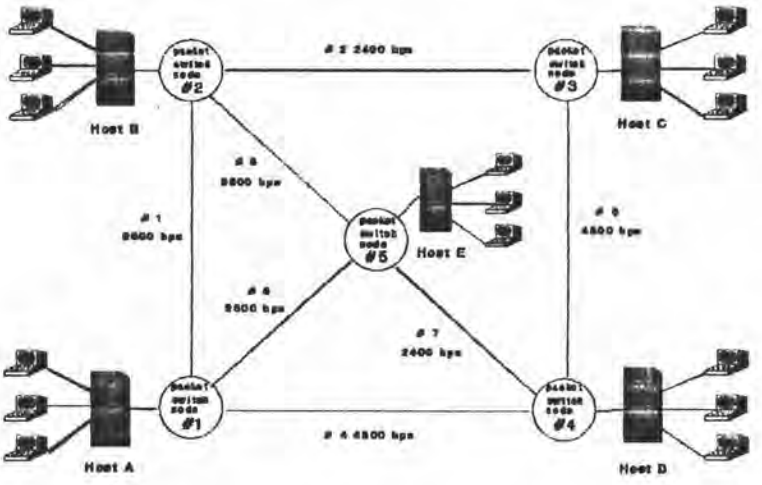
บทที่ 5

โปรแกรมต้นแบบ

5.1 ข้อกำหนดทั่วไป

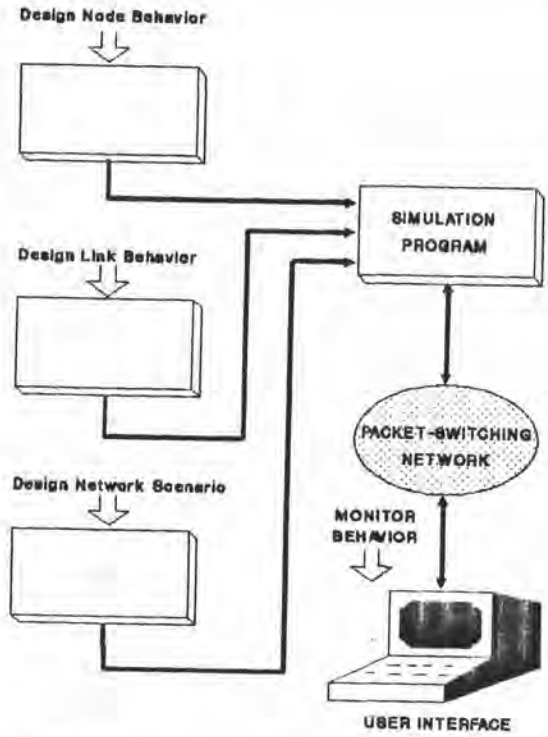
การออกแบบโปรแกรมต้นแบบนี้จะต้องสอดคล้องกับส่วนประกอบแบบจำลองของโนด ตามรูปที่ 3.11 โดยพิจารณาว่าในแต่ละโนดประกอบด้วยหน่วยการทำงาน 3 หน่วย ดังกล่าวไว้ในหัวข้อที่ 3.2 คือ หน่วยเชื่อมต่อกับผู้ใช้ หน่วยเชื่อมต่อกับเครือข่าย และหน่วยควบคุมโนด การทำงานของโนดจะอยู่ภายใต้ขอบเขตข้อกำหนดแบบจำลองของโนดดังกล่าวไว้ในหัวข้อ 3.1.3 และการทำงานของแต่ละหน่วยในโนดจะแสดงด้วยแผนภูมิสถานะดังกล่าวไว้ในหัวข้อ 3.3

โปรแกรมจะประกอบด้วยซอฟต์แวร์โมดูลหลักที่พัฒนาขึ้นเป็นแบบจำลองทางซอฟต์แวร์ เพื่อแทนการทำงานของแบบจำลองของโนดที่สร้างขึ้น และเมื่อนำแบบจำลองของโนดมาประกอบเป็นเครือข่ายดังรูป 5.1 ตามสมมุติฐานที่กำหนดขึ้น ได้เขียนซอฟต์แวร์ควบคุมการทำงานของทุกโนดเพื่อการจำลองการทำงานแบบขนานตามที่กล่าวไว้ในบทที่ 4 รายละเอียดของซอฟต์แวร์โมดูลหลักจะกล่าวในหัวข้อถัดไป



