมาะสมในเว็บเพจนั้นและทำการเพิ่ม URL นั้นลงที่ไฟล์ Black List แต่ถ้าเปรียบเทียบกลุ่มคำในไฟล์ “wordList” จนหมดแล้วน้ำหนักไม่เกินที่กำหนดไว้ ฟังก์ชั่นจะส่งข้อมูลเว็บเพจไปทำงานต่อที่พร็อกซี่ในขั้นตอนที่ 2 ในส่วนของการรับ HTTP Response

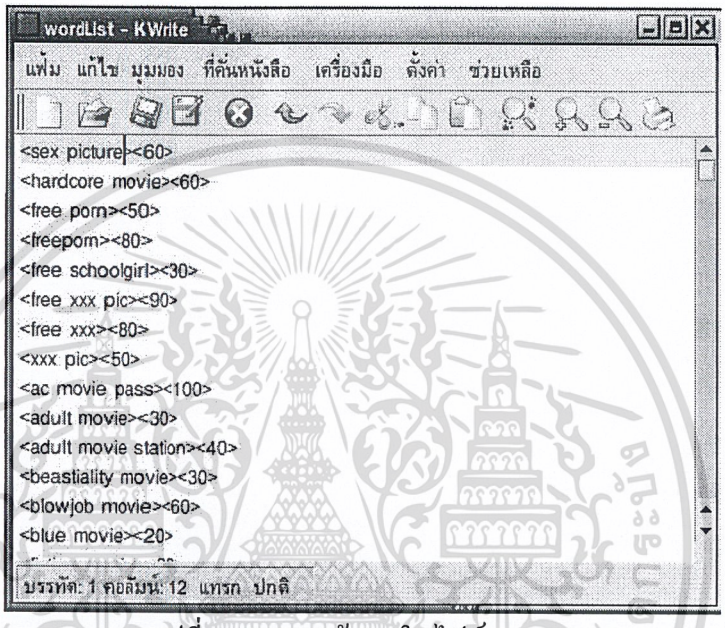
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



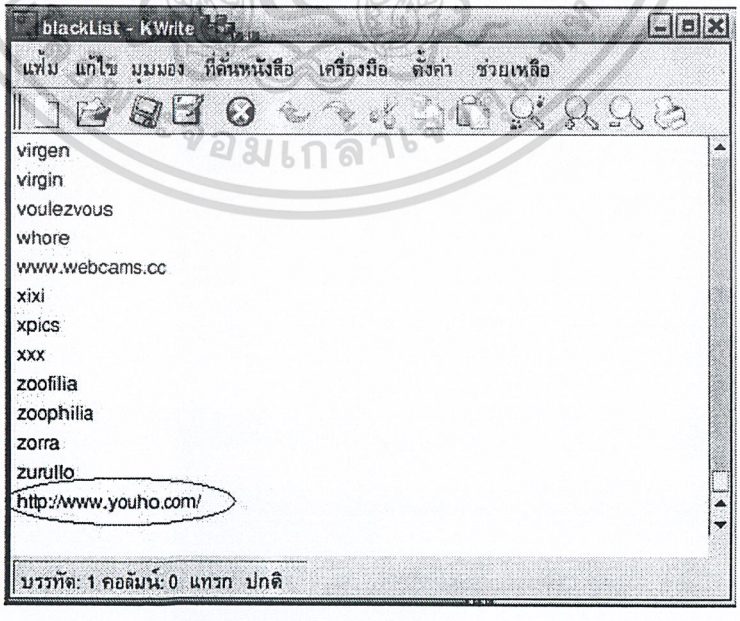
รูปที่ 3.16 แสดงผังการทำงานในส่วนของการกรองเนื้อหาเมื่อได้รับ HTTP Response

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ “wordList” จะเก็บคำหรือกลุ่มคำที่ไม่เหมาะสม และน้ำหนักของคำหรือกลุ่มคำนั้น รายชื่อคำในไฟล์ “wordList” ส่วนหนึ่งนำมาจากไฟล์ “weighted” ซึ่งอยู่ในโปรแกรม dansguardian และอีกส่วนหนึ่งนำมาจากคำที่อยู่ใน META TAG ของเว็บเพจที่ไม่เหมาะสม ข้อมูลในไฟล์ “wordList” แสดงดังรูปที่ 3.17



รูปที่ 3.17 แสดงข้อมูลในไฟล์ “wordList”



รูปที่ 3.18 แสดงข้อมูลที่เพิ่มขึ้นใน Black List เมื่อพบว่าเป็นเว็บเพจที่มีเนื้อหาไม่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

