บทที่ 2

ทฤษฎีและความรู้พื้นฐาน

ในบทนี้อธิบายถึงแบบจำลองของจดหมายอิเล็กทรอนิกส์ และกล่าวถึงโปรโตคอลต่าง ๆ ที่เกี่ยวข้องกับการรับส่งจดหมาย เช่น พีโอพี และไอเอ็มเอพีโปรโตคอล ซึ่งใช้ในการรับหรืออ่านจดหมาย เอสเอ็มทีพีโปรโตคอลที่ใช้สำหรับการส่งจดหมาย รวมถึงรูปแบบมาตรฐานของเอกสารแบบเออาร์พีเอ อินเตอร์เน็ต และเอ็มไอเอ็มอี ที่สนับสนุนการรับส่งจดหมายที่มีอักขระ 8 บิท

แบบจำลองของจดหมายอิเล็กทรอนิกส์ (Marshall T. Rose, 1993)

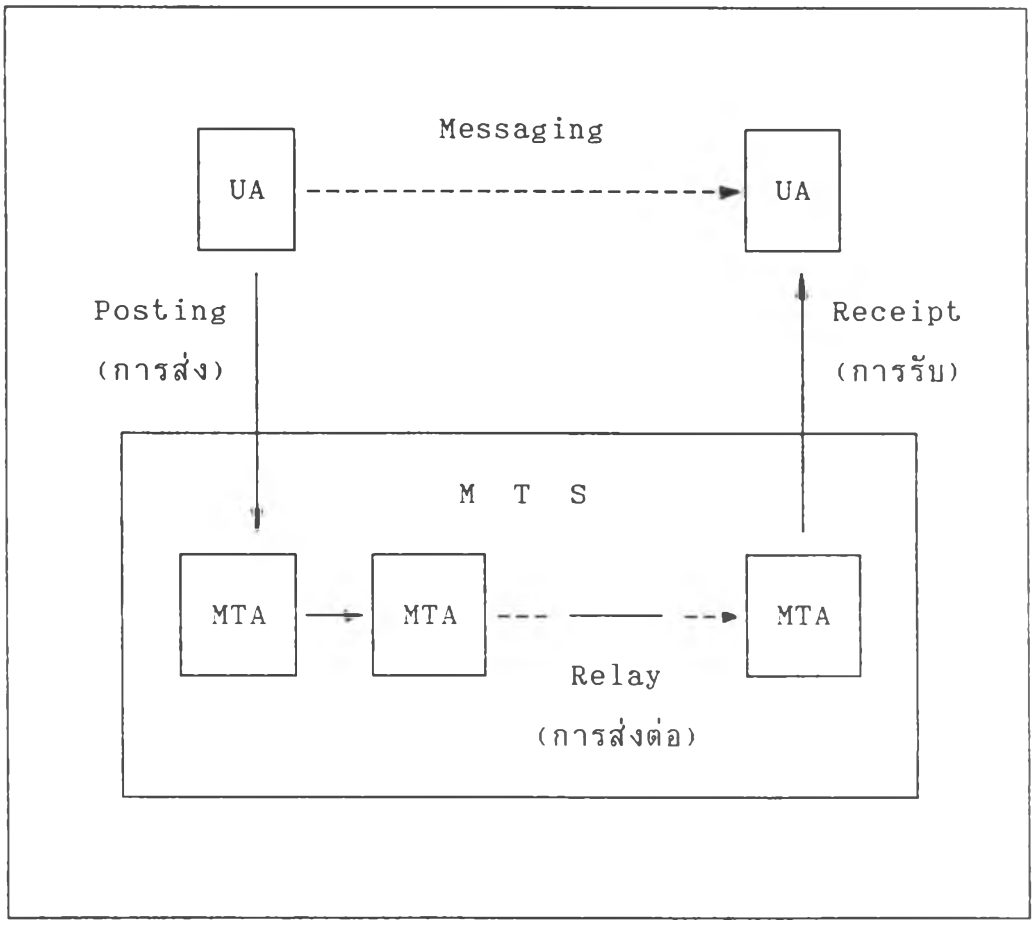
ข้อความ (messages) จะถูกส่งไปโดยระบบส่งข้อมูล (message transfer system หรือ MTS) ซึ่งประกอบไปด้วยตัวแทนระบบส่งข้อมูล (message transfer agent หรือ MTA) จำนวนอย่างน้อย 1 ชุด ระบบส่งข้อมูลจะถูกกระจายออกไป โดยไม่อยู่ภายใต้การจัดการใด ๆ แต่การรวบรวมตัวแทนระบบส่งข้อมูล จะถูกควบคุมภายใต้การจัดการเดียวกัน ณ ที่สุดขอบของระบบ โดยตัวแทนผู้ใช้ (user agent หรือ UA) จะเป็นเสมือนผู้ใช้งาน และติดต่ออยู่กับตัวแทนระบบส่งข้อมูลเฉพาะที่ (local MTA) ดังที่แสดงไว้ในรูป 2.1 แบบจำลองของจดหมายอิเล็กทรอนิกส์

ในมุมมองของระบบส่งข้อมูล ข้อความของเอกสารที่ถูกส่ง จะเรียกว่าเนื้อความ (content) และสารสนเทศที่เกี่ยวข้องที่แนบไปกับข้อความนั้นจะเรียกว่าซองจดหมาย (envelope)

ตามทฤษฎีแล้วระบบส่งข้อมูล ไม่สนใจโครงสร้างของเนื้อความที่ส่งออก แต่เป็นโครงสร้างที่ได้มีการตกลงกันไว้ล่วงหน้า ระหว่างตัวแทนผู้ใช้ แม้ไม่มีข้อกำหนดที่แน่นอนของโครงสร้างนี้ แต่มีรูปแบบที่ใช้ทั่วไป คือ ส่วน

ควบคุมสารสนเทศ เรียกว่าส่วนซองของจดหมาย (header) และส่วนข้อมูลสารสนเทศ เรียกว่าส่วนข้อความของจดหมาย (body) หรือกล่าวอีกนัยหนึ่งคือ

1. ซองจดหมาย จะมีความหมายต่อตัวแทนระบบส่งข้อมูล
2. ส่วนหัวของจดหมาย จะมีความหมายต่อตัวแทนผู้ใช้
3. ส่วนข้อความของจดหมาย จะมีความหมายต่อผู้ใช้งาน (users) ซึ่งอาจเป็นคนหรือโปรแกรม



รูปที่ 2.1 แบบจำลองของจดหมายอิเล็กทรอนิกส์

เมื่อมีการส่งข้อความจากผู้ใช้คนหนึ่งเพื่อไปยังอีกผู้หนึ่ง มีขั้นตอนดังต่อไปนี้

1. เริ่มแรกผู้ส่งต้องกำหนดจุดหมายปลายทางของผู้รับให้แก่ตัวแทนผู้ใช้

2. ตัวแทนผู้ใช้ จะใส่ที่อยู่ (address) ทั้งของผู้ส่งและผู้รับลงบนตัวซองจดหมายและนำส่งผ่านไปที่ ตัวแทนระบบส่งข้อมูล ซึ่งเกี่ยวข้องกับโปรโตคอลของการส่ง โดยจะตรวจสอบความถูกต้องของที่อยู่และรูปแบบของข้อความในเอกสาร

3. เมื่อโปรโตคอลของการส่งทำงานได้สำเร็จ ทางตัวแทนระบบส่งข้อมูลจะเป็นผู้รับช่วงของการส่งเอกสารต่อไป และเมื่อการส่งมีความผิดพลาด จะแจ้งกลับไปยังผู้ส่งเกี่ยวกับข้อความที่ผิดพลาดนั้น

4. เมื่อตัวแทนระบบส่งข้อมูลรับช่วงในการส่งเอกสารต่อแล้ว ต้องพิจารณาว่าสามารถส่งเอกสารนั้นให้แก่ผู้รับโดยตรงได้หรือไม่ โดยทำการส่งเอกสารนั้น ให้แก่ตัวแทนผู้ใช้ของผู้รับ โดยใช้โปรโตคอลของการส่งถึงผู้รับในเครื่องปลายทาง หากไม่สามารถส่งให้แก่ผู้รับโดยตรงได้ จะจัดหาตัวแทนระบบส่งข้อมูลอื่น ที่อยู่ใกล้กับผู้รับเพื่อฝากเอกสารนั้นผ่านต่อไป และทำเช่นนี้ต่อไปจนกว่าสามารถส่งเอกสารไปถึงปลายทางได้ หรือตัวแทนระบบส่งข้อมูลพิจารณาว่าไม่สามารถส่งต่อไปได้อีกแล้ว

เมื่อเอกสารถูกส่งออกไปจากผู้ส่ง ผู้ใช้จะไม่มีสิทธิเหนือเอกสารนั้นอีกต่อไป ระบบส่งข้อมูลจะเป็นผู้รับผิดชอบต่อไปในช่วงเวลาของการส่งออกไปจนถึงเวลาของการส่งถึงผู้รับ

