Network Layer Design Issues

- Fundamentals of Internetworking
  - Concept of Universal Network
  - Internetworking with Bridges
- Network Level Internetworking
  - Network Layer Functions
    - Addressing
    - Routing
    - Congestion Control

Fundamentals of Internetworking

- Internetworking allows users of different networks to cooperate across network boundaries
- Cooperation is motivated by:
  - Exchange of Electronic mail
  - Banking and remote transaction
  - Rapid dissemination of pertinent information
  - Market forecasting
- In essence, Internetworking is motivated by the fact that value to user of many networks is strongly correlated with the number of other users that can be reached

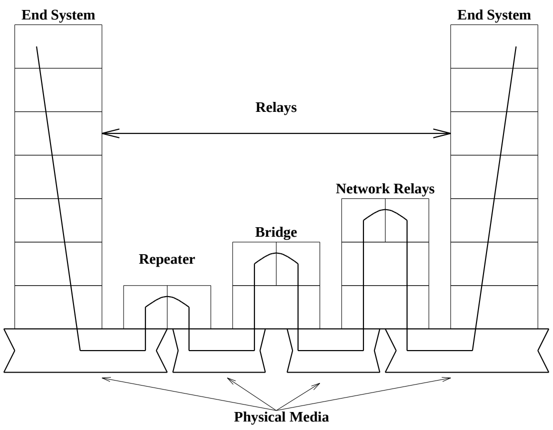
Requirements for Network Interconnection

- The main goal of internetworking is to provide transparent access to all facilities of each network by users of other networks
  - Concept of universal or metanetwork
- A few requirements are necessary
  - Provide communication between networks, at least the equivalent of the OSI physical or data link layer
  - Provide routing between users
  - Provide an accounting service to keep track of the use of various network resources

Relays, Bridges, Routers and Gateways

- A layer-(n) relay is a device that interconnects two systems not directly connected to each other (OSI Ref. Model)
  - Layer-(n) relay shares a common layer-(n) with other systems, but does not participate in layer-(n+1) protocol
- Current terminology, however
  - Repeater : Physical layer relay
  - Bridge : Data link layer relay
  - Router : Network layer relay
  - Gateway : Any relay at higher layer than network layer
- Terminology is non uniformly used in the literature

Relays, Bridges, Routers and Gateways



Network Interconnection Issues

- **Development of appropriate approaches depend upon resolution of a variety of issues**
  - **Addressing and Naming issues**
  - **Routing techniques, Error Control, Flow Control**
  - **Information Quanta, Packet Sizes**
  - **Connection-oriented versus connectionless**
  - **Timings, Interrupts**
  - **Access Control**
- **Appropriate answers depend on degree of compatibility among protocols**

Networking with Bridges

- **Simplest of internetworking devices Connects LANs**
  - **Usually, protocols for the physical and medium access layers are identical**
    - **Minimal processing is required of a bridge**
  - **For different protocols either translation or encapsulation is used**

Bridged Networks

- **Advantages over single large LAN**
  - **Higher reliability due to partitioning**
  - **Improved performance depending on traffic pattern**
  - **Improved security**
  - **Better accommodation of geographical restrictions**

Bridge Functions

- Consider a bridge connection LAN A to LAN B
- Read all frames transmitted on LAN A, accept those addressed to machines on LAN B.
- Using MAC protocol for B, retransmit frames onto B.
- Do the same for B-to-A traffic.

Bridge Requirements

- No modification to content or format of frames
- Enough buffer space to handle peak loads
- Some addressing and routing intelligence
  - May have to forward frames intended for machines not directly connected to either LAN
    - Learning functionalities

Bridge Routing
Fixed Routing

- Routing table developed off-line
- Appropriate entries stored at each bridge
- Simple and widely used
- May be impractical in a dynamic network

Bridge Routing
Source Routing

- Sending station determines the route that the frame will follow
- Routing information included in frame
- Bridges read the information and determine if they should forward the frame
- No routing tables at bridges
- Learning Routes
  - Manual loading
  - Query neighbors
  - Dynamic route discovery procedure using ‘‘all routes’’ message

Bridge Routing
Transparent Routing

- Bridges automatically develop routing table that changes in response to dynamic conditions
- Frame forwarding: keep list of LAN ids for each port, indicating which LANs can be reached through that port
- Address learning: when a frames arrives on a particular port, bridge updates its forwarding database appropriately
  - Timer eliminates unused entries
- Spanning tree algorithm: bridges exchange routing information in order to configure network routing paths as a spanning tree