รูปที่ 5.1 การต่อเชื่อมระหว่างโนดต่าง ๆ ในเครือข่าย

การออกแบบโปรแกรมสามารถแบ่งออกเป็น 3 หัวข้อ ดังรูปที่ 5.2 โดยมีรายละเอียดการออกแบบพอสังเขปดังนี้



รูปที่ 5.2 แสดงหัวข้อในการออกแบบโปรแกรม

5.1.1 การออกแบบพฤติกรรมของโนด (Design Node Behavior)

เป็นการออกแบบที่เกี่ยวข้องกับหน้าที่การทำงานของโนดดังนี้

- การจัดการบัฟเฟอร์ของโนด
- การจัดคิวและการเก็บแพ็คเก็ตในบัฟเฟอร์
- การส่งแพ็คเก็ตไปยังหน่วยเชื่อมต่อกับเครือข่ายภาคส่ง
- การตรวจสอบเลขที่โนดปลายทางของแพ็คเก็ต
- การเลือกเส้นทางในการส่งแพ็คเก็ต
- การตรวจสอบชนิดของแพ็คเก็ต
- การตรวจสอบความผิดพลาดของการรับส่งแพ็คเก็ต
- การสร้างและการแปรรหัสคำสั่งของแพ็คเก็ตควบคุม (RR, RNR,ReTx)

5.1.2 การออกแบบพฤติกรรมการเชื่อมโยงข้อมูล

เป็นการออกแบบที่เกี่ยวข้องกับการรับส่งข้อมูล ดังนี้

- การรับแพ็กเก็ตจาก Host
- การส่งแพ็กเก็ตไปยัง Host
- การส่งแพ็กเก็ตไปยังโนดอื่น
- การรับแพ็กเก็ตจากโนดอื่น
- การจัดการความผิดพลาดในการรับแพ็กเก็ตจากโนดอื่น
- การเคลียร์บัฟเฟอร์ของหน่วยเชื่อมต่อกับผู้ใช้และหน่วยเชื่อมต่อกับเครือข่าย

5.1.3 การออกแบบการแสดงผลการทำงานของเครือข่าย (Design Network Scenario)

เป็นการออกแบบการเก็บข้อมูลสถิติที่ได้จากการทำงานของโปรแกรมและการแสดงผลต่าง ๆ ออกทางหน้าจอ โดยมีการเก็บข้อมูลสถิติไว้ 10 หัวข้อ ดังนี้

- Packet Generation
- Network Throughput
- Mean Packet Delay
- Number of Error
- Number of Transmitted Packets
- Number of received Packets
- Link Utilization
- Current Packets in Queue
- Mean Packets in Queue
- Maximum Packets in Queue

การแสดงผลทางหน้าจอสามารถดูผลได้ 2 ลักษณะคือ ภาวะข้อความ (text mode) และภาวะกราฟิก (graphic mode) สำหรับภาวะข้อความสามารถดูพฤติกรรมการทำงานของแต่ละโมดอย่างละเอียด ส่วนภาวะกราฟิกเป็นการดูผลการทำงานของแต่ละโมดอย่างสรุป รายละเอียดในการแสดงผลจะกล่าวในภายหลังใน บทที่ 6

5.2 ข้อกำหนดเบื้องต้นในการสร้างซอฟต์แวร์โมดูล

5.2.1 ข้อกำหนดพื้นฐาน

ในหัวข้อนี้จะกล่าวถึงข้อกำหนดในการสร้างซอฟต์แวร์โมดูล และหลักในการสร้างข้อมูลเชิงวัตถุ (Object)