ในการส่งเอกสารนั้น ตัวแทนผู้ใช้ ทั้งของฝ่ายผู้ส่งและผู้รับไม่จำเป็นจะต้องเชื่อมต่อกันโดยตรง (on-line) พร้อมกันทั้ง 2 ฝ่ายตลอดการส่งออก การส่งผ่าน และการส่งถึง ในความเป็นจริง มีเพียงโหนดที่รับผิดชอบต่อเอกสารในขณะนั้น กับโหนดที่จะรับช่วงดูแลต่อไปเท่านั้นที่ต้องเชื่อมต่อถึงกันเพื่อส่งเอกสาร

จะพบว่ามี 3 โปรโตคอลที่เกี่ยวข้อง ดังนี้ (Marshall T. Rose, 1993)

1. โปรโตคอลสำหรับข้อความ (a messaging protocol) ใช้ระหว่างสองตัวแทนผู้ใช้

2. โปรโตคอลการส่งผ่าน (a relay protocol) ใช้ระหว่างสองตัวแทนระบบส่งข้อมูล

3. โปรโตคอลการส่งออกและการส่งถึง (a submission/delivery protocol) ใช้ระหว่างตัวแทนระบบส่งข้อมูล และตัวแทนผู้ใช้

พีโอพีโปรโตคอลเวอร์ชัน 3 (POP version 3) (Marshall T. Rose, 1993)

สำหรับโหนดเล็ก ๆ ในอินเตอร์เน็ต (internet) มักไม่เหมาะที่จะให้มีระบบส่งข้อมูล เช่น เวิคร์สเตชั่น อาจไม่มีเนื้อที่ในดิสค์หรือหน่วยความจำที่เพียงพอเพื่ออนุญาติให้ เอสเอ็มทีพี เซิร์ฟเวอร์ (simple mail transfer protocol (SMTP) server) และระบบส่งจดหมายอิเล็กทรอนิกส์เฉพาะบริเวณที่เกี่ยวข้อง ได้ถูกเก็บและทำงานได้อย่างต่อเนื่องในทำนองเดียวกัน อาจต้องเสียค่าใช้จ่ายสูงเพื่อให้คอมพิวเตอร์ส่วนบุคคลเชื่อมต่อกับเครือข่ายในแบบไอพี (internet protocol (IP)) เป็นเวลานาน ๆ

พีโอพี โปรโตคอล หรือโพสท์ออฟฟิสโปรโตคอล (post office protocol) เป็นระบบที่ช่วยจัดการกับจดหมายอิเล็กทรอนิกส์บนโหนดที่มีขนาดเล็ก และช่วยสนับสนุนตัวแทนผู้ใช้ ให้ผู้ใช้ทำงานได้สะดวก พีโอพี 3 จึงได้มีบทบาทในการอนุญาติให้เวิร์คสเตชั่น สามารถเข้าถึงและนำออกจดหมายอิเล็กทรอนิกส์ที่อยู่บนเซิร์ฟเวอร์โฮส (server host)ได้

การทำงานพื้นฐานของพีโอพีโปรโตคอล เวอร์ชัน 3

เมื่อเชิร์ฟเวอร์หรือโฮส (หมายถึง โฮสที่เสนอการให้บริการพีโอพี 3) เริ่มให้การบริการ พีโอพี 3 นั้นจะรอรับฟังสัญญาณที่ ทีซีพี พอร์ท 110 เมื่อไคล์แอนท์ (client หมายถึง เครื่องที่ทำการขอใช้บริการพีโอพี 3) ต้องการขอใช้บริการ จะทำการสร้างการเชื่อมต่อแบบทีซีพีกับเชิร์ฟเวอร์ เมื่อเชื่อมต่อกันได้แล้ว พีโอพี 3 เชิร์ฟเวอร์จะส่งคำสั่งทักทายไป จากนั้นทั้งไคล์แอนท์และพีโอพี 3 เชิร์ฟเวอร์ จะแลกเปลี่ยนคำสั่งและการตอบรับ จนกระทั่งสิ้นสุดการเชื่อมต่อกัน

คำสั่งของพีโอพี 3 ประกอบด้วยคำหลัก (keyword) และอาจตามด้วยอาร์กิวเมนต์ (argument) ทุกคำสั่งจะจบด้วย 'CRLF' (คือ รหัสสำหรับขึ้นบรรทัดใหม่)

การตอบรับของพีโอพี 3 ประกอบด้วยตัวบอกความสำเร็จ (success indicator) และคำหลัก อาจตามด้วยข้อมูลเพิ่มเติม ทุกการตอบรับจบด้วย 'CRLF' ส่วนตัวบอกความสำเร็จมี บวก ('+OK') และลบ ('-ERR')

การตอบรับนั้นอาจเป็นหลายบรรทัด หลังจากบรรทัดแรกที่ตอบกลับมาตามด้วย 'CRLF' แล้ว บรรทัดถัด ๆ ไปที่จบลงด้วย 'CRLF' จะตามกันมา แต่บรรทัดสุดท้ายจะจบลงด้วย \046 ('.' หรือเรียกว่า termination octet) และ 'CRLF'

กรณีที่การตอบรับจบด้วย 'CRLF.CRLF' ตัวไคล์แอนท์จะตรวจดูว่าบรรทัดนั้นเริ่มต้นด้วย '.' หรือเปล่า และหากมีค่าตามหลังที่ไม่ใช้ 'CRLF' ก็จะยกเลิกคำสิ้นสุดนั้นเสีย หากค่าตามหลังเป็น 'CRLF' แสดงว่าเป็นการสิ้นสุดการตอบรับจากเชิร์ฟเวอร์ และจะไม่นับว่าบรรทัดที่เก็บ '.CRLF' เป็นส่วนหนึ่งของค่าตอบด้วย

เมื่อการเชื่อมต่อกับทีซีพีเริ่มขึ้น และพีโอพี 3 เซิร์ฟเวอร์ได้ส่งคำทักทายมาให้แล้ว เป็นการเข้าสู่สภาวะ authorization โดยในสภาวะนี้ตัวไคล์แอนท์จะแสดงตัวเองต่อพีโอพี 3 เซิร์ฟเวอร์ เมื่อถูกต้องแล้วตัวเซิร์ฟเวอร์ต้องการใช้ทรัพยากร ที่เกี่ยวข้องกับที่เก็บจดหมายอิเล็กทรอนิกส์ของไคล์แอนท์ และเข้าสู่สภาวะทรานส์แซกชั่น (transaction) ซึ่งในสภาวะนี้ ตัวไคล์แอนท์ต้องการ การกระทำจากพีโอพี 3 เซิร์ฟเวอร์ เมื่อไคล์แอนท์สิ้นสุดการถามตอบแล้ว ก็จะเข้าสู่สภาวะอัพเดท (update) ในสภาวะนี้พีโอพี 3 เซิร์ฟเวอร์จะคืนทรัพยากรที่ต้องใช้ในสภาวะทรานส์แซกชั่นคืนกลับการเชื่อมต่อของทีซีพีเป็นอันสิ้นสุด ดังรูปที่ 2.2 แสดงถึงคำสั่งที่ใช้ในพีโอพี 3 และตัวอย่างการทำงานของพีโอพี 3 ได้แสดงไว้ในรูปที่ 2.3

คำสั่งของพีโอพี 3 มีดังนี้

(คำสั่งเบื้องต้น)

ในสภาวะ authorization

USER	name	ระบุผู้ขอใช้บริการไปยังเซิร์ฟเวอร์
PASS	string	ระบุรหัสผ่าน
QUIT		สิ้นสุดการติดต่อ

ในสภาวะ transaction

STAT		ดูสถานะของตู้จดหมาย
LIST	[msg]	ดูสถานะของจดหมายที่เก็บอยู่
RETR	msg	ขอจดหมายอิเล็กทรอนิกส์จากเซิร์ฟเวอร์
DELE	msg	ลบจดหมายอิเล็กทรอนิกส์
NOOP		ไม่มีการทำงานใด ๆ
LAST		บอกจดหมายลำดับสูงสุดในตู้จดหมาย

รูปที่ 2.2 คำสั่งของพีโอพี 3

```
   RSET                    ยกเลิกการลบจดหมายและค่าสุดท้าย

ในสภาวะ update
   QUIT                    สิ้นสุดการทำงาน

(คำสั่งที่เป็น option)
ในสภาวะ authorization
   APOP name password ใช้ algorithm เพื่อแปลงรหัสผ่าน
                           และมีผลเหมือน USER + PASS

ในสภาวะ transaction
   TOP  msg  n             ขอ n บรรทัดแรกของจดหมาย
```

รูปที่ 2.2 (ต่อ) คำสั่งของพีโอพี 3

```
ตัวอย่างการทำงานของ พีโอพี 3

S: <wait for connection on TCP port 110>
C: <open connection>
S: +OK POP3 server ready <189.67052@dbc.mtview.ca.us>
C: USER mrose
S: +OK mrose is a real user
C: PASS secret
```

