

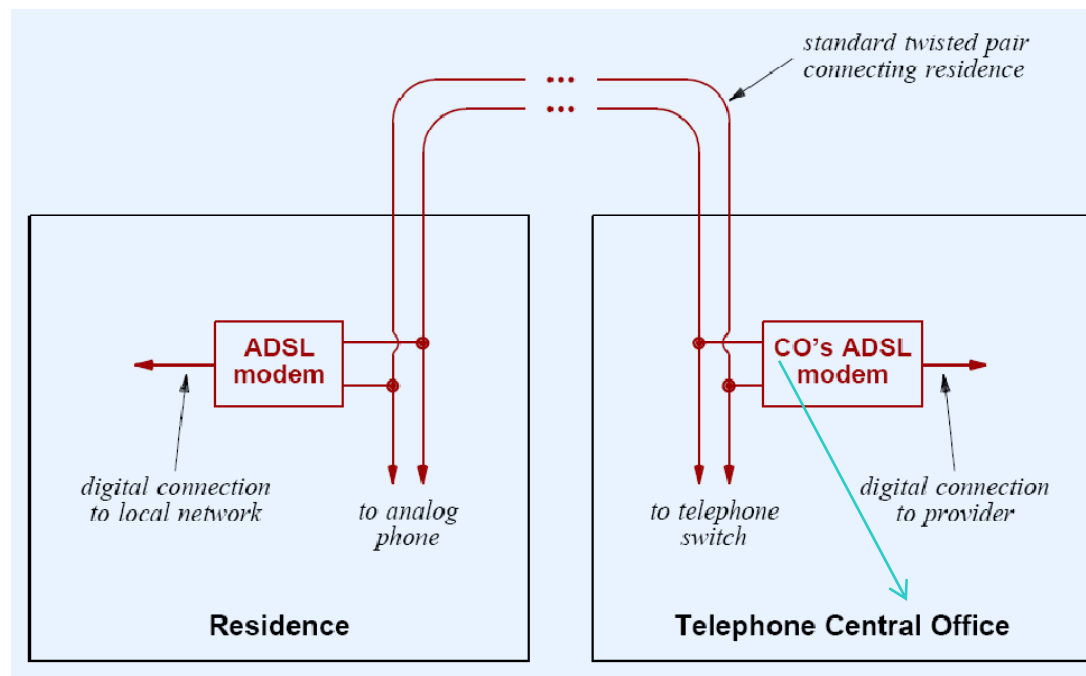
Digital Local Loop Technologies

- *Integrated Services Digital Network (ISDN)*
 - Handles voice and data
 - Relatively high cost for low bandwidth → (Skip)
- *Digital Subscriber Line (DSL)*
- *Cable modems*
- *Hybrid Fiber Coax (HFC)*

Asymmetric Digital Subscriber Line (ADSL)

- Popular DSL variant (xDSL)
- Runs over conventional POTS wiring
- Higher (lower) capacity down (up) stream
- Uses frequencies above POTS (Plain Old Telephone Service)

Illustration Of ADSL Wiring



No wiring change
No preemption due to
Separate freq. operating

DMT in the TEXT
(pp.190)
SDSL,HDSL,VSDL
In the TEXT

- Downstream can reach 6.4 Mbps
- Upstream can reach 640 Kbps

Cable Modems

- Superior shielding over twisted pair
- Send/receive over CATV wiring
- Use FDM
- Group of subscribers in neighborhood share bandwidth (by Time Division Multiplexing)
- No way for upstream → Dial Up modem for upstream (dual path)
- Two way cable modem is under way → VOD, Interactive TV

Hybrid Fiber Coax (HFC)

(optical fiber + coax) – 단계적

- Wiring scheme for cable to allow digital access optical fiber
 - Highest bandwidth
 - Extends from central office to neighborhood concentration points → Trunk with Optical Fiber
- Coaxial cable
 - Less bandwidth
 - Extends from neighborhood concentration point to individual subscribers (e.g., residence) → Feeder Circuit to individuals
- Spectrum (50–450MHz: TV (6MHz for each channel), 450–750MHz: Upstream, 5–50MHz: Downstream)
- Run Optical to the residence → FTTC (Fiber to the Curb)
(needs twisted pair wiring for voice installation in addition)

Head, Tail End Modem & Wireless Alternatives & Satellite systems



TEXT

Summary (SKIP)

