COMPUTER NETWORKS

- Chapter 3. Data Link Layer 3

王昊翔
WANG Haoxiang
hxwang@scut.edu.cn

School of Computer Science & Engineering
国家双语教学试点项目 广东省精品课

Contents of this lecture

- Learn protocol verification
  - Finite state machine models
  - Petri net models
- Learn Example DLL protocol
  - HDLC
  - PPP

Protocol verification

- Due to the complexity of various protocols, it is important that they are properly verified for correctness.
- The verification should also determine if it is possible for deadlocks or other problems to occur in the protocol.
- There are several different methods of protocol verification.
  - Finite state machine models
  - Petri net models

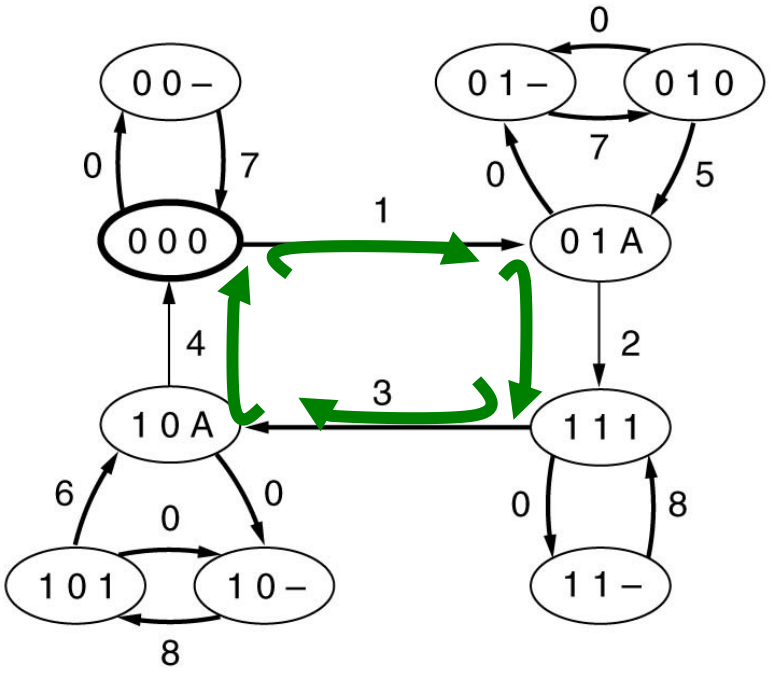
Finite State Machine Models

- Each protocol machine (i.e., sender or receiver) is always in a specific state at every instant of time.
  - All the states are denoted as nodes.
- The state of the complete system is the combination of all the states of the two protocol machines and the channel.
- From each state, there are zero or more possible transitions to other states. Transitions occur when some event happens.
  - All the transitions are denoted as directed arcs.
- Initial state corresponds to the description of the system when it starts running, or at some convenient starting place shortly thereafter.
- Reachability analysis
  - which states are reachable and which are not
  - detect a variety of errors in the protocol specification

An Example Of A Finite State Machine Model

- Protocol 3: 2 protocol machine and channel, total 16 states
- Each state is labeled by three characters, SRC
- S is
  - 0: frame 0 is sent
  - 1: frame 1 is sent
- R is
  - 0: frame 0 is expected
  - 1: frame 1 is expected
- C is
  - 0: frame 0 is on the channel
  - 1: frame 1 is on the channel
  - A: ack frame is on the channel
  - -: the channel is empty

An Example Of A Finite State Machine Model (cont'd)



(a)

Transition	Who runs?	Frame accepted	Frame emitted	To network layer
0	–	(frame lost)		–
1	R	0	A	Yes
2	S	A	1	–
3	R	1	A	Yes
4	S	A	0	–
5	R	0	A	No
6	R	1	A	No
7	S	(timeout)	0	–
8	S	(timeout)	1	–

(b)

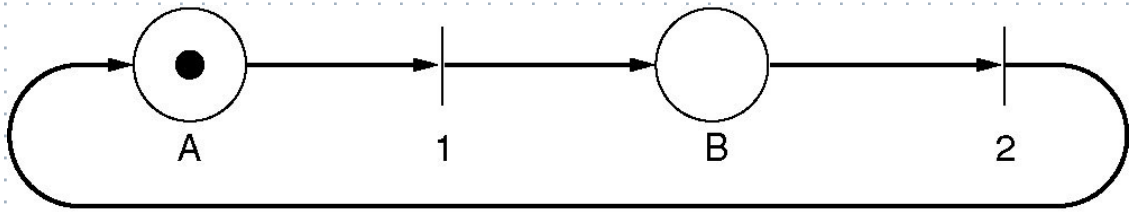
- **(a) State diagram for protocol 3. (b) Transmissions.**