รูปที่ 2.3 ตัวอย่างการทำงานของพีโอพี 3

```
S: +OK mrose's maildrop has 2 messages (320 octets)
C: STAT
S: +OK 2 320
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S: <the POP3 server sends message 1>
S: .
C: DELE 1
S: +OK message 1 deleted
C: RETR 2
S: +OK 200 octets
S: <the POP3 server sendds message 2>
S: .
C: DELE 2
S: +OK message 2 deleted
C: QUIT
S: +OK dewey POP3 server signing off (maildrop empty)
C: <close connection>
S: <wait for next connection>
```

โดย S: หมายถึงเซิร์ฟเวอร์ และ C: หมายถึงไคล์แอนท์

<ข้อความ> หมายถึงการกระทำขั้นตอนพิเศษ

รูปที่ 2.3 (ต่อ) ตัวอย่างการทำงานของพีโอพี 3

ไอเอ็มเอพีโปรโตคอล เวอร์ชัน 3 (IMAP version 3) (James Rice, 1991)

โดยที่ตัวเมล์เซิร์ฟเวอร์ (mail server) มีหน้าที่เป็นตัวกลางระหว่างผู้ใช้กับแหล่งเก็บข้อความหรือจดหมายอิเล็กทรอนิกส์ และเมล์เลอร์อื่น ๆ โปรโตคอลที่ใช้ในการเข้าถึงเมล์เซิร์ฟเวอร์นั้น จะถูกใช้เพื่อดึงจดหมายเหล่านั้นออกมา ทั้งยังสามารถเปลี่ยนสถานะภาพของจดหมายนั้น ๆ ได้ และจัดการกับตู้จดหมาย (mailbox) ได้ การจัดของโปรโตคอลในลักษณะนี้ทำให้ผู้ใช้และตัวไคล์แอนท์ ไม่จำเป็นต้องรู้ว่าจดหมายเหล่านั้น ถูกจัดเก็บอย่างไรในเมล์เซิร์ฟเวอร์ และยังใช้งานได้กับเซิร์ฟเวอร์อื่น ๆ ได้ด้วย

ไอเอ็มเอพี 3 หรืออินเตอร์แอคทีฟเมล์แอสเซสโปรโตคอล เวอร์ชัน 3 (IMAP หรือ Interactive Mail Access Protocol Version 3) เป็นโปรโตคอลสำหรับการเข้าถึงและจัดการกับจดหมายอิเล็กทรอนิกส์ เมื่อใช้บนทีซีพี ไอเอ็มเอพีเซิร์ฟเวอร์จะรอรับสัญญาณที่ ทีซีพี พอร์ท 220 และมีคุณสมบัติที่สำคัญดังนี้

1. สามารถค้นหาข้อความที่ต้องการ เช่น วันที่ ที่อยู่ (address) สถานะภาพของจดหมาย (status flag) หรือจากเนื้อความของจดหมายบนเครื่องเซิร์ฟเวอร์ก่อนที่จะส่งผ่านมายังไคล์แอนท์ แทนการส่งจดหมายทั้งหมดมาให้ไคล์แอนท์ค้นหาเอง

2. มีคำสั่งสำหรับใช้งานที่สะดวก โดยที่ผู้ใช้ไม่ต้องมีความรู้เกี่ยวกับการจัดเก็บจดหมายอิเล็กทรอนิกส์ตามมาตรฐานของ อาร์เอฟซี 822

3. สนับสนุนการจัดการกับตู้จดหมาย ทั้งยังถูกออกแบบสำหรับการทำงานแบบออนไลน์ โดยให้ความสำคัญกับเรื่องของเรียลไทม์ (real time) และการเข้าถึงแหล่งข้อมูลได้พร้อม ๆ กันในเวลาเดียวกัน

4. ทั้งไคล์แอนท์และเซิร์ฟเวอร์ จะมีความสัมพันธ์ตลอดการเชื่อมต่อของไอเอ็มเอพี และสถานะภาพของจดหมายและที่เก็บจดหมาย จะถูกเก็บไว้ที่เซิร์ฟเวอร์ ซึ่งผู้ใช้สามารถเข้าถึงที่เก็บจดหมายจากเครื่องไคล์แอนท์ ใด ๆ ได้

5. ไอเอ็มเอพีได้ถูกออกแบบให้ครอบคลุมการทำงานทั้งแบบออนไลน์และแบบออฟไลน์

การทำงานของไอเอ็มเอพี 3

การทำงานของไอเอ็มเอพี 3 ประกอบด้วยลำดับของคำถามหรือคำสั่งจากไคล์แอนท์ และการตอบคำถามหรือตอบรับจากเซิร์ฟเวอร์ โดยคำตอบจากเซิร์ฟเวอร์เป็นแบบอะซิงโครนัส (asynchronous) และเป็นอิสระไม่ต้องลำดับคำตอบตามคำถามจากไคล์แอนท์ ทั้งคำถามและคำตอบจะถูกนำหน้าด้วยรหัสจากตัวอักษรหรือตัวเลขที่ไม่ซ้ำกัน เช่น A0001, A0002, A0003,.. ซึ่งเรียกว่าแท็ค (tag) โดยในการตอบคำถามของไคล์แอนท์นั้น เซิร์ฟเวอร์จะให้แท็คที่เหมือนกับคำถามจากไคล์แอนท์ คำสั่งของไอเอ็มเอพีแสดงไว้ในรูปที่ 2.4

โซลิซิทเท็ด (solicited) คือ ข้อมูลหรือคำตอบที่ถูกส่งจากเซิร์ฟเวอร์ เพื่อตอบคำถามของไคล์แอนท์ โดยมีแท็คที่เหมือนกัน ส่วนอันโซลิซิทเท็ด (unsolicited) คือ ข้อมูลหรือคำตอบจากเซิร์ฟเวอร์ ที่ไม่ใช่เพื่อการตอบคำถามของไคล์แอนท์ และจะใช้อักขระ '*' นำหน้า เช่น การตอบรับด้วย 'OK' เมื่อเริ่มต้นการเชื่อมต่อระหว่างไคล์แอนท์และเซิร์ฟเวอร์ เพื่อเป็นการทักทายและรอรับคำถามหรือคำสั่งจากไคล์แอนท์ต่อไป ตัวอย่างการทำงานของไอเอ็มเอพี แสดงไว้ในรูปที่ 2.5 และเซิร์ฟเวอร์จะมีอิสระในการประมวลผลหาคำตอบต่อหลายคำถามของไคล์แอนท์ โดยไม่จำเป็นต้องเรียงลำดับ ดังตัวอย่างที่แสดงไว้ในรูปที่ 2.6

เมื่อเริ่มการเชื่อมต่อระหว่างไคล์แอนท์และเชิร์ฟเวอร์ ไคล์แอนท์จะรอคำทักทายจากเชิร์ฟเวอร์และส่งคำสั่ง 'LOGIN' เพื่อระบุชื่อผู้ขอใช้และรหัสผ่าน เมื่อได้คำตอบ 'OK' จากเชิร์ฟเวอร์แล้ว ไคล์แอนท์จะส่งคำถาม 'SELECT' เพื่อเลือกตู้จดหมายที่ต้องการ โดยมี 'INBOX' เป็นตู้จดหมายประจำสำหรับผู้ขอใช้ทั่วไป และเชิร์ฟเวอร์จะให้คำตอบโดยส่งสถานะภาพของจดหมายบนเชิร์ฟเวอร์ที่ใช้งานได้ จำนวนของจดหมายทั้งหมด และจำนวนของจดหมายใหม่ ตามด้วย 'OK' ส่วนไคล์แอนท์สามารถใช้คำสั่ง 'FETCH' สำหรับเปิดอ่านจดหมายต่อไป

การดึงข้อมูลหรือจดหมาย อาจแบ่งเป็น 3 กลุ่ม ดังนี้

1. ข้อมูลที่เกี่ยวกับจดหมายอิเล็กทรอนิกส์จากที่เก็บจดหมาย

- ข้อมูลภายใน (internal data) เช่น วันและเวลา ที่จดหมายถูกจัดเก็บในตู้จดหมาย
- ขนาดของจดหมายอิเล็กทรอนิกส์
- สถานะภาพของจดหมายอิเล็กทรอนิกส์

2. ข้อมูลที่เกี่ยวกับการส่งถึงผู้รับ เช่น ชื่อผู้ส่ง รายชื่อผู้รับ หมายเลขของจดหมาย และหัวข้อเรื่องของจดหมายอิเล็กทรอนิกส์ ซึ่งเป็นข้อมูลจากส่วนของซองจดหมายอิเล็กทรอนิกส์ (envelope) นั่นเอง

