บทที่ 2

ทฤษฎีและเอกสารที่เกี่ยวข้อง

ปัจจุบันมีการใช้งานโปรแกรมประยุกต์ต่างๆบนเครือข่ายคอมพิวเตอร์ที่ทำงานบนระบบปฏิบัติการวินโดวส์กันอย่างมากมาย ซึ่งโปรแกรมเหล่านี้ส่วนใหญ่พัฒนาขึ้นมาจากการใช้วินโดวส์ซ็อคเก็ตและใช้ชุดโปรโตคอล TCP/IP เป็นโปรโตคอลหลักในการติดต่อผ่านเครือข่าย ในบทนี้กล่าวถึงหลักการสำคัญที่เกี่ยวกับวินโดวส์ซ็อคเก็ตและชุดโปรโตคอล TCP/IP ในส่วนที่จำเป็นและเกี่ยวข้องกับการจัดทำงานวิจัยนี้ดังรายละเอียดในหัวข้อ 2.1-2.3

2.1 วินโดวส์ซ็อคเก็ต

วินโดวส์ซ็อคเก็ตหรือวินซ็อค (Windows Sockets, WINSOCK) เป็นโปรแกรมช่องทางแบบเปิดในการติดต่อไปยังเครือข่ายสำหรับใช้ในการพัฒนาโปรแกรมประยุกต์ต่างๆ บนเครือข่ายภายใต้ระบบปฏิบัติการวินโดวส์

คำว่าช่องทางแบบเปิดในที่นี้มีความหมายเช่นเดียวกันกับระบบเปิดอื่นๆ กล่าวคือ ข้อกำหนด (Specifications) ของวินซ็อคนี้จะถูกสร้างขึ้นโดยความร่วมมือกันของกลุ่มบริษัทที่พัฒนาซอฟต์แวร์ระบบเครือข่าย โดยข้อกำหนดนี้จะประกอบไปด้วยแฟ้มข้อมูลเฮดเดอร์ (Header File) และโปรแกรมไลบรารี (Library Program) ซึ่งสามารถนำมาใช้งานได้โดยไม่ต้องเสียค่าใช้จ่ายและค่าลิขสิทธิ์ใดๆ เพื่อให้ผู้พัฒนาโปรแกรมสามารถใช้พัฒนาโปรแกรมสำหรับติดต่อกันระหว่างเครื่องคอมพิวเตอร์บนเครือข่ายได้

วินซ็อคเอพีไอ (WINSOCK API, Windows Sockets API, Windows Sockets Application Programming Interface, WSA) ประกอบไปด้วยกลุ่มของโปรแกรมดำเนินงานต่างๆ ที่ให้เรียกใช้โครงสร้างข้อมูล และระเบียบแบบแผนการใช้งาน โดยที่วินซ็อค API ได้จัดเตรียมมาตรฐานที่ใช้ในการเข้าถึงบริการต่างๆ บนเครือข่ายไว้ให้

ข้อกำหนดของวินซ็อคได้แบ่งออกเป็น 2 ส่วนด้วยกัน คือ ส่วนของการเชื่อมต่อกับโปรแกรมประยุกต์บนเครือข่าย และส่วนของการเชื่อมต่อกับโปรโตคอลสแตก (Protocol Stack) บนเครือข่าย โดยในส่วนของการเชื่อมต่อกับโปรแกรมประยุกต์บนเครือข่ายนั้นตัววินซ็อคได้จัดเตรียมช่องทางใน

การติดต่อกับผู้ใช้ รูปแบบ (Format) และการกระจาย (Parse) ข้อมูล สำหรับในส่วนของการเชื่อมต่อกับโปรโตคอลสแตก วินซ็อคจะทำการส่งและรับข้อมูลโดยใช้ทรานสปอร์ตโปรโตคอล (Transport Protocol) ไดรว์เวอร์ (Drivers) และสื่อนำสัญญาณ (Network Media)

วินซ็อคอาจจะเปรียบได้กับตัวแจ็ค (Jack) ซึ่งถือว่าเป็นมาตรฐานอันหนึ่งในการติดต่อไปยังเครื่องใช้โทรศัพท์ หรืออาจจะเปรียบได้กับปลั๊กไฟฟ้าสำหรับเชื่อมต่อให้อุปกรณ์ไฟฟ้าสามารถใช้งานได้ โดยไม่คำนึงว่าไฟฟ้าที่ใช้ได้มาจากแหล่งพลังงานใด ซึ่งอาจจะเป็นจากแหล่งพลังงานน้ำ พลังงานนิวเคลียร์ หรือจากแหล่งพลังงานอื่นๆ และในทำนองเดียวกันกับวินซ็อค โปรแกรมประยุกต์ที่ใช้งานอยู่ก็ไม่จำเป็นว่าต้องเลือกใช้โทโพโลยี (Topology) ใดในการใช้งาน ไม่ว่าจะเป็นอีเธอร์เน็ต (Ethernet) โทเก้นริง (Token Ring) หรือว่าโปรโตคอลสแตก TCP/IP จะพัฒนามาจากบริษัทใด

การใช้งานวินซ็อคมีข้อดีดังนี้คือ

- มีความเป็นมาตรฐาน ทั้งนี้เนื่องจากวินซ็อคได้ถูกพัฒนาขึ้นมาจากความร่วมมือจากหลายๆ บริษัทผู้ผลิตซอฟต์แวร์เครือข่ายภายใต้ข้อกำหนดอันเดียวกัน ทำให้มีความเป็นมาตรฐานจึงทำให้ผู้พัฒนาโปรแกรมประยุกต์บนเครือข่ายสามารถที่จะเลือกใช้วินซ็อคที่ผลิตจากบริษัทต่างๆ ได้อย่างอิสระ
- ความสามารถในการแปลงรหัสโปรแกรม (Source Code) เนื่องจากวินซ็อคได้ทำการพัฒนามาจาก Berkley Sockets API ซึ่งเป็นช่องทางในการติดต่อไปยังเครือข่ายภายใต้ระบบปฏิบัติการยูนิกซ์ (UNIX) ดังนั้นจึงสามารถที่จะทำการแปลงรหัสโปรแกรมประยุกต์บนเครือข่ายที่พัฒนาโดย Berkeley Socket บนยูนิกซ์ มาใช้กับวินซ็อคที่ทำงานบนระบบปฏิบัติการวินโดวส์ได้
- สนับสนุน Dynamic Linking ตัว Dynamic Link Libraries (DLLs) เป็นคุณลักษณะที่สำคัญอันหนึ่งของไมโครซอฟต์วินโดวส์ ซึ่งเป็นไลบรารี (Library) ที่ประกอบไปด้วย Executable Procedure โดยโปรแกรมประยุกต์จะทำการ Link กับตัว DLL ขณะตอนทำงาน (Run Time) ซึ่งมีผลดีดังนี้ คือ