Petri Net Models

- A Petri net has four basic elements: places(库所), transitions（变迁）, arcs（弧）, and tokens（标记）.
  - A place represents a state which (part of) the system may be in. (circle)
  - The current state indicated by the token (heavy dot)
  - A transition is indicated by a horizontal or vertical bar.
  - Each transition has zero or more input arcs coming from its input places, and zero or more output arcs, going to its output places.

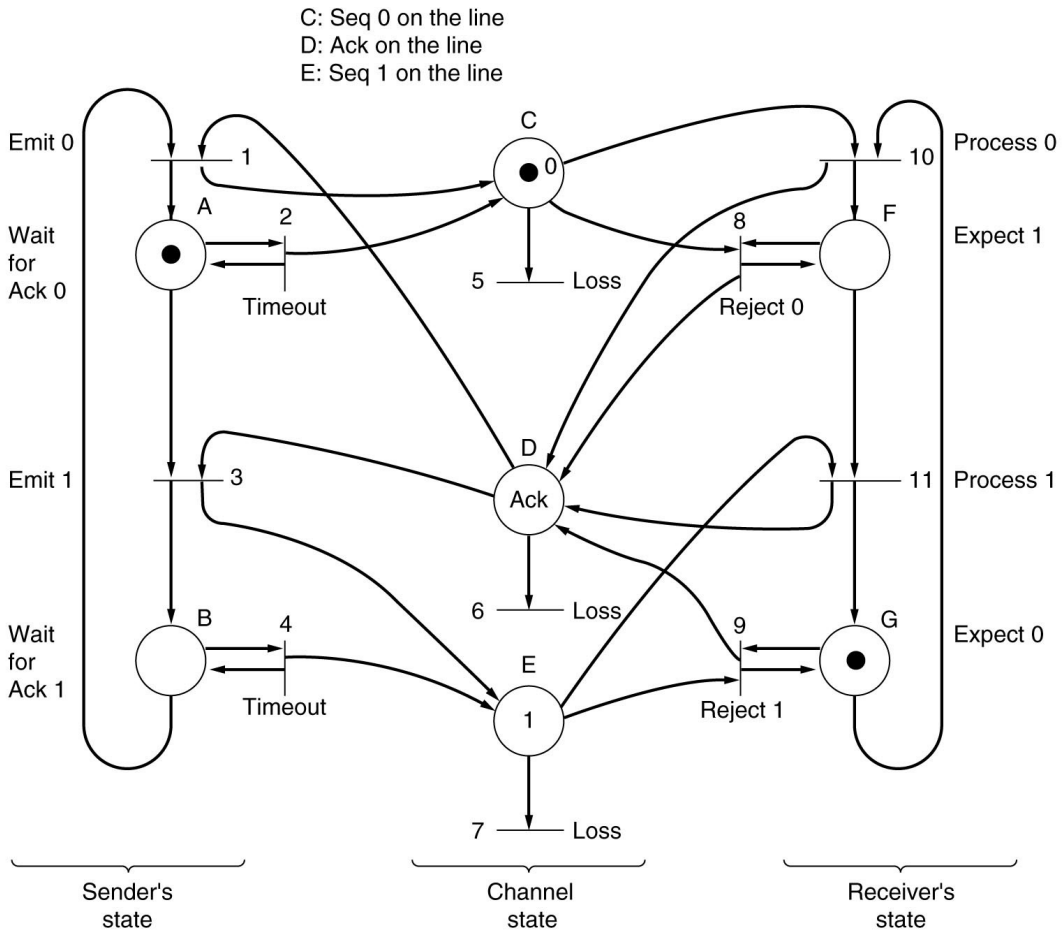
Petri Net Models (cont'd)

- A transition is **enabled** （激活的）if there is at least one input token in each of its input places.
- Any enabled transition may **fire** （激发）at will, removing one token from each input place and depositing a token in each output place.
- Petri net can be used to detect protocol failures in a way similar to the use of finite state machines.
- Petri net can be represented in convenient algebraic form（代数形式） resembling a grammar.