Fundamentals Of Network Layer Protocols

- Main purpose of Network Layer Protocol :
  - Relieve the transport layer from the dependency on the operational characteristics and topology of the of the transmission facilities
  - The network becomes totally transparent to upper layer protocols

Network Layer Protocols

- The network layer is, conceptually, the most complex layer in the reference model
  - Peer processes at all nodes must cooperate to provide the required network services
- The network layer may provide two types of services:
  - Connection-oriented service or
  - Connectionless service

Network Layer Protocols

- The network layer provides functionalities that completes the definition of the interaction between the host and the network
  - Defines the basic unit of transfer across the network
  - Defines Naming and Addressing
  - Provides services to achieve routing and congestion control
  - Allows for Internetworking to support universal communication service

Addressing in Internet Environments

- Communication between users require several forms of addressing
  - Typically,
    - Physical Address
    - Data Link Address
    - Network Address
    - Higher layer addresses for end to end communications

Network Level Naming and Addressing

- To deliver information, the network layer needs to know the location of the network node
  - Each node is assigned a network address, which designates a connection of a node to a network
- Most applications allow mnemonic specification of machine names. e.g.,
  rlogin speedy.pitt.edu
- Actually, names and addresses are just different forms of the same thing.

Flat Names

- The network layer may define a flat name space
  - Each machine name consisted of a sequence of characters without any further structure
    - names are convenient and short
- Disadvantages
  - Conflicts, central authority for adding new names, name-address bindings change frequently, too much traffic to a single site, hard to keep an updated list at multiple sites,

Hierarchical Names

- The set of nodes is decomposed into domains, the domains into subdomains, and so on
- Decentralization of naming mechanism through delegation of name space authority to individual organizations.
  - Used in the Internet, where hierarchical machine names are assigned according to the structure of organizations that obtain authority for parts of the name space, not necessarily according to the structure of the physical network interconnections.

Mapping Names to Addresses

- Nodes use the Domain Name System (DNS) to map node names network addresses
- DNS is distributed in that servers at multiple sites cooperate in solving the mapping problem
  - DNS is efficient in that most names are mapped locally
    - Name server, program that supplies name-to-address translation, often executing on a dedicated processor.
    - Name resolver, client software, which may communicate with more than one name server in resolving a name
    - Resolver must know how to contact at least one name server
    - Every name server knows the address of at leas one root server

Name Resolution

- Resolving names by working down a list of name servers
  - Most name resolution refers to local names (don't want to translate these in a distributed manner)
    - May flood nodes at root of the tree
    - Vulnerable to failures of root nodes

Name Caching

- Each server maintains a cache of recently used names as well as a record of where the mapping information was obtained
  - Upon receiving request, server:
    - checks to see if it has jurisdiction
    - checks the cache and reports (nonauthoritative, with name of responsible server)
    - if not there, must check with authoritative server

Physical Addresses

- At the physical level, nodes must use physical addresses in order to communicate
  - Network addresses are essentially virtual addresses.
  - The virtual address must be mapped to the physical address.
    - Direct mapping
    - Dynamic mapping

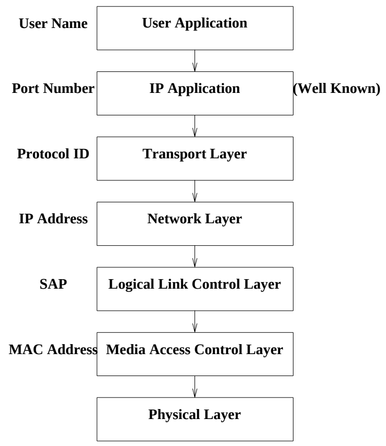
Physical Addresses
Direct Mapping

- If administrator can assign physical addresses, network address may be trivially converted to physical address
  - Example is proNET token ring (hostid = physical address)
- if administrator cannot assign physical addresses, network address may be converted to physical address using a mapping table

Physical Addresses
Dynamic Mapping

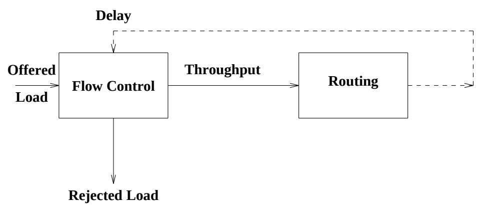
- Sending node must determine physical address before transmission
  - Address Resolution Protocol performs this function
    - Sender broadcasts a query packet containing the network address
    - The designated node responds with its physical address
  - Address resolution cache used to improve performance (reduce number of broadcasts)
  - When requesting address, include your address in query
  - When booting, broadcast your address