- Technologies exist that span long distances
 - Leased analog lines (require modems)
 - Leased digital circuits (require DSU/CSUs)
- Digital circuits
 - Available from phone company
 - Cost depends on distance and capacity
 - Popular capacities called T1 and T3
 - Fractional T1 also available

Summary (continued)

- High capacity circuits available
 - Popular capacities known as OC-3, OC-12
- Local loop refers to connection between central office and subscriber
- Local loop technologies include
 - DSL (especially ADSL)
 - Cable modems



PART VII



Wide Area Networks (WANs), Routing, and Shortest Paths

Motivations

- Connect multiple computers
(LAN → MAN → WAN)
- WAN:LAN → Scalability and Capacity
- Span large geographic distance
- Cross public right-of-way
 - Streets
 - Buildings
 - Railroads

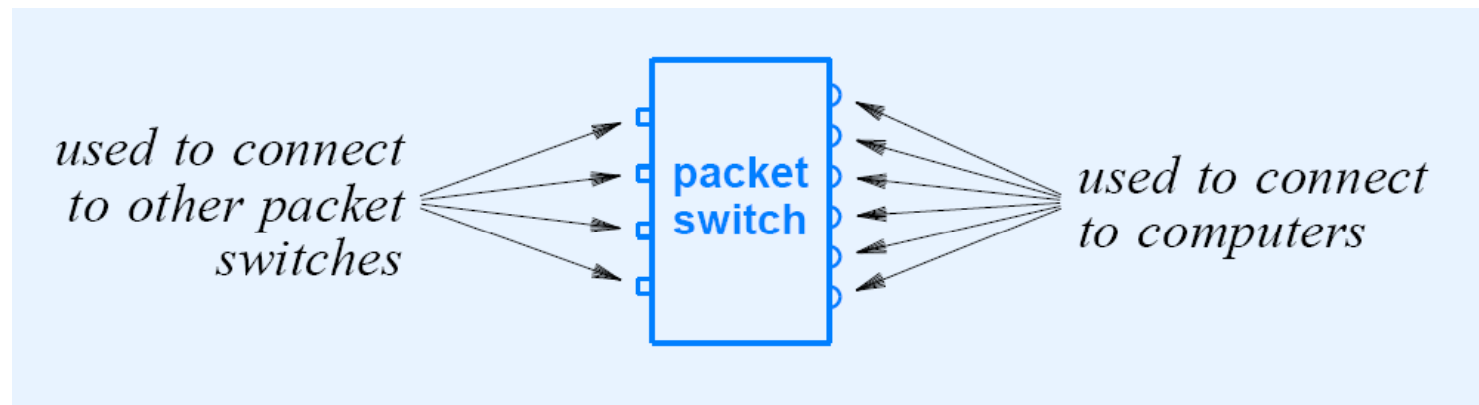
Building Blocks (WAN)

- Packet switches instead of Point-to-point long-distance connections

Packet Switch

- Hardware device
(Processor + I/O + Memory) → ASIC
- Two types of I/O Connect to
 - Other packet switches (Fast I/O)
 - Computers (Slow I/O)
- Forwards packets
- Uses addresses

