High-Level Data Link Control

- This class of data link layer protocols includes High-level Data Link Control (HDLC), Link Access Procedure Balanced (LAPB) for X.25, Link Access Procedure for D-channel (LAPD) for ISDN, and Logic Link Control (LLC) for FDDI
- The frame format is:

  Note that address and control can be extended to 16 bits, so that sequence number is 7-bit

	Flag	Address	Control	Data	FCS	Flag
bits	8	8	8	variable	16	8

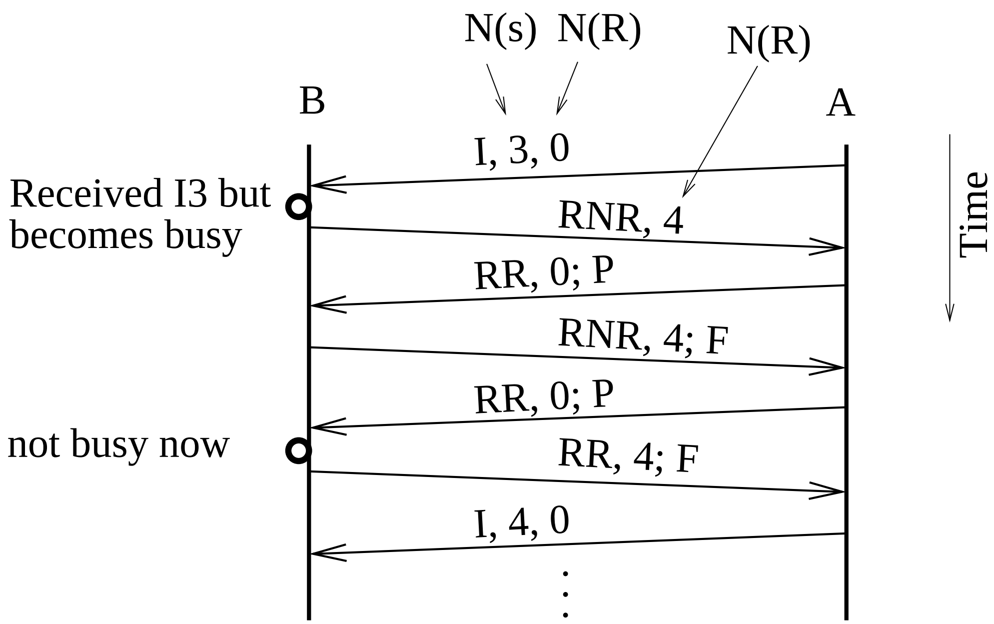
bit position	1	2	3	4	5	6	7	8
Information:	0	N(S)			P/F	N(R)		
Supervisory:	1	0	S		P/F	N(R)		
Unnumbered:	1	1	M		P/F	M		

N(S)=send sequence number
N(R)=receive sequence number
S=supervisory function bits
M=unumbered function bits
P/F=poll/final bit

- **Frame flag**: 01111110, so bit stuffing is used
- **Address**: For multipoint operation, it is used to identify the terminal that transmits or receives the frame and, in point-to-point link, it is used to distinguish Commands from Responses (2nd bit for C/R: 0/1). The address is now extended to 16 bits (1st bit indicates long/short 16/8 bits)
- **Checksum**: FCS contains the remainder of a 16-bit CRC calculation of the frame. It may be extended to 32 bits, using a 32-bit CRC
- **Control**: Three types of frames, I, S and U frames. The old protocol uses a sliding window with 3-bit sequence number and the maximum window size is $N = 7$. The control field is now extended to 16 bits with 7-bit sequence number

HDLC (continue)

- **I-frames**: carry user data. Additionally, flow and error control (ACKs) may be piggybacked inside the control field of an I-frame using N(R). P/F bit is P for command and F for response
- **S-frames**: provide flow/error control when piggybacking is not used. These include
  - Receive Ready (RR, S bits=00)
  - REJect (REJ, S bits=01)
  - Receive Not Ready (RNR, S bits=10)
  - Possibly, Selective REJect (SREJ, S bits=11)
- P/F bit: when set to P forces the other machine to response immediately rather than waiting for an I-frame to piggybacking
- Illustration: Busy condition

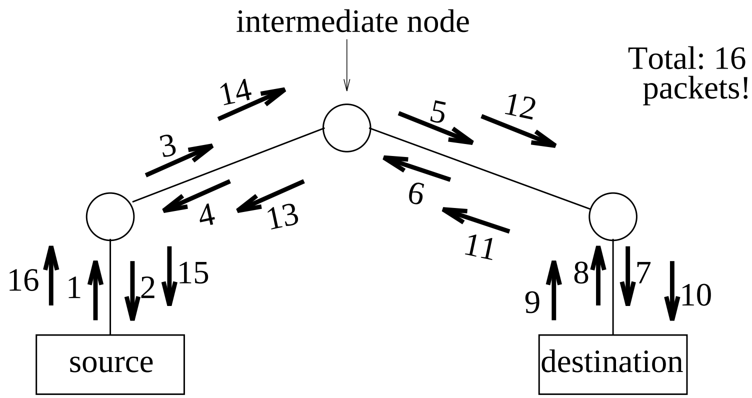


- **U-frames**: are mainly for control purposes, i.e. establishing and terminating connections, unnumbered ACK, etc., but can also be used to carry data for unacknowledged connectionless services, and type is defined by M bits