เหตุผลในการสร้างซอฟต์แวร์โมดูลของแบบจำลองของโมดให้มีโครงสร้างข้อมูลเชิงวัตถุ ก็เพื่อกำหนดโครงสร้างต้นแบบสำหรับแต่ละหน่วยได้นำไปใช้ โดยไม่ต้องคำนึงถึงราย ละเอียดโครงสร้างภายในของต้นแบบ เป็นการมองโปรแกรมแต่ละโมดูลเป็นวัตถุ ทางด้านภาษาแบบเชิงวัตถุจะกำหนดต้นตระกูลขึ้นมาเป็นแม่แบบสำหรับผู้สืบสกุลต่าง ๆ ไปขยายระเบียบวิธี (Method) ของตนเอง และสามารถสืบทอดต่อไปได้ ในการพัฒนาเราใช้ภาษาเทอร์โบปาสคาล มีการกำหนดแนวทางการเขียนโปรแกรมเชิงวัตถุมีลักษณะคล้ายระเบียน (Record) ต่างกันตรงที่เขตข้อมูล (Field) ของภาษาเชิงวัตถุอาจเป็นข้อมูลหรือโปรแกรมย่อยก็ได้ ให้กำหนดไว้เฉพาะส่วนหัวเท่านั้น และให้มีความสามารถต่าง ๆ ดังนี้

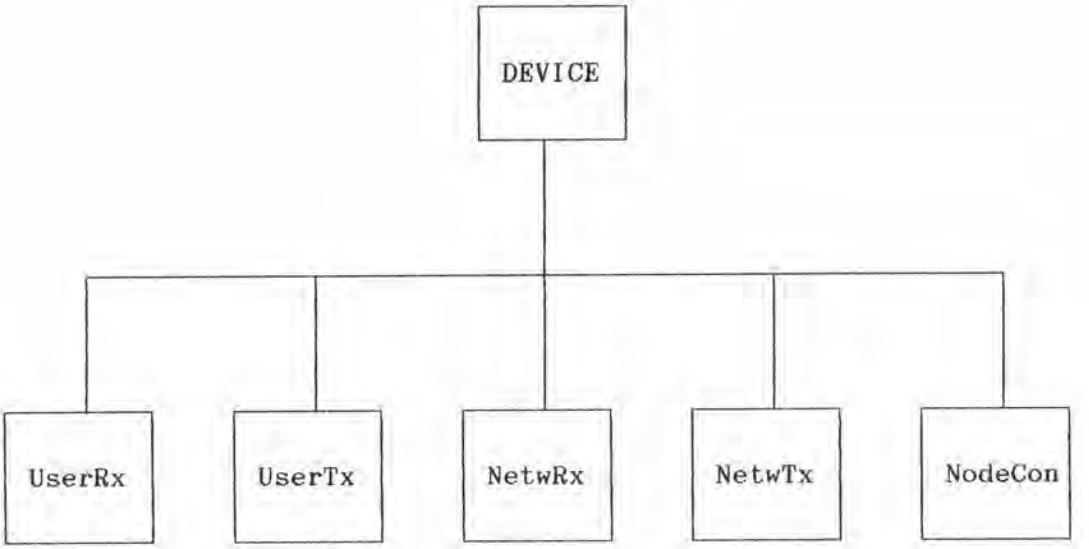
- ให้สืบทอดกันได้ เช่น โมดูล UserRx สืบทอดมาจาก โมดูล Device
- เขตข้อมูลที่เป็นข้อมูล คือ ลักษณะและเขตข้อมูลที่เป็นโปรแกรมย่อย คือ ระเบียบวิธี (Method) หรือพฤติกรรม (Behavior)
- เมื่อกำหนดส่วนหัวของโปรแกรมย่อยที่เป็นระเบียบวิธีแล้ว ให้เขียนโปรแกรมย่อยนั้นโดยใช้ลักษณะ ดังนี้

  ชื่อวัตถุ.ชื่อโปรแกรมย่อย

- ผู้สืบสกุลสามารถใช้ข้อมูลและโปรแกรมย่อยของบรรพบุรุษ (Ancestor) ได้แต่บรรพบุรุษจะใช้ข้อมูลและโปรแกรมย่อยของผู้สืบสกุลไม่ได้

- ลักษณะของแต่ละวัตถุถือเป็นข้อมูลแบบตัวแปรร่วม (Global) จะคงอยู่ตลอดการใช้โปรแกรม แต่จะเป็นตัวแปรเฉพาะที่ (Local variable) ของตระกูลนั้น คือจะรู้จักกันในผู้สืบทอดเท่านั้น