Illustration Of A Packet Switch

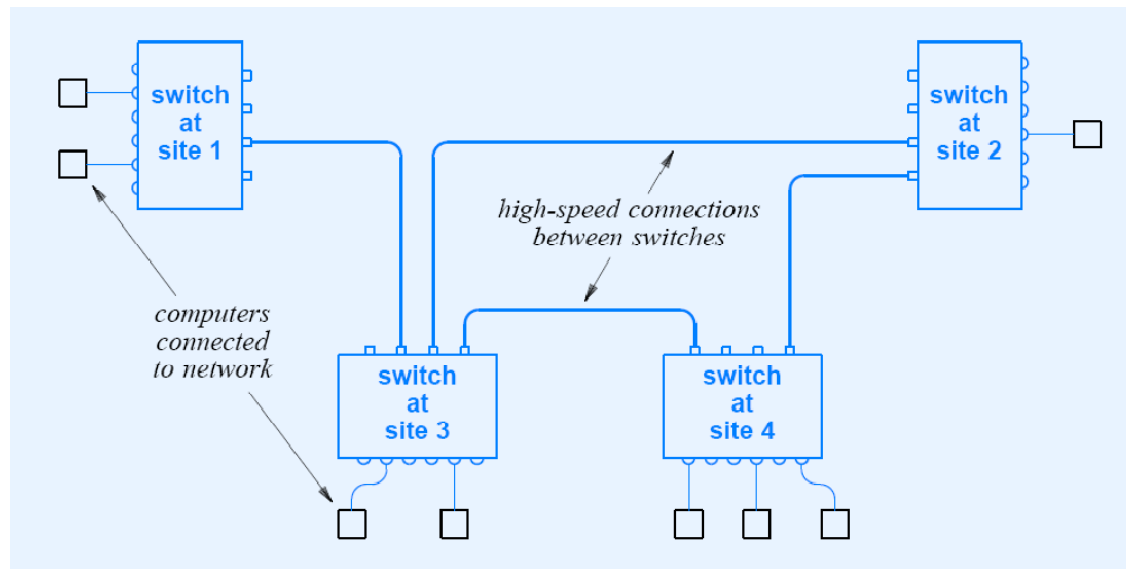


- Special-purpose computer system
 - CPU
 - Memory
 - I/O interfaces
 - Firmware

Building A WAN

- Place one or more packet switches at each site
- Interconnect switches
 - LAN technology for local connections
 - Leased digital circuits for long-distance connections

Illustration of a WAN



- Interconnections depend on
 - Estimated traffic
 - Reliability needed (Redundancy)

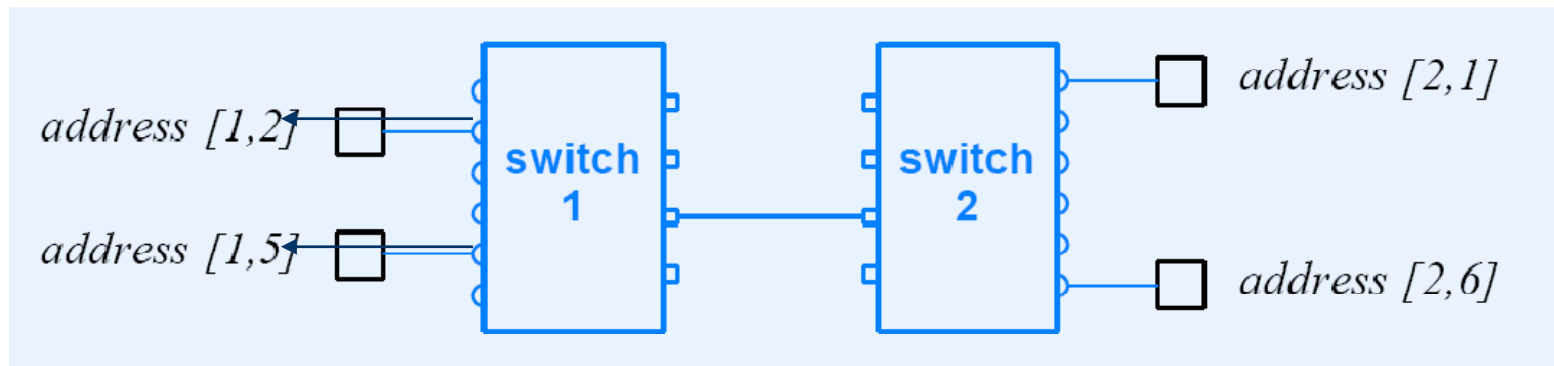
Store and Forward

- Basic paradigm used in packet switched network
- Packet
 - Sent from source computer
 - Travels switch-to-switch
 - Delivered to destination
- Switch
 - *Stores* packet in memory (→ then interrupt processor)
 - Examines packet's destination address
 - *Forwards* packet toward destination (→ by selecting a proper interface)