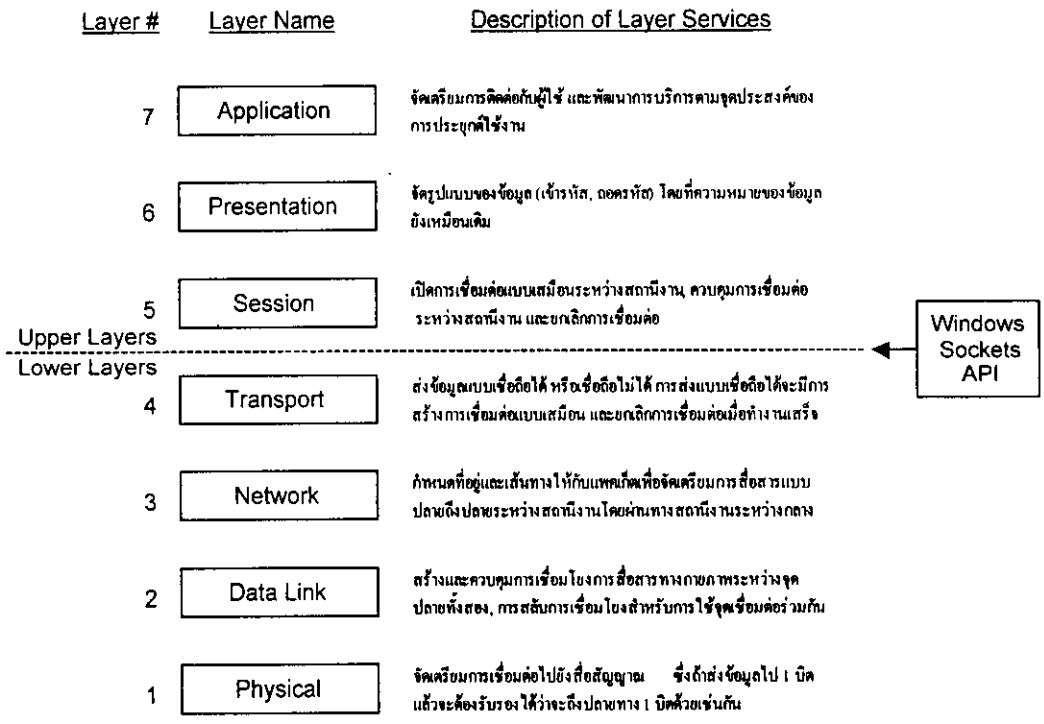
  -เมื่อโปรแกรมประยุกต์หลายๆ โปรแกรมใช้ DLL ร่วมกันทำให้ประหยัดหน่วยความจำ

  -เนื่องจากโปรแกรมประยุกต์จะเป็นอิสระแยกจากตัว DLL ดังนั้นจึงสามารถที่จะทำการปรับปรุง DLL โดยที่จะไม่ส่งผลต่อโปรแกรมประยุกต์

  -ตัว DLL ทุกตัวที่สร้างขึ้นจะสนับสนุน API ดังนั้นโปรแกรมประยุกต์ที่สร้างขึ้นจึงสามารถนำ DLL ไปใช้ได้

2.2 โมเดลเครือข่ายวินซ็อค

เป็นที่ทราบกันว่า OSI (Open Systems Interconnection) เป็นโมเดลเครือข่ายที่แสดงถึงการมองของระบบเครือข่ายทั้งหมด โมเดลนี้จะช่วยลดความซับซ้อนของเครือข่ายให้มองเป็นองค์ประกอบต่างๆ ที่สำคัญ และแสดงให้เห็นว่าองค์ประกอบเหล่านี้สามารถรวมเข้ากันเป็นโครงร่าง (Framework) ได้ ในการใช้โครงร่างดังกล่าวสามารถที่จะกำหนดและแสดงให้เห็นแนวคิดของเครือข่ายและการใช้คำศัพท์เฉพาะได้ ในหัวข้อนี้จะใช้โมเดลนี้อธิบายเปรียบเทียบกับโมเดลเครือข่ายของวินซ็อค



ภาพประกอบ 2-1 โมเดลเครือข่าย OSI

ที่มา: Roberts, Dave., 1995 : 12

การอธิบายเนื้อหาทางวิชาการในระบบเครือข่ายคอมพิวเตอร์โดยทั่วไปมีการใช้คำว่าโปรโตคอลสแตก (Protocol Stack) ซึ่งคือกลุ่มของโปรโตคอลที่สัมพันธ์กัน โดยจะเปรียบเทียบว่าเป็นสแตกของกล่องหลายๆ ใบ ที่นำไปแสดงในโมเดล OSI อย่างเป็นลำดับชั้น ในระดับล่างจะสนับสนุนการทำงานในระดับบนด้วยบริการที่ได้มีการจัดเตรียมไว้ ในโมเดลของวินซ็อคจะเรียกระดับล่างนี้ว่าเนตเวิร์คซิสเต็ม (Network System) และระดับบนว่า วินซ็อคแอพพลิเคชั่น (WINSOCK Application)

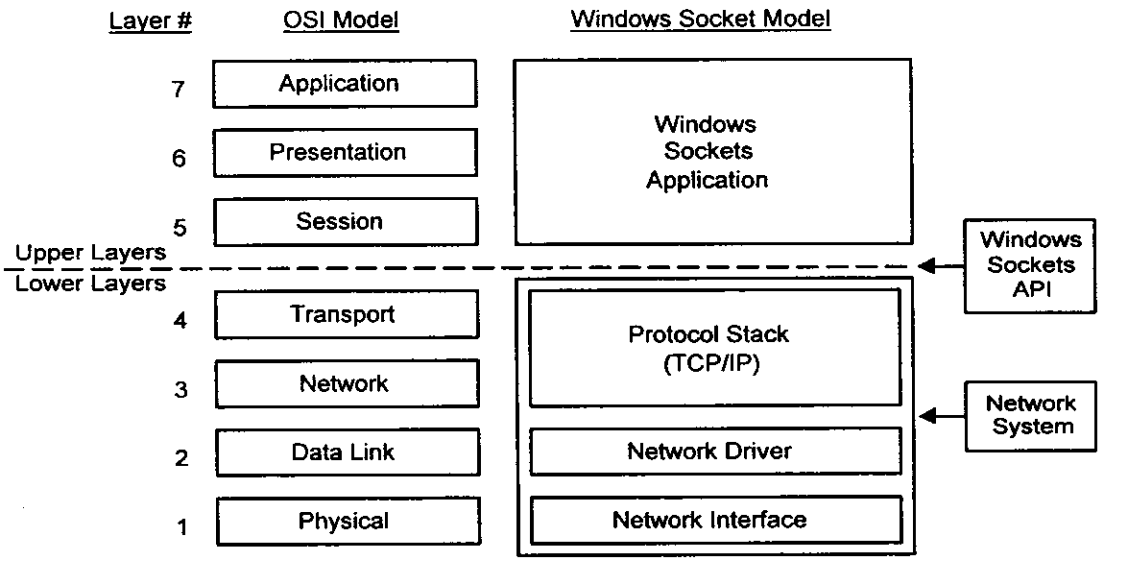
โมเดลเครือข่ายวินซ็อคประกอบด้วย 3 ส่วนดังนี้ คือ

- WINSOCK Application จัดเตรียมฟังก์ชันการทำงานในลำดับชั้นระดับบน ซึ่งคือลำดับชั้นที่ 5-7 เมื่อเปรียบเทียบกับโมเดล OSI

- Network System จัดเตรียมฟังก์ชันการทำงานในลำดับชั้นระดับล่าง ซึ่งคือลำดับที่ 1-4 เมื่อเปรียบ เทียบกับโมเดล OSI