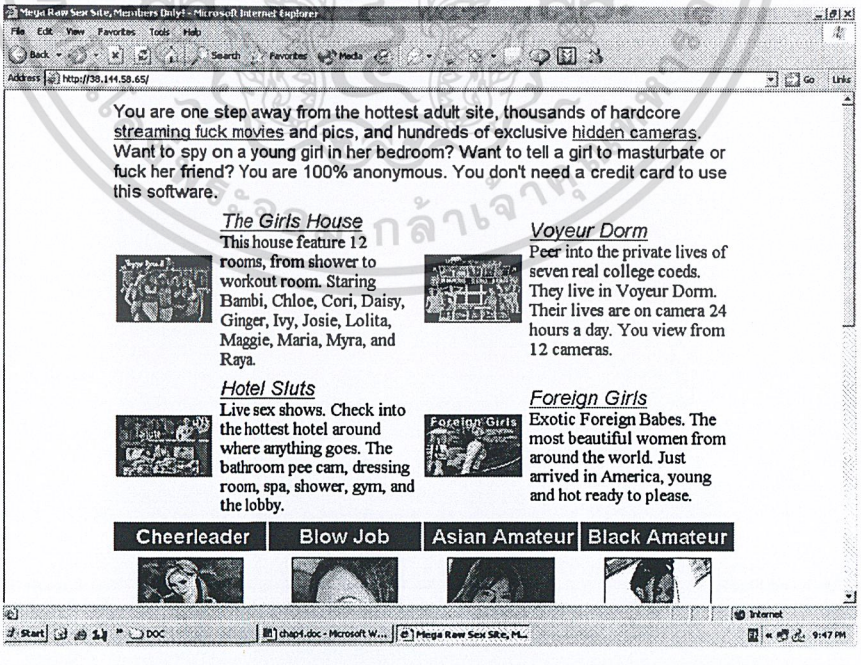
ผลการทดลองแบ่งออกเป็น 2 ส่วน คือ ผลการทดลองทางฝั่งของเครื่องไคลเอนต์ และผลการทดลองทางฝั่งของเครื่องที่ติดตั้งโปรแกรม Web Host Blocking System

4.1 ผลการทดลองทางฝั่งของเครื่องไคลเอนต์

ส่วนนี้จะแสดงถึงเนื้อหาที่ผู้ใช้งานฝั่งไคลเอนต์จะพบบนบราวเซอร์ เพื่อให้เห็นผลการทดลองตามเงื่อนไขของฟังก์ชั่นต่าง ๆ ที่ได้พัฒนาขึ้นมา โดยจะแบ่งการทำงานของแต่ละฟังก์ชันตามเหตุการณ์ต่าง ๆ ดังนี้

4.1.1 การปิดกั้นจากการกรอง URL ที่ไม่เหมาะสม

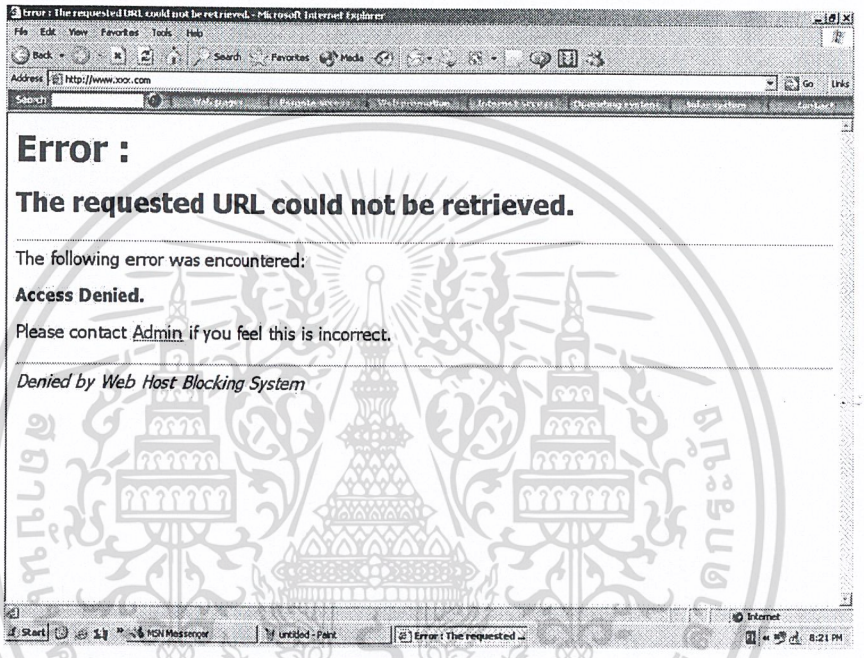
เมื่อมีการร้องขอการเข้าถึงเว็บไซต์ที่ไม่เหมาะสม (www.xxx.com) จากเครื่องลูกข่าย เมื่อยังไม่ได้ติดตั้งฟังก์ชั่นกรอง URL Black List ก็จะปรากฏเนื้อหาจากเว็บเซิร์ฟเวอร์ที่เราร้องขอบนตัวบราวเซอร์ของเครื่องลูกข่ายดังรูปที่ 4.1