Addressing in a WAN

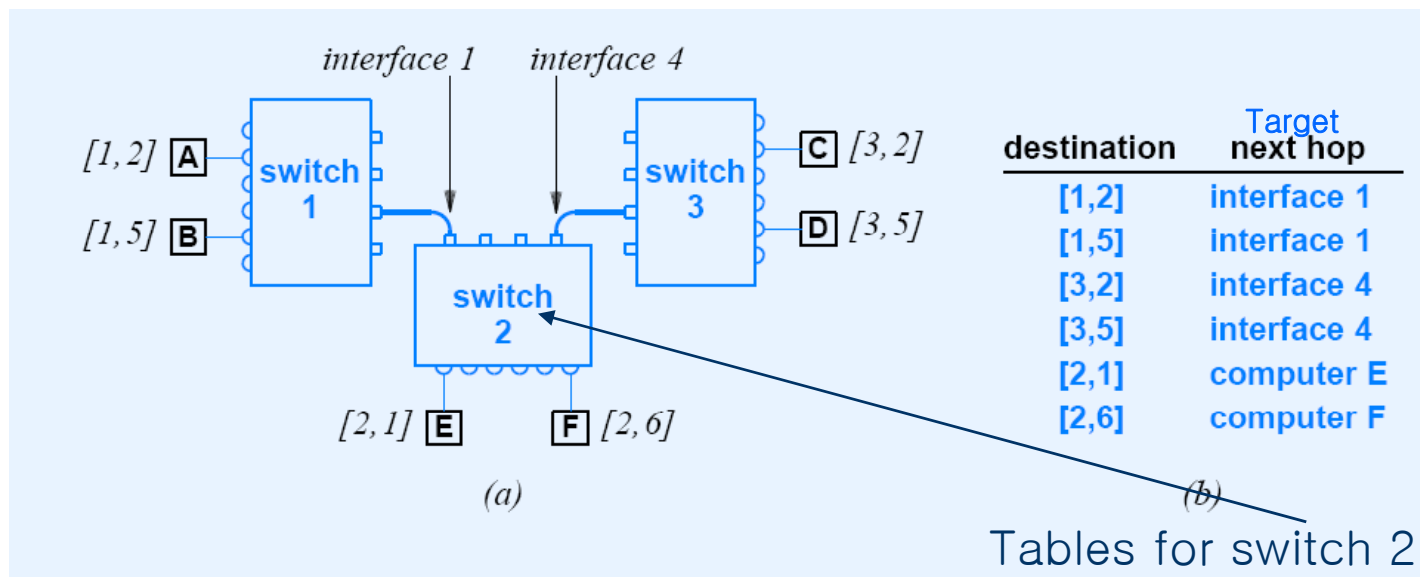
- Need
 - Unique address for each computer
 - Efficient forwarding
- Two-part address (Hierarchical Addressing)
 - Packet switch number
 - Computer on that switch

Illustration Of WAN Addressing



- Two-part address encoded as integer ([single](#))
 - High-order bits for switch number
 - Low-order bits for computer number

Next-Hop Forwarding



- Performed by packet switch
- Uses table of routes
- Table gives *next hop*

- **Source independence: doesn't depend on the source or the previous path the "packet to go" has taken**
- **Only the destination in the packet is extracted for next hop forwarding**

Forwarding Table Abbreviations

Destination	Next Hop
(1, anything)	interface 1
(3, anything)	interface 4
(2, anything)	local computer