Complete Naming Scheme



Routing and Flow Control

- Routing determines paths for data flowing between source and destination pairs
- Flow control regulates traffic to avoid congestion



Routing

- The principal task of the network layer is to accept packets from the source node and deliver them to the destination node.
- Generally, more than one route is possible
  - A routing function must be performed
  - A number of desirable attributes for routing function
    - Correctness, Simplicity, robustness, Stability, Fairness, and Optimality

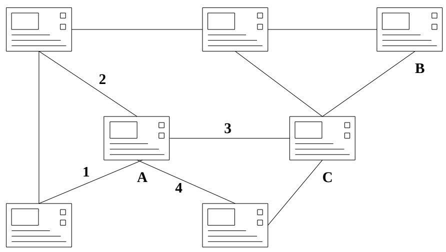
Dimensions of a Routing Task

- In connection-oriented service, routing decisions are only made when a new virtual circuit is being set.
- In datagram service, the decision is made anew at the arrival of every new packet.

Routing with Datagrams

- Each node maintains a routing table
  - Entries in the routing table may be based on estimates of congestion in the network
  - Entries may contain alternate links for routing
    - Alternate links can be selected randomly, in a round robin fashion, or used to balance the traffic load

Datagram Routing Tables



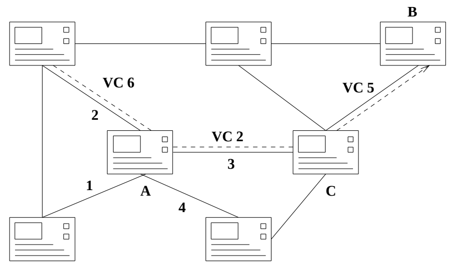
Destination	Link	Fraction
B	2	0.6
	3	0.4
C	3	0.8
	4	0.2

Routing Table at Node A

Routing with Virtual Circuits

- **Routing tables for virtual tables identify circuit numbers instead of destinations**
  - ○ **Entries in the routing table specify for each incoming link and incoming virtual circuit number, the corresponding outgoing link and outgoing virtual-circuit number**
    - □ **Nodes assign numbers independently, so numbers may change along the path**

Virtual Circuit Routing Tables



IN		OUT	
Link Number	VC Number	Link Number	VC Number
2	6	3	2
1	5	4	7
⋮	⋮	⋮	⋮

Routing Table at Node A

Routing Algorithms

- **The main design goals of a routing algorithm are :**
  - ○ **Simplicity and Low overhead**
  - ○ **Robustness and Stability**
  - ○ **Rapid Convergence**
  - ○ **Flexibility**
  - ○ **Optimality**

Simplicity

- **Routing algorithms must offer their functionalities Efficiently**
  - ○ **Minimum of software and utilization overhed**
- **Efficiency is important, particularly when the software is implemented to run over a computer with limited resource capabilities**

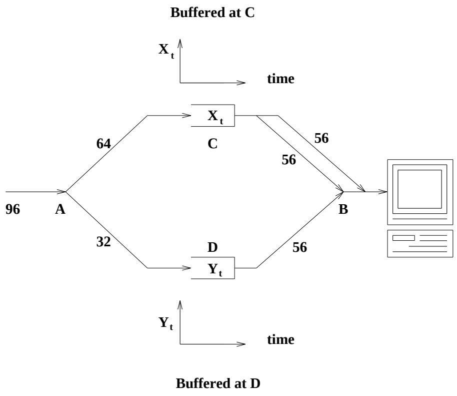
Robustness

- Routing algorithms must perform correctly despite unforeseen circumstances
  - Hardware failures
  - High load conditions
  - Incorrect implementations
- Unstable routing algorithms can cause considerable problems when they fail at the network junction points

Robustness

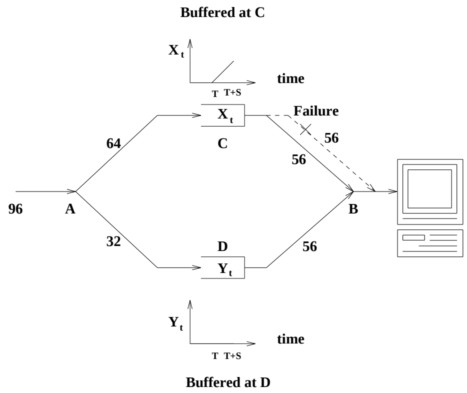
- Robustness requires three tasks:
  - Monitoring network status
  - Update routing decision to reflect new changes
  - Enforce new decisions to react as fast as possible to network changes