A Petri net with two places and two transitions.

A Petri Net Model For Protocol 3



1: BD → AC
2: A → A
3: AD → BE
4: B → B
5: C →
6: D →
7: E →
8: CF → DF
9: EG → DG
10: CG → DF
11: EF → DG

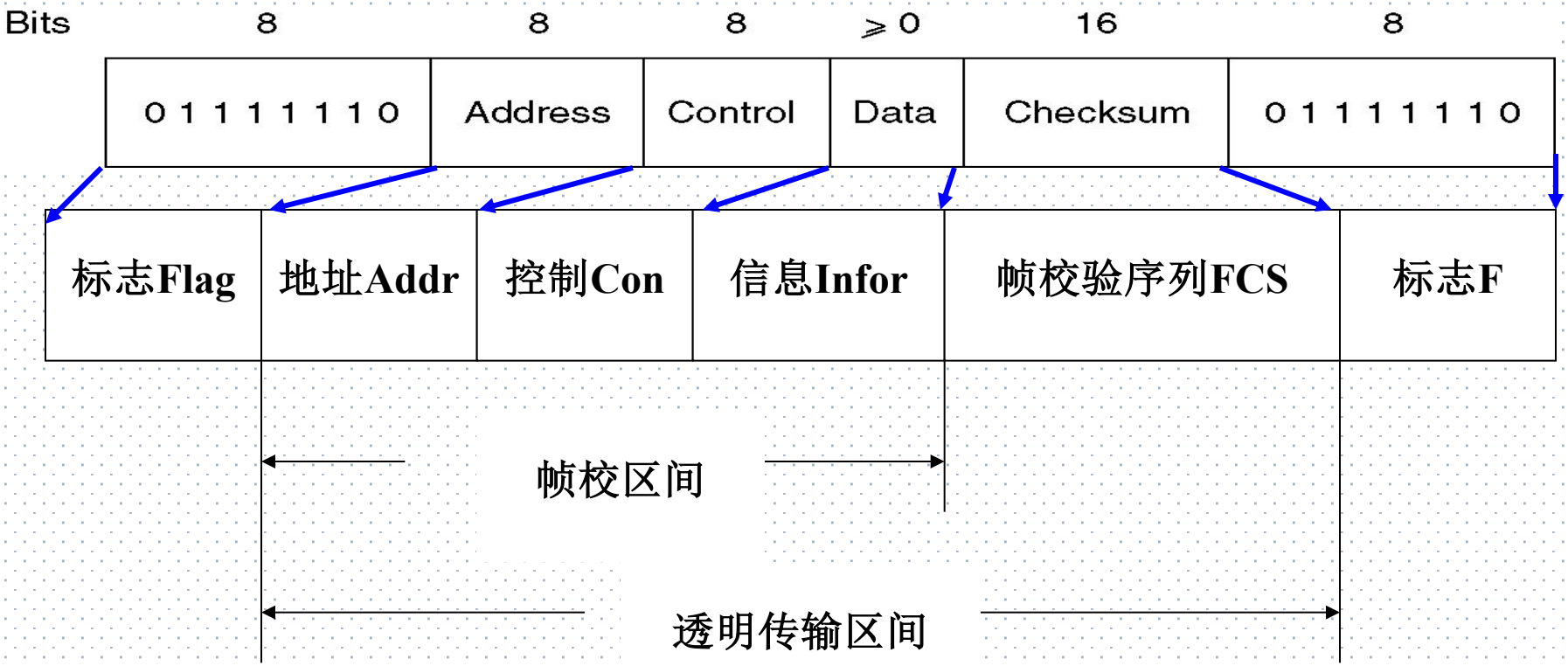
Example DLL protocol

- HDLC
  - High-Level Data Link Control
- PPP
  - The Point-to-Point Protocol

History of HDLC

- IBM introduced SDLC – Synchronous Data Link Control – and submitted it to ANSI and ISO for acceptance as US and International standards.
- ANSI modified it to be ADCCP – Advanced Data Communication Control Procedure
- ISO modified it to be HDLC – High-level Data Link Control.
- CCITT modified HDLC for its LAP (Link Access Procedure) but later modified it again to LAPB.
- They are all very similar, with only minor (but annoying) differences between them.