ความสามารถในภาษาเชิงวัตถุจะมีประโยชน์และมีส่วนช่วยให้การเขียนซอฟต์แวร์โมดูลสำหรับแบบจำลองของโนดไม่ต้องคำนึงถึงรายละเอียดของข้อมูลหรือโปรแกรมย่อย สามารถสืบทอดและขยายขีดความสามารถเพิ่มเติมได้ง่าย ในการพัฒนาโปรแกรมเราจึงกำหนดต้นตระกูล คือ Device ให้มีโครงสร้างเชิงวัตถุ และผู้สืบทอดระดับที่ 1 มี UserRx, UserTx, NetwRx, NetwTx, NodeCon ดังรูปที่ 5.3



รูปที่ 5.3 โครงสร้างเชิงวัตถุ และผู้สืบทอดระดับที่ 1

5.2.2 โครงสร้างข้อมูลเชิงวัตถุของ Device

ในส่วนโครงสร้างข้อมูลของ Device จะประกอบด้วย

- Event Time เป็นข้อมูลของเวลาที่ถูกใช้ไปในการทำกิจกรรมหนึ่ง เพื่อแต่ละหน่วยจะเป็นผู้ตรวจสอบค่าของเวลา Used Time ว่าถูกใช้ครบตามเวลาที่ระบุไว้สำหรับแต่ละกิจกรรมแล้วหรือยัง

- Ready เป็นข้อมูลแบบตรรกะบอกสภาวะความพร้อมในการทำงานของแต่ละหน่วยในโนด

- Buffer เป็นพื้นที่เก็บข้อมูลแพ็กเก็ตชั่วคราวเพื่อรอการถ่ายแพ็กเก็ตไปให้หน่วยอื่น โดยมี PktPtr เป็นตัวชี้

- HavePkt เป็นข้อมูลบ่งชี้สถานะ ว่าหน่วยนั้นมีแพ็กเก็ตอยู่

- Msg เป็นข้อมูลชนิดสายอักขระ

- โปรแกรมย่อย writemsg เป็นโปรแกรมที่เขียนข้อความชนิดอักขระออกหน้าจอในบรรทัดที่กำหนด

- โปรแกรมย่อย writeevent เป็นโปรแกรมที่เขียนข้อความชนิดอักขระเช่นเดียวกับโปรแกรมย่อย writemsg แต่จะเขียนออกหน้าจอภายในหน้าต่างที่กำหนด โดยตรวจสอบว่าจะแสดงผลหน้าจอเป็นวินโดใด

- โปรแกรมย่อย writedevmsg เป็นโปรแกรมย่อยที่ใช้เขียนข้อความชนิดอักขระออกหน้าจอภายในกรอบหน้าต่าง สำหรับแสดงสถานะการทำงานของแต่ละหน่วยของวินโดนั้น

โครงสร้างข้อมูลของ Device แสดงได้ดังนี้

```
Device  = Object
              EventTime : Real;
              Ready : Boolean;
              Buffer : Packet;
              HavePkt : Boolean;
              msg : string;
              procedure  writemsg
              procedure  writeevent
              procedure  writedevmsg
          end;
```

5.2.3 การสร้างแพ็กเก็ตอย่างสุ่ม

ในหัวข้อนี้จะกล่าวถึงข้อกำหนดในการสร้างแพ็กเก็ต โครงสร้างข้อมูลของแพ็กเก็ต พารามิเตอร์ต่าง ๆ ที่ใช้ในการจำลองการทำงานของเครือข่ายแพ็กเก็ตที่สร้างขึ้นจะสอดคล้องกับแพ็กเก็ตที่ใช้รับส่งข้อมูลจริง ดังได้กล่าวไปแล้วในหัวข้อโครงสร้างของแพ็กเก็ต บทที่ 2 หัวข้อ 2.2.3

(ก) ข้อกำหนดในการสร้างแพ็คเก็ต

- การกำหนดชนิดของแพ็คเก็ต มีอยู่ 2 ชนิด คือ แพ็คเก็ตข้อมูล และแพ็คเก็ตควบคุมแพ็คเก็ตควบคุมมีอยู่ 3 ลักษณะ ได้แก่ RR Packet, RNR Packet, ReTx Packet