Robustness Example



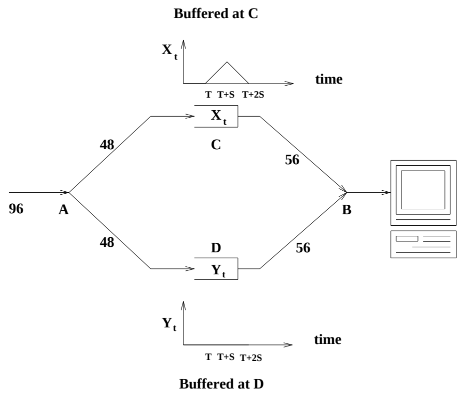
- Buffers do not accumulate bits, since C and D rate of transmission is faster than input stream bit rates

Robustness Example
Link Failure



- Buffering occurs at node C, since the arrival rate is higher than the transmission rate

Robustness Example
Routing Decision Adjustment



- After S units of time, A learns of the failure and updates routing decisions to take failure into consideration
  - Buffers at C begin to empty

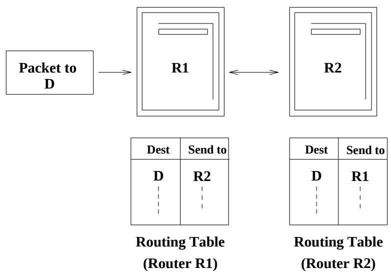
Robustness

- The time S to learn failure has two implications:
  - Buffers at C must be able to store S $\times$ 8 Kbps
  - Bits are delayed at node C during the failure
    - It takes (S $\times$ 8 Kbps) /(56 Kbps) to clear the backlog
    - For S=1, the buffer must hold 8,000 bits and the delay is S/7 = 140 ms

Stability and Rapid Convergence

- Convergence is the process of agreements by the routers on the optimal routes
  - Dialogue among routers is carried by update messages
  - Routing updates stimulate recalculations of optimal routes
- Routing algorithms that converge slowly can cause loops or network outages

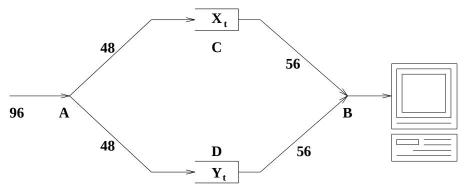
Slow Convergence
and
Routing Loops

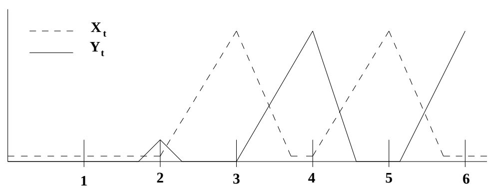


Flexibility

- Flexibility refers to the capability of the routing algorithm to adapt quickly and accurately to a variety of network circumstances
- Flexible routing algorithms can adapt easity to :
  - Changes in network bandwidth
  - Router queue size
  - Network delay
  - Link failure

Stability





Routing Desirable Characteristics
Fairness

- Routing must accommodate traffic with different quality of services (QoS)
- Packets within the same class should suffer similar delays

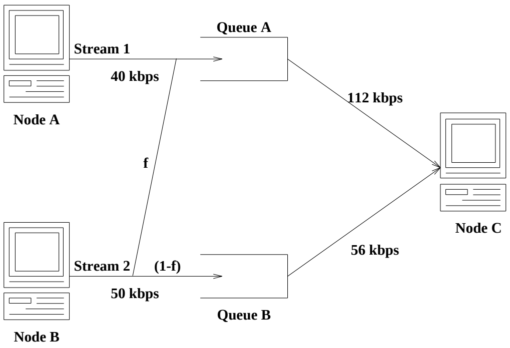
Fairness

- A commonly used formula to compute the delay, D, a packet suffers at a node is:

$$D = \frac{1}{(\mu - \lambda)}$$

  - μ is the transmission rate (packets per sec)
  - λ is the packet arrival rate (packets per sec)

Fairness



- Stream Packet Size : 1,000 bits

Fairness

- Using the delay expression, we can write:

$$D_A = \frac{1}{112 - \lambda_A}$$

Where $\lambda_A = 40 + f \times 50$ Packets/Sec

$$D_A = \frac{1}{72 - f \times 50}$$

- The average packet delay of Stream 1, $D_1 = D_A$

Fairness

- Similarly, using the delay expression, we can write:

$$D_B = \frac{1}{56 - \lambda_B}$$

Where $\lambda_B = (1 - f) \times 50$ Packets/Sec

$$D_A = \frac{1}{6 + f \times 50}$$

- The average packet delay of Stream 2, $D_2 = f \times D_A + (1 - f) \times D_B$