HDLC frame-structure

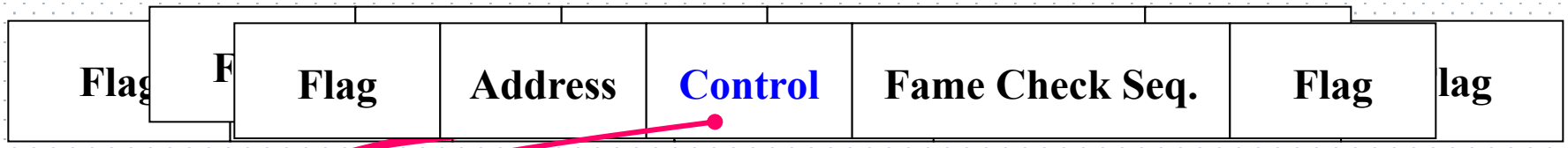


HDLC frame-structure(cont'd)

- **Flag sequence**
  - Identify start or end of the frame
  - Bit stuffing for transparency.
- **Address field**
  - Identify one of the terminals (on lines with multiple terminals)
  - Distinguish commands from response (for point-to-point lines)
- **Control field**
  - For sequence numbers, acknowledgements, and other purposes

HDLC frame-structure(cont'd)

- **Control field**
  - **For sequence numbers, acknowledgements, and other purposes**



Bit Seq. No.	1	2	3	4	5	6	7	8
I frame	0	N（S）			P/F	N（R）		
S frame	1	0	S		P/F	N（R）		
U frame	1	1	M		P/F	M		

HDLC – Frame Type

- **Frame Types**
  - **Information Frame(信息帧)**
  - **Supervisory Frame (监控帧)**
  - **Unnumbered Frame (无编号帧)**
- **The contents of the Control field for three kind frames**

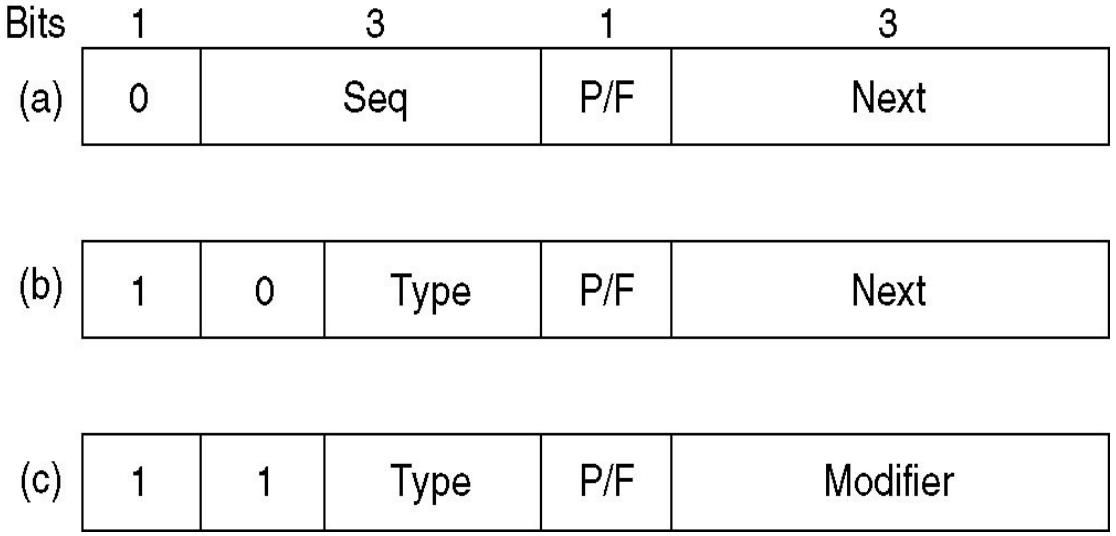


Figure 3-25 Control field of
(a) An Information frame.
(b) A Supervisory frame.
(c) An Unnumbered frame.