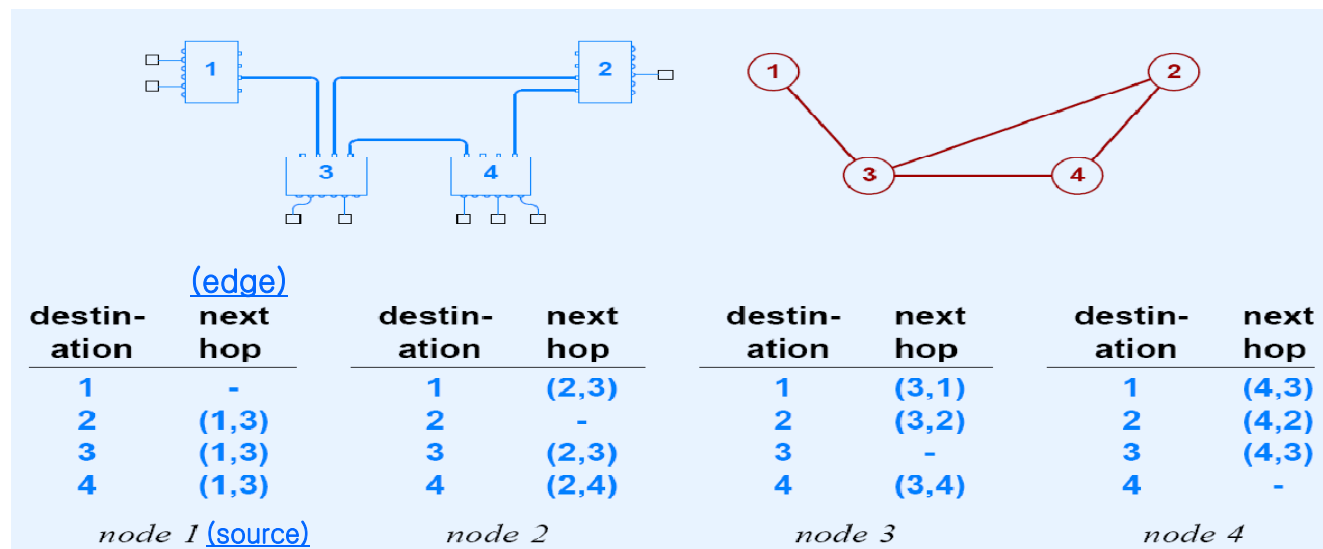
Packet
Switch
Number

- Many entries point to same next hop
(Only destination counts: First part → to arrive at packet switch, Second part → to arrive at the destination computer)
- Can be condensed (*default*)
- Improves lookup efficiency
(Computation time (indexing), storage space saving (one per packet switch))

Source Of Routing Table Information

- Manual
 - Table created by hand
 - Useful in small networks
 - Useful if routes never change
- Automatic routing
 - Software creates/updates table
 - Needed in large networks
 - Changes routes when failures occur

Relationship of Routing To Graph Theory



Many duplication in the table
 → Default routing w/o duplicated next hop entry
 (in the text Fig. 13.8)

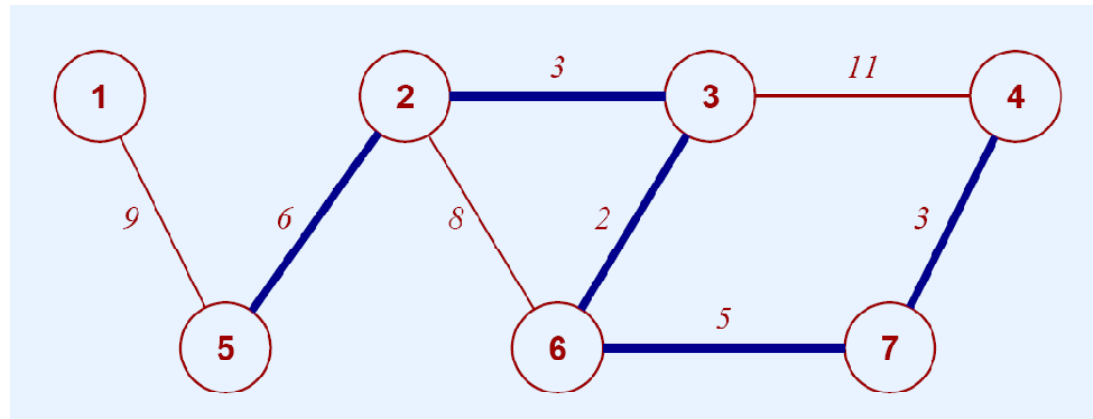
Most next hop for most destination as default route → *

- Graph
 - *Node* models switch (packet switch)
 - *Edge* models connection

Shortest Path Computation

- Algorithms from graph theory
(Dijkstra's Algorithms)
 - Compute the entries for a routing table
 - Find the distance along a shortest path from a node to each other) – will also find next hop routing table
- No central authority
(distributed computation) (@each router)
- A switch
 - Must learn route to each destination
 - Only communicates with directly attached neighbors

Illustration of Minimum Weight Path



- Label on edge represents “distance”
- Possible distance metric → to be weighted
 - Geographic distance
 - Economic cost
 - Inverse of capacity
- Darkened path is minimum 4 (*node*) to 5 (*node*)

Algorithms for Computing Shortest Paths

(Distributed Computing case)

- *Distance Vector (DV)*
 - Switches exchange information in their routing tables
- *Link-state*
 - Switches exchange link status information (if link is up or down)
- Both used in practice

Distance Vector

- Periodic, two-way exchange between neighbors
- During exchange, switch sends
 - List of pairs 자기를 통하여 가면 거리가 얼마이다 정보
 - Each pair gives (*destination, distance*)
- Receiver
 - Compares each item in list to local routes
 - Changes routes if better path exists

Distance Vector Algorithm

Sending switch

(Algorithm 13.2 in text) – see next slide for insight

(Each switch periodically sends the list of
(destination, distance) pairs from its routing table to all neighbors.)