Fairness

- A fair routing algorithm must select f so that:

$$D_2 = D_1 \Rightarrow D_A = D_B$$

  - Solving for f results in:

$$f = 0.66$$

Optimality

- Optimality refers to the capability of the routing algorithm to find the ‘‘Best Route’’
  - It is strictly defined with respect to the routing metrics and metric weightings
- May be difficult to achieve

Routing Metrics

- Path length, sum of the costs associated with links
  - Frequently reduces to hop count
- Reliability, link bit error rate, usually assigned by network administrator
- Delay, as the amount time required to move packets from source to destination
  - Delay depends on link bandwidth, router queues and network load
- Communication costs to reduce expenditures

Optimality

- Consider the previous network and assume that the objective is to minimize the average packet delay
- The average delay, D, can be expressed as:

$$D = p_1 D_1 + P_2 D_2$$

Where

- p_i is the proportion of packets from stream i, $i = 1,2$
  - $p_1 = \frac{40}{90}$ and $p_2 = \frac{50}{90}$

Optimality

- Taking the derivative of the delay expression D with respect to f and setting it to 0 produces the value of f that minimizes the average network delay
  - Solving the equation for f results in:

$$f \approx 0.465$$

- The value of f that minimizes the average delay is not the same as the value that achieves routing fairness

Routing Algorithm Classification

- Routing algorithms can be classified as :
  - Static or Dynamic
  - Single path or Multipath
  - Flat or Hierarchical
  - Intradomain or Interdomain
  - Link State or Distance Vector

Static Routing

- Static routing algorithms use a mapping predefined by the network administrator to determine routes
  - Routing table mappings do not change, unless the administrator changes them
  - Static routing algorithms are simple to design
  - They only work well in environments where network traffic is predictable and network design is simple

Dynamic Routing

- Dynamic routing algorithms adjust, in real time, to the current network condition
- Dynamic changes are conveyed by routing update messages
  - New routes may result upon exchange of update messages
- Dynamic routing may be supplemented by static default routes, ''last resort''

Single Path or Multipath

- Routing algorithms may support multipaths to the same destination
  - Multipath permits traffic multiplexing over multiple links
- Multipath algorithms may improve network throughput

Flat Routing

- In a flat routing algorithms, all routers are peers
  - Routing algorithm operates on a flat name space

Hierarchical Routing

- A group of routers form what amounts to a Routing Backbone
- Hierarchical routing is usually defined based on the concept of Autonomous Systems (ASs), or Areas
  - Backbone routers can communicate between domains, while other routers are restricted to within their domains
- Hierarchical routing mimics the organization of most companies
  - Efficient support for the company traffic pattern

Host or Router Intelligent

- Source routing assumes that the end node determines the entire route to destination
  - Routers merely act as store and forward nodes
- Router intelligent algorithms rely entirely on the router to determine the path to destination

Host or Router Intelligent Trade-Off

- Trade-off is route optimality versus traffic overhead
- Source routing can achieve optimality, since end node computes all paths to destination
  - Optimality may very to reflect a particular system's definition
- Discovering all routes, however, may be costly

Intradomain or Interdomain

- Intradomain algorithms are optimized to work within a domain
- Interdomain algorithms are optimized to work between domains
  - The nature of these algorithms are different
  - An ‘‘optimal’’ interdomain routing algorithm may not necessarily be optimal or even perform well between domains

Routing Techniques
Path Optimality Principle

- ‘‘If router J is in the optimal path from I to K, then the optimal path from J to K also falls along the same route’’
- As a direct consequence, the set of optimal routes from all sources to a given destination form a tree routed at the destination
  - The tree is refereed to as ‘‘Sink Tree’’

Routing Techniques
Bellman-Ford-Moore Algorithm

- The network is represented as a set of nodes with N elements
- D_i represents the minimum distance for reaching node i from a specified origin
- C_{ij} represents cost (distance) of the link between node i and node j
  - $C_{ii} = 0$,
  - $C_{ij} = \infty$, if node i and node j are not adjacent

Shortest Path Algorithms
Bellman-Ford-Moore Algorithm

- The objective is to find optimum routes from a given source node, say 1, to all other nodes in the network
- The principle of optimality can be expressed as:

$$D_i = \min_j \left\{ D_j + C_{ij} \right\}, \text{ for } all\ i \neq 1$$

$$D_1 = 0$$

Bellman-Ford-Moore Algorithm

- The algorithm iterates on the number of ''hops'' in the path
  - First find optimal paths from a prescribed source node to all other nodes subject to the constraint no path contains more than one hop, then finding optimal paths subject to the constraint no path contains more than two hops, ... three hops, etc.
- The shortest path subject to the constraint no path contains more than h hops is called the shortest ($\leq h$) path

