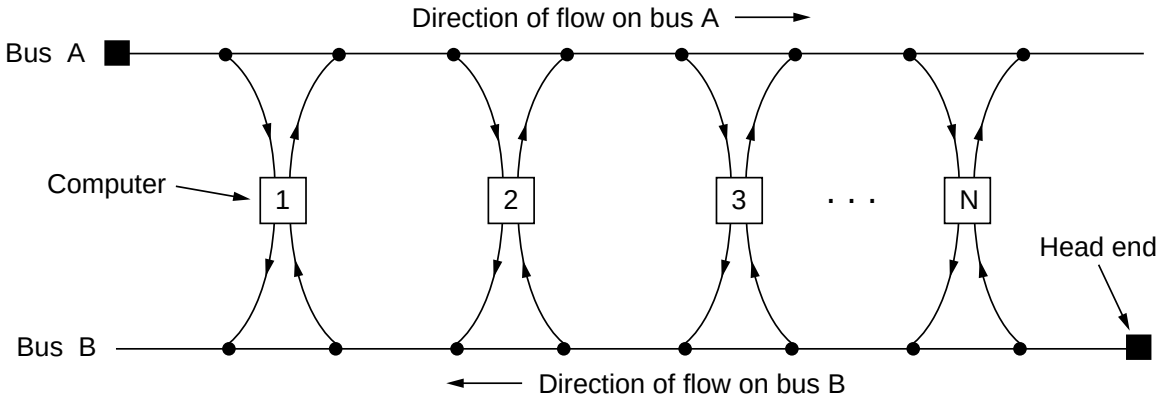
Distributed Queue Dual Bus

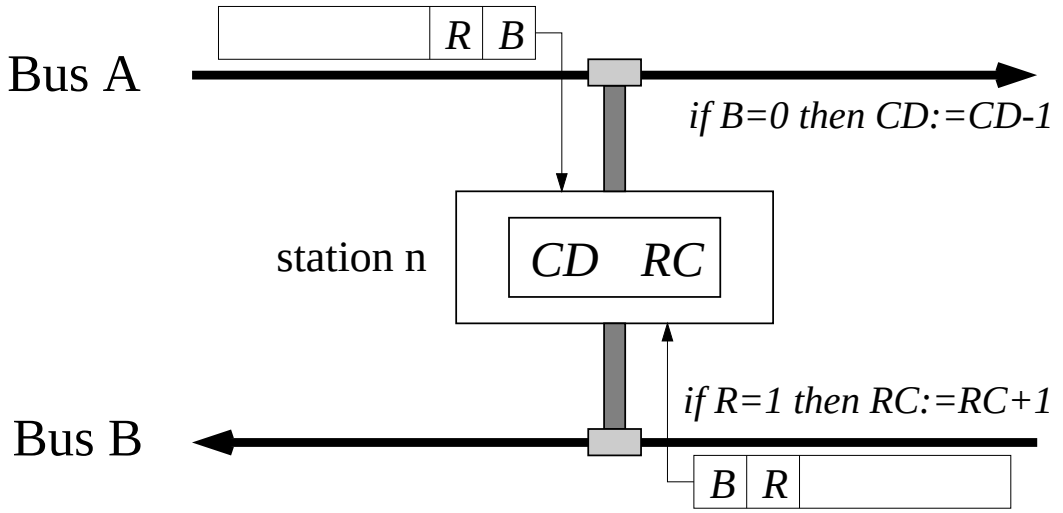
- IEEE 802.3 to 802.5 protocols are only suited for "small" LANs. They cannot be used for very large but non-wide area networks. IEEE 802.6 DQDB is designed for MANs
- It can cover an entire city, up to 160 km at a rate of 44.736 Mbps
  - Basic rule: if you want to send something to one of your right-hand neighbours, use upper bus A; otherwise, lower bus B
  - Direction of flow on a bus points to downstream. Fixed-size 53-byte cells with 44-byte payload are used, similar to ATM
  - Stream of cells flows down on a bus. Each cell has a busy (B) bit and request (R) bit. If a cell is occupied, its B bit is 1. You make a request by setting a cell's R bit (if it is zero) to 1



- Unlike 802.3 to 802.5 where a user transmits at first chance, DQDB MAC is non greedy
- Users queue up in the order they became ready to send and transmit in FIFO order
- This is achieved without a central queue control, hence called distributed queue
- The key of this MAC protocol is: **be polite to your downstream stations, and let them have go first if they requested before yours**