- การกำหนดโนดต้นทางผู้ส่งและโดนปลายทางผู้รับ

- การกำหนดเลขที่แต่ละแพ็คเก็ต

(ข) โครงสร้างข้อมูลของแต่ละแพ็คเก็ต

โครงสร้างข้อมูลของแพ็คเก็ตข้อมูลสร้างโดยเทคนิคเชิงวัตถุ เพื่อให้แต่ละโนดสามารถใช้โครงสร้างข้อมูลนี้ร่วมกันได้ แต่ละแพ็คเก็ตจะต่อเป็นแบบเชื่อมโยง มีตัวแปร Next เป็นตัวชี้ไปยังแพ็คเก็ตต่อไป ตัวแปร SA เป็นชื่อของโนดต้นทางผู้ส่ง ตัวแปร DA เป็นชื่อของโนดปลายทางผู้รับ ตัวแปร Name เป็นการกำหนดชื่อและเลขที่ของแต่ละแพ็คเก็ตโดย ประกอบด้วยพารามิเตอร์ชนิดคำ (Word) ดังนี้ Group, No.และ TotalPktInGroup ในส่วนของ Group และ No ถือเป็นเลขที่ของแพ็คเก็ต

ในการสร้างแพ็คเก็ตอย่างสุ่มขึ้นมาแต่ละครั้งตามจำนวนเท่ากับ TotlaPktInGroup แพ็คเก็ตชุดนั้นจะถูกตั้งชื่อเลขที่กลุ่มและลำดับที่ของ แต่ละแพ็คเก็ตในโปรแกรมมีการกำหนดค่า MaxPktInGroup เป็นค่าคงที่ที่กำหนดขึ้นมาว่าจะสามารถสร้าง แพ็คเก็ตจำนวนสูงสุดที่สามารถสร้างได้

สมมุติกำหนดค่า MaxPktInGroup เท่ากับ 4 แสดงว่าในการสร้างแพ็คเก็ตขึ้นมาอย่างสุ่มในแต่ละครั้งจะสร้างแพ็คเก็ตได้ตั้งแต่ 1-4 แพ็คเก็ต เมื่อสุ่มค่า TotalPktInGroup (ตั้งแต่ 1-4 แพ็คเก็ต) สมมุติได้เท่ากับ 3 การสร้างแพ็คเก็ตก็จะกำหนดค่าเลขที่ Group จะเพิ่มครั้งละ 1 ต่อการสร้างแพ็คเก็ต 1 ชุด จะได้แพ็คเก็ตต่าง ๆ ดังนี้คือ G1N1, G1N2 และ G1N3 และแพ็คเก็ตชุดต่อไปก็จะมีเลขที่ Group เท่ากับ 2 และถ้า TotalPktInGroup = 4 จะได้แพ็คเก็ตต่าง ๆ ดังนี้คือ G2N1, G2N2, G2N3, G2N4 เป็นต้น

(ค) พารามิเตอร์ที่ใช้ได้แก่

- ตัวแปร PktType มี 2 ชนิด คือ Control และ Data

- ตัวแปร Control Code มี 4 ชนิด คือ ReceiveReady, ReceiveNotReady และ ReTxPkt Received Ready มี 2 ลักษณะดังกล่าวไว้ในหัวข้อ 3.2

- ตัวแปร Life Time คือตัวแปรที่เก็บค่าอายุของแพ็คเก็ตที่อยู่ในเครือข่ายเป็นตัวแปรชนิดจำนวนจริง

- ตัวแปรบ่งชี้ Error เป็นตัวแปรชนิดตรรกะแบบบูล ว่าเป็นจริงหรือเท็จ ถ้าเป็นจริงแสดงว่า แพ็คเก็ตนั้นมีความผิดพลาดเกิดขึ้น

โครงสร้างข้อมูลของแพ็คเก็ตแสดงในรูปโปรแกรมได้ดังนี้

```
Type
      TypeofPkt = (Control, Data) ;
      ControlCodeType = (ReceiveReady, ReceiveNotReady, ReTxPkt) ;
      PacketPtr       = ^Packet ;
      Packet          = object
        Next  :  PacketPtr ;
        SA,DA :  NodeName ;
        Group,No,TotalPktInGroup : word ;
        name  :  String[10] ;
        ControlCode : ControlCodeType ;
        GCode,NCode : word ;
        PktType     : TypeOfPkt ;
        LifeTime    : Real ;
        Error       : Boolean ;
      end ;
```