- WINSOCK API จัดการให้ลำดับชั้นระดับบนสามารถติดต่อไปยังลำดับชั้นระดับล่างได้

ในภาพประกอบ 2-2 ลำดับชั้นระดับบนจะเป็นโปรแกรมประยุกต์ที่เป็นส่วนที่ทำการติดต่อกับผู้ใช้ ถัดลงมาจะเป็นวินซ็อค API โดยส่วนวินซ็อค API นี้โปรแกรมประยุกต์จะใช้ทำการส่งและรับข้อมูลไปยังลำดับชั้นถัดไปคือ Network System ในภาพนี้เส้นประจะแทนตำแหน่งของวินซ็อค API ซึ่งจะอยู่ระหว่าง Network System กับ WINSOCK Application ซึ่งจะตรงกันกับการแบ่งของโมเดล OSI ที่แบ่งเป็นลำดับชั้นระดับบน กับ ลำดับชั้นระดับล่าง



ภาพประกอบ 2-2 เปรียบเทียบระหว่างโมเดล OSI กับ โมเดลของวินซ็อค

ที่มา: Roberts, Dave., 1995 : 15

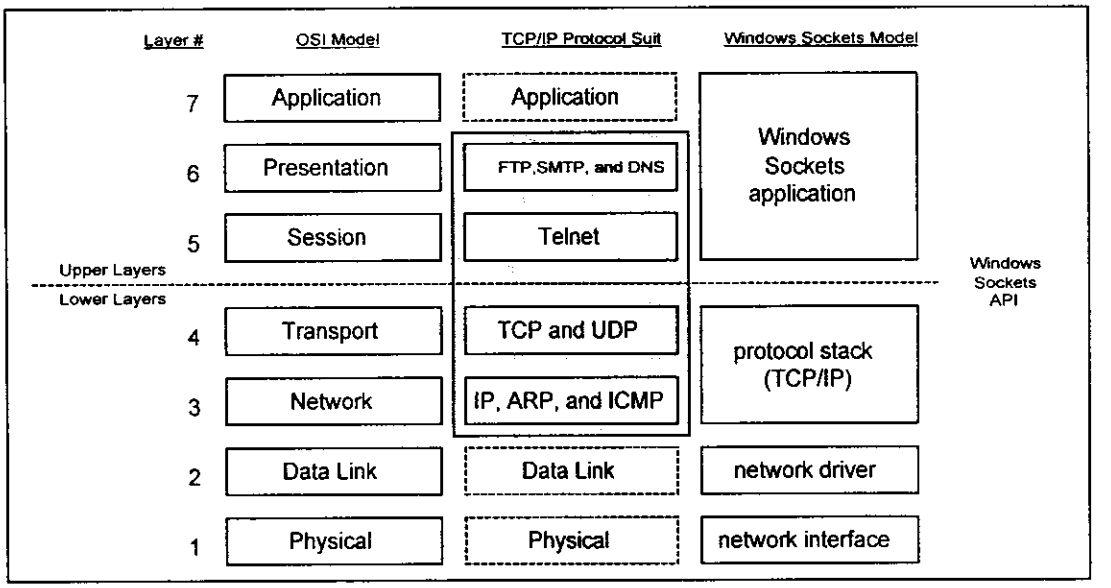
การออกแบบโดยทั่วไปของวินซ็อค API จะอนุญาตให้มีการเข้าถึงบริการต่างๆของโปรโตคอลเครือข่ายที่มีอยู่เป็นจำนวนมาก ข้อกำหนดของวินโดวส์ซ็อคเก็ตเวอร์ชัน 1.1 จะเฉพาะเจาะจงไปที่ชุด

ของโปรโตคอล TCP/IP ดังนั้นโปรแกรมประยุกต์ต่างๆ ที่ใช้วินซ็อคในการพัฒนาจะต้องรองรับโปรโตคอล TCP/IP ด้วยเช่นกัน

จากภาพประกอบ 2-3 ชุดของโปรโตคอล TCP/IP จะสัมพันธ์กับโมเดล OSI และโมเดลของวินซ็อค จะเห็นว่าชุดของโปรโตคอล TCP/IP ไม่ได้มีเพียงแค่โปรโตคอล TCP และ IP เท่านั้น ตลอดจนในชั้น Data Link และชั้น Physical ไม่ได้เป็นส่วนหนึ่งของชุดโปรโตคอล TCP/IP นอกจากนี้ชั้น Application ก็ไม่อยู่ในชุดของโปรโตคอล TCP/IP ด้วยเช่นกัน

ในระดับชั้นบนจะมีโปรโตคอลประยุกต์ ซึ่งได้แก่ Telnet, FTP และ SMTP ซึ่งเป็นโปรโตคอลที่สำคัญที่ชุดของโปรโตคอล TCP/IP ได้เตรียมไว้ นอกจากนี้ยังมีโปรโตคอลอื่นๆ อีกเช่น Archie, Gopher และ Talk เป็นต้น แต่โดยทั่วไปจะพิจารณาว่า Telnet, FTP และ SMTP เป็นส่วนหนึ่งของ TCP/IP ทั้งนี้เนื่องจากได้เตรียมการให้บริการขั้นพื้นฐานต่างๆ ไว้เป็นส่วนมาก เช่น จดหมายอิเล็กทรอนิกส์, การถ่ายโอนแฟ้มข้อมูล และ การเข้าใช้งานในระยะไกล เป็นต้น

นอกจากนี้วินซ็อค API ยังจัดเตรียมการเข้าถึงการบริการของโปรโตคอลที่อยู่ในระดับชั้น Transport และ Network อีกด้วย



ภาพประกอบ 2-3 เปรียบเทียบโปรโตคอล TCP/IP กับ โมเดล OSI และ โมเดลของวินซ็อค

ที่มา: Roberts, Dave., 1995 : 33

2.3 โปรโตคอลเทลเน็ต

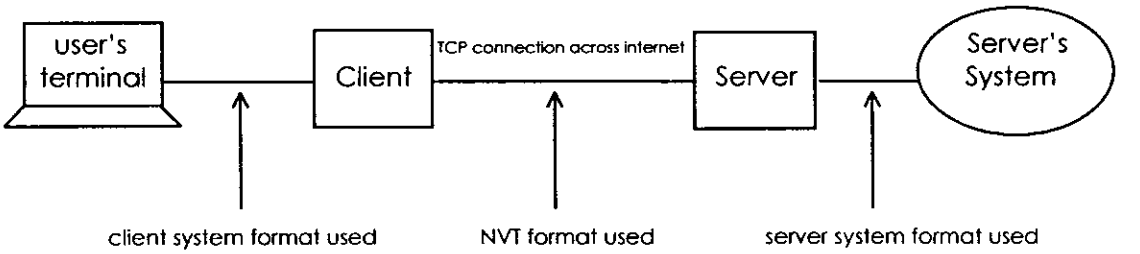
จุดประสงค์ของโปรโตคอลเทลเน็ตคือให้ผู้ใช้สามารถเข้าไปใช้บริการจากคอมพิวเตอร์ที่อยู่ในเครือข่ายได้โดยไม่ต้องนั่งอยู่หน้าเครื่องนั้นโดยตรง โดยโปรโตคอลเทลเน็ตจัดเตรียมการติดต่อสื่อสารกันแบบ 2 ทิศทาง (Bi-directional) ข้อกำหนดดังกล่าวเป็นเป้าหมายหลักที่จะกำหนดวิธีการที่เป็นมาตรฐานในการติดต่อสื่อสารกันระหว่างคอมพิวเตอร์