รูปที่ 4.1 แสดงเนื้อหาที่ปรากฏบนบราวเซอร์ของเครื่องไคลเอนต์เมื่อร้องขอเว็บไซต์ที่ไม่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการติดตั้งฟังก์ชันกรอง URL Black List ที่เครื่องพร็อกซี่ แล้วเครื่องลูกข่ายมีการร้องขอการเข้าถึงเว็บไซต์ดังกล่าว (www.xxx.com) อีกครั้ง Web Host Blocking System จะทำการกรอง URL ซึ่งได้มี www.xxx.com อยู่ในไฟล์ "blackList" อยู่แล้ว ทำให้การร้องขอจากบราวเซอร์ของเครื่องลูกข่ายถูกปิดกั้น โดยระบบจะส่ง Dropping Message ในรูปแบบ html มายังบราวเซอร์ของเครื่องลูกข่ายดังรูปที่ 4.2

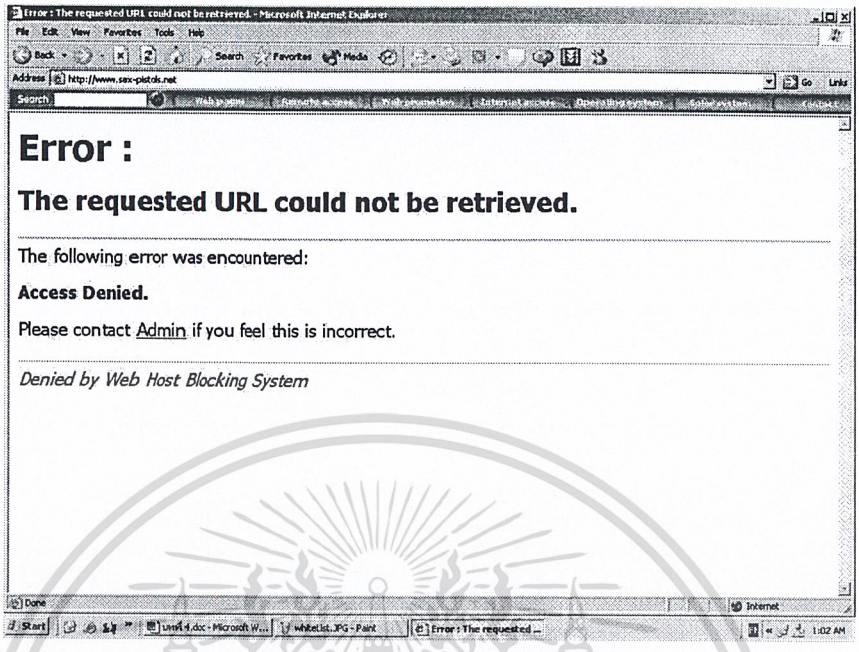


รูปที่ 4.2 แสดง Dropping Message บนบราวเซอร์ของเครื่องไคลเอนต์เมื่อเครื่องไคลเอนต์ร้องขอการเข้าถึงเว็บไซต์ที่ไม่เหมาะสมเมื่อทำการติดตั้งฟังก์ชันกรอง URL Black List

4.1.2 การยอมรับเว็บไซต์ที่เหมาะสม แต่มี URL สื่อไปในทางที่ไม่เหมาะสม

เมื่อทำการติดตั้งฟังก์ชันสำหรับกรอง URL Black List แล้วจะเกิดปัญหาคือ จะมีเว็บไซต์ส่วนหนึ่งที่เป็นเว็บไซต์เหมาะสมแต่จะถูกปิดกั้นจากฟังก์ชันกรอง URL Black List เนื่องจากใน URL นั้นประกอบด้วยคำที่ถือว่าไม่เหมาะสม เช่น www.sex-pistols.net เพราะฉะนั้นจึงต้องทำการตรวจสอบก่อนว่าเป็น URL ของเว็บไซต์ที่เหมาะสมหรือไม่ ซึ่งรายชื่อเว็บไซต์เหล่านี้จะเก็บอยู่ในไฟล์ "whiteList" ถ้าเว็บไซต์ที่ร้องขออยู่ในไฟล์นี้ก็ไม่ต้องส่งข้อมูลเข้าสู่ฟังก์ชันกรอง URL Black List ซึ่งฟังก์ชันที่ตรวจสอบเว็บไซต์ที่เหมาะสมกลุ่มนี้คือ ฟังก์ชันกรอง URL White List

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดง Dropping Message บนบราวเซอร์ของเครื่องไคลเอนต์ เมื่อไคลเอนต์ทำการร้องขอเว็บไซต์ที่เหมาะสม แต่มี URL ที่สื่อไปในทางที่ไม่เหมาะสม

รูปที่ 4.4 แสดงเนื้อหาที่ปรากฏบนบราวเซอร์ของเครื่องไคลเอนต์
เมื่อทำการร้องขอเว็บไซต์ที่เหมาะสม แต่มี URL ที่สื่อไปในทางที่ไม่เหมาะสม
หลังจากที่ติดตั้งฟังก์ชันกรอง URL White List แล้ว