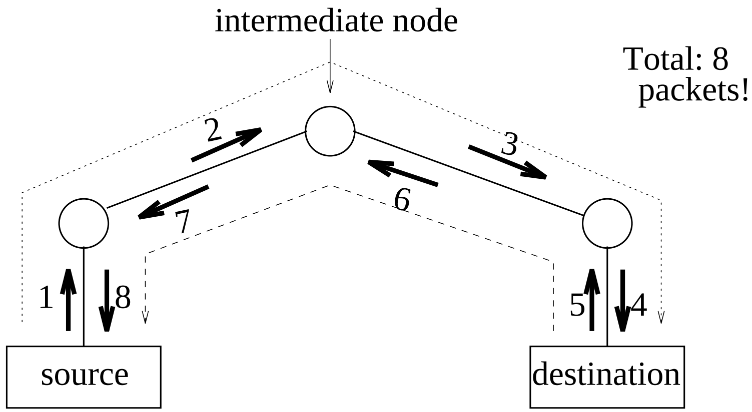
  Examples: Set Asynchronous Balanced Mode (SABM) for resetting the line, DISConnect (DISC) for terminating logical link connection, and Unnumbered Acknowledgement (UA)

Frame Relay

- Having examined a class of **full** data link layer protocols, let us see a **bare** DLL protocol
- Let us first look into traditional packet switching network, X.25, which employs full data link layer functions
  - Consider flow of data link frames required for transmission of a single data packet from source to destination and return of an ACK packet
  - At each hop through network, DLL protocol involves the exchange of a data frame and an ACK, and carries out full flow/error control for each virtual circuit



  - This is **very robust**, but **time consuming** and not feasible for high speed, and it may be unnecessary for high-quality, reliable links
- Now consider an alternative strategy, call frame relay:
  - At each hop, data link layer **does not** involved in flow/error control
  - With such a bare data link layer, lower delay and higher throughput can be achieved
  - This kind of strategy does relay on links being highly reliable, hardly anytime goes wrong



Frame Relay (continue)

- Frame relay frame format is very similar to HDLC except of **no control field** and, crucially, **no sequence number**

	Flag	Address	Data	FCS	Flag
bits	8	16	variable	16	8

  - There is only one frame type for carrying user data
  - It is **not possible to perform flow/error control** since there are no sequence numbers
  - Data link layer only does: separating frames using flags; forcing data transparency by bit stuffing; checking frames for errors, and frames with errors are simply discarded
  - Error and flow control, if implemented, is left for higher layer between end users to do (data link layer does not get involved)

- Full data link layer protocols make sense for unreliable networks
  - With frequent errors, they can be dealt with more efficiently at each (local) link level where errors occur
  - Otherwise, error recovery will be required frequently at higher layer, which may cost more

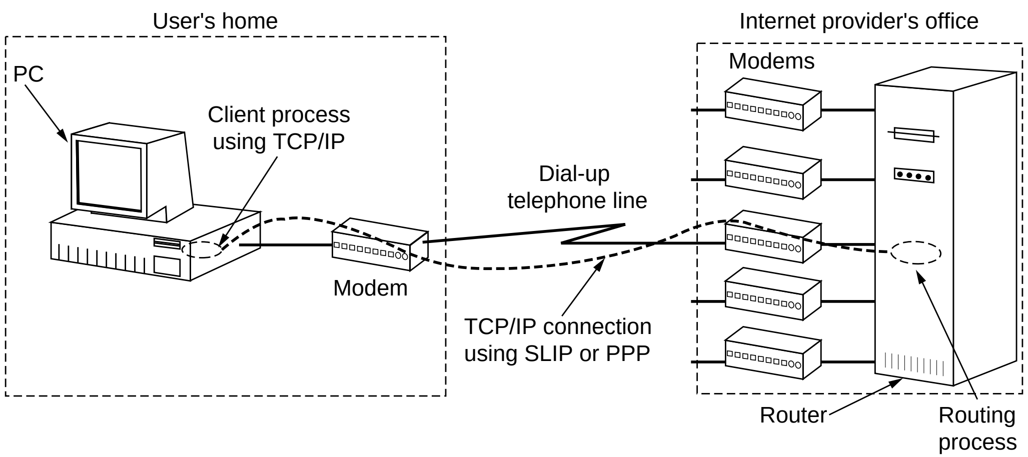
- Increasingly, as networks get more and more reliable and are designed for higher speed, bare data link layer protocols are adopted more and more
  - This achieves considerable overhead savings in software processing
  - Occasional errors may left to higher layer to deal with on an end-to-end base