5.3 โครงสร้างข้อมูลซอฟแวร์โมดูลของหน่วยเชื่อมต่อกับผู้ใช้

5.3.1 โครงสร้างข้อมูลซอฟต์แวร์โมดูลของหน่วยเชื่อมต่อกับผู้ใช้ภาครับ

หน่วยเชื่อมต่อกับผู้ใช้ภาครับมี 3 สถานะการทำงาน ดังได้กล่าวไปแล้วในหัวข้อ 3.3.2 (ก) "การทำงานของหน่วยเชื่อมต่อกับผู้ใช้ภาครับ" ดังนั้นในส่วนของโครงสร้างข้อมูลนี้ นอกจากจะมีโปรแกรมย่อยของสถานะการทำงานทั้ง 3 สถานะแล้ว ยังจะต้องมี Event เป็นตัวบ่งชี้สถานะการทำงานของหน่วยเชื่อมต่อกับผู้ใช้ภาครับในขณะเวลานั้น ๆ และมีโปรแกรมย่อย ExecEvent ทำหน้าที่ในการตรวจสอบว่า สถานะการทำงานต่อไปจะอยู่ที่สถานะใดตามตัวบ่งชี้ Event แล้วก็จะเรียกโปรแกรมย่อยของสถานะนั้น มาทำงานต่อไป

โครงสร้างข้อมูลของหน่วยเชื่อมต่อกับผู้ใช้ภาครับมีรายละเอียด ดังนี้

```
Type
    UserRxEvent   =  (URxWait, URxRxPkt, URxWaitNodeCon);
    UserRx  =  Object (Device)
                        Event : UserRxEvent;
                        Procedure ExecEvent (Event : UserRxEvent);
                        Procedure Wait;
                        Procedure RxPkt (Source : NodeName);
                        Procedure WaitNodeCon;
               End;
```

5.3.2 โครงสร้างข้อมูลซอฟแวร์โมดูลของหน่วยเชื่อมต่อกับผู้ใช้ภาคส่ง

หน่วยเชื่อมต่อกับผู้ใช้ภาคส่งมี 2 สถานะการทำงาน ดังได้กล่าวไปแล้วในหัวข้อ 3.3.2 (ข) "การทำงานของหน่วยเชื่อมต่กับผู้ใช้ภาคส่ง" ดังนั้นในส่วนของโครงสร้างของข้อูลนี้ นอกจากจะมีโปรแกรมย่อยของสถานะการทำงานทั้ง 2 สถานะแล้ว ยังต้องมี Event เป็นตัวบ่งชี้สถานะการทำงาน และโปรแกรมย่อย ExecEvent ทำหน้าที่ในการตรวจสอบสถานะการทำงานและเรียกโปรแกรมย่อยของสถานะนั้นมาทำงาน โครงสร้างข้อมูลของหน่วยเชื่อมต่อกับผู้ใช้ภาคส่ง มีรายละเอียดดังนี้

```
Type
    UserTxEvent = (UTxWait, UTxTxPkt);
    UserTx  = Object (Device)
                    Event : UserTxEvent;
                    Procedure ExecEvent (Event:UserTxEvent);
                    Procedure TxPkt;
                    Procedure Wait;
                end;
```

5.4 โครงสร้างข้อมูลซอฟแวร์โมดูลของหน่วยเชื่อมต่อกับเครือข่าย

5.4.1 โครงสร้างข้อมูลซอฟแวร์โมดูลของหน่วยเชื่อมต่อกับเครือข่ายภาครับ

หน่วยเชื่อมต่อกับเครือข่ายภาครับมี 3 สถานะการทำงาน ดังได้กล่าวไปแล้วในหัวข้อ 3.3.3 (ก) "การทำงานของหน่วยเชื่อมต่อกับเครือข่ายภาครับ" ดังนั้นในส่วนของโครงสร้างข้อมูลนี้ นอกจากจะมีโปรแกรมย่อยของสถานะการทำงานทั้ง 3 สถานะแล้ว ยังมี Event เป็นตัวบ่งชี้สถานะการทำงานและโปรแกรมย่อย ExecEvent ทำหน้าที่ในการตรวจสอบสถานะการทำงานและเรียกโปรแกรมย่อมของสถานะการทำงานนั้นมาทำงาน โครงสร้างข้อมูลของหน่วยเชื่อมต่อกับเครือข่ายภาครับ มีรายละเอียด ดังนี้

```
Type
    NetwRxEvent  = (NRxWait, NRxRXPkt, NRxWaitNodeCon);
    NetwRx  =  Object (Device)
                    Event : NetwRxEvent;
                    Procedure ExecEvent (Event : NetwRxEvent);
                    Procedure Wait;
                    Procedure RxPkt;
                    Procedure WaitNodeCon;
                end;
```