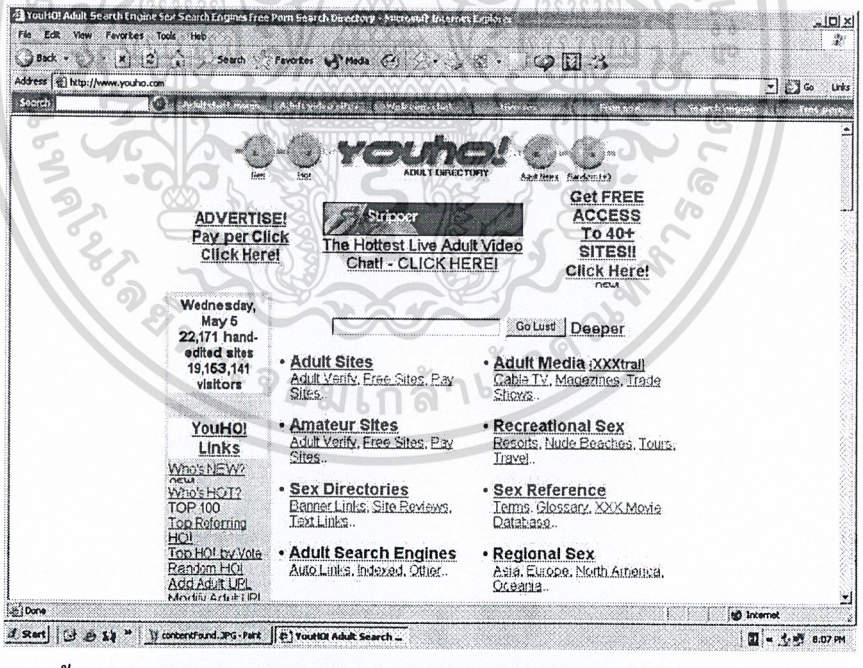
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การปิดกั้นจากการกรองเนื้อหาที่ไม่เหมาะสม

กรณีนี้จะเกิดขึ้นก็ต่อเมื่อระบบไม่มีรายชื่อ URL ของเว็บไซต์ที่ไม่เหมาะสมนั้น ทำให้ไม่สามารถปิดกั้นการร้องขอการเข้าถึงเว็บไซต์ไม่เหมาะสมนั้นได้จากการกรอง URL จึงต้องมีการกรองเนื้อหาที่ส่งมาจากเว็บเซิร์ฟเวอร์ก่อนที่จะส่งไปยังไคลเอนต์ ซึ่งจะตรวจสอบคำที่พบบนเว็บเพจกับคำในไฟล์ที่เก็บคำที่ไม่เหมาะสมที่ชื่อว่า "wordList" ซึ่งแต่ละคำจะมีน้ำหนักประจำตัวของตัวเอง โดยจะบวกน้ำหนักของคำที่พบเข้าด้วยกัน ซึ่งหากน้ำหนักรวมที่คำนวณได้มากกว่าน้ำหนักที่ตั้งไว้เพื่อเป็นเกณฑ์ในการตัดสินใจว่าเป็นเว็บไซต์ที่ไม่เหมาะสมหรือไม่ ก็จะทำการปิดกั้นการส่งข้อมูลนั้นทันที

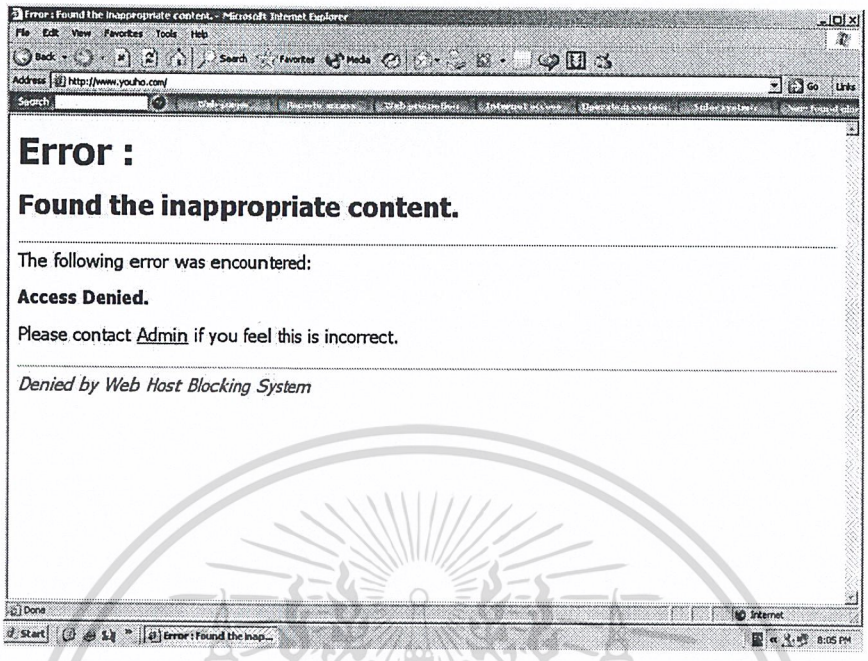
เมื่อมีการร้องขอเว็บไซต์ www.youho.com ซึ่งไม่อยู่ในไฟล์ "blackList" การร้องขอนั้นก็จะถูกส่งต่อไปยังเว็บเซิร์ฟเวอร์ทำให้สามารถเข้าถึงเว็บไซต์ที่ไม่เหมาะสมได้ แสดงดังรูปที่ 4.5