HDLC – Information Frame

- Information Frame(信息帧)
  - Seq :N(S)
    - Sending frame sequence number
  - Next:N(R)
    - Piggybacked acknowledgement
      - Piggybacking the number of the first frame not yet received (i.e., the next frame expected), not the number of the last frame received correctly.
  - P/F
    - Poll/Final(查询/结束)
    - Used when a computer is polling a group of terminals

HDLC – Supervisory Frame

- Supervisory Frame (监控帧)
  - Type 0 (bit3-4: 0 0) receive ready
    - RR frame= acknowledgement frame
    - Used when there is no reverse traffic to use for piggybacking
  - Type 1 (bit3-4: 0 1)，like protocol5
    - RNR=Negative acknowledgement frame
    - The Next field indicates the first frame in sequence not received correctly
  - Type 2 (bit3-4: 1 0):RECEIVE NOT READY
    - Acknowledges all frames up to but not including Next
    - Tells the sender to stop sending
  - Type 3 (bit3-4: 1 1):SELECTIVE REJECT
    - Calls for retransmission of only the frame specified

HDLC – Frame structure

Flag	Address	Control	Information	Fame Check Seq.	Flag

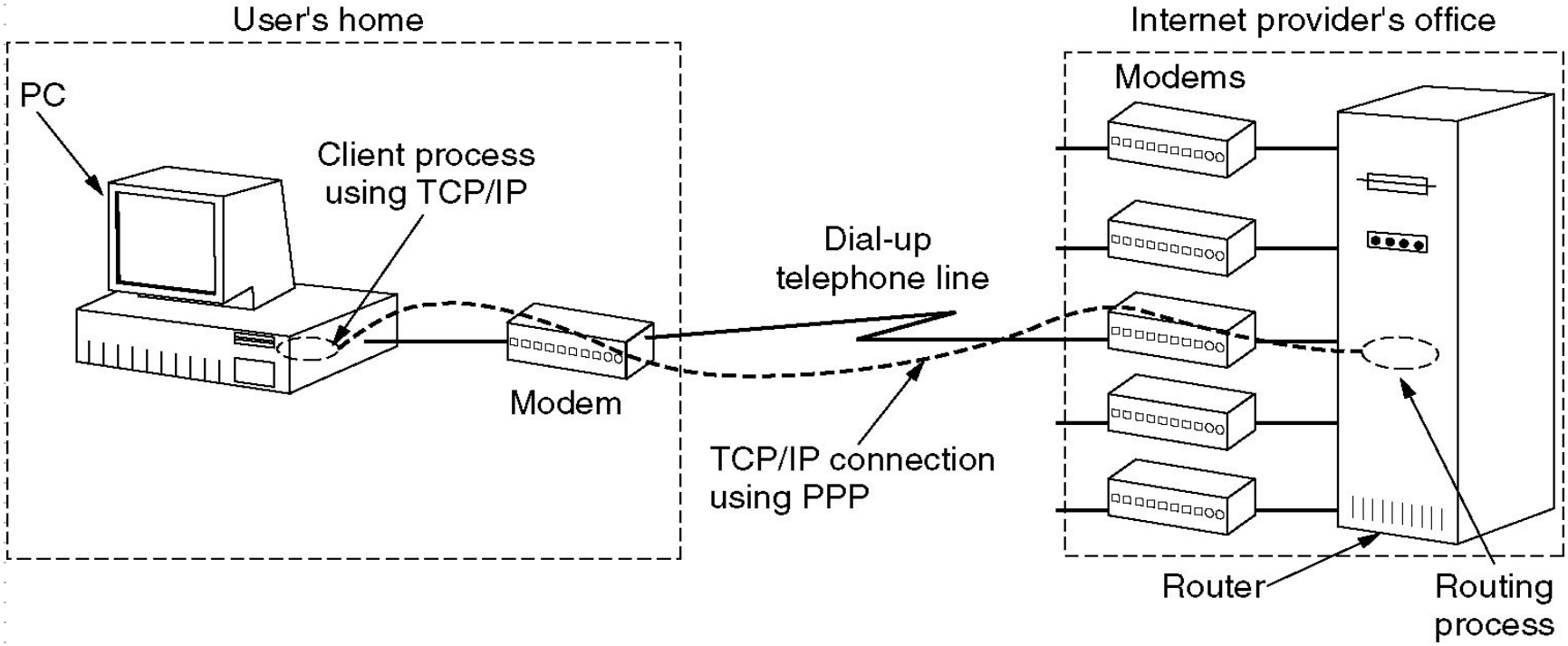
- Frame Structure
  - Data field
    - Contain any information
    - May be arbitrarily long
    - The efficiency of the checksum falls off with increasing frame length
  - Checksum field
    - Cyclic redundancy code: 16bit – CRC: $x^{16}+x^{12}+x^5+1$