5.4.2 โครงสร้างข้อมูลซอฟแวร์โมดูลของหน่วยเชื่อมต่อกับเครือข่ายภาคส่ง

หน่วยเชื่อมต่อกับเครือข่ายภาคส่งมี 3 สถานะการทำงาน ดังได้กล่าวไปแล้วในหัวข้อ 3.3.3 (ข) "การทำงานของหน่วยเชื่อมต่อกับเครือข่ายภาคส่ง" ดังนั้นในส่วนของโครงสร้างข้อมูลนี้ นอกจากจะมีโปรแกรมย่อยของสถานะการทำงานทั้ง 3 สถานะแล้ว ยังมี Event เป็นตัวบ่งชี้สถานะการทำงาน และโปรแกรมย่อย ExectEvent ทำหน้าที่ในการตรวจสอบสถานะการทำงาน และโปรแกรมย่อยของสถานะการทำงานนั้นมาทำงาน โครงสร้างข้อมูลของหน่วยเชื่อมต่อกับเครือข่ายภาคส่ง มีรายละเอียด ดังนี้

```
Type
     NetwTxEvent = (NTxWait, NTxTxPkt, NTxEndSending);
     NetwTx      = Object (Device)
                         Event : NetwTxEvent;
                         procedure ExecEvent (Event:NetwTxEvent);
                         procedure Wait;
                         procedure TxPkt;
                         procedure EndSending;
                   end;
```

5.5 โครงสร้างข้อมูลซอฟแวร์โมดูลของหน่วยควบคุมโนด

หน่วยควบคุมโนดมี 10 สถานะการทำงาน ดังได้กล่าวไปแล้วในหัวข้อ 3.3.4 "การทำงานของหน่วยควบคุมโนด"ดังนั้นในส่วนของโครงสร้างข้อมูลนี้ นอกจากจะมีโปรแกรมย่อยของสถานะการทำงานทั้ง 10 แล้ว ยังมี Event เป็นตัวบ่งชี้สถานะการทำงาน และโปรแกรมย่อย ExecEvent ทำหน้าที่ในการตรวจสอบสถานะการทำงานและเรียกโปรแกรมย่อยของสถานะการทำงานนั้นมาทำงาน และก็ยังมีตัวแปร ChkQtN, Con_is_working_at_port และ Routing_assign_port_no เป็นชื่อของโนด โดยมีตัวแปร ChkQtN สำหรับไว้ตรวจสอบว่า โนดนั้นมีช่องสื่อสารใดต่ออยู่และ Con_is_working_at_port จะแสดงถึงว่าหน่วยควบคุมโนดกำลังทำงานหรือตรวจสอบว่าอะไรอยู่ที่ช่องสื่อสารนั้น และ Routing_assign_port_no จะแสดงถึงการนำแพ็กเก็ตนั้นไปต่อคิวที่ช่องสื่อสารนั้นและ NRxQuit เป็นตัวแปรแบบชนิดตรรกะที่

แจ้งว่าหน่วยเชื่อมต่อกับเครือข่ายภาครับทำงานได้หรือไม่ และ PktFromUser เป็นตัวแปรแบบชนิดตรรกะที่แจ้งว่าแพ็คเก็ตนั้นรับจากผู้ใช้หรือไม่

โครงสร้างข้อมูลของหน่วยควบคุมโนด มีรายละเอียดดังนี้

```
Type
    NodeConEvent = (Nodemanager,ChkError,ChkPktType, Prerouting,
                    ChkAvailable, CallRouting, InsertPktInQ,
                    SendConPkt, ProcessConPkt, ClearBuffer);
    NodeCon  = Object (Device);
                    Event : NodeConEvent;
                    ChkQtN : NodeName;
                    Con_is_working_at_port : NodeName;
                    Routing_assign_port_no : Nodename;
                    NRxQuit : Boolean;
                    PktFromUser : Boolean;
                    procedure  ExecEvent (Event:NodeConEvent);
                    procedure NodeManager;
                    procedure MovePkt;
                    procedure ChkError;
                    procedure ChkPktType;
                    procedure Prerouting;
                    procedure ChkAvailable;
                    procedure CallRouting;
                    procedure InsertPktInQ;
                    procedure SendConPkt;
                    procedure ProcessConPkt;
                    procedure ClearBuffer;
                end;
```