Distributed Queue Dual Bus (continue)

- How the MAC works: consider transmission on bus A (on bus B is similar), each user has
  - A RC that counts number of requests from its downstream stations
  - A CD that counts number of outstanding requests issued before its own request
  - Note that the downstream requests come from bus B, as you becomes "downstream" on bus B

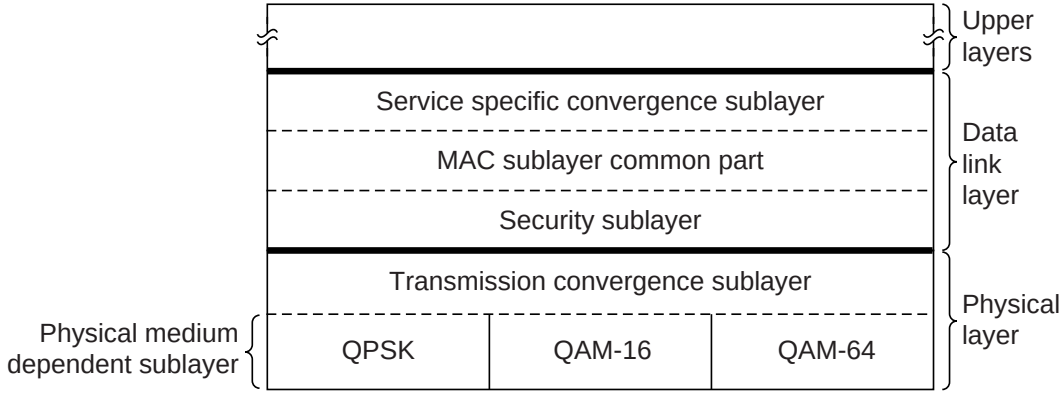


- If a user wants to send something to its downstream (on bus A), it sets first available request bit in a cell on bus B to 1, and copies RC to CD
- Assume that the value of CD at this moment is x. Then there are x requests from the station's downstream, and it has to let x free cells pass to the downstream
- Also, the user must let its upstream stations on bus A know its request, that is why it uses bus B to make a reservation
- Each time a non-busy cell passes by, CD is reduced by 1. When CD drops to zero, the user can take the next non-busy cell on bus A to transmit

Broadband Wireless

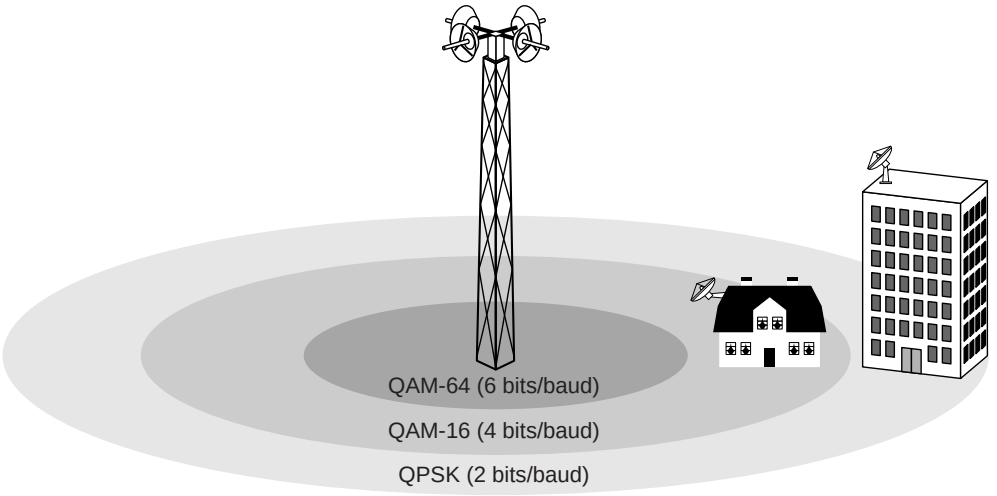
- IEEE 802.16 air interface for fixed broadband wireless access systems, also called wireless MAN or wireless local loop, has protocol stack:

  It provides multimegabits wireless services for voice, Internet, movies on demand, etc.