2.3.1 แนวคิดของโปรโตคอลเทลเน็ต

การเชื่อมต่อของเทลเน็ตใช้โปรโตคอล TCP โดยโปรโตคอลเทลเน็ตมีแนวความคิดอยู่ด้วยกัน 3 ส่วนหลัก ๆ คือ

1. จอภาพเสมือนทางเครือข่าย (Network Virtual Terminal : NVT)

เมื่อเทลเน็ตมีการเชื่อมต่อกันเกิดขึ้นจะถือว่าแต่ละข้างของการเชื่อมต่อมีการจำลองตัวเองเป็น NVT โดย NVT เป็นแนวความคิดที่ได้จัดเตรียมมาตรฐานโดยทำการแทนข้อมูลที่เป็นกลางในการสื่อสารระหว่างกัน แนวความคิดดังกล่าวทำให้ทั้งเครื่องคอมพิวเตอร์แม่ข่าย (Server) และเครื่องคอมพิวเตอร์ลูกข่าย (Client) ไม่จำเป็นต้องเก็บข้อมูลต่างๆ ของแต่ละจอภาพทั้งหมด ทั้งเครื่องคอมพิวเตอร์แม่ข่ายและเครื่องคอมพิวเตอร์ลูกข่ายสามารถทำการแปลงข้อมูลที่ได้รับมาเป็นข้อมูลที่ตัวเองเข้าใจได้ และทำนองเดียวกันสามารถทำการแปลงข้อมูลให้เป็นมาตรฐานตามแบบ NVT ที่ผู้รับสามารถเข้าใจแล้วจึงทำการส่งออกไปในเครือข่าย รายละเอียดการทำงานและมาตรฐานต่างๆ ของ NVT จะได้กล่าวต่อไปในหัวข้อ 2.3.2



ภาพประกอบ 2-4 จอภาพเสมือนทางเครือข่าย

ที่มา: Comer, Douglas., 1991 : 369

2. การต่อรองทางเลือก (Option Negotiation)

เครื่องที่ให้บริการอาจต้องการที่จะจัดเตรียมการให้บริการเพิ่มเติมนอกเหนือจาก NVT และผู้ใช้เองก็ต้องการบริการที่เพิ่มขึ้นมานี้ด้วย ในโปรโตคอลเทลเน็ตจึงได้มีทางเลือก (Option) หลายทางเลือกด้วยกัน โดยจะใช้กับรหัสคำสั่ง DO, DON'T, WILL, WON'T เพื่อที่จะให้ทั้งผู้ใช้บริการและผู้ให้บริการทำการใช้ทางเลือกต่างๆ ที่เพิ่มขึ้นมานี้ได้

ในการใช้งานทางเลือกที่เพิ่มขึ้นมานี้คู่ของการเชื่อมต่อข้างใดข้างหนึ่งหรือทั้งคู่จะต้องทำการร้องขอ (Request) ทางเลือกที่ต้องการ โดยอีกข้างหนึ่งของการเชื่อมต่ออาจจะทำการตอบรับหรือปฏิเสธการร้องขออันนั้น ซึ่งถ้าการร้องขอได้รับการตอบรับ ทางเลือกที่ขอไปก็จะมีผลในทันที แต่ถ้าได้รับการปฏิเสธ การเชื่อมต่อก็ยังคงมีมาตรฐานตามแบบ NVT

ตัวอย่างของการต่อรองทางเลือก เช่น ถ้าฝ่ายที่ขอต่อรองต้องการแสดงอักขระและส่งคำสั่งไปแจ้งให้อีกฝ่ายหนึ่งทราบโดยใช้ IAC WILL ดังนี้

IAC WILL ECHO

หมายถึงฝ่ายส่งจะแสดงอักขระ โดยอีกฝ่ายสามารถตอบด้วย

IAC DO ECHO

หมายถึงให้ทำการแสดงอักขระได้ หรืออาจตอบกลับด้วย

IAC DON'T ECHO

หมายถึงไม่ให้แสดงอักขระ

ในบางครั้งอาจต้องการต่อรองทางเลือกย่อยเพิ่มเติมอีก กล่าวคือเมื่อการต่อรองทางเลือกหลักเสร็จสิ้นแล้วก็จะมีการต่อรองทางเลือกย่อยในรายละเอียดตามมา เรียกกลไกนี้ว่าการต่อรองทางเลือกย่อย (Option Subnegotiation) ตัวอย่างของกลไกนี้เช่น การต่อรองเพื่อกำหนดชนิดของจอภาพเป็นต้น

การต่อรองทางเลือกย่อยใช้รหัส SB (Subnegotiation) เป็นรหัสนำ ตามด้วยรหัสทางเลือกและปิดท้ายด้วยรหัส SE (End of Subnegotiation) ตัวอย่างต่อไปนี้แสดงการต่อรองทางเลือกย่อยเพื่อกำหนดชนิดจอภาพ

IAC WILL terminal type

หมายถึงเครื่องคอมพิวเตอร์ลูกข่ายขอกำหนดชนิดจอภาพ

IAC DO terminal type

หมายถึงเครื่องคอมพิวเตอร์แม่ข่ายตอบตกลง

IAC SB Terminal-type SEND IAC SE

หมายถึงเครื่องคอมพิวเตอร์แม่ข่ายขอให้ไคลเอนแจ้งชนิดจอภาพ

IAC SB Terminal-type IS vt100 IAC SE

หมายถึงเครื่องคอมพิวเตอร์ลูกข่ายส่งจอภาพชนิด vt100 กลับมา

ตารางข้างล่างนี้จะสรุปความหมายของรหัสคำสั่งที่ใช้ในการต่อรองทางเลือก

ชื่อรหัส	ค่าฐานสิบ	ความหมาย
WILL	251	ผู้ส่งคำสั่งนี้จะกระทำทางเลือกที่ได้เลือกไว้ หรือยืนยันที่จะกระทำทางเลือก
WONT	252	ผู้ส่งคำสั่งนี้จะหยุดกระทำทางเลือกที่ได้เลือกไว้ หรือปฏิเสธที่จะกระทำทางเลือก
DO	253	ผู้ส่งคำสั่งนี้ร้องขอให้อีกฝ่ายกระทำทางเลือก หรือยืนยันที่จะกระทำทางเลือก
DONT	254	ผู้ส่งคำสั่งนี้ร้องขอให้อีกฝ่ายหยุดกระทำทางเลือก หรือยืนยันที่จะไม่กระทำทางเลือก
SB	250	กำหนดการต่อรองย่อย
SE	240	จบการต่อรองย่อย

ตาราง 2-1 Option Negotiation

ที่มา: Postel, Jon. and Reynolds, Joyce., 1983., RFC 854 : 14

3. การสมมาตรกันของทางเลือกที่ได้ทำการต่อรองกันไว้ (Symmetry of the Negotiation)

กล่าวคือกลไกในการทำงานของโปรโตคอลเทลเน็ตสามารถที่จะกระทำได้ในทิศทางใดทิศทางหนึ่งของจุดปลายของการเชื่อมต่อทั้งสองได้