5.6 การเชื่อมประสานระหว่างโปรแกรมกับผู้ใช้

ในการศึกษาถึงการทำงานของเครือข่ายการสื่อสารข้อมูลแบบแพ็กเก็ตนั้น ขึ้นอยู่กับองค์ประกอบหน้าที่การทำงานของโนด (Node Function) และหน้าที่ในการติดต่อสื่อสาร (Link Function) โดยมีพารามิเตอร์ต่าง ๆ เป็นตัวกำหนด ซึ่งมีผลต่อข้อมูลสถิติต่าง ๆ ที่ได้จากการทำงานของโปรแกรม ดังนั้นจึงได้มีการออกแบบโปรแกรมให้สามารถเปลี่ยนแปลงค่าพารามิเตอร์ต่างๆ เพื่อศึกษาและวิเคราะห์การทำงาน 2 ทาง ดังรูปที่ 5.3 โดยมีรายละเอียดพอสังเขปดังนี้

การเชื่อมประสานระหว่างโปรแกรมกับผู้ใช้ได้มีการอำนวยความสะดวกสำหรับผู้ใช้ที่ไม่มีความรู้เบื้องต้นในการใช้ภาษาชั้นสูงให้สามารถแก้ไขพารามิเตอร์ได้โดยสะดวกไม่ต้องทำการคอมไพล์โปรแกรมใหม่ โดยมีพารามิเตอร์ต่าง ๆ ดังนี้

- Packet Size เป็นขนาดของแพ็กเก็ต มีหน่วยเป็น บิทต่อแพ็กเก็ต
- Maximum Storage เป็นขนาดของบัฟเฟอร์ มีหน่วยเป็นแพ็กเก็ตต่อโนด
- Maximum Packet in Group เป็นขนาดสูงสุดของการสร้างแพ็กเก็ตให้แพ็กเก็ตทีละหลาย ๆ แพ็กเก็ตต่อการสุ่มหนึ่งครั้ง แล้วจึงส่งเข้าไปในเครือข่าย ซึ่งขึ้นอยู่กับสถานะการทำงานของโนดนั้น ๆ เช่น มีเนื้อที่ในบัฟเฟอร์ของโนดว่างพอที่จะให้บริการหรือยัง

นอกจากนี้ผู้ใช้ยังสามารถที่จะกำหนดลักษณะการทำงานของการทำงานของโปรแกรมแบบภาวะข้อความ ดังนี้

- การกำหนดช่วงเวลาของการทำงานของโปรแกรม
- การทำงานของโปรแกรมแบบปกติโดยอัตโนมัติ
- การทำงานของโปรแกรมเป็นขั้นตอน (Step Run) ให้สามารถทำการรันโปรแกรมในลักษณะดูผลการทำงานและการเปลี่ยนสถานะทีละอุบกรณ์
- การหน่วงเวลาการทำงานของโปรแกรม (Program Delay Time) ให้หน่วงเวลาในการแสดงผลออกหน้าจอให้ช้าลงเท่านั้นไม่มีผลกระทบต่อการจำลอง
- การกำหนดเวลาที่ต้องการสังเกตในแต่ละครั้ง (Observed Time) เป็นความต้องการให้โปรแกรมทำงาน โดยแต่ละอุบกรณ์จะทำงานที่ละช่วงเวลาที่ต้องการสังเกต เพื่อดูผลการเปลี่ยนสถานะในช่วงเวลาที่เป็นคาบ ๆ ที่ละคาบ

- การเลือกการแสดงผลทางหน้าจอของข้อมูลสถิติในภาวะข้อความ โดยผู้ใช้สามารถเลือกหัวข้อการแสดงผลที่สนใจต่อการศึกษาได้ 3 หัวข้อจากรายการต่าง ๆ ที่แสดงไว้ 10 หัวข้อ สามารถเลือกหรือเปลี่ยนได้ตลอดเวลาขึ้นอยู่กับความสนใจของผู้ใช้ ดังจะกล่าวในรายละเอียดใน บทที่ 6