และเมื่อใส่ฟังก์ชันของการกรองเนื้อหาบนเว็บเพจ ก่อนที่ข้อมูลจะถูกส่งไปยังไคลเอนต์ก็จะถูกปิดกั้นทันทีเมื่อน้ำหนักรวมของคำที่พบนั้นมากกว่าน้ำหนักที่เรากำหนดไว้แล้ว แสดงดังรูปที่ 4.6



รูปที่ 4.5 แสดงเนื้อหาที่ปรากฏบนบราวเซอร์ของเครื่องไคลเอนต์เมื่อทำการร้องขอเว็บไซต์ที่ไม่เหมาะสมที่ โดยไม่มีการกรองเนื้อหาก่อนส่งมายังไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดง Dropping Message บนบราวเซอร์ของเครื่องไคลเอนต์
ที่ได้จากการกรองเนื้อหาบนเว็บเพจก่อนที่จะส่งมายังไคลเอนต์

4.2 ผลการทดลองทางฝั่งของเครื่องที่ติดตั้งโปรแกรม Web Host Blocking System

ส่วนนี้เป็นส่วนของ Graphic User Interface (GUI) ซึ่งแสดงดังรูปที่ 4.5 แบ่งออกเป็น 3 ส่วน คือ 1. ส่วนของการเพิ่ม, ลบหรืออัพเดท URL หรือ IP Address ที่เป็นของเว็บไซต์ที่ไม่เหมาะสม 2. ส่วนของการเพิ่ม, ลบหรืออัพเดท URL หรือ IP Address ที่เป็นของเว็บไซต์ที่เหมาะสม 3. ส่วนของการเพิ่ม, ลบหรืออัพเดท คำหรือกลุ่มคำที่ไม่เหมาะสมที่ใช้ในการกรองเนื้อหาบนเว็บเพจ

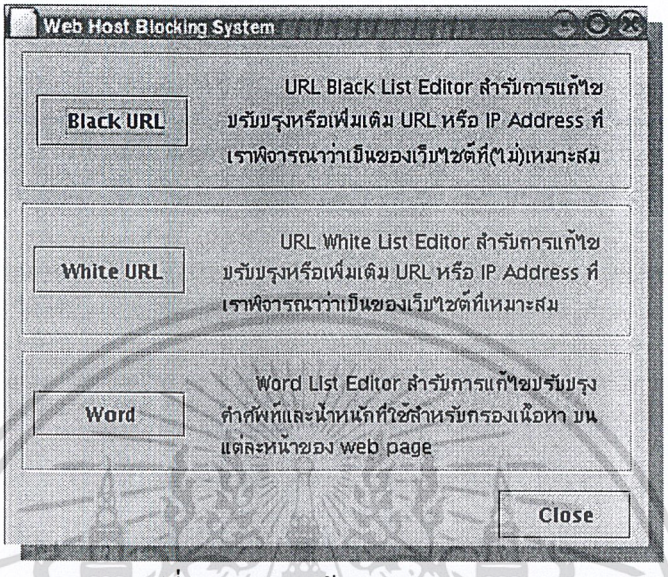
ในส่วนของ GUI นี้จะทำให้ผู้ดูแลระบบสามารถเพิ่ม, ลบหรืออัพเดท URL Black List, URL White List และ Word List ได้สะดวกมากขึ้น