3. ข้อมูลที่เกี่ยวกับเนื้อความของจดหมายอิเล็กทรอนิกส์

```
คำสั่ง :
  tag NOOP
  tag LOGIN user password
  tag LOGOUT
  tag SELECT mailbox
  tag CHECK
  tag EXPUNGE
  tag COPY sequence mailbox
  tag FETCH sequence data
  tag STORE sequence data value
  tag SEARCH criteria
  tag BBOARD bboard
  tag FIND (BBOARDS / MAILBOXES) pattern
  tag READONLY
  tag READWRITE
  tag SELECT.VERSION (major_version minor_version)
  tag SELECT.FEATURE features
  tag SUPPORTED.VERSIONS
  tag FLAGS
  tag SET.FLAGS

การตอบ :
  */tag FLAGS flag_list
  */tag SEARCH (numbers) (criteria)
  */tag EXISTS
```

รูปที่ 2.4 คำสั่งและการตอบของไอเอ็มเอพี 3

```
  */tag RECENT
  */tag EXPUNGE
  */tag STORE data
  */tag FETCH data
  */tag BBOARD bboard_name
  */tag MAILBOX non_inbox_mailbox_name
  */tag SUPPORTED.VERSIONS version_data
  */tag READONLY
  */tag READWRITE
  */tag OK text
  */tag NO text
  */tag BAD text
  */tag BYE text
  */tag FLAGS flag_list
  */tag READONLY

การตอบ : (เฉพาะโซลิซิทเท็ดเท่านั้น)
  tag COPY message_number

การตอบ : (เฉพาะอันโซลิซิทเท็ดเท่านั้น)
  + text
```

รูปที่ 2.4 (ต่อ) คำสั่งและการตอบของไอเอ็มเอพี 3

ตัวอย่างการทำงานของ ไอเอ็มเอพี 3

ไคล์แอนท์	เซิร์ฟเวอร์
	{รอการเชื่อมต่อ}
{เริ่มการเชื่อมต่อ} -->	
	<-- * OK IMAP3 Server Ready
	{รอคำสั่ง}
A001 SUPPORTED.VERSIONS -->	
	<-- * SUPPORTED.VERSIONS ((2 0) (3 0 EIGHT.BIT.TRANSPARENT AUTO.SET.SEEN TAGGED.SOLICITED)) A001 OK Supported versions returned.
	{รอคำสั่ง}
A002 SELECT.VERSION (3 0) -->	
	<-- A002 OK Version 3.0 Selected.
	{รอคำสั่ง}
A002 SELECT.FEATURES TAGGED.SOLICITED -->	
	<-- A002 OK Features Selected.
	{รอคำสั่ง}
A003 LOGIN Fred Secret -->	

รูปที่ 2.5 ตัวอย่างการทำงานของไอเอ็มเอพี 3

```
                          <-- A003 OK User Fred logged
                                   in
                                {รอคำสั่ง}
A004 SELECT INBOX         -->
                          <-- A004 FLAGS (Meeting
                                   Notice \Answered
                                   \Flagged \Deleted
                                   \Seen
                          <-- A004 19 EXISTS
                          <-- A004 2 RECENT
                          <-- A004 OK Select complete
                                {รอคำสั่ง}
A005 FETCH 1:19 ALL       -->
                          <-- A005 1 Fetch (.....)
                                   ...
                          <-- A005 18 Fetch (.....)
                          <-- A005 19 Fetch (.....)
                          <-- A005 OK Fetch complete
                                {รอคำสั่ง}
A006 FETCH 8 RFC822.TEXT -->
                          <-- A006 8 Fetch (RFC822.TEXT
                                        {893}
                              ...893 characters of
                                     text...
                          <-- )
                          <-- A006 OK Fetch complete
```

รูปที่ 2.5 (ต่อ) ตัวอย่างการทำงานของไอเอ็มเอพี 3

```
                                    {รอคำสั่ง}
A007 STORE 8 +Flags \Deleted -->
                          <-- A007 8 Store (Flags
                                   (\Deleted \Seen))
                          <-- A007 OK Store complete
                                   {รอคำสั่ง}
A008 EXPUNGE              -->
                          <-- A008 19 EXISTS
                          <-- A008 8 EXPUNGE
                          <-- A008 18 EXISTS
                          <-- A008 Expunge complete
                                   {รอคำสั่ง}
A009 LOGOUT               -->
                          <-- A009 BYE IMAP3 server
                                   quitting
                          <-- A009 OK Logout complete
{หยุดการเชื่อมต่อ}        -->                <-- {หยุดการเชื่อมต่อ}
                                   {กลับไปจุดเริ่มต้นใหม่}

โดย {ข้อความ} หมายถึงการกระทำขั้นตอนพิเศษ
```

รูปที่ 2.5 (ต่อ) ตัวอย่างการทำงานของไอเอ็มเอพี 3

ตัวอย่างการทำงานของ ไอเอ็มเอพี 3 เมื่อคำสั่งและการตอบไม่พร้อมกัน

ไคล์แอนท์	เซิร์ฟเวอร์
	{รอการเชื่อมต่อ}
{เริ่มการเชื่อมต่อเหมือนตัวอย่างดังรูปที่ 2.5}	
	<-- A004 19 EXISTS
	<-- A004 2 RECENT
	<-- A004 OK Select complete
	{รอคำสั่ง}
A005 SEARCH RECENT	-->
	<-- A005 SEARCH (18 19)
	(RECENT)
	<-- A005 OK Search complete
A006 FETCH 18:19 ALL RFC822.TEXT	
A007 STORE 18:19 +FLAGS (\SEEN)	
A008 FETCH 1:17 ALL	-->
	<-- A006 18 Fetch
	(...RFC822.TEXT)
A009 STORE 18 +FLAGS (\DELETED)	
	<-- A006 19 Fetch
	(...RFC822.TEXT)
	<-- A006 OK Fetch complete
	<-- A007 18 STORE (Flags
	(\Seen))

รูปที่ 2.6 ตัวอย่างการทำงานของไอเอ็มเอพี 3
เมื่อคำสั่งและการตอบไม่พร้อมกัน

```
A010 STORE 19 +FLAGS (\DELETED)
                        <-- A007 19 STORE (Flags
                                 (\Seen))
                        <-- A007 OK Store complete
                        <-- A008 1 Fetch (.........)
                                  ...
                        <-- A008 16 Fetch (.........)
                        <-- A008 17 Fetch (.........)
                        <-- A008 OK Fetch complete
                        <-- A009 18 STORE (Flags
                                 (\Seen \Deleted))
                        <-- A009 OK Store complete
                        <-- A010 19 STORE (Flags
                                 (\Seen \Deleted))
                        <-- A010 OK Store complete
                             (รอคำสั่ง)
                        <-- * EXISTS 23
                        <-- * RECENT 4
                        <-- * SEARCH (20 21 22 23)
                               (RECENT)
A011 FETCH 20:23 ALL RFC822.TEXT
```

รูปที่ 2.6 (ต่อ) ตัวอย่างการทำงานของไอเอ็มเอพี 3
เมื่อคำสั่งและการตอบไม่พร้อมกัน

ในวิทยานิพนธ์นี้ใช้ ไอเอ็มเอพี เวอร์ชัน 2 ด้วยเหตุผลที่ไม่อาจหาไอเอ็มเอพี เวอร์ชัน 3 ได้ แต่เนื่องจากไอเอ็มเอพี เวอร์ชัน 3 เป็น superset กับไอเอ็มเอพี เวอร์ชัน 2 และในการทำวิจัยนี้ไม่ได้ใช้ feature

เกินไอเอ็มเอพี เวอร์ชัน 2 จึงใช้ได้

ข้อแตกต่างระหว่าง พีโอพี และ ไอเอ็มเอพี โปรโตคอล (Terry Gray, 1995)

ในการเข้าถึงและนำออกของจดหมายอิเล็กทรอนิกส์ ที่อยู่บนเซิร์ฟเวอร์ มีด้วยกัน 3 แบบดังนี้

1. แบบออฟไลน์ (offline operation)

เมื่อข้อความหรือจดหมายถูกส่งมาเก็บไว้ที่เซิร์ฟเวอร์ ผู้รับจดหมายจะใช้เครื่องคอมพิวเตอร์ส่วนบุคคล หรือเครื่องเวิร์คสเตชั่น ซึ่งเรียกว่าไคล์แอนท์ โดยจะเชื่อมต่อเข้ากับเซิร์ฟเวอร์ที่เก็บจดหมายอิเล็กทรอนิกส์นั้น และทำการถ่ายโอนจดหมายที่ส่งถึงผู้รับทั้งหมด เพื่อเก็บไว้บนเครื่องไคล์แอนท์ และทำการลบจดหมายเหล่านั้นออกจากเซิร์ฟเวอร์ ดังนั้นจดหมายจะถูกเปิดอ่านหรือจัดการโดยเครื่องไคล์แอนท์

2. แบบออนไลน์ (online operation)

จดหมายอิเล็กทรอนิกส์ทั้งหมด ยังถูกเก็บไว้ที่เซิร์ฟเวอร์ตลอดเวลา และถูกจัดการหรือเปิดอ่าน โดยเครื่องไคล์แอนท์ที่เชื่อมต่อกับเซิร์ฟเวอร์นั้น ซึ่งจดหมายเหล่านั้นอาจถูกเข้าถึงหรือจัดการ โดยเครื่องไคล์แอนท์หลายเครื่องในเวลาเดียวกันหรือต่างเวลากัน