- **Physical layer** operates in 10 to 66 GHz range, and base has multiple antennas, each pointing at a separate sector

  For close-in subscribers, 64QAM is used, so typical 25 MHz spectrum offers 150 Mbps; for medium-distance subscribers, 16QAM is used; and for distant subscribers QPSK is used



- **Data link layer** consists of three sublayers
  - Security sublayer manages encryption, decryption, and key management, crucial for privacy and security
  - Service-specific convergence replaces logical link control, providing seamlessly interface for network layer that may have both datagram protocols and ATM

802.16 MAC Sublayer Protocol

- 802.16 MAC sublayer is completely **connection oriented** to provide quality-of-service guarantees for telephony and multimedia, and MAC frames occupy integral number of physical layer time slots

  Each frame is composed of subframes, and the first two are downstream and upstream maps
  - These two maps tell what is in which time slot and which time slots are free
  - Downstream map also contains system parameters to inform new users as they come on-line

- **Downstream** channel: base simply decides what to put in which subframe

- **Upstream** channel: there are competing subscribers and its allocation is tied to class of service
  - Constant bit rate: dedicate certain time slots to each connection and bandwidth is fixed through the connection, providing typical telephone channel service
  - Real-time variable bit rate: for compressed multimedia and other soft real-time applications in which bandwidth needed each instant may vary

    Base polls subscriber at fixed interval to ask how much bandwidth is needed this time
  - Non-real-time variable bit rate: for non-real-time heavy transmissions such as large file transfers

    Base polls subscribers often at non rigidly defined intervals to see who needs this service
  - Best-efforts: no polling and subscriber contends for bandwidth with others

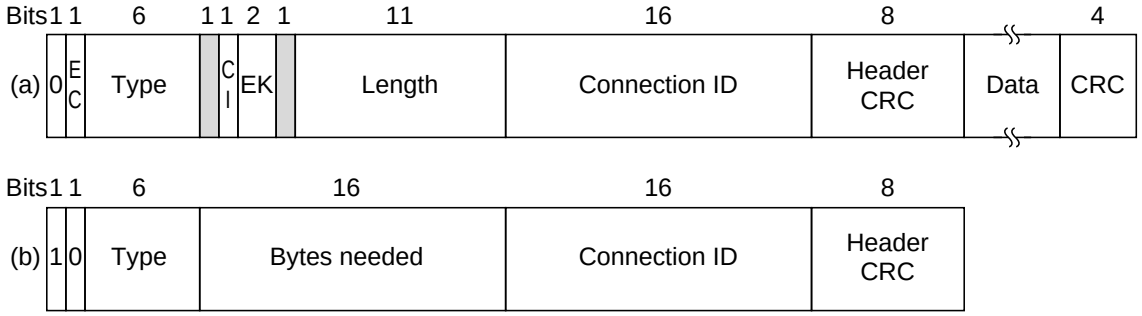
    Requests for bandwidth are done in time slots marked in upstream map as available for contention. Successful request will be noted in next downstream map, and unsuccessful subscriber has to wait a random period of time before try again

802.16 Frame Structure

- Generic frame (a) and bandwidth request (a control) frame (b):