4.2.1 Black URL

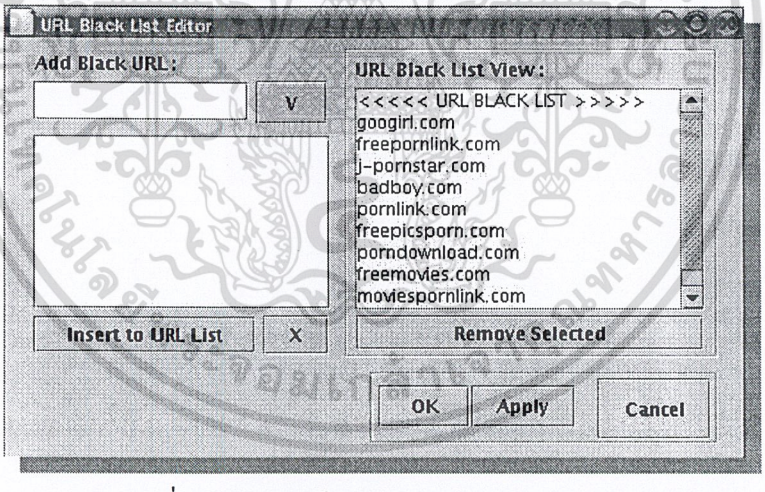
ส่วนนี้ไว้สำหรับเพิ่ม, ลบหรืออัพเดท URL หรือ IP Address ที่เป็นของเว็บไซต์ที่ไม่เหมาะสม จากรูปที่ 4.7 เมื่อคลิ๊กที่ปุ่ม “Black URL” จะแสดงหน้า URL Black List Editor ดังรูปที่ 4.8 ในหน้านี้จะมีช่องให้ใส่ URL ที่ไม่เหมาะสม จากนั้นคลิ๊กปุ่ม “Insert to URL List” เพื่อเพิ่ม URL เข้าไปในไฟล์ “blackList” หรือเราสามารถลบ URL ออกจากไฟล์ได้โดยเลือกจากรายชื่อที่แสดงในช่อง URL Black List View ในช่องของ URL Black List View นี้เป็นการนำรายชื่อ URL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ไม่เหมาะสมจากไฟล์ “blackList” มาแสดง เมื่อเพิ่มหรือลบ URL เรียบร้อยแล้ว คลิ๊กปุ่ม “OK” หรือ “Apply” เพื่อบันทึกข้อมูล หรือคลิ๊กปุ่ม “Cancel” เพื่อยกเลิก



รูปที่ 4.7 แสดงหน้าแรกของ GUI



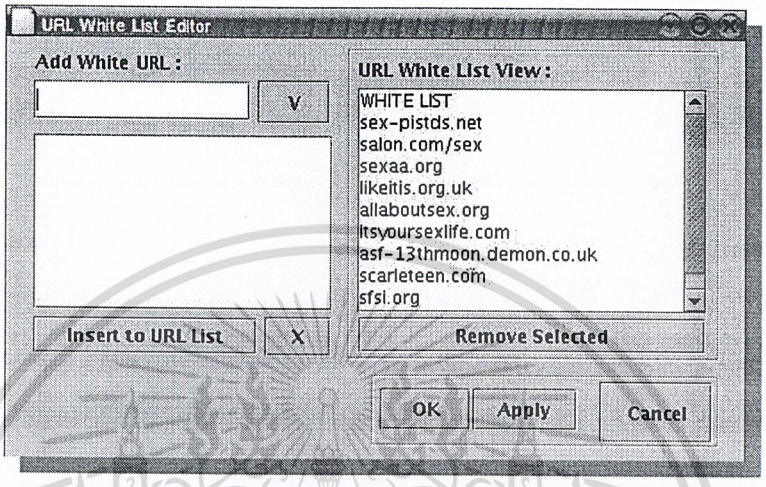
รูปที่ 4.8 แสดงหน้า URL Black List Editor

4.2.2 White URL

ส่วนนี้ไว้สำหรับเพิ่ม, ลบหรืออัพเดท URL หรือ IP Address ที่เป็นของเว็บไซต์ที่เหมาะสม จากรูปที่ 4.7 เมื่อคลิ๊กที่ปุ่ม “White URL” จะแสดงหน้า URL White List Editor ดังรูปที่ 4.9 ในหน้านี้จะมีช่องให้ใส่ URL ที่เหมาะสม จากนั้นคลิ๊กปุ่ม “Insert to URL List” เพื่อเพิ่ม URL เข้าไปในไฟล์ “whiteList” หรือเราสามารถลบ URL ออกจากไฟล์ได้โดยเลือกจากรายชื่อที่แสดงในช่อง URL White List View ในช่องของ URL White List View นี้เป็นการนำรายชื่อ URL ที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากไฟล์ “whiteList”มาแสดง เมื่อเพิ่มหรือลบ URL เรียบร้อยแล้ว คลิ๊กปุ่ม “OK” หรือ “Apply” เพื่อบันทึกข้อมูล หรือคลิ๊ก
ปุ่ม “Cancel” เพื่อยกเลิก