3. แบบดีสคอนเน็กเต็ด (disconnected operation)

เครื่องไคล์แอนท์ที่จัดการกับจดหมายอิเล็กทรอนิกส์นั้น เชื่อมต่อกับเซิร์ฟเวอร์ และทำการเลือกเฉพาะจดหมายที่ต้องการเท่านั้น มาเก็บไว้ในหน่วยความจำชั่วคราว (cache) บนเครื่องไคล์แอนท์ และจะยุติการเชื่อมต่อกับเซิร์ฟเวอร์ แต่จะทำการเชื่อมต่ออีกครั้งภายหลังเมื่อต้องการใช้งาน ส่วนผู้ใช้จะจัดการหรือเปิดอ่านจดหมายนั้น จากหน่วยความจำชั่วคราวบนเครื่องไคล์แอนท์ วิธีนี้จะต่างกับแบบออฟไลน์ ตรงที่ตัวจดหมายยังคงเก็บไว้บนเซิร์ฟเวอร์ ซึ่งภายหลังจะทำการเชื่อมต่ออีก เพื่อเปลี่ยนสถานะภาพของจดหมายระหว่างเซิร์ฟเวอร์ และหน่วยความจำชั่วคราวบนเครื่องไคล์แอนท์ ทั้งนี้ผู้ใช้งานอาจเชื่อมต่อกับเซิร์ฟเวอร์ ด้วยเครื่องไคล์แอนท์ใด ๆ ในเวลาเดียวกันหรือต่างกัน

โปรโตคอลที่ใช้สำหรับจัดการกับจดหมายอิเล็กทรอนิกส์ ที่อยู่บนเซิร์ฟเวอร์ มีดังนี้

1. ดีเอ็มเอสพี (DMSP) หรือ ดีสทรีบิวเต็ดเมล์ซีสเต็มโปรโตคอล (distributed mail system protocol)

โดยที่ดีเอ็มเอสพีได้ถูกเริ่มกำหนดไว้ในเอกสารของ อาร์เอฟซี 984 เมื่อเดือนเมษายน 1986 แต่ไม่ได้รับการสนับสนุนมากนัก เพราะส่วนใหญ่ถูกจำกัดไว้ใช้สำหรับโปรแกรมประยุกต์ชื่อ พีซีเมล์ (PCMail) ซึ่งถูกนำมาใช้แบบดีสคอนเน็กเต็ด

2. พีโอพี (POP) หรือ โพสท์ออฟฟิสโปรโตคอล (post office protocol)

พีโอพีได้ถูกกำหนดไว้ในเอกสารของ อาร์เอฟซี 918 เมื่อเดือนตุลาคม 1984 โดยถูกออกแบบให้สนับสนุนการทำงานแบบออฟไลน์ และ

ได้มีการแก้ไขมาตรฐานนี้จนเป็น พีโอพี 3 ดังปรากฏในเอกสาร อาร์เอฟซี 1725 ทั้งยังได้ปรับปรุงเพื่อให้ทำงานในแบบออนไลน์ และแบบดีสคอนเน็กเต็ด แต่อยู่ภายใต้ข้อจำกัดหลายประการ

3. ไอเอ็มเอพี (IMAP) หรือ อินเตอร์แอคทีฟเมล์แอสเซส โปรโตคอล (interactive mail access protocol)

ได้เริ่มกำหนดตั้งแต่ปี 1986 แต่ถูกบรรจุไว้ในเอกสาร อาร์เอฟซี 1064 เมื่อเดือนกรกฎาคม 1988 โดยถูกออกแบบให้ใช้งานแบบออนไลน์ และได้ถูกพัฒนาให้มีหน้าที่สำหรับครอบคลุมการทำงานของ พีโอพี ในแบบออฟไลน์ และในเวอร์ชัน 4 จะถูกออกแบบให้สนับสนุนการทำงานแบบดีสคอนเน็กเต็ด ของดีเอ็มเอสพี อีกด้วย

สิ่งที่เหมือนกันระหว่าง พีโอพี และ ไอเอ็มเอพี

1. จัดการกับจดหมายอิเล็กทรอนิกส์บนเซิร์ฟเวอร์เท่านั้น ส่วนการส่งจดหมายใช้โปรโตคอลเอสเอ็มทีพี

2. ใช้งานกับจดหมายที่ถูกนำส่งและเก็บไว้ยังเมล์เซิร์ฟเวอร์ ซึ่งเป็นเครื่องเซิร์ฟเวอร์ที่เปิดใช้งานตลอดเวลา และถูกใช้ร่วมกัน

3. สามารถเข้าถึงจดหมายอิเล็กทรอนิกส์ ได้จากเครื่องไคล์แอนท์ใด ๆ

4. สามารถเข้าถึงจดหมายอิเล็กทรอนิกส์ ได้จากทุกที่บนเครือข่ายสื่อสาร

5. สนับสนุนการจัดการจดหมายอิเล็กทรอนิกส์แบบออฟไลน์ คือ การโอนย้ายจดหมายออก และลบทิ้งจากเซิร์ฟเวอร์ได้

6. สนับสนุนการมีเลขหมายประจำจดหมายอิเล็กทรอนิกส์ เพื่อใช้ในการจัดการแบบดีสคอนเน็กเต็ดได้

7. สามารถถูกพัฒนาขึ้นได้ง่าย เพราะมีโปรแกรมต้นฉบับให้

8. มีการพัฒนาขึ้นเพื่อใช้งานอย่างกว้างขวาง บนเครื่องไคล์แอนท์ที่เป็นเครื่องคอมพิวเตอร์ส่วนบุคคล เครื่องคอมพิวเตอร์แบบ Macintosh และบนเครื่องยูนิกซ์

9. มีการพัฒนาขึ้นในเชิงพาณิชย์อย่างกว้างขวาง

10. เป็นโปรโตคอลที่เปิดกว้าง และถูกกำหนดโดยหน่วยงานอินเตอร์เน็ต (Internet IETF)

11. เป็นต้นแบบของโปรโตคอลสำหรับอินเตอร์เน็ต จึงไม่จำเป็นต้องมีเมล์เกตเวย์ (mail gateways)

ข้อดีของ พีโอพี

1. เป็นโปรโตคอลที่ใช้งานง่าย และสะดวกในการพัฒนาโปรแกรมประยุกต์

2. ถูกนำมาใช้พัฒนาเป็นโปรแกรมประยุกต์อย่างกว้างขวางมาก่อน

ข้อดีของ ไอเอ็มเอพี ที่เหนือกว่า พีโอพี

สามารถแบ่งออกได้เป็น 3 ด้าน ดังนี้

1. มีคำสั่งสำหรับจัดการกับจดหมายอิเล็กทรอนิกส์ ในที่รับจดหมายเข้า (inbox) มากกว่า

- สามารถเพิ่มหรือเก็บจดหมายอิเล็กทรอนิกส์ ไว้ที่ตู้เก็บจดหมายใด ๆ ได้
- สามารถกำหนดสถานะภาพของจดหมายอิเล็กทรอนิกส์เองได้
- สนับสนุนการแก้ไขจดหมายอิเล็กทรอนิกส์พร้อม ๆ กันได้

และจะทำการแจ้งให้ผู้ใช้ร่วมกันในขณะนั้นทราบ

- สามารถแจ้งให้ผู้ใช้ทราบ เมื่อมีจดหมายใหม่เข้ามา

2. สนับสนุนการจัดการกับตู้เก็บจดหมายอิเล็กทรอนิกส์หลายชุดได้

- สามารถบริหารตู้เก็บจดหมายอื่น ๆ ได้ นอกจากที่รับจดหมายเข้าเท่านั้น

- สามารถจัดการกับตู้เก็บจดหมายใด ๆ ได้ เช่น การเลือกตู้เก็บจดหมายใด ๆ การสร้างใหม่ การลบทิ้ง และการเปลี่ยนชื่อตู้เก็บจดหมายอิเล็กทรอนิกส์

- สนับสนุนตู้เก็บจดหมายในรูปของลำดับชั้น (hierarchies)

- เหมาะสำหรับการเข้าถึงข้อมูลประเภทอื่น ที่ไม่ใช้จดหมายอิเล็กทรอนิกส์ เช่น เน็ตนิวส์ (netnews)

3. มีคุณสมบัติที่เหมาะสม ตามสภาพของการเชื่อมต่อของเครือข่ายสื่อสาร

- สามารถเลือกพิจารณา เฉพาะโครงสร้างบางส่วนของจดหมายอิเล็กทรอนิกส์ แทนการนำออกมาทั้งหมด

- สามารถเลือกดูเฉพาะบางส่วนของ เอ็มไอเอ็มอี (MIME) ได้

- การค้นหาจดหมายอิเล็กทรอนิกส์ จะทำการค้นหาบนเซิร์ฟเวอร์เอง เพื่อลดปริมาณการส่งผ่านของข้อมูลทั้งหมดบนเครือข่ายสื่อสาร