Bellman-Ford-Moore Algorithm

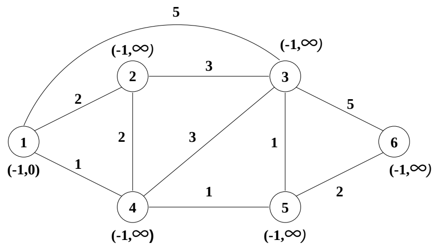
- Based on the shortest ($\leq h$) path concept, the algorithm can be expressed as follows:
  - Let $D_i^{(h)}$ be the shortest ($\leq h$) path from **1** to i
  - For each successive $h \geq \mathbf{0}$, compute

$$D_i^{(h+1)} = \min_j \left\{ D_j^{(h)} + C_{ji} \right\}, \text{ for } all\ i \neq 1$$

$$D_i^{(0)} = \begin{cases} 0 & i = 1 \\ \infty & i \neq 1 \end{cases}$$

  - Connect i with the predecessor node, j, for which the minimum is obtained, possibly eliminating any connection formed during an earlier iteration
  - Continue until no more changes occur in the next iteration

BFM Algorithm
Example



Q={1}

BFM Algorithm
Complexity Analysis

- Worst case computational requirements?
  - For a network of N nodes, a path can contain at most $N-1$ links
    - Hence, $N-1$ iterations, over the number of hops, before convergence
  - For each iteration, the minimization is performed for $N-1$ nodes
  - For each node, the iteration involves at most $N-1$ alternatives
- Thus, worst case computational requirements is $O(N^3)$
  - More careful counting results in $O(mL)$, where m is the number of iteration required to converger, and L the number of links

Shortest Path Algorithms
Dijkstra Algorithm

- **The objective is to find shortest paths from a specified source, say 1, to all other destinations**
- **The approach is similar to BFM, but iterates on the length of a path**
  - **Let** N_c **be the set of nodes currently connected with the destination, and** D_j **the current estimate of distance of node** j **from the destination**
    - **Initially,** N_c **= {1},** $D_1 = 0$**, and** $D_j = \infty$**, for** $i \neq 1$

Dijkstra Algorithm

- **Step 1: Update labels**
  - **For all** j *not in* N_c**, set**

$$D_j = \min_{i \in N_c} \left\{ D_j, D_i + C_{ij} \right\}$$

    - **If min is the same as previous** D_j**, leave the tentative connection of** j **to a successor unchanged**
    - **If min is different, connect** j **to the successor** i**, for which the minimum is obtained**

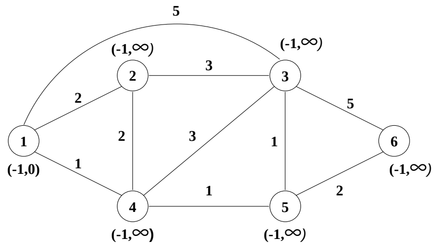
Dijkstra Algorithm

- **Step 2: Find next closest node**
  - **Find** i **not in** N_c**, for which**

$$D_i = \min_{j \text{ not in } N_c} \{D_j\}$$

    - **Set** $N_c = N_c \cup \{i\}$**, and permanently connect** i **to the appropriate successor**
  - **If** $N_c = N$ **then stop, else return to Step 1**

Dijkstra Algorithm
Example



Q={1}

Dijkstra Algorithm
Complexity Analysis

- **Each step requires a number of computation proportional to the number of nodes N**
- **Steps are repeated $N-1$ times**
  - **In the worst case, the computational cost is $O(N^2)$, better than $O(N^3)$**
    - **A significant reduction over BFM, if the network is small and dense with a high ratio of number of links to number of nodes**
    - **Usually, however, $L \ll N(N-1)$, resulting in $m \ll N$, and $mL \ll N^3$**

Distributed Asynchronous
BFM Routing Algorithm

- **One of the main reason of the popularity of BFM is its suitability for implementation in a distributed manner**
- **A precise implementation of algorithm requires synchronization among the nodes**
  - **All nodes must start at the same time**
  - **All nodes must abort and restart if a link status changes while computation is running**

Distributed Asynchronous
BFM Routing Algorithm

- **Practical implementation uses an asynchronous approach which eliminates the need for synchronization by computing distances at regular intervals**
  - **Convergence occurs to correct distances if no changes occur during route update time**