ผู้ที่ทำการออกแบบทางเลือกจะไม่จำกัดขนาดความยาวคำสั่งของการต่อรองทางเลือกเพื่อให้ง่ายต่อการใช้งาน ทั้งนี้เนื่องจากจะมีบางทางเลือก เช่น การต่อรองทางเลือกย่อย ที่จะมีความยาวของคำสั่งมากเป็นพิเศษ

โดยสรุปเมื่อระบบข้างใดข้างหนึ่งของการเชื่อมต่อทำการส่ง WILL *XXX* จะเป็นการเสนอให้เริ่มต้นกระทำทางเลือก *XXX* แล้วอีกข้างหนึ่งจะตอบกลับไปโดยใช้ DO *XXX* หรือ DON'T *XXX* สำหรับแจ้งการตอบรับและตอบปฏิเสธตามลำดับ ในทำนองเดียวกันการส่งคำสั่ง DO *XXX* จะเป็นการร้องขอเพื่อให้อีกข้างหนึ่งกระทำทางเลือก *XXX* โดย WILL *XXX* และ WON'T *XXX* จะเป็นการแจ้งการตอบรับและตอบปฏิเสธตามลำดับ

2.3.2 จอภาพเสมือนทางเครือข่าย (Network Virtual Terminal : NVT)

NVT คือฮาร์ดแวร์เชิงลอจิกที่ทำการส่งอักขระในแบบ 2 ทิศทาง NVT ประกอบไปด้วย พรินเตอร์ (Printer) และแป้นพิมพ์ (Keyboard) พรินเตอร์จะเป็นส่วนที่ทำการตอบสนองต่อข้อมูลที่ได้รับโดยการแสดงผลออกทางจอภาพ ส่วนแป้นพิมพ์จะเป็นการส่งข้อมูลออกไปในขณะที่มีการเชื่อมต่อของเทลเน็ต

2.3.2.1 การส่งข้อมูล

แม้ว่าการเชื่อมต่อของเทลเน็ตโดยแท้ที่จริงแล้วจะเป็นแบบฟูลดูเพล็ก (Full Duplex) แต่ NVT จะมองเหมือนกับว่าเป็นแบบฮาล์ฟดูเพล็ก (Half-duplex) โดยมีการจัดการในแบบไลน์บัฟ-เฟอร์โหมด (Line-buffer Mode) เพื่อให้การส่งข้อมูลเป็นไปอย่างเรียบร้อยจึงได้กำหนดค่าโดยปริยาย (Default) ของการส่งข้อมูลขึ้นดังนี้

1. ข้อมูลจะมีการเก็บรวบรวมที่ผู้ให้บริการจนกว่าจะมีความพร้อมในการส่งข้อมูลหรือจนกว่าจะมีการส่งสัญญาณว่าจะทำการส่งข้อมูล ซึ่งสัญญาณนี้อาจจะถูกส่งมาจากโปรเซส (Process) หรือผู้ใช้ สาเหตุของการเก็บรวบรวมข้อมูลก่อนทำการส่งเนื่องจากจะต้องมีการแปลงข้อมูลให้อยู่ในแบบ NVT ก่อนที่จะทำการส่งออกไป

2. เมื่อเสร็จสิ้นกระบวนการส่งข้อมูลไปยังจอภาพ และไม่มีคิวข้อมูล (Queue Input) จากแป้นพิมพ์ สำหรับประมวลผลต่อไปแล้วจะต้องมีการส่งทางเลือก GA (Go Ahead) โดยที่ GA จะถูกส่งจากจอภาพตรงตำแหน่งสุดท้ายของบรรทัดคำสั่ง คำสั่งนี้ออกแบบมาเพื่อช่วยผู้ใช้งานจอภาพที่เป็นแบบฮาล์ฟดูเพล็ก หรือที่เรียกว่า Lockable Terminal เช่น IBM 2741 เป็นต้น ซึ่งจอภาพชนิดนี้ไม่อนุญาตให้ผู้ใช้ป้อนข้อมูลในขณะที่มีการแสดงผลลัพธ์ออกทางจอภาพ จอภาพจะมีการเปลี่ยนการควบคุมจากผู้ใช้ที่มีการป้อนข้อมูลผ่านแป้นพิมพ์ไปยังจอภาพเมื่อผู้ใช้ทำการพิมพ์ข้อมูลเสร็จสิ้น แต่อย่างไรก็ตามเครื่องที่ให้บริการต้องการที่จะแจ้งให้ผู้ใช้ทราบว่าเมื่อใดที่ทำการแสดงผลออกทางจอภาพ

แล้ว เพื่อที่จะได้เปลี่ยนการควบคุมไปยังแป้นพิมพ์ของผู้ใช้ เพื่อสนับสนุนการทำงานดังกล่าวจึงได้มีการพัฒนาทางเลือก GA ขึ้นมา โดยเมื่อเครื่องคอมพิวเตอร์แม่ข่ายพบว่าไม่มีผลลัพธ์ใดๆ ที่จะแสดงออกทางจอภาพ จะส่งทางเลือก GA ไปยังเครื่องคอมพิวเตอร์ลูกข่าย ทางเครื่องคอมพิวเตอร์ลูกข่ายก็จะเปลี่ยนการควบคุมกลับไปยังแป้นพิมพ์ของผู้ใช้ สำหรับระบบที่ไม่ต้องการใช้ทางเลือก GA ก็จะมีการใช้ทางเลือก SGA (Suppress Go Ahead) เพื่อระงับการใช้ทางเลือก GA

โดยแท้ที่จริงแล้วในหลายๆระบบปฏิบัติการ จะไม่มีวิธีการที่จะทราบได้ว่าเมื่อใดที่เครื่องคอมพิวเตอร์แม่ข่ายได้สั่งให้แสดงผลลัพธ์ออกทางจอภาพเสร็จสิ้นแล้ว หรือว่ามีโปรเซสที่กำลังทำงานอยู่รอรับข้อมูลจากแป้นพิมพ์ ดังนั้นส่วนใหญ่ในการพัฒนาโปรแกรมเทลเน็ตจะไม่มีการใช้คำสั่ง GA

2.3.2.2 มาตรฐานในการแทนฟังก์ชันควบคุม (Standard Representation of Control Functions)

ในทุกระบบจะมีกลไกในการทำงานต่างๆ เช่น ในระบบยูนิกซ์ ถ้าต้องการให้โปรแกรมหยุดการทำงานก็จะกด CONTROL-C ซึ่งในการส่งข้อมูลไปยังเครื่องคอมพิวเตอร์แม่ข่ายจะไม่ทำการส่ง CONTROL-C ไป แต่จะเป็นการแทนด้วยมาตรฐานที่ได้กำหนดไว้แล้วทำการส่งไป โดยมาตรฐานในการแทนฟังก์ชันควบคุมจะมีอยู่ด้วยกัน 5 ฟังก์ชันดังรายละเอียดดังนี้

Interrupt Process (IP)

เป็นมาตรฐานที่ใช้สำหรับการยกเลิกการทำงานของโปรเซส

Abort Output (AO)

เป็นมาตรฐานที่ใช้ในการยกเลิกการสั่งผลลัพธ์ไปแสดงยังจอภาพเมื่อโปรเซสทำงานเสร็จ

Are You There (AYT)