นอกจากนี้ยังมีไลบรารี่ (libraries) ของไอเอ็มเอพี ที่สนับสนุนการพัฒนาเป็นโปรแกรมประยุกต์

เอ็มไอเอ็มอี (MIME) (Borenstein & Freed, 1993)

มาตรฐานที่กำหนดในเอกสารอาร์เอฟซี 822 ที่เกี่ยวกับรูปแบบของจดหมายอิเล็กทรอนิกส์ จะระบุรายละเอียดของส่วนหัวจดหมายเท่านั้น โดยกำหนดให้ส่วนเนื้อความของจดหมาย เป็นเพียงชุดของตัวอักษรแบบแอสกี้ที่ใช้เป็นรหัสแบบ 7 บิทเท่านั้น ทำให้ไม่สามารถส่งข้อความที่ใช้รหัสแบบ 8 บิท เช่น ภาษาไทยได้

เอ็มไอเอ็มอี หรือ มัลติเพอร์โพสอินเตอร์เน็ตเมล์เอ็กซ์เทนชั่น (multipurpose internet mail extensions) ได้ถูกเสนอขึ้น เพื่อให้สามารถส่งข้อความจดหมายที่เป็นรหัสแบบ 8 บิทได้ โดยกำหนดไว้ในเอกสารอาร์เอฟซี 1521 โดยเพิ่มกลไกนอกเหนือจากที่มีการระบุตามเอกสารอาร์เอฟซี 822 ดังนี้

1. เพิ่มประโยค 'MIME-Version' ไว้กับส่วนซองของจดหมาย โดยใช้หมายเลขของเวอร์ชันเป็นตัวกำกับมาตรฐานที่ใช้ตามเอ็มไอเอ็มอี เช่น เวอร์ชัน 1.0 เป็นไปตามมาตรฐานของ อาร์เอฟซี 1521 ให้แก่ตัวแทนผู้ใช้ที่มีการสนับสนุนวิธีของเอ็มไอเอ็มอี ดังตัวอย่างในรูปที่ 2.7

รูปแบบของประโยค

Version := "MIME-Version" ":" 1*DIGIT "." 1*DIGIT

ตัวอย่าง MIME-Version : 1.0

รูปที่ 2.7 รูปแบบของประโยค 'MIME-Version'

2. เพิ่มประโยค 'Content-Type' ไว้กับส่วนของของจดหมาย เพื่อระบุประเภทของส่วนเนื้อจดหมาย ทั้งยังมีการระบุประเภทย่อย ๆ ของเนื้อความ เพื่อระบุรูปแบบของข้อมูล นอกจากนี้ยังมีพารามิเตอร์ เพื่อเป็นส่วนขยายของประเภทย่อยอีกด้วย ทั้งนี้ก็เพื่อบอกตัวแทนผู้ใช้ให้เลือกกลไกที่เหมาะสมทั้งการรับและส่งข้อความเหล่านั้น ดังตัวอย่างในรูปที่ 2.8

รูปแบบของประโยค

```
content := "Content-Type" ":" type "/" subtype
           *(";" parameter)
           ; case-insensitive matching of type and -
           ; subtype
type    :=  "application" / "audio"
          / "image"       / "message"
          / "multipart"   / "text"
          / "video"       / "extension-token"
          ; All values case-insensitive
extension-token := x-token / iana-token
iana-token := <a publicly-defined extension token,
               registered with IANA>
x-token := <The two characters "X-" or "x-" followed,
            with no intervening white space, by any
            token>
subtype   := token ; case-insensitive
parameter := attribute "=" value
attribute := token ; case-insensitive
```

รูปที่ 2.8 รูปแบบของประโยค 'Content-Type'

```
value      := token / quoted-string
token      := 1*<any (ASCII) CHAR except SPACE, CTLs,
                 or tspecials>
tspecials := "(" / ")" / "<" / ">" / "@"
           / "," / ";" / ":" / "\" / <">
           / "/" / "[" / "]" / "?" / "="
           ; Must be in quoted-string,
           ; to use within parameter values
```

ตัวอย่าง

```
Content-Type: text/plain; charset=us-ascii
```

รูปที่ 2.8 (ต่อ) รูปแบบของประโยค 'Content-Type'

ประเภทต่าง ๆ ของประโยค 'Content-Type' มีดังนี้

- ประเภท 'text'

ใช้กับสารสนเทศแบบข้อความทั่วไป ซึ่งอาจเป็นกลุ่มตัวอักษรชุดใด ๆ (character set) และมี 'plain' เป็นประเภทย่อย รูปแบบของข้อความประเภทนี้ ไม่ต้องใช้ซอฟท์แวร์พิเศษเพื่อช่วยในการอ่าน นอกจากการชี้ไปยังตารางของกลุ่มตัวอักษรชุด ตามที่ระบุไว้ในพารามิเตอร์ เช่น us-ascii หรือ ISO-8859

- ประเภท 'multipart'

ใช้กับส่วนเนื้อข้อความ ที่ประกอบด้วยข้อความหลายประเภทต่างชนิดกัน แต่ต้องการให้รวมเป็นข้อความเพียงชุดเดียว โดยมีประเภทย่อยพื้นฐานคือ 'mixed' นอกจากนี้ยังมี 'alternative' สำหรับข้อความที่เหมือนกัน แต่แสดงด้วยรูปแบบที่ต่างกัน 'parallel' สำหรับข้อความที่ต้องการเปิดอ่าน หรือใช้งานพร้อม ๆ กัน และ 'digest' สำหรับข้อความที่แต่ละส่วนเป็นข้อความประเภท 'message'

- ประเภท 'message'

ใช้กับส่วนเนื้อข้อความที่ภายในก็เป็นตัวจดหมายเอง โดยมีประเภทย่อยพื้นฐานคือ 'rfc822' ทั้งยังมีประเภทย่อย 'partial' สำหรับส่วนย่อยของข้อความที่ถูกแบ่งออกเนื่องจากมีขนาดใหญ่มาก และประเภทย่อย 'External-body' สำหรับอ้างถึงเนื้อความจากแหล่งภายนอก

- ประเภท 'image'

ใช้สำหรับส่วนเนื้อข้อความที่เป็นรูปภาพ ซึ่งจำเป็นต้องมีอุปกรณ์ช่วยในการแสดงผล เช่น จอภาพกราฟฟิค เครื่องพิมพ์ และเครื่องโทรสาร และมีประเภทย่อย คือ 'jpeg' และ 'gif'

- ประเภท 'audio'

ใช้สำหรับส่วนเนื้อข้อความที่เป็นรูปเสียงหรือคำพูด ซึ่งจำเป็นต้องมีอุปกรณ์ช่วยในการแสดงผล เช่น ลำโพง หรือโทรศัพท์ โดยมีประเภทย่อย 'basic'

- ประเภท 'video'

ใช้สำหรับส่วนเนื้อข้อความ ที่เป็นรูปภาพที่มีการเคลื่อนไหว ซึ่งจำเป็นต้องมีทั้งฮาร์ดแวร์และซอร์ฟแวร์ช่วยในการแสดงผล และมีประเภทย่อย คือ 'mpeg'

- ประเภท 'application'

ใช้สำหรับการส่งข้อมูลที่เป็นโปรแกรมประยุกต์ หรือข้อมูลที่เป็นแบบไบนารีผ่านทางจดหมายอิเล็กทรอนิกส์ โดยมีประเภทย่อย 'octet-stream' สำหรับข้อมูลแบบไบนารี และ 'PostScript' สำหรับข้อมูลที่เป็นโพสท์สคริปท์

3. เพิ่มประโยค 'Content-Transfer_Encoding' ไว้กับส่วนซองของจดหมาย เพื่อระบุการเข้ารหัสเพิ่มเติม สำหรับใช้ส่งผ่านข้อความไปยังกลไกการส่งที่มีข้อจำกัดด้านรูปแบบของข้อมูลและกลุ่มของตัวอักษร แม้นว่าประโยค 'Content-Type' จะสามารถส่งผ่านข้อความในรูปแบบของข้อมูล 8 บิท หรือแบบไบนารีก็ตาม แต่ยังไม่สามารถส่งผ่านไปยังโปรโตคอลของการส่งได้เช่น ตามมาตรฐาน อาร์เอฟซี 821 จะกำหนดการส่งในรูปแบบของข้อมูล 7 บิทเท่านั้น และต้องมีความยาวไม่เกิน 1,000 ตัวอักษร ดังนั้นจึงต้องมีการเข้ารหัสข้อมูลนั้นอีกครั้ง โดยมีกลไกการเข้ารหัสล่วงหน้า คือ '7bit'