Routing Algorithms
Practical Implementations

- **Modern computer networks use dynamic routing algorithms rather than static ones**
- **Two dynamic algorithms are frequently used**
  - **Distance Vector**
  - **Link State Protocol**

Distance Vector and Link State

- Distance Vector algorithms (BFM) call for each router to send all or some portion of its routing table but only to its neighbors
- Link State algorithms (Dijkstra) flood link state information to all nodes
  - Each router sends only portion of the routing tables that describes the state of its own links
    - Link State algorithms converge more rapidly, so they are less prone to routing loops
    - Link State algorithms may be computationally expensive

Distance Vector Routing

- Basic Idea:
  - Each router starts out with a distance 0 for those networks it is attached to
  - Each router transmits its distance vector to each of its neighbor periodically or whenever information changes
  - Each router calculates its own distance vector based on the information received from each neighbor

Distance Vector

- Gateway Z, attached to Net 1 and Net 2

Destination	Distance	Route
Net 1	0 (hops)	Direct
Net 2	0 (hops)	Direct

- At a given time

Destination	Distance	Route
Net 1	0 (hops)	Direct
Net 2	0 (hops)	Direct
Net 3	9 (hops)	Gateway A
Net 4	5 (hops)	Gateway B
Net 5	7 (hops)	Gateway X
Net 6	3 (hops)	Gateway D
Net 7	3 (hops)	Gateway X

Distance Vector

- Z receives update message from X

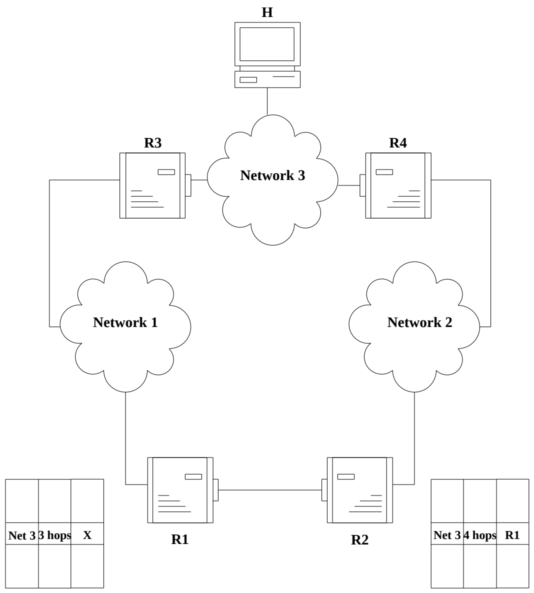
Destination	Distance
Net 1	2 (hops)
Net 3	5 (hops)
Net 4	7 (hops)
Net 5	6 (hops)
Net 6	10 (hops)
Net 7	4 (hops)
Net 8	4 (hops)

Distance Vector

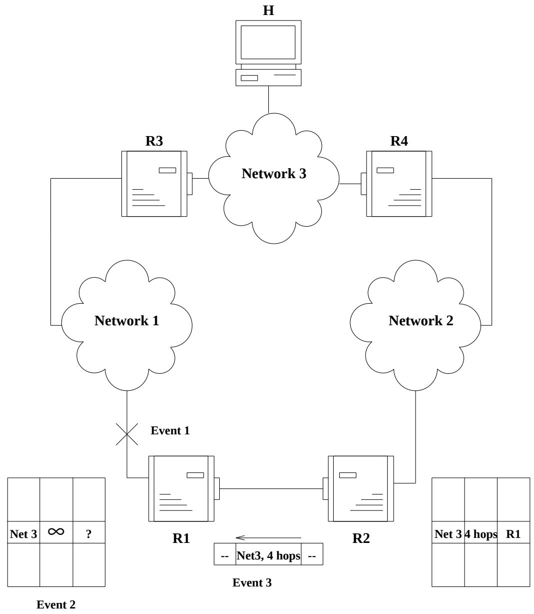
- Z updates its table

Destination	Distance	Route
Net 1	0 (hops)	Direct
Net 2	0 (hops)	Direct
Net 3	6 (hops)	Gateway X (•)
Net 4	5 (hops)	Gateway B
Net 5	7 (hops)	Gateway X
Net 6	3 (hops)	Gateway D
Net 7	5 (hops)	Gateway X (•)
Net 8	5 (hops)	Gateway X (•)

Distance Vector Shortcomings
Infinite Loop



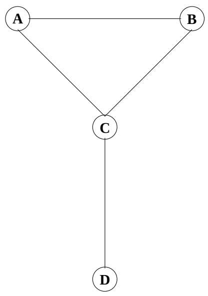
Distance Vector Shortcomings
Infinite Loop