Three Commands Provided By Protocols

- **DISC (DISConnect) - allows a machine to announce that it is going down (e.g., for preventive maintenance).**
- **SNRM (Set Normal Response Mode) - allows a machine that has just come back on-line to announce its presence and force all the sequence numbers back to zero.**
  - **HDLC and LAPB have an additional command,** SABM **(Set Asynchronous Balanced Mode).**
  - SABME **and** SNRME **are the same as** SABM **and SNRM**
- **FRMR (FRaMe Reject) - indicate that a frame with a correct checksum but impossible semantics arrived.**

DLL in the internet

- **Point-to-Point Communication**
  - **Router-Router leased line connection**
  - **Dial-up host-router connection**



Point-to-Point Protocol

- Defined in RFC 1661 and further elaborated on in several other RFCs (e.g., RFCs 1662 and 1663).
- PPP provides three features:
  - *A framing method* , The frame format also handles error detection.
  - *A link control protocol* for bringing lines up, testing them, negotiating options, and bringing them down.
    - This protocol is called LCP (Link Control Protocol).
  - *A way to negotiate network-layer options* in a way that is independent of the network layer protocol to be used. The method chosen is to have a different NCP (Network Control Protocol) for each network layer supported.

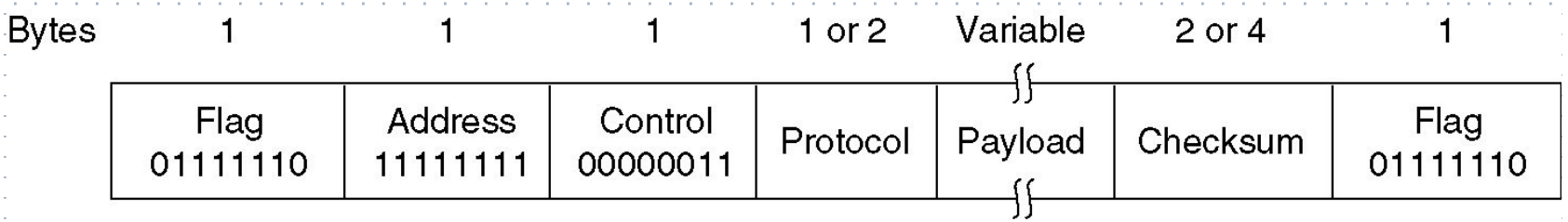
Typical Scenario: Connecting A Home PC To Internet Service Provider

1. **Physical connection setup phase:**
   - **The PC calls the provider's router via a modem.**
   - **The router's modem answers the phone and establishes a physical connection.**
2. **Data link layer options negotiation phase:**
   - **The PC sends the router a series of LCP packets in the payload field of one or more PPP frames. These packets and their responses select the PPP parameters to be used.**
3. **Network layer options negotiation phase:**
   - **A series of NCP packets are sent to configure the network layer and to assign an IP address for the PC (if the PC wants to run a TCP/IP protocol stack).**
4. **Data communication phase:**
   - **The PC sends and receives IP packets over the established connection.**
5. **Connection release phase:**
   - **When the PC is finished, NCP is used to tear down the network layer connection and free up the IP address.**
   - **The LCP is used to shut down the data line layer connection.**
   - **The computer tells the modem to hang up the phone, releasing the physical connection.**

PPP Frame Format

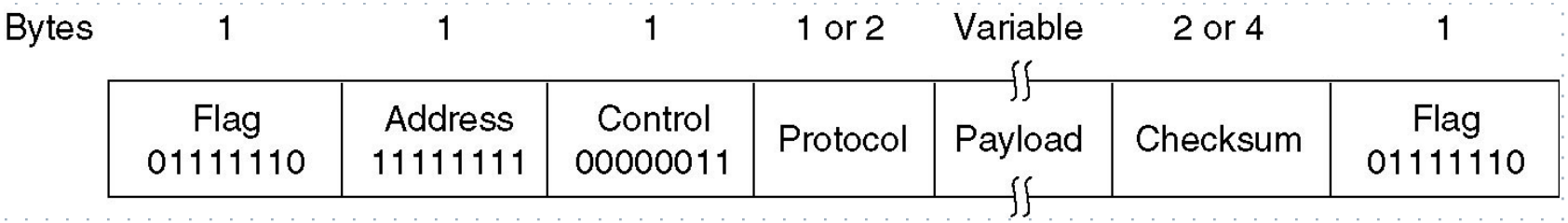
- **Main differences**
  - **PPP is character-oriented while HDLC is bit oriented.**
  - **PPP uses byte stuffing on dial-up modem lines, so all frames are an integral number of bytes.**