เป็นมาตรฐานที่ใช้ในการตรวจสอบเครื่องคอมพิวเตอร์แม่ข่ายที่ให้บริการอยู่ในขณะนั้นว่ากำลังทำงานอยู่หรือไม่ ทั้งนี้เนื่องจากในบางครั้งเครื่องคอมพิวเตอร์แม่ข่ายอาจจะใช้เวลาในการตอบสนองช้า อาจจะเนื่องมาจากระบบทำงานหนักหรือว่าจากสาเหตุอื่นๆ

Erase Character (EC)

เป็นมาตรฐานที่ใช้ทำการลบอักขระออกจากจอภาพ

Erase Line (EL)

เป็นมาตรฐานที่ใช้ทำการลบข้อมูลทั้งหมดตรงตำแหน่งบรรทัดปัจจุบันของอินพุท

2.3.2.3 สัญญาณซิงค์ของเทลเน็ต (Telnet "SYNCH" Signal)

ในขณะที่กำลังมีการแสดงผลลัพธ์ออกทางจอภาพของผู้ใช้งานและผู้ใช้ต้องการที่จะหยุดการแสดงผลลัพธ์นั้น เทลเน็ตมีกลไกที่หยุดการทำงานของโปรเซส (IP) และไม่ทำการส่งผลลัพธ์ไปแสดงยังจอภาพของผู้ใช้งาน (AO) ดังที่ได้กล่าวมาแล้ว แต่กระบวนการดังกล่าวจะมีข้อมูลที่ค้างอยู่ในบัฟเฟอร์ระหว่างผู้ใช้บริการกับเครื่องคอมพิวเตอร์แม่ข่าย ซึ่งได้แก่คำสั่งที่ผู้ใช้งานพิมพ์ไปแล้วแต่ยังไม่ได้ทำการประมวลผล และข้อมูลที่ยังไม่ได้แสดงผลออกทางจอภาพ

ในการเคลียร์ข้อมูลที่ยังคงค้างอยู่ในบัฟเฟอร์ดังกล่าวเทลเน็ตจะมีกลไกที่เรียกว่า "ซิงค์" สัญญาณการซิงค์จะประกอบไปด้วยทีซีพีเออร์เจ้นท์ (TCP Urgent) และรหัสคำสั่งเทลเน็ตที่ชื่อดาต้ามาร์ค (Data Mark, DM) โดยทีซีพีเออร์เจ้นท์จะถูกใช้เพื่อเป็นการแจ้งให้ทราบว่าให้ทำการข้าม ข้อมูลที่อยู่ในสายข้อมูลในทันทีที่ได้รับจนกว่าจะพบรหัสคำสั่ง DM โดยรหัสคำสั่ง DM จะเป็นตัวบ่งบอกว่าจบการซิงค์แล้ว และให้ผู้รับทำการประมวลผลข้อมูลที่อยู่ในสายส่งข้อมูลตามปกติ

ในสถานะการทำงานปกติ DM จะไม่มีการทำงาน แต่ในสถานะรีบด่วนตัว DM จะเป็นสัญญาณที่แจ้งให้ทราบว่าให้ทำการจบการประมวลผลโดยเร็วที่สุด

โดยปกติกระบวนการซิงค์จะรวมเข้าด้วยกันกับอินเตอร์รัพโปรเซส ดังนั้นลำดับ IAC IP IAC DM จะถูกส่งไปเป็นทีซีพีเออร์เจ้นท์

2.3.2.4 เครื่องพิมพ์ และแป้นพิมพ์เสมือน (NVT Printer and Keyboard)

เครื่องพิมพ์เสมือนทำหน้าที่ในการแสดงผล โดยทั่วไปแล้วเครื่องพิมพ์เสมือนก็คือจอภาพ เครื่องพิมพ์เสมือนประกอบด้วย USASCII จำนวน 95 ตัว (รหัส 32-126) คอนโทรลโค้ดของ USASCII จำนวน 33 ตัว (รหัส 0-31 และ 127) และที่เหลืออีก 128 ตัว (รหัส 128-255) จะไม่มีการอ้างอิงถึง ตาราง 2-2 จะแสดงความหมายรหัสควบคุมการแสดงผลของเครื่องพิมพ์เสมือน

ชื่อรหัส (ชื่อย่อ)	ค่าฐานสิบ	ความหมาย
NULL (NUL)	0	ไม่มีการกระทำใดๆ
BELL (BEL)	7	ทำการส่งเสียง
Back Space (BS)	8	ย้ายเคอร์เซอร์ไปทางซ้าย 1ตำแหน่ง
Horizontal Tab (HT)	9	ย้ายเคอร์เซอร์ไปทางขวาตรงตำแหน่งแท็บถัดไป
Line Feed (LF)	10	ย้ายเคอร์เซอร์ไปยังบรรทัดถัดไป
Vertical Tab (VT)	11	ย้ายเคอร์เซอร์ลงข้างล่างตรงตำแหน่งแท็บถัดไป
Form Feed (FF)	12	ย้ายเคอร์เซอร์ไปยังตำแหน่งบนสุดของหน้าถัดไป
Carriage Return (CR)	13	ย้ายเคอร์เซอร์ไปยังตำแหน่งริมซ้ายของบรรทัด

ตาราง 2-2 รหัสควบคุมการแสดงผลตามแบบเครื่องพิมพ์เสมือน

ที่มา: Postel, Jon. and Reynolds, Joyce., 1983., RFC 854 : 10-11

นอกจากนี้เครื่องพิมพ์เสมือนยังได้กำหนดมาตรฐานในการจบบรรทัดด้วยอักขระ 2 ตัว มีลำดับเป็น CR-LF เมื่อผู้ใช้ทำการกดคีย์ซึ่งสอดคล้องกันกับการจบของบรรทัดบนจอภาพของผู้ใช้ ซึ่งอาจจะเป็น Enter หรือ Return เป็นต้น เครื่องคอมพิวเตอร์ลูกข่ายก็จะทำการแปลงคีย์ที่กดนั้นไปเป็น CR-LF เพื่อทำการการส่งออกไปยังเครื่องคอมพิวเตอร์แม่ข่าย ทางเครื่องคอมพิวเตอร์แม่ข่ายก็จะทำการแปลง CR-LF ให้อยู่ในรูปแบบที่เครื่องคอมพิวเตอร์แม่ข่ายเข้าใจได้ว่านี่คือการจบของบรรทัด

สำหรับแป้นพิมพ์เสมือนทำหน้าที่ในการสร้างอักขระ โดยทั่วไปแล้วแป้นพิมพ์เสมือนก็คือแป้นพิมพ์ที่ใช้งาน แป้นพิมพ์เสมือนประกอบด้วย USASCII ทั้งหมด 128 ตัว

นอกจากนี้แป้นพิมพ์เสมือนยังเป็นส่วนที่ทำการสร้างรหัสคำสั่งที่จะใส่เข้าไปในรหัสคำสั่งของเทลเน็ต ดังตาราง 2-3