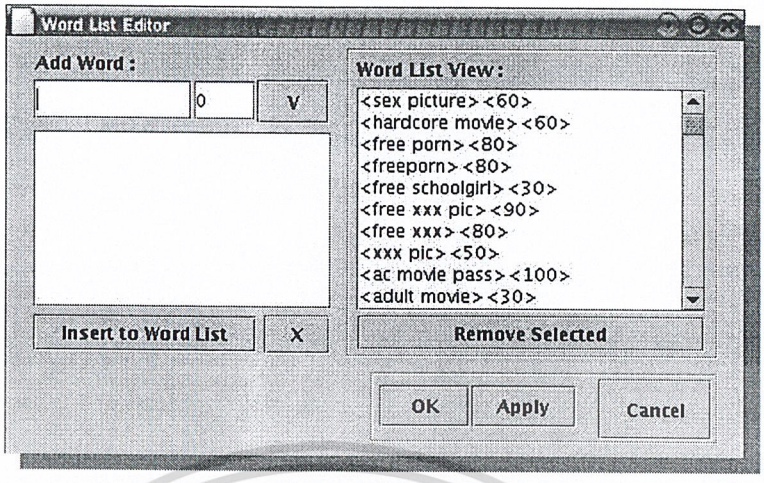


รูปที่ 4.9 แสดงหน้า URL White List Editor

4.2.3 Word

ส่วนนี้ไว้สำหรับเพิ่ม, ลบหรืออัพเดท คำหรือกลุ่มคำที่ไม่เหมาะสมที่ใช้ในการกรองเนื้อหาบนเว็บเพจ จากรูปที่ 4.7 เมื่อคลิ๊กที่ปุ่ม “Word” จะแสดงหน้า Word List Editor ดังรูปที่ 4.10 ในหน้านี้จะมีช่องให้ใส่คำหรือกลุ่มคำที่ไม่เหมาะสม และน้ำหนักของคำหรือกลุ่มคำนั้น จากนั้นคลิ๊กปุ่ม “Insert to Word List” เพื่อเพิ่ม คำหรือกลุ่มคำและน้ำหนักเข้าไปในไฟล์ “wordList” หรือเราสามารถลบคำหรือกลุ่มคำและน้ำหนักออกจากไฟล์ได้โดยเลือกจากรายชื่อที่แสดงในช่อง Word List View ในช่องของ Word List View นี้เป็นการนำคำหรือกลุ่มคำ และน้ำหนักของคำหรือกลุ่มคำจากไฟล์ “wordList” มาแสดง เมื่อเพิ่มหรือลบคำหรือกลุ่มคำเรียบร้อยแล้ว คลิ๊กปุ่ม “OK” หรือ “Apply” เพื่อบันทึกข้อมูล หรือคลิ๊กปุ่ม “Cancel” เพื่อยกเลิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงหน้า Word List Editor

4.3 ประสิทธิภาพของการกรอง

เนื่องจากโปรแกรมจะกรองข้อมูลที่รับมาจากเซิร์ฟเวอร์ก่อน แล้วส่งต่อไปยังไคลเอนต์ ถ้าการกรองใช้เวลานานจะทำให้เครื่องไคลเอนต์ต้องรอเป็นเวลานานด้วยเช่นกัน ดังนั้นเวลาที่ใช้ในการกรองจึงสำคัญมาก ยิ่งใช้เวลาในการกรองน้อยเท่าไรยิ่งดี แต่การกรองเนื้อหาทั้งเว็บเพจนั้นใช้เวลานานพอสมควร จึงเกิดแนวคิดที่จะกำหนดขนาดของข้อมูลที่จะกรอง เพื่อจะได้ใช้เวลาในการกรองให้น้อยลงและทำให้การกรองมีประสิทธิภาพมากขึ้น ประกอบกับเว็บที่ไม่เหมาะสมต่าง ๆ จะมี META TAG ที่บรรจุคำหรือกลุ่มคำที่สอดคล้องกับเว็บไซต์นั้น เพื่อไว้เป็นอินเด็กซ์ (index) สำหรับเว็บไซต์ที่ให้บริการด้าน Search Engine ต่างๆ ซึ่งคำหรือกลุ่มคำเหล่านี้ล้วนเป็นคำหรือกลุ่มคำที่ไม่เหมาะสมทั้งนั้น เราจึงนำ META TAG มาใช้ให้เป็นประโยชน์ โดยโปรแกรมจะกำหนดขนาดของข้อมูลที่จะกรองโดยเริ่มจากส่วนต้นของข้อมูลที่รับมาซึ่งจะบรรจุ META TAG อยู่ การกำหนดขนาดของข้อมูลที่จะกรองนั้นทำให้การกรองมีประสิทธิภาพมากขึ้นและใช้เวลาน้อยลงมาก ดังนั้นจึงไม่จำเป็นที่จะต้องกรองเนื้อหาทั้งเว็บเพจ

ตัวอย่าง META TAG ของ www.ninenine.com ซึ่งเป็นเว็บไซต์ที่ไม่เหมาะสม

<META NAME="description" CONTENT="NineNine. Free Porn. Period.">

<META NAME="keywords" CONTENT="NineNine, free, porn, teen, teens, lesbians, lesbian, video, videos, story, stories, pic, pictures, pics, search, XXX, list, nude, naked, fuck, fucking, sex, cum, porn, boobs, tits, pussy, pussies">

จากตัวอย่างจะเห็นว่าใน META TAG ของเว็บไซต์ที่ไม่เหมาะสมจะบรรจุคำที่ไม่เหมาะสมอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 แสดงเวลาในการเข้าถึงเว็บไซต์ต่างๆ ซึ่งกำหนดจำนวนตัวอักษรที่กรองต่างกัน