Given: (destination node, next hop edge)

a local routing table, a weight for each link that connects to another switch, and an incoming routing message

Compute:

an updated routing table

Method:

Maintain a *distance* field in each routing table entry;

Initialize routing table with a single entry that has the *destination* equal to the local packet switch,

the *next-hop* unused, and the *distance* set to zero;

Repeat forever {

wait for the next routing message to arrive over the network from a neighbor; Let *N* be the sending switch; for each entry in the message {

Newly arrived
message

Let *V* be the destination in the entry and let *D* be the distance; [Next slide](#)

Compute *C* as *D* plus the weight assigned to the link over which the message arrived;

Examine and update the local routing table:

if (no route exists to *V*) {

add an entry to the local routing table for destination

V with next-hop *N* and distance *C*;

} else if (a route exists that has next-hop *N*) {

replace the distance in existing route with *C*; (가장 최근의 것으로 update)

} else if (a route exists with distance greater than *C*) {

change the next-hop to *N* and distance to *C*;

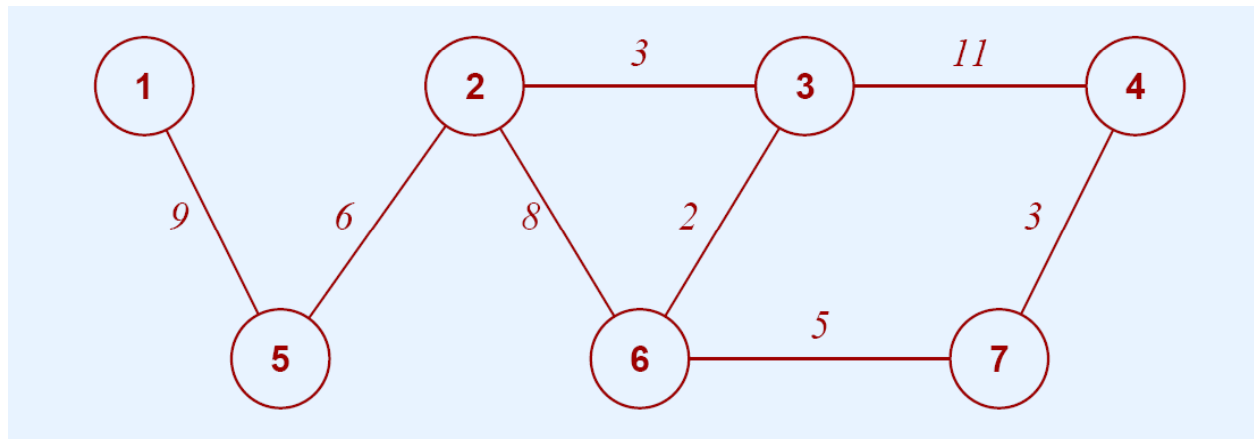
}

}

Distance Vector Intuition

- Let
 - N be neighbor that sent the routing message
 - V be destination in a pair
 - D be distance in a pair
 - C be D plus the cost to reach the sender
- If no local route to V or local route has cost greater than C , install a route with next hop N and cost C
- Else ignore pair