ชื่อรหัส (ชื่อย่อ)	ค่าฐานสิบ	ความหมาย
Break (BRK)	243	กดปุ่ม Break
Interrupt Process (IP)	244	ยกเลิกโปรเซส
Abort Output (AO)	245	ยกเลิกการแสดงผลที่ยังค้างอยู่
Are You There (AYT)	246	สอบถามว่าเครื่องคอมพิวเตอร์แม่ข่ายยังคงทำงานอยู่หรือไม่
Erase Character (EC)	247	ลบอักขระ
Erase Line (EL)	248	ลบบรรทัด

ตาราง 2-3 รหัสควบคุมของแป้นพิมพ์เสมือน

ที่มา: Postel, Jon. and Reynolds, Joyce., 1983., RFC 854 : 12-13

2.3.3 โครงสร้างคำสั่งของเทลเน็ต (Telnet Command Structure)

โครงสร้างคำสั่งของเทลเน็ตอย่างน้อยที่สุดจะประกอบด้วย 2 ไบต์ ซึ่งก็คือ "Interpret as Command" (IAC) ตามด้วยรหัสคำสั่ง ส่วนคำสั่งที่เป็นการต่อรองการใช้เงื่อนไขประกอบด้วย 3 ไบต์ โดยไบต์ที่ 3 จะเป็นรหัสทางเลือก

ไบต์ที่ 1	ไบต์ที่ 2	ไบต์ที่ 3
IAC	รหัสคำสั่ง	รหัสทางเลือก

ภาพประกอบ 2-5 โครงสร้างคำสั่งของเทลเน็ต

ตารางข้างล่างนี้จะเป็นรหัสคำสั่งในเทลเน็ต ซึ่งรหัสคำสั่งในเทลเน็ตจะต้องอยู่หลัง IAC จึงจะมีความหมายดังตารางข้างล่าง

ชื่อย่อ	ค่าฐานสิบ	ความหมาย
SE	240	จบการต่อรองย่อย
NOP	241	ไม่มีการกระทำใดๆ
DM	242	ทำการซิงค์
BRK	243	อักขระ BRK ของ NVT
IP	244	ยกเลิกโปรเซส
AO	245	ยกเลิกการแสดงผลที่ยังค้างอยู่
AYT	246	สอบถามว่าเครื่องคอมพิวเตอร์แม่ข่ายยังคงทำงานอยู่หรือไม่
EC	247	ลบอักขระ
EL	248	ลบบรรทัด
GA	249	กำหนดการทำงานแบบฮาล์ฟดูเพล็ก
SB	250	กำหนดการต่อรองย่อย
WILL (รหัสทางเลือก)	251	ยืนยันว่าจะดำเนินการตามรหัสทางเลือก
WON'T (รหัสทางเลือก)	252	ปฏิเสธว่าจะไม่ดำเนินการตามรหัสทางเลือก
DO (รหัสทางเลือก)	253	ร้องขอให้อีกฝ่ายดำเนินการตามรหัสทางเลือก
DON'T (รหัสทางเลือก)	254	ร้องขอให้หยุดดำเนินการตามรหัสทางเลือก
IAC	255	คำสั่งแจ้งว่าสิ่งที่ตามมาไม่ใช้ข้อมูลแต่เป็นรหัสคำสั่ง

ตาราง 2-4 รหัสคำสั่งในเทลเน็ต

ที่มา: Postel, Jon. and Reynolds, Joyce., 1983., RFC 854 : 14

2.3.4 การสถาปนาการเชื่อมต่อ (Connection Establishment)

การเชื่อมต่อของเทลเน็ตใช้โปรโตคอล TCP โดยจะใช้หมายเลขพอร์ต 23 ในการสร้างการเชื่อมต่อ

2.3.5 ทางเลือกของเทลเน็ต (Telnet Option)

2.3.5.1 ทางเลือกการส่งข้อมูลแบบไบนารี (Telnet Binary Transmission Option)

การใช้ทางเลือกนี้จะเปลี่ยนสถานะการทำงานจาก NVT ซึ่งใช้ 7 บิท มาเป็นสถานะไบนารี ซึ่งใช้ 8 บิท ดังจะอธิบายรายละเอียดอื่นๆ ได้ดังนี้

ก. ชื่อคำสั่งและค่าฐานสิบ

ทางเลือกนี้มีชื่อคำสั่งคือ TRANSMIT-BINARY และมีค่าฐานสิบคือ 0

ข. ความหมายของคำสั่ง

ทางเลือกนี้มีคำสั่งที่ใช้งานและความหมายของแต่ละคำสั่งดังนี้

IAC WILL TRANSMIT-BINARY

ผู้ส่งคำสั่งนี้จะร้องขอ หรือเป็นการยืนยัน ที่จะเริ่มต้นทำการส่งอักขระในแบบ 8 บิท และต้องการให้ผู้รับแปลอักขระที่ได้รับในแบบ 8 บิท

IAC WON'T TRANSMIT-BINARY

ในกรณีที่การเชื่อมต่อยังคงส่งข้อมูลเป็นแบบไบนารี การส่งคำสั่งนี้จะหมายถึงความต้องการที่จะให้หยุดการส่งแบบไบนารีและให้ผู้รับแปลอักขระที่ได้รับในแบบ NVT ASCII แทน

แต่ในกรณีที่การเชื่อมต่อไปไม่ได้เป็นแบบไบนารี การส่งคำสั่งนี้จะเป็นการปฏิเสธการส่งข้อมูลโหมดไบนารีและยังคงส่งข้อมูลในโหมดปัจจุบันที่ทำงานอยู่ขณะนั้น

การส่งแบบไบนารีจะกระทำได้ก็ต่อเมื่อมีการร้องขอจากฝ่ายหนึ่งและอีกฝ่ายหนึ่งต้องทำการตอบสนองกลับมาด้วย

IAC DO TRANSMIT-BINARY

ผู้ส่งคำสั่งนี้จะทำการร้องขอ หรือยืนยัน ให้ทำการส่งข้อมูลแบบ 8 บิท และแปลอักขระที่ได้รับในแบบ 8 บิท

IAC DON'T TRANSMIT-BINARY

ถ้าการเชื่อมต่อยังคงมีการส่งข้อมูลแบบไบนารี การส่งคำสั่งนี้จะหมายถึงความต้องการที่จะให้ทำการส่งข้อมูลเป็นแบบ NVT ASCII และให้ผู้รับแปลความหมายอักขระที่ได้รับในรูปแบบ NVT แต่ถ้าการเชื่อมต่อไม่ได้มีการส่งข้อมูลในโหมดไบนารี การส่งคำสั่งนี้ก็จะไม่มีผลอะไร กล่าวคือการเชื่อมต่อก็จะยังคงส่งข้อมูลในโหมดปัจจุบันที่ทำงานอยู่ขณะนั้น

การส่งแบบไบนารีจะกระทำได้ก็ต่อเมื่อมีการร้องขอจากฝ่ายหนึ่งและอีกฝ่ายหนึ่งต้องทำการตอบสนองกลับมาด้วย

ค. ค่าโดยปริยาย

หากไม่มีการร้องขอใดๆ การเชื่อมต่อจะไม่มีการส่งข้อมูลแบบไบนารี

2.3.5.2 ทางเลือกการแสดงอักขระ (Telnet Echo Option)