	ชื่อเว็บไซต์	1,024 ตัวอักษร	16,384 ตัวอักษร
เว็บทั่วไป	www.sanook.com	15 วินาที	2 นาที
	www.madoo.com	12 วินาที	2 นาที
	www.kapook.com	28 วินาที	3 นาที
เว็บไม่เหมาะสม	www.ninenine.com	7 วินาที	8 วินาที
	www.link-fabulous.com	9 วินาที	8 วินาที
	www.youho.com	11 วินาที	10 วินาที

จากตารางจะเห็นว่าการเข้าถึงเว็บไซต์ที่ไม่เหมาะสมนั้นจะใช้เวลาใกล้เคียงกัน เนื่องจากอัลกอริทึมที่ใช้ในการกรองจะบวกน้ำหนักของคำที่พบไปเรื่อย ๆ จนกว่าน้ำหนักที่พบนั้นจะเกินน้ำหนักที่กำหนดไว้ ดังนั้นเวลาที่ใช้จึงใกล้เคียงกัน เพราะถ้าน้ำหนักเกินกว่าที่กำหนดไว้ก็จะออกจากการกรองทันที แต่การเข้าถึงเว็บไซต์ทั่วไปนั้นเวลาจะต่างกันมาก เนื่องจากเว็บไซต์ทั่วไปมีคำที่ไม่เหมาะสมไม่เกินน้ำหนักที่กำหนด จึงต้องกรองจนกว่าจะจบถึงจะออกจากการกรอง ดังนั้นการที่กำหนดขนาดของข้อมูลที่ใช้กรองจึงเป็นประโยชน์มากสำหรับการเข้าถึงเว็บไซต์ทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

ระบบปิดกั้นเว็บไซต์ที่ไม่เหมาะสมนั้นแบ่งฟังก์ชั่นการทำงานออกเป็นสองส่วนคือ ส่วนของโปรแกรมในการกรองเนื้อหาต่าง ๆ พร้อมทั้งดำเนินการปิดกั้นการเข้าถึงเว็บไซต์ที่ไม่เหมาะสม สามารถ ปิดกั้นเว็บไซต์ที่ไม่เหมาะสมได้อย่างมีประสิทธิภาพ และส่วนติดต่อกับผู้ใช้งานนั้นผู้ใช้สามารถใช้งานได้ง่าย เนื่องจากเป็นรูปแบบของ GUI จึงสามารถแก้ไขข้อมูลที่จะนำมาพิจารณาในการปิดกั้นได้ ไม่ว่าจะเป็น URL หรือ ข้อมูลคำศัพท์ พร้อมมีข้อความอธิบายการใช้งานจึงทำให้เพิ่มความสะดวกสำหรับผู้ใช้งานได้เป็นอย่างดี

5.2 ปัญหาที่เกิดขึ้นระหว่างการทดลอง

5.2.1 การทดสอบทุกครั้งจะต้องเชื่อมโยงกับเครือข่ายอินเตอร์เน็ต

5.2.2 การทดสอบแต่ละครั้งที่ไม่ใช่เวลาเดียวกันอาจทำให้ได้รับผลไม่เท่ากัน เนื่องจากความไม่มีเสถียรภาพของเครือข่ายอินเตอร์เน็ต

5.3 แนวทางการพัฒนาโครงงาน

5.3.1 สามารถกรองเนื้อหาในเว็บเพจที่เป็นรูปภาพได้

5.3.2 สามารถกำหนดค่าน้ำหนักรวมที่จะใช้พิจารณาเพื่อปิดกั้นเว็บไซต์ และกำหนดความละเอียดในการกรองเนื้อหาของเว็บไซต์ได้จาก user interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Ari Luotonen. Web Proxy Server. Prentice Hall. First Edition

[2] Thai Computer Emergency Response Team (ThaiCERT), "ความรู้พื้นฐานเกี่ยวกับไฟร์วอลล์" [online] .Available : http://thaicert.nectec.or.th/paper/firewall/. 2003

[3] สถาบันราชภัฏเชียงราย, "Domain System" [online].Available : http://learning.ricr.ac.th/datacomm/Subjectnew/Less10.html. 2003

[4] A Premier Technical Information Directory, "Hypey Text Transfer Protocol Tutorials" [online].Available : http://www.techtutorials.info/nhttp.html 2003

[5] Great Internet Mersenne Prime Search (GIMPS), "The GNU C Programming Tutorial --> strstr" [online].Available : http://crasseux.com/books/ctutorial/String-library-functions .html. 2003

[6] Christian Chamas, Thierry Lecrog, "Boyer-Moore algorithm" [online].Available : http://www-igm.univ-mlv.fr/~lecroq/string/node14.html. 2003

[7] สันติ ศรีลาศักดิ์, วรวุฒิ เที่ยงธรรม.เจาะประเด็น งานเขียนโปรแกรมบนลีนุกซ์. ออฟเซ็ทเพรส. พิมพ์ครั้งที่ 1, 2542

[8] Sebastien Ailleret, "BannerKiller" [online].Available : http://pauillac.inria.fr/~ailleret/prog/ bannerkiller/index-eng.html. 2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้