  Header is followed by an optional payload and an optional frame CRC

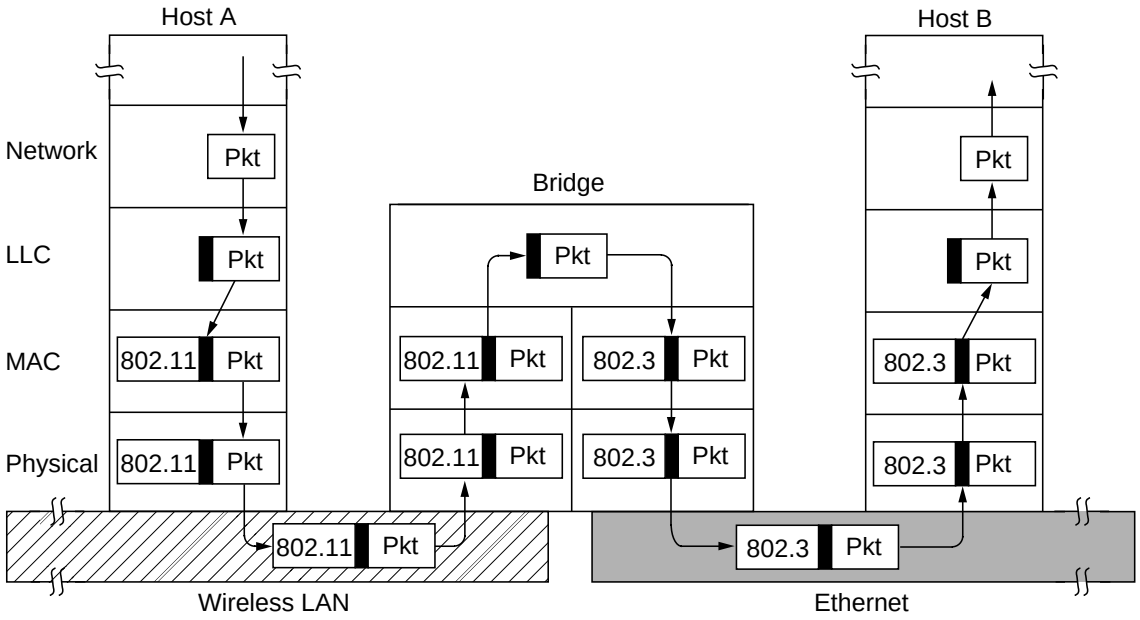
  Control frames have no payload



- Generic frame starts with a bit 0
  - EC bit: tells whether payload is encrypted
  - Type: identifies frame type, telling whether packing and fragmentation are presented
  - CI: indicates presence or absence of final checksum
  - EK: tells which encryption keys is used (if any)
  - Length: gives complete length of frame, including header
  - Connection identifier: tells which connection this frame belongs to
  - Header CRC: checksum over header only, using $x^8 + x^2 + x + 1$
- Control frame starts with a bit 1: 2nd and 3rd bytes form a 16-bit number telling how much bandwidth is needed to carry specific number of bytes
- Why optional frame CRC: no attempt is made to retransmit real-time frames $\longrightarrow$ Why bother with a CRC if no retransmission? Also error correction is present in physical layer (channel coding)

Bridging LANs

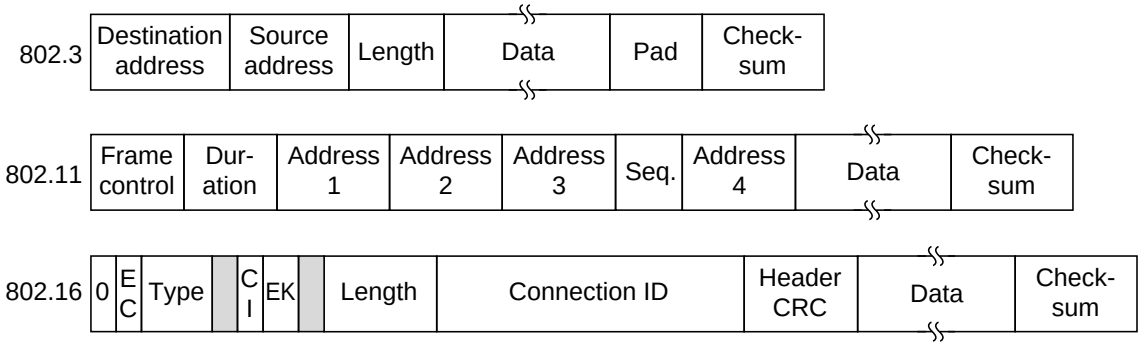
- Many LANs can be connected by **bridges**, which operate at data link layer level, i.e. they do not examine the network layer header and simply pass network packets, unlike a router
- There are many situations in which bridges are used
- A bridge must know MAC protocols used by LANs connected to it, so it can "talk" to each of them
- Consider two LANs, L_A and L_B, connected by a bridge, who must:
  - Know the MAC protocol for L_A, read all frames transmitted on L_A, and accept those addressed to hosts on L_B
  - Use the MAC protocol for L_B to retransmit those frames onto L_B
  - Do the same for L_B-to-L_A traffic



- Simple or is it? $\longrightarrow$ If source LAN and destination LAN are the same, bridging them will in deed be a simple task. If, however, two LANs are of different standards, there are some serious difficulties

Bridging Problems

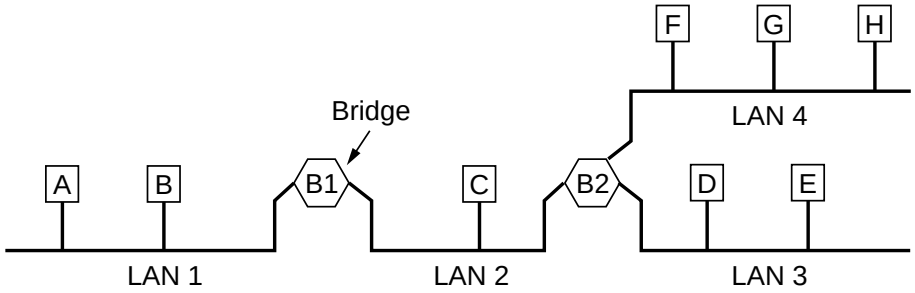
- Building a bridge between various 802 LANs (and MANs) can encounter some serious difficulties. Here we focus on 802.3, 802.11 and 802.16, as they are most widely used
- Different frame formats: copying between different LANs requires reformatting, which takes CPU times, requires a new checksum calculation, and possibly introduces undetected errors