ทั้งการเข้ารหัสแบบ '7bit', '8bit' และ 'binary' จะไม่มีการเข้ารหัสจริงในการส่งผ่าน ส่วนการเข้ารหัสแบบ 'quoted-printable' นั้น รหัสตัวอักษรที่มีค่าเกินกว่า 127 จะถูกแปลงให้เป็นเลขฐาน 16 และมีเครื่องหมาย '=' อยู่ข้างหน้า และการเข้ารหัสแบบ 'base64' จะทำการเข้ารหัสข้อความทุก 24 บิท หรือตัวอักษรที่มีรูปแบบเป็น 8 บิท จำนวน 3 ตัวอักษรเรียงกัน ให้เป็นชุดของตัวอักษรที่มีขนาด 6 บิท จำนวน 4 ตัวอักษร

เรียงจากซ้ายไปขวา รูปแบบและตัวอย่างของประโยคนี้ แสดงไว้ในรูปที่ 2.9

รูปแบบของประโยค

```
encoding := "Content-Transfer-Encoding" ":" machanism
mechanism :=  "7bit"   ; case-insensitive
            / "quoted-printable"
            / "base64"
            / "8bit"
            / "binary"
            / x-token
```

รูปที่ 2.9 รูปแบบของประโยค 'Content-Transfer-Encoding'

4. เพิ่มประโยค 'Content-ID' และ 'Content-Description' จะใช้สำหรับเมื่อต้องการอ้างอิงระหว่างส่วนเนื้อจดหมาย และเมื่อต้องการกำกับคำอธิบายไว้ ตามลำดับ แต่ประโยคทั้ง 2 นี้อาจละไว้ก็ได้ รูปแบบของประโยคนี้ แสดงไว้ในรูปที่ 2.10

รูปแบบของประโยค

```
  id := "content-ID" ":" msg-id
  description := "Content-Description" ":" *text
```

รูปที่ 2.10 รูปแบบของประโยค 'Content-ID'
และ 'Content-Description'

รูปแบบมาตรฐานของเอกสารแบบเออาร์พีเอ อินเตอร์เน็ต (ARPA Internet)

มาตรฐานนี้เป็นการกำหนดรูปแบบของเอกสาร (text messages) ที่ใช้ส่งระหว่างผู้ใช้คอมพิวเตอร์ในลักษณะของจดหมายอิเล็กทรอนิกส์ (David H. Crocker, 1982)

เอกสารนั้นถูกมองเป็นรูปแบบของซองจดหมาย และตัวเนื้อความ โดยที่ซองจดหมายนั้น ประกอบไปด้วยข้อมูลที่จำเป็นสำหรับใช้ในการส่ง (transmission) และการมอบ (delivery) ส่วนข้อความในเอกสาร ประกอบด้วยข่าวสารหรือเนื้อความที่ต้องการมอบให้แก่ผู้รับปลายทาง

รูปแบบของข้อกำหนดใช้ในรูปของ Backus-Naur Form (BNF) แต่มีส่วนเพิ่มคือ repetition และ "Local" alternative รายละเอียดมีดังรูปที่ 2.11

Backus-Naur Form (BNF)

1. RULE NAMING
   ชื่อนั้นจะเป็นชื่อเต็ม ไม่มีเครื่องหมาย "<" และ ">" ประกอบอยู่ และเป็นอักษรตัวใหญ่ (Uppercase)
2. RULE1 / RULE2 : ALTERNATIVES
   เป็นการให้เลือกอย่างใดอย่างหนึ่ง
3. (RULE1 RULE2) : LOCAL ALTERNATIVES
   Element ภายในวงเล็บถือเป็นเพียง 1 Element

รูปที่ 2.11 Backus-Naur Form (BNF)

4. *RULE : REPETITIVE

   รูปแบบเต็มคือ <l>*<m>element หมายถึงจำนวน element เกิดขึ้นได้อย่างน้อย <l> และอย่างมาก <m> เมื่อไม่มีการให้ค่าแก่ <l> และ <m> หมายถึง zero และ Infinity ตามลำดับ

5. [RULE] : OPTIONAL

   เป็นตัวเลือก

6. NRULE : SPECIFIC REPETITION

   เป็นการระบุจำนวน element ที่แน่นอนคือ <n>(element) ซึ่งมีค่าเหมือน <n>*<n>(element)

7. #RULE : LISTS

   รูปแบบเต็มคือ <l>#<m>element หมายถึงจำนวน element อย่างน้อย <l> และอย่างมาก <m> ที่ถูกขั้นด้วย "," จำนวน 1 ตัวหรือมากกว่า

8. ; COMMENT

   โดยเริ่มจากหลังเครื่องหมาย ";" จนสุดบรรทัด

รูปที่ 2.11 (ต่อ) Backus-Naur Form (BNF)

รูปแบบของเอกสารที่ใช้ในจดหมายอิเล็กทรอนิกส์อาจแบ่งเป็น 2 ส่วน คือ ส่วนหัวและส่วนข้อความ ซึ่งถูกคั่นด้วย blank line รายละเอียดหรือฟิลด์ภายในของส่วนหัว อาจไม่ต้องเรียงลำดับก่อนหลังได้ ยกเว้นแต่ส่วนข้อความนั้น ต้องอยู่หลังจากส่วนหัวเสมอ แต่ได้ถูกแนะนำให้ฟิลด์ในส่วนหัวมีการเรียงลำดับไว้

รายละเอียดของส่วนหัว (Header)

```
message    = field *(CRLF *text) ; Everything after
                                 ;  first null line
                                 ;  is message body
fields     =   dates             ; Creation time,
               source            ;  author id & one
             1*destination       ;  adddress required
              *optional-field    ;  others optional
source     = [ trace ]           ; net traversals
               originator        ; original mail
             [ resent ]          ; forwarded
trace      =   return            ; path to sender
             1*received          ; receipt tags
return     = "Return-path" ":" route-addr
                                 ; return address
received   = "Received"    ":"   ; one per relay
                [["from" domain]  ; sending host
                 ["by"   domain]  ; receiving host
                 ["via"  atom]    ; physical path
                *("with" atom)    ; link/mail protocol
                 ["id"   msg-id]  ; receiver msg id
                 ["for"  addr-spec] ; initail form
              ";"        date-time  ; time received
originator  =  authentic         ; authenticated addr
                  ["Reply-To" ":" 1#address])
```

รูปที่ 2.12 รายละเอียดของส่วนหัวของเอกสาร

```
authentic    =  "From"    ":"    mailbox
                                  ; Single author
             / ("Sender" ":"    mailbox
                                  ; Actual submittor
                "From"    ":" 1#mailbox)
                                  ; Multiple authors
                                  ;  or not sender
resent       =  resent-authentic
                   ["Resent-Reply-To"  ":" 1#address])
resent-authentic
             =  "Resent-From"       ":"    mailbox
             / ("Resent-Sendder"    ":"    mailbox
                "Resent-From"       ":" 1#mailbox
dates        =   orig-date         ; Original
               [ resent-date ]     ; Forwarded
orig-date    = "Date"                ":"    date-time
resent-date  = "Resent-Date"         ":"    date-time
destination  = "To"          ":" 1#address ; Primary
             / "Resent-To"   ":" 1#address
             / "cc"          ":" 1#address ; Secondary
             / "Resent-cc"   ":" 1#address
             / "bcc"         ":"  #address ; Blind cc
             / "Resent-bcc"  ":"  #address
```

รูปที่ 2.12 (ต่อ) รายละเอียดของส่วนหัวของเอกสาร

```
optional-field =
        / "Message-ID"        ":"   msg-id
        / "Resent-Message-ID" ":"   msg-id
        / "In-Reply-To"       ":"  *(phrase / msg-id)
        / "References"        ":"  *(phrase / msg-id)
        / "Keywords"          ":"  #phrase
        / "Subject"           ":"  *text
        / "Comments"          ":"  *text
        / "Encrypted"         ":" 1#2word
        / extension-field          ; To be definedd
        / user-defined-field       ; May be pre-empted
msg-id        = "<" addr-spec ">" ; Unique message id
extension-field
              = <Any field which is defined in a
                 document published as a formal
                 extension to this specification;
                 none will have names beginning with
                 the string "X-">
user-defined-field
              = (Any field which has not been defined
                 in this specification or published
                 as an extension to this
                 specification; names for such fields
                 must be unique and may be pre-empted
                 by published extensions>
```

รูปที่ 2.12 (ต่อ) รายละเอียดของส่วนหัวของเอกสาร

เอสเอ็มทีพี (SMTP) (Jonathan B. Postel, 1982)

เอสเอ็มทีพี หรือ ซิมเพิลเมล์ทรานส์เฟอร์โปรโตคอล (simple mail transfer protocol) เป็นโปรโตคอลที่ใช้สำหรับการส่งผ่าน คือ ส่งผ่านตัวซองจดหมายและเนื้อความในจดหมาย ในส่วนของซองจดหมายนั้น บรรจุส่วนที่สำคัญดังนี้

1. ที่อยู่ของผู้ส่ง (originator) ซึ่งเป็นตัวสร้างซองจดหมายนี้
2. ที่อยู่ของผู้รับ ซึ่งอาจมีมากกว่า 1 รายก็ได้ และระบุโหมดของการส่งมอบด้วย