Example Of Distance Vector Routing

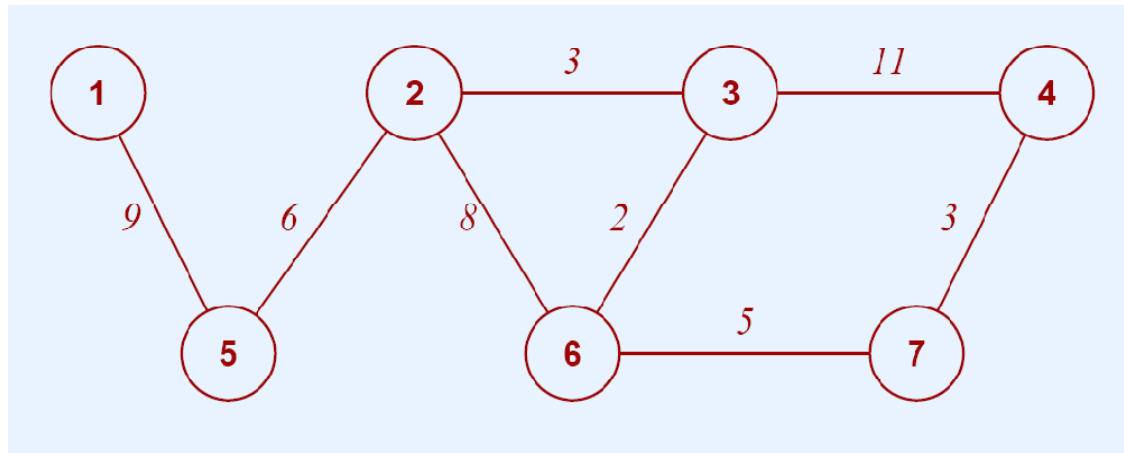


- Consider transmission of one DV message
- Node 2 sends to nodes 3, 5, and 6 *(to all neighbors)*
- Node 6 installs cost 8 route to node 2
- Later, node 3 sends update *(to node 6 after node 2 sent message)*
- Node 6 changes route to make node 3 the next hop for destination 2 *(since to go node 2 directly – cost 8, through node 3 – cost 2+3=5)*

Link-State Routing

- Overcomes instabilities in DV
- Pair of switches periodically
 - Test link between them
 - Broadcast link status message
- Switch
 - Receives status messages
 - Computes new routes
 - Uses Dijkstra's algorithm

Example Of Link-State Information



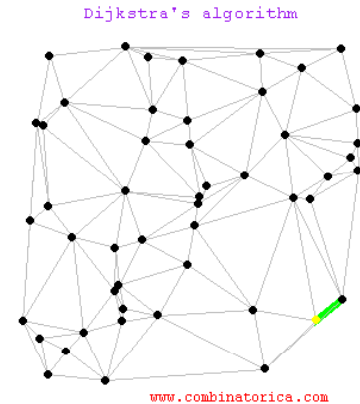
- Assume nodes 2 and 3
 - Test link between them
 - Broadcast information
- Each node
 - Receives information
 - Re-computes routes as needed

Dijkstra's Shortest Path Algorithm

- Input
 - Graph with weighted edges
 - Node n ($\rightarrow$ *source node*)
- Output
 - Set of shortest paths from n to each node
 - Cost of each path
- Called *Shortest Path First (SPF)* algorithm

Dijkstra's Algorithm

(Algorithm 13.1 in Text)



Given:

a graph with a nonnegative weight assigned to each edge and a designated source node

Compute:

the shortest distance from the source node to each other node and a next-hop routing table

Method:

Initialize set S to contain all nodes except the source node;

Initialize array D so that $D[v]$ is the weight of the edge from the source to v if such an edge exists, and *infinity* otherwise;

Initialize entries of R so that $R[v]$ is assigned v if an edge exists from the source to v, and zero otherwise;

while (set S is not empty) {

choose a node u from S such that $D[u]$ is minimum;

if ($D[u]$ is *infinity*) {

no path exists to nodes in S; quit;}

}

delete u from set S;

for each node v such that (u,v) is an edge {

if (v is still in S) {

$c = D[u] + \text{weight}(u,v)$;

if ($c < D[v]$) {

$R[v] = u$;

$D[v] = c$;