ใน NVT จะประกอบไปด้วยพริ้นเตอร์และแป้นพิมพ์ ซึ่งโดยปกติแล้วอักขระที่พิมพ์บนแป้นพิมพ์จะแสดงผลตรงพริ้นเตอร์ของผู้ใช้ แต่ในสถานการณ์ที่มีการประมวลผลแบบทันทีทันใดที่เครื่องคอมพิวเตอร์ที่อยู่ระยะไกล ซึ่งอักขระที่ส่งไปประมวลผลนั้นจะถูกส่งไปเพื่อควบคุมการแสดงผลออกทางพรินเตอร์ ในการที่จะสนับสนุนการทำงานดังกล่าวจะต้องทำการตกลงให้อักขระที่พิมพ์ไปบนแป้นพิมพ์นั้นให้ทำการแสดงผลยังเครื่องคอมพิวเตอร์ที่อยู่ระยะไกล ดังจะอธิบายรายละเอียดอื่นๆ ได้ดังนี้

ก. ชื่อคำสั่งและค่าฐานสิบ

ทางเลือกนี้มีชื่อคำสั่งคือ ECHO และมีค่าฐานสิบคือ 1

ข. ความหมายของคำสั่ง

ทางเลือกนี้มีคำสั่งที่ใช้งานและความหมายของแต่ละคำสั่งดังนี้

IAC WILL ECHO

ผู้ส่งคำสั่งนี้จะทำการร้องขอเพื่อทำการเริ่มต้น หรือยืนยันที่จะแสดงอักขระที่ได้รับกลับไปยังผู้ที่ทำการส่งอักขระมา

IAC WON'T ECHO

ผู้ส่งคำสั่งนี้ต้องการที่จะหยุดหรือปฏิเสธที่จะแสดงอักขระที่ได้รับกลับไปยังผู้ที่ทำการส่งอักขระมา

IAC DO ECHO

ผู้ส่งคำสั่งนี้จะร้องขอไปยังผู้รับคำสั่งนี้ให้ทำการแสดงผลอักขระ หรือ ให้ผู้รับคำสั่งนี้ทำการยืนยันที่จะทำการแสดงผลอักขระกลับไปยังผู้ที่ทำการร้องขอมา

IAC DON'T ECHO

ผู้ส่งคำสั่งนี้ต้องการที่จะให้ผู้รับคำสั่งนี้ทำการหยุดการแสดงผลอักขระ

ค. ค่าโดยปริยาย

หากไม่มีการร้องขอใดๆ จะไม่มีการแสดงผลอักขระ

2.3.5.3 ทางเลือกการระงับ GA (Telnet Suppress Go Ahead Option)

ทางเลือก GA ได้ออกแบบมาเพื่อรองรับการทำงานในแบบฮาล์ฟดูเพล็ก แต่การเชื่อมต่อของเทลเน็ตจะเป็นแบบฟูลดูเพล็กจึงไม่มีความจำเป็นที่จะต้องใช้ทางเลือก GA จึงมีความต้องการที่จะให้การเชื่อมต่อของเทลเน็ตยกเลิกการใช้ทางเลือก GA ดังจะอธิบายรายละเอียดอื่นๆ ได้ดังนี้

ก. ชื่อคำสั่งและค่าฐานสิบ

ทางเลือกนี้มีชื่อคำสั่งคือ SUPPRESS-GO-AHEAD และมีค่าฐานสิบคือ 3

ข. ความหมายของคำสั่ง

ทางเลือกนี้มีคำสั่งที่ใช้งานและความหมายของแต่ละคำสั่งดังนี้

IAC WILL SUPPRESS-GO-AHEAD

ผู้ส่งคำสั่งนี้จะทำการยกเลิก หรือยืนยันที่จะยกเลิกการใช้ทางเลือก GA ในขณะที่มีการส่งข้อมูล

IAC WON'T SUPPRESS-GO-AHEAD

ผู้ส่งคำสั่งนี้จะทำการใช้ทางเลือก GA หรือยังคงใช้ทางเลือก GA ต่อไป ในขณะที่มีการส่งข้อมูล

IAC DO SUPPRESS-GO-AHEAD

ผู้ส่งคำสั่งนี้ร้องขอให้ทำการยกเลิกการใช้ทางเลือก GA หรือเป็นการยืนยันที่จะยกเลิกการใช้ทางเลือก GA ในขณะที่มีการส่งข้อมูล

IAC DON'T SUPPRESS-GO-AHEAD

ผู้ส่งคำสั่งนี้ต้องการให้ใช้ทางเลือก GA หรือยังคงทำการใช้ทางเลือก GA ต่อไปในขณะที่มีการส่งข้อมูล

ค. ค่าโดยปริยาย

หากไม่มีการร้องขอใดๆ จะมีการใช้ทางเลือก GA

2.3.5.4 ทางเลือกการกำหนดชนิดของจอภาพ (Telnet Terminal Type Option)

ทางเลือกนี้มีไว้เพื่อให้เครื่องคอมพิวเตอร์แม่ข่ายตรวจชนิดจอภาพของเครื่องคอมพิวเตอร์ลูกข่าย ดังจะอธิบายรายละเอียดอื่นๆ ได้ดังนี้

ก. ชื่อคำสั่งและค่าฐานสิบ

ทางเลือกนี้มีชื่อคำสั่งคือ TERMINAL-TYPE และมีค่าฐานสิบคือ 24

ข. ความหมายของคำสั่ง

ทางเลือกนี้มีคำสั่งที่ใช้งานและความหมายของแต่ละคำสั่งดังนี้

IAC WILL TERMINAL-TYPE

ผู้ส่งคำสั่งนี้จะทำการส่งชนิดของจอภาพไปในภายหลังที่มีการต่อรองทางเลือกย่อย

IAC WON'T TERMINAL-TYPE

ผู้ส่งคำสั่งนี้จะปฏิเสธที่จะส่งชนิดของจอภาพ

IAC DO TERMINAL-TYPE

ผู้ส่งคำสั่งนี้จะทำการการรับชนิดของจอภาพในภายหลังที่มีการต่อรองทางเลือกย่อย

IAC DON'T TERMINAL-TYPE

ผู้ส่งคำสั่งนี้จะปฏิเสธที่จะรับชนิดของจอภาพ

IAC SB TERMINAL-TYPE SEND IAC SE

เครื่องคอมพิวเตอร์แม่ข่ายจะทำการร้องขอให้เครื่องคอมพิวเตอร์ลูกข่ายทำการส่งชนิดของจอภาพ (ค่าฐานสิบของ SEND คือ 1)

IAC SB TERMINAL-TYPE IS ... IAC SE

เครื่องคอมพิวเตอร์ลูกข่ายส่งชนิดของจอภาพไปยังเครื่องคอมพิวเตอร์แม่ข่าย (ค่าฐานสิบของ IS คือ 0)

ค. ค่าโดยปริยาย

หากไม่มีการร้องขอใดๆ จะไม่มีการส่งชนิดของจอภาพ

ในบทนี้ก็ได้ทราบถึงถึงหลักการสำคัญของวินซ็อค ซึ่งเป็นเครื่องมือที่ใช้ในการพัฒนางานวิจัยนี้ ตลอดจนทราบถึงกลไกการทำงานของโปรโตคอลเทลเน็ต ต่อไปจะกล่าวถึงการวิเคราะห์และออกแบบระบบเพื่อให้รองรับการทำงานของโปรโตคอลเทลเน็ต