ทั้งนี้ ที่อยู่ของซองจดหมาย อาจไม่มีความสัมพันธ์กับที่อยู่ของผู้รับที่ปรากฏบนส่วนหัวของจดหมาย และในความเป็นจริง ในส่วน To: บนส่วนหัวของจดหมายอาจไม่ได้ระบุที่อยู่ของผู้รับได้ด้วย

โหมดต่าง ๆ ของการส่งมอบของเอสเอ็มทีพี มีดังต่อไปนี้

1. การส่งมอบจดหมายไปยังตู้เก็บจดหมายของผู้รับ
2. แสดงข้อความของจดหมายนั้นบนหน้าจอของผู้รับ ซึ่งขณะนั้นกำลังทำงานอยู่
3. หากผู้รับกำลังทำงานอยู่ ให้แสดงข้อความของจดหมายนั้น หรือให้เก็บจดหมายไว้ในตู้เก็บจดหมายของผู้รับ
4. ส่งมอบจดหมายไว้ในตู้เก็บจดหมายของผู้รับ และหากขณะนั้นผู้รับกำลังทำงานอยู่ ให้เลือกแสดงข้อความในจดหมาย หรือให้เก็บไว้ตู้เก็บจดหมาย

การแสดงข้อผิดพลาด (Error Report)

การแสดงข้อผิดพลาดเกิดขึ้นได้ก็ต่อเมื่อ ตัวแทนระบบส่งข้อมูลพิจารณาว่าไม่สามารถส่งผ่าน หรือส่งมอบเอกสารให้แก่ผู้รับ 1 รายหรือมากกว่าได้ และจะสร้างซองจดหมาย และข้อความของเอกสารขึ้นใหม่ โดยซองจดหมายใหม่นี้ไม่มีชื่อผู้ส่ง มีชื่อผู้รับเพียง 1 ราย และที่อยู่ของผู้ส่งจากซองจดหมายเดิม

ส่วนข้อความของเอกสารประกอบด้วย ข้อความแสดงข้อผิดพลาด ส่วนของ From: จะเป็นที่อยู่ของตัวแทนระบบส่งข้อมูล ที่แจ้งข้อผิดพลาด ส่วนของ To: จะระบุที่อยู่ของผู้ส่งจากซองจดหมายเดิม รวมทั้ง Date: นอกจากฟิลด์เหล่านี้แล้ว อาจมีการนำข้อความของจดหมายเดิมมารวมได้ด้วยกัน

การทำงานของเอสเอ็มทีพี โปรโตคอล

โดยที่เอสเอ็มทีพีเซิร์ฟเวอร์คอยรับสัญญาณที่ ทีซีพี พอร์ต 25 เมื่อไคล์แอนท์ขอการสร้างการเชื่อมต่อทีซีพีกับเอสเอ็มทีพีเซิร์ฟเวอร์ ไคล์แอนท์ต้องคอยการตอบรับจากเซิร์ฟเวอร์ และเมื่อเซิร์ฟเวอร์ทำการตอบรับและส่งสถานะภาพของตัวแทนระบบส่งข้อมูลเฉพาะที่ ทำให้ทั้งคู่ได้สถานะภาพที่สมบูรณ์ ไคล์แอนท์จะแสดงตัวและเริ่มการส่งเอสเอ็มทีพี ทรานส์แซคชั่น ;เพื่อตอบรับกัน และเมื่อสิ้นสุด จะส่งคำสั่งให้ยุติการติดต่อ และคอยการตอบรับจากเซิร์ฟเวอร์ เมื่อจบการเชื่อมต่อทีซีพี ทางไคล์แอนท์ปิดการติดต่อทีซีพีเช่นกัน รูปแบบทั่วไปของเอสเอ็มทีพี แสดงไว้ดังรูปที่ 2.13 ส่วนตัวอย่างการทำงานของเอสเอ็มทีพีแสดงไว้ดังรูปที่ 2.14

รูปแบบของเอสเอ็มทีพี

```
; case-insensitive matching of srting-literals
command  ::= "HELO "         domain          CRLF
           / "MAIL FROM: "   route-addr      CRLF
           / "MAIL FROM: "   "<>"            CRLF
           / "RCPT TO: "     route-addr      CRLF
           / "DATA"                          CRLF
           / "NOOP"                          CRLF
           / "RSET"                          CRLF
           / "QUIT"                          CRLF
           / optional
optional ::= "SEND FROM: "   route-addr      CRLF
           / "SOML FROM: "   route-addr      CRLF
           / "SAML FROM: "   route-addr      CRLF
           / "VRFY "         string          CRLF
           / "EXPN "         string          CRLF
           / "TURN"                          CRLF
           / "HELP"     [" " string]         CRLF
response ::= *(code ["-"][" " *text] CRLF)
               code       [" " *text] CRLF
code     ::= ("1" / "2" / "3" / "4" / "5")
             ("0" / "1" / "2" /     / "5")
             1DIGIT
string   ::= 1*(stext / quoted-pair)
```

รูปที่ 2.13 รูปแบบของเอสเอ็มทีพี

```
stext      ::= <any character, not including space or
                specials>
text       ::= <any character, including bare CR and
                bare LF, but not including CRLF>
```

รูปที่ 2.13 (ต่อ) รูปแบบของเอสเอ็มทีพี

รหัสของการตอบรับ

รหัสที่ตอบรับประกอบด้วยเลข 3 หลักดังนี้

หลักที่หนึ่ง (completion) เป็นการระบุความสำเร็จของคำสั่ง

(1) positive preliminary เซิร์ฟเวอร์พร้อมที่จะทำตามคำสั่ง แต่รอการยืนยันจากไคล์แอนท์

(2) positive completion เซิร์ฟเวอร์ทำตามคำสั่งได้สำเร็จ

(3) positive intermediate เซิร์ฟเวอร์พร้อมที่จะทำตามคำสั่ง แต่ต้องรอข้อมูลเพิ่มเติมจากไคล์แอนท์

(4) transient negative เซิร์ฟเวอร์ไม่สามารถทำตามคำสั่งได้สำเร็จ เนื่องจากเกิดปัญหาชั่วคราว ภายหลังไคล์แอนท์สามารถส่งคำสั่งมาใหม่ได้

(5) permanent negative เซิร์ฟเวอร์ไม่สามารถทำตามคำสั่งได้สำเร็จ เนื่องจากเกิดปัญหาอย่างถาวร ภายหลังไคล์แอนท์สามารถส่งคำสั่งมาใหม่ได้ แต่ต้องเปลี่ยนอาร์กิวเมนต์

หลักที่สอง (category) เป็นการระบุเหตุผลของความสำเร็จหรือความผิดพลาด

(0) syntax เป็นสาเหตุจากไวยกรณ์ของคำสั่งไม่ถูกต้อง

(1) informational เป็นสาเหตุจากข้อมูลจากคำสั่งของไคล์แอนท์

(2) connections เป็นสาเหตุจากการบริการเครือข่าย

(5) application-specific เป็นสาเหตุจากตัวแทนระบบส่งข้อมูล (MTA)

หลักที่สาม (instance) ใช้แยกความแตกต่างเมื่อ หลักที่หนึ่งและหลักที่สองเหมือนกัน

ตัวอย่างการทำงานของเอสเอ็มทีพี

```
S:  {wait for connection onTCP port 25}
C:  {Open connection to server}
S:  220 dbc.mtview.ca.us SMTP service ready
C:  HELO baiji.dbc.mtview.ca.us
S:  250 dbc.mtview.ca.us say hello to baiji.dbc.mtvi
    ew.ca.us
C:  MAIL FROM: <mross@dbc.mtview.ca.us>
S:  250 sender ok
C:  RCPT TO: <nsb@thumper.bellcore.com>
S:  250 recipient ok
```

รูปที่ 2.14 ตัวอย่างการทำงานของเอสเอ็มทีพี

```
C:  RCPT TO: <galvin@tis.com>
S:  250 recipient ok
C:  RCPT TO: <pvm@darpa.mil>
S:  250 recipient ok
C:  DATA
S:  354 Enter mail, end with "." on a line by itself
C:  To: Reviewers: ;
C:  Subject: Want to review a book?
C:  Date: Sun, 12 Apr 1992 11:50:40 -0700
C:  Message-ID: <18597.703104640@dbc.mtview.ca.us>
C:  From: Marshall Rose <mrose@dbc.mtview.ca.us>
C:
C:  As you may have heard, I am undertaking the ....
    <<in the interests of brevity, the majority of
    the body of the original message has been
    removed in this figure ... >>
C:  /mtr
C:  .
S:  250 message sent
c:  QUIT
S:  221 dbc.mtview.ca.us closing connection
C:  <closes connection>
S:  <closes connection>
```

โดย S: หมายถึงเซิร์ฟเวอร์ และ C: หมายถึงไคลแอนท์
{ข้อความ} หมายถึงการกระทำขั้นตอนพิเศษ

รูปที่ 2.14 (ต่อ) ตัวอย่างการทำงานของเอสเอ็มทีพี