- **The PPP full frame format for unnumbered mode operation.**



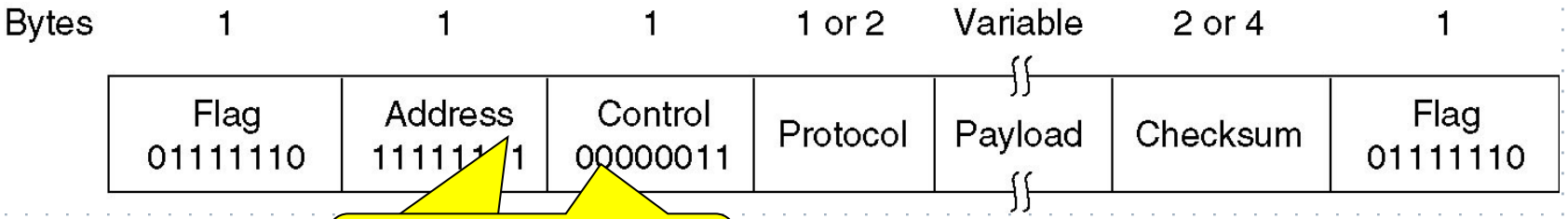
PPP Frame Format(cont'd)

- **Begin with a spcial byte-01111110 (same as HDLC)**
  - 若封装在PPP帧中的数据出现0x7E字节，则用2字节序列0x7D、0x5E取代；
  - 若出现0x7D字节，则用2字节序列0x7D、0x5D取代；



PPP Frame Format(cont'd)

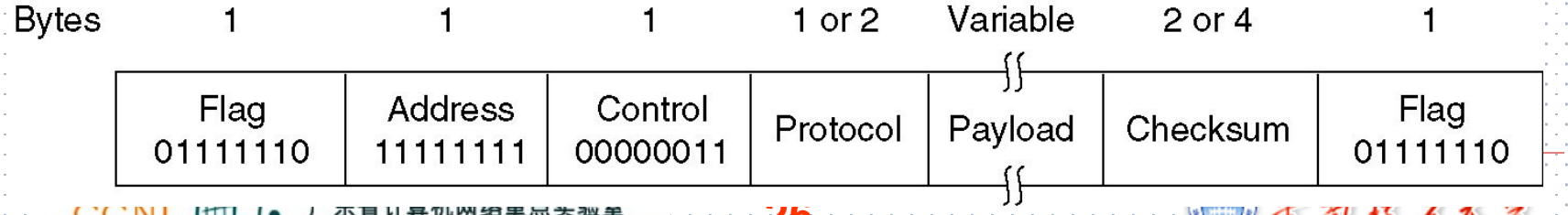
- **Address Fieled**
  - always set to the binary value 11111111
- **Control field**
  - default value is 00000011, which indicates an unnumbered frame.
- Considering the address and the control field are constant in default, so the LCP can negotiate to **leave those two fields out.**



双方协商认同后，可省略

PPP Frame Format(cont'd)

- **Protocol - tell what kind of packet is in the Payload field**
  - **The default size of the Protocol field is 2 bytes, can be negotiated down to 1 byte using LCP.**
    - When protocol=0x0021,the payload is IP packet.
    - When protocol= 0xc021,the payload is LCP packet.
    - When protocol= 0x8021, the payload is NCP packet.
- **Payload - variable length, up to some negotiated maximum, default is 1500 bytes.**
- **Checksum - normally 2 bytes (but can be 4 bytes)**
- **Closing flag – same as starting flag**



Summary of this lecture

- Learn protocol verification
  - Finite state machine models
  - Petri net models
- Learn Example DDL protocol
  - HDLC
    - Frame format
  - PPP

Thanks !