Data Link Layer at ATM

- To achieve high speed, data link layer functions must be down to a **minimum** → Data link layer in ATM does very little
  - Part of transmission convergence sublayer in ATM perform DLL functions
- **Sending cells**: ATM layer takes 48-byte cell payload, generates the cell header except HEC, passes them to TC
  - TC then generates HEC (layer 2 function), and cells go to physical layer for transmission
  - TC also does some physical layer functions, matching bit stream to the speed of physical medium by inserting idle cells
- **Receiving cells**: from incoming bit stream, TC locates cell boundary, then
  - Checks header, discards cells with invalid header, passes cells to ATM layer
  - Note: The 48-byte payload in cell is not checked for error (no checksum for payload), and there is no sequence number
  - There is no flow/error control at data link layer level

Data Link Layer in Internet

- Note that for HDLC class, frame relay and ATM, we are dealing with "uniform" network, i.e. machines at the two ends of a link support same protocols
- Internet consists of various "networks", i.e. they may support different network layer protocols, and even links within may be very different
- A data link layer protocol for Internet needs to be able to deal with these issues
- Internet connection, in practice, is built up on point-to-point links
  - Organisations' routers are connected to outside world's routers via point-to-point leased lines (router-router)
  - Home users connect to outside Internet routers via cable modems and dial-up telephone lines (host-router)
  - For either cases of connections, some data link layer protocol is required



- Internet uses the point-to-point protocol (PPP): Its handles error detection, supports multiple protocols, allows IP addresses to be negotiated at connection time, permit authentication

Point-to-Point Protocol

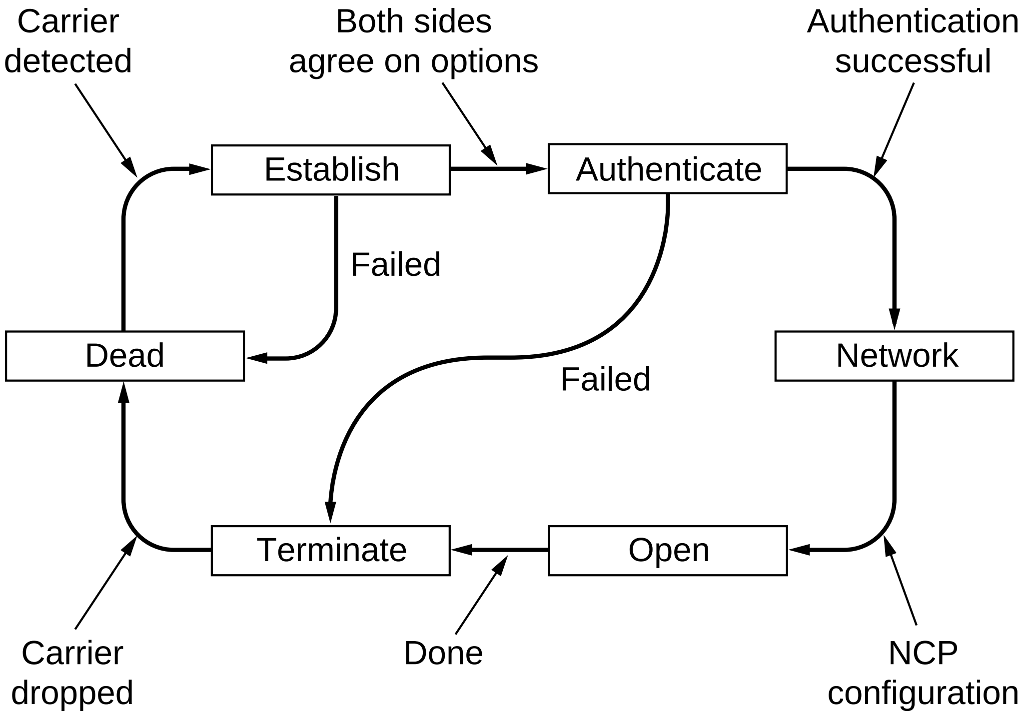
- PPP frame format is:
  - Frame flag: 01111110, and byte stuffing is used, i.e.

