The Network Layer

• Move packets from source all the way to destination.
  - must know network topology
  - need to choose appropriate paths

• Provides services to the transport layer.

• Two kinds of services: connectionless or connection-oriented

• Two camps (where to put complexity):
  - Internet: subnet is unreliable; hosts should do error control, flow control; network service should be connectionless; complexity at the transport layer.

  - Telecom: subnet should provide a reliable, connection-oriented service; complexity at the network layer.
Virtual Circuit and Datagram

- **Connection - virtual circuit**
  - need connection setup.
  - a route is chosen for all packets.
  - need virtual circuit number field in the header.
  - router builds a routing table; one entry for each VC.
  - virtual circuit has local significance only.

- **Independent packets - datagram**
  - contain full destination address in the packet header.
  - route decision is done per packet.
  - router has a routing table, one entry for each destination router.
Routing algorithm

- Decides which output line an incoming packet should be transmitted on.
- Goals: minimize mean packet delay or maximize network throughput. However, these two are in conflict since operating near capacity implies a long queuing delay.
- Instead, minimize number of hops => improve delay and reduce bandwidth consumed => improve throughput.
- Two classes of routing algorithms:
  - Static algorithms: run in advance, off-line.
  - Adaptive (dynamic) algorithms:
Routing algorithm

- Optimality principle: if a router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route.

```
I   J   K
```

- Consequence of optimality principle => The set of optimal routes from all sources to a given destination form a tree rooted at the destination - sink tree.

- The goal of all routing algorithms is to find and use the sink tree for all routers.
Static Routing

- **Shortest Path Routing**
  - metrics: hops, distance, comm. Cost, mean queue length, delay, etc.
  - Dijkstra algorithm:

- **Flooding**
  - incoming packets is sent out on every outgoing line except the one it arrived on.
  - generate vast number of duplicate packets.
  - can use hop count to limit duplicate packets
  - or by keeping track of which packets have been flooded.
    each router needs a list per source router telling which sequence number originating at that source have already been seen.
Static Routing

• Flow-based routing

  - Consider both topology and link load.
  
  - Assume the link capacity and traffic flow from i to j are known.
  
  - Once a routing algorithm is chosen, can compute the mean packet delay on that line from queuing theory.
Adaptive routing

- **Distance vector routing**
  - each router maintains a table giving the best known distance to each destination and which line to use to get there.
  - tables are updated by exchanging info. with neighbors.
  - distributed Bellman-Ford routing algorithm
  - distributed Ford-Fulkerson algorithm
  - original ARPANET routing algorithm (RIP)
  - old routing table is not used in updating.

- **Count-to-infinity problem**
  - converge slowly, especially with bad news.

- **Split Horizon Hack**
  - distance to X is not reported on the line that packets to X are sent on.
Adaptive routing

- **Link State Routing**
  - Discover its neighbors and learn their network addresses:
    - send Hello packet on each point-to-point line; neighbors reply.
  - Measure the delay or cost to each of its neighbors:
    - send echo packet over the line that the other side is required to send back immediately.
  - Construct a packet telling all it has just learned.
    - The hard part is when to build the packet.
  - Send the packet to ALL other routers.
    - use flooding.
  - Compute the shortest path to every other router.
    - E.g., use dijkstra algorithm.
Hierarchical routing

- As network size grows, routing tables grows.
- Divide routers into regions.
- Each router knows all the details about routing within its region and knows nothing about the internal structure of other region.
- Advantage: less storage requirement.
- Penalty: longer path.
Routing for Mobile Hosts

• How to locate a mobile user?
• All users have a home location.
• Home agent: keep track of roaming away “local” users.
• Foreign agent: keep track of visiting users.
• When a mobile user enters a new area,
  - it registers with the foreign agent.
  - foreign agent contacts the mobile’s home agent.
  - home agent checks the security information, then Ack.
  - foreign agent registers the mobile user after receiving the Ack.

• When a packet is to send to a mobile user:
  - intercepted by home agent; home agent tunnels packet to foreign agent; home agent informs sender to send to foreign agent; sender sends packet via foreign agent to mobile user.
Broadcast Routing

- Hosts need to send one packet to each destination.
- Can send each one separately.
- Can use flooding.
- Use multi-destination routing
- Use spanning tree: router needs to know some spanning trees.

- **Reverse Path Forwarding**: When a broadcast packet arrives at a router, the router checks to see if the packet arrived on the line that is normally used for sending the packets to the source of the broadcast.
  - If so, forward the packet.
  - If not, discard the packet.