}

}

}

}

Find a nearest node
from source node

Next hop is u
i.e. go u to go to v

Distance to v
since c is shorter
compared to existing one

Distance
Vector

Next Hop

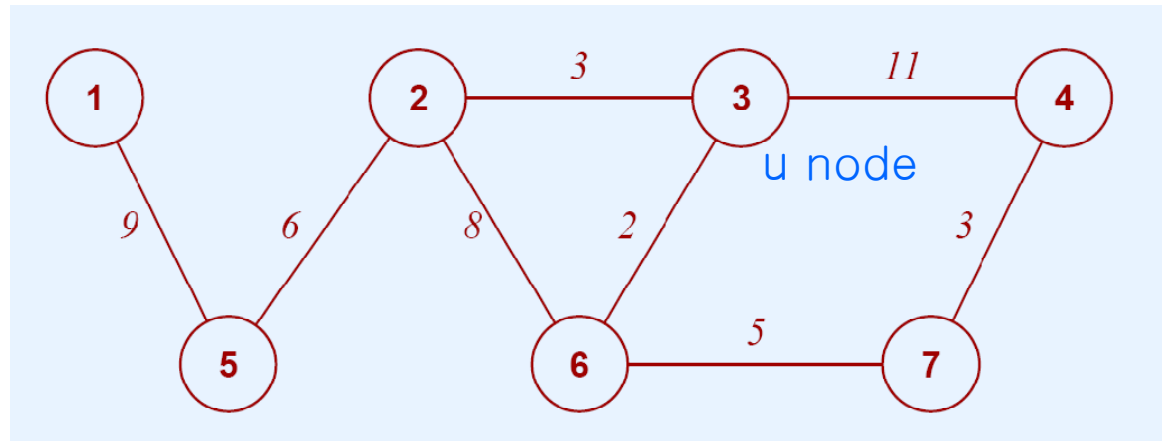
Algorithm Intuition

- Start with self as source node
- Move outward
- At each step
 - Find node u such that it
 - * Has not been considered
 - * Is “closest” to source
 - Compute
 - * Distance from u to each neighbor v
 - * If distance shorter, make path from u go through v

Base (반복적으로 계속 outward)

Result Of Dijkstra's Algorithm

The same example as before



- Example routes from node 6
 - To 3, next hop = 3, cost = 2
 - To 2, next hop = 3, cost = 5 ($C = D(3) + \text{edge}(3,2) = 5 < D(v) = D(2) = 8$)
 - To 5, next hop = 3, cost = 11
 - To 4, next hop = 7, cost = 8

Early WAN Technologies

- ARPANET
 - Historically important in packet switching
 - Fast when invented; slow by current standards
- X.25
 - Early commercial service
 - Still used
 - More popular in Europe

Recent WAN Technologies

- SMDS (Switched Multi-megabit Data Service)
 - Offered by phone companies
 - Not as popular as Frame Relay
- Frame Relay
 - Widely used commercial service
 - Offered by phone companies
- ATM

Chapter 15 (Partly 15.14–15.16)

Primary Performance Measures

- Delay
- Throughput

Delay

- Time required for one bit to travel through the network
- Four types (causes)
 - Propagation delay (proportional to distance)
 - Switching delay (choose the next hop etc.)
 - Access delay (CSMA/CD etc.)
 - Queuing delay (due to store and forward)
- Intuition: “length” of the pipe

Throughput

- Number of bits per second that can be transmitted (bps, Mbps, Gbps)
- Capacity
- Intuition: “width” of the pipe
(throughput = bandwidth)

Components Of Delay

- Fixed (nearly constant)
 - Propagation delay
 - Switching delay
- Variable
 - Queuing delay
 - Depends on throughput

Relationship Between Delay And Throughput

- When network idle
 - Queuing delay is zero
- As load on network increases
 - Queuing delay rises
- Load defined as ratio of throughput to capacity
 - Called utilization

Relationship Between Delay and Utilization

- Define
 - D_0 to be the propagation and switching delay
 - U to be the utilization ($0 \leq U \leq 1$)
 - D to be the total delay

- Then

$$D = D_0 / (1 - U)$$

e.g. half utilization $U=1/2$, delay doubles

- High utilization known as congestion

Practical Consequence

Any network that operates with a utilization approaching 100% of capacity is doomed (disastrous delay).

Delay–Throughput Product

- Delay
 - Time to cross network
 - Measured in seconds
- Throughput
 - Capacity
 - Measured in bits per second
- Delay * Throughput
 - Measured in bits
 - Gives quantity of data “in transit”

Jitter

- Variance in Delay
- Critical in real time audio and video
- Isochronous Network: zero jitter

(Asynchronous Network: Opposite to isochronous
→ requires additional protocol action to audio/video
since it is cheaper)

Summary

- Two performance measures
 - Delay
 - Throughput
- Delay and throughput interact
- Queueing delay increases as utilization increases
- Delay X Throughput
 - Measured in bits
 - Gives total data “in transit”



PART IX



Protocols and Protocol Layering

Protocol

- Agreement about communication
- Specifies
 - Format of messages
 - Meaning of messages
 - Rules for exchange
 - Procedures for handling problems

Need For Protocols

- Hardware is low level
- Many problems can occur
 - Bits corrupted or destroyed
 - Entire packet lost
 - Packet duplicated
 - Packets delivered out of order