- Different rates: when a high-speed LAN pumps frames onto a low-speed one, the bridge can simply hope that it will not run out of buffer memory. Timeout at higher layer of a fast source LAN may also cause serious bridge-as-bottleneck problem
- Frame length too long for target LAN: frame cannot be split into small pieces, as all MAC protocols do not have provision for reassembling frames out of smaller units. Basically, frames that are too long to be forwarded are simply discarded
- 802.11 and 802.16 support encryption, but 802.3 does not: if a wireless station uses data link layer encryption, there is no way to decrypt it when it arrives over Ethernet
- 802.11/802.16 provide quality of service (e.g. constant bit rate), but 802.3 does not have this concept: traffic from 802.11 or 802.16 loses its quality of service when passing over to Ethernet

Transparent Bridges

- Bridges should be completely transparent: simply plug in and walk away, no hardware and software changes are required
- Bridge must decide what to do with an incoming frame: discards it if destination and source LAN addresses in the frame are the same, or forwards it to destination LAN

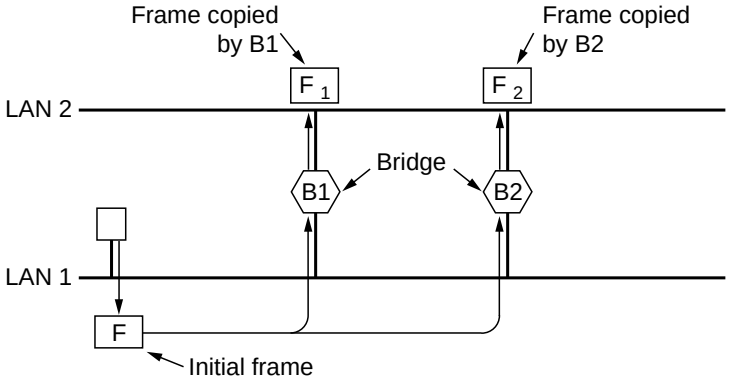


- Each bridge maintains a hash table, listing all the possible destination address/output line (LAN) pairs. When a bridge is first plug in, its hash table is empty, and this is how it figures out the configuration and keeps updating its hash table:
  - **Flooding**: incoming frames with unknown destinations are simply forwarded to all other LANs connected to the bridge, e.g. a frame coming from host C to bridge $B2$ with an unknown destination to $B2$ will be forwarded to LAN3 and LAN4
  - **Backward learning**: bridge gradually learns which interface it can reach a host and builds up its hash table from incoming frames' source addresses, e.g. a frame from A reaches bridge $B2$. From source address of this frame, $B2$ knows A is reachable via LAN2
  - **Timeout**: whenever a hash table entry is made, arrival time of the frame is noted in the entry. Whenever a frame whose source address already in the table arrives, its entry is updated with this new time. Periodically, bridge gets ride of those entries more than a few minutes old

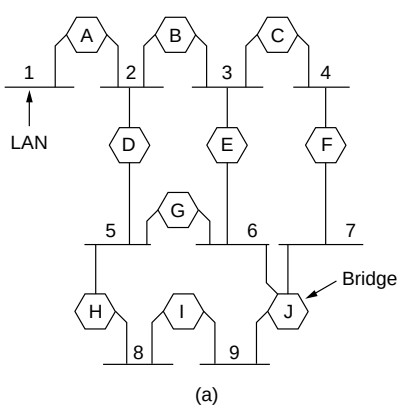
Spanning Tree

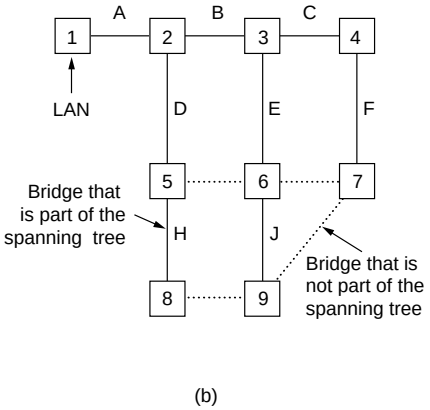
- To increase reliability, two or more bridges may be used in parallel between pairs of LANs, but this topology creates looping problem