Bytes	1	1	1	1 or 2	Variable	2 or 4	1
	Flag 01111110	Address 11111111	Control 00000011	Protocol	Payload	Checksum	Flag 01111110

  - If flag pattern appears in data stream, a special escape byte (ESC) is added before it, and if ESC appears inside data, it also needs byte stuffing
  - Address: 11111111, which means all stations can accept the frame → This avoids issue of assigning data link addresses
  - Control: 00000011 is a default value, indicating unnumbered frame
  - Note there is no sequence number, so data link layer does not do flow/error control, but PPP has FSC and is able to detect errors
  - Protocol field: tells what kind of packet is in payload → Protocols starting with a 0 bit are network protocols such as IP, IPX, OSI, and others; starting with a 1 bit are used to negotiate other protocols, and these include LCP and different NCPs
- PPP provides link control protocol (LCP) for controlling line (setup, testing, negotiating options, shut down)
- PPP can negotiate network-layer options → having a different network control protocol (NCP) for each different network layer supported
- Network address is not fixed but assigned dynamically at connection time

PPP Operation

- PPP is multiprotocol framing mechanism suitable for use over modems, HDLC lines, SONET, and other physical layers
- Operation of PPP:
  - Start with physical line state *Dead*, after physical connection is established, the line moves to *Establish*
  - LCP option negotiation then begins, which lead to *Authenticate* if successful, and two parties can now check each other's identities if so desired
  - *Network* phase is next entered, and appropriate NCP protocol is invoked to configure network layer



  - If configuration successful, *Open* state is reached and data transport can take place
  - After data transport is completed, the line moves to *Termination* phase, and from there back to *Dead* when carrier is dropped

PPP Operation (continue)

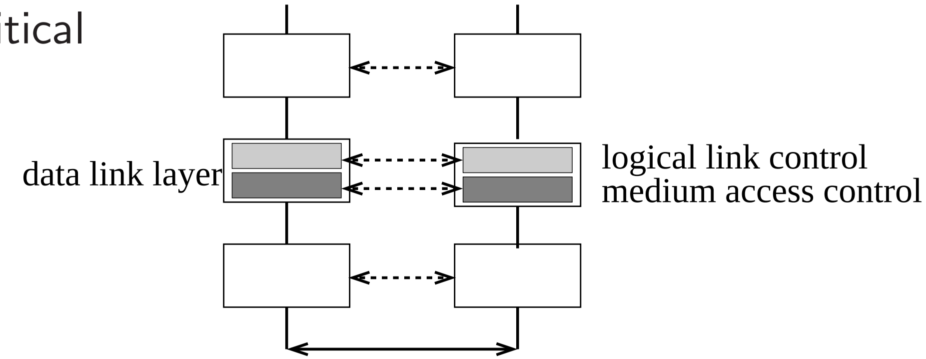
As a specific example, consider how a home user sets up an Internet connection to an Internet provider

1. *Dead* To *Establish* – Call the provider's router via a modem to set up a physical connection
2. To *Authenticate* – Send a number of LCP packets, embedded inside PPP frames, to negotiate PPP connection:
   - The maximum payload size in data frames
   - Do authentication (e.g. ask for password)
   - Monitor the link quality (e.g. how many frames did not get through)
   - Compress headers (which is useful for slow links between fast computers)
3. To *Network* – Send a number of NCP packets, embedded inside PPP frames, to negotiate the network layer configuration

   For example your PC wants to run TCP/IP, the NCP for IP is used to do IP address assignment
4. To *Open* – Your PC is now an Internet host, able to send and receive IP packets
5. To *Termination* – When you have finished,
   - The NCP shuts down the network layer connection, free IP address
   - The LCP shuts down the data link layer connection
   - Finally, your PC tells the modem to hang up the phone, releasing physical layer connection

Medium Access Control Overview

- We have so far discussed data link layer at **point-to-point** connection
- For broadcast networks, such as LANs, many users **share same medium** and collision avoidance is critical
- Typically, data link layer is divided into: Logical Link Control (LLC) sublayer, which does usual point-to-point connection issues, and **Medium Access Control** (MAC) sublayer: which deals with how to access medium (channel)



- There are three strategies for medium access or channel allocation
  - **Contention**: a user simply tries to use channel and does something when collision happens
  - **Round-robin**: each user is in turn given opportunity to use channel
  - **Reservation**: a user places a reservation for a channel
- In practice, a MAC may adopt one or more these strategies. For example, in GSM system, a user gains access to system by making a reservation on a channel but the reservation is made via a signalling channel and access to the signalling channel is through contention
- Multiple LANs can be connected via bridges (switches), which operate in data link layer, i.e. they do not examine the network layer header and simply pass network packets (unlike a router)

Summary

- Key design consideration is to have or not to have full data link layer functions
- Full data link layer functions (flow/error control etc.) depend on having sequence number
- Full data link layer class discussed include HDLC, LAPB, LAPD, LLC
- Bare data link layer class discussed include frame relay, ATM, Internet PPP
- Special issues for Internet: ability to support multiple protocols (not an issue in "single" network), and this is achieved via a protocol type field in frame header