  For example, frame F with an unknown destination will be copied by bridge $B1$ to LAN2 as $F1$ and by bridge $B2$ to LAN2 as $F2$; frame $F1$ will be copied back to LAN1 by $B2$, and $F2$ copied back to LAN1 by $B1$. This cycle goes on forever



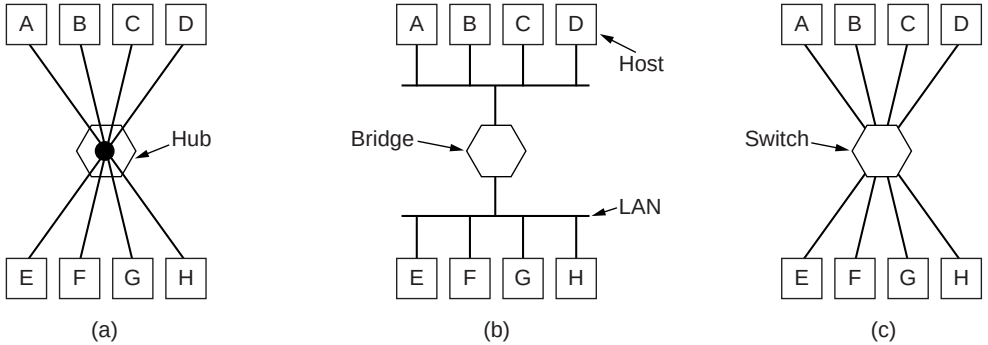
- Solution is to make it a **spanning tree**: some potential connection between LANs are ignored in the interest of constructing a logically loop-free topology, and the procedure is a graph optimisation algorithm
  - Choose a bridge as the root, usually, the bridge with the lowest ID number
  - Find a tree of shortest paths from the root to every bridge and LAN
  - If a bridge or LAN fails or there are some other changes in connection, a new spanning tree is computed





Switching Devices

- A classification of various switching devices to get frames from one cable segment to another
- **Repeater**: is a device connecting two cable segments. It is in physical layer and operates on bit level, i.e. detects bits from signal waveform, amplifies and re-transmits them



- **Hub**: joints number of input lines electrically. All lines coming into a hub operates at same speed, and frames arriving from any one line are sent out on all the others
  - If two frames arrive at same time, they collide, just as in Ethernet, so the entire hub forms a single collision domain
  - Hub does not examine or use 802 addresses in any way, so it is in physical layer
- **Bridge**: connects two or more LANs. It examines address in frame header to see where to send the frame. So Bridge is in data link layer, may have line cards for different network types and different speeds, and each line is its own collision domain
- **Switch**: is similar to bridge in that both route on frame addresses. In fact, two terms are often used interchangeable. Main difference is that switch is most used to connect individual hosts
- **Router**: is in network layer. Router does not care frame header/trailer, but examines network address in packet header to see where to send the packet

Network Layer Overview

- Network Layer is concerned with getting packets from source all the way to destination. This is where we start to deal with **end-to-end** users
- Getting to destination may require making many hops through many links: compare this with data link layer, which essentially deals with issues related to two ends of a link
- What network layer does is to provide service to transport layer $\longrightarrow$ facilities for getting data from source to destination, thus routing is a basic function of network layer. The design goals:
  - Services should be independent of subnets technology and implementation
  - Transport layer should be shielded from subnets technology and implementation
  - Network addresses should be unique, even across LANs and WANs
- Two kinds of services are offered: connection-oriented and connectionless, with two implementations respectively
  - **Virtual circuit**: a complete route is set up in advance, all packets go through the same route
  - **Datagram**: each packet is independently routed on the way to the destination
- Other important topics include: **congestion control**, **internetworking**, network layer in Internet, and network layer in ATM networks

Summary

- IEEE 802.6 distributed queue dual bus (a MAN standard) medium access control protocol
- IEEE 802.16 broadband wireless: medium access control sublayer protocol, service classes, and frame structure
- Bridge for connecting LANs: operates at data link layer level, and some serious difficulties encountered in bridging 802.x/802.y

  Transparent bridges: how it self learns (backward learning)

  Spanning tree bridges
- Definitions and differences in repeater, hub, bridge, switch, and router
- Network layer overview