Infinite Loop Fixes

- Split Horizons
  - Never send information about a route back in the direction from which it came
  - Helps prevent two-node routing loops
  - The solution may fail in some cases

Split Horizon



Initially, both A and B have a distance 2 to D, and C has a distance 1
Distances are computed based on C's advertizements
If CD link goes down, C realizes that it cannot get to D
Neither A nor B tell C about distances to D
A, however, may hear from B it can go to D in 2
Both nodes will gradually count to infinity

Split Horizon with ''Poison Reverse''

- **Split horizon with poison reverse improves distance vector convergence over simple split horizon by advertising all network IDs to all neighbors**
  - **But those network IDs learned from a given direction are advertised to the same direction with a metric of ∞, indicating that the network is unavailable**
- **Poison reverse has no benefit beyond split horizon in a single path internetwork**
- **In a multipath internetwork, split horizon with poison reverse greatly reduces count to infinity and routing loops**

Infinite Loop Fixes
Poison Reverse Updates and Hold Down

- **Increases in routing metrics generally indicates routing loops**
  - **When increase is observed, send ''Poison Reverse'' updates to remove route and place it in ''Hold Down''**
    - **''Hold Down'' causes discarding of any changes affecting recently removed routes for a period of time**
- **Prevent larger routing loops, but may slow down convergence**

Distance Vector Shortcomings

- **Slow convergence when routes change frequently**
  - **Information cannot propagate through a node until that node recalculates its routing information**
  - **Many routers may become nonfunctional as a result of the delay**
- **May cause gateways to have wrong information**
- **Does not scale properly to large networks**
- **May require large message exchange before stability**

Link State Routing

- The algorithm requires that each participant has complete network topology information
- Basic steps include:
  - ○ Discover neighbors and learn their network addresses
  - ○ Measure the delay or cost to each of the neighbors
  - ○ Construct a packet containing the information just learned
  - ○ Send the packet to all other routers
  - ○ Compute the shortest path to every router

Link State Routing

- Each participant actively tests the status of all links to neighbor gateways
  - □ This accomplished by periodically exchanging ''alive'' messages with neighbors
- ○ Each participant forms and propagates the link status information to all other participant using a Link State Packet
  - □ Periodically,
  - □ When a neighbor becomes active,
  - □ When a cost of a link changes (after a link failure)
- ○ Armed with a complete map of the topology, participants compute routes to each destination

Link Cost

- Routers are expected to have a reasonable estimate of the delay to each of the neighbors
  - ○ Can be achieved by measuring the round trip time of a special ''ECHO'' request packet sent over the link to the neighbor
    - □ In response, the neighbor is expected to send an ''ECHO'' reply immediately
    - □ For better results, the test can be conducted several times and averaged out

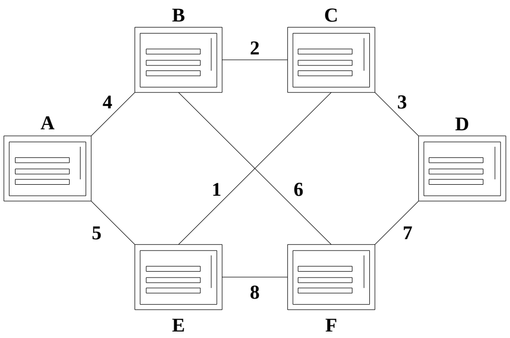
Link Cost Measurement
Load Factor

- Load can be factored in by starting the timer when the ''ECHO'' packet is queued, not when it reaches the front of the queue
- Should the load be taken into consideration?
  - ○ Including traffic-induced delays in the measurements helps the router select the unloaded links over the loaded ones with the same bandwidth
  - ○ Factoring the load in the delay measurements may cause routing table oscillations which leads to undesirable erratic routing behavior

Link State Protocol Packets

- **Link State Packet contains the following information:**
  - **Identity of the sender**
  - **Sequence Number, incremented for each new packet sent**
  - **Age of the packet**
  - **list of the neighbors and the delay to reach them**

Link State Protocol Packets Example



A	
Seq #	
Age	
B	4
E	5

B	
Seq #	
Age	
A	4
C	2
F	6

C	
Seq #	
Age	
B	2
D	3
E	1

D	
Seq #	
Age	
C	3
F	7

E	
Seq #	
Age	
A	5
C	1
F	8

F	
Seq #	
Age	
B	6
D	7
E	8

Link State Routing Sequence Numbers and Age Fields

- **Routers can get confused if sequence numbers wrap around**
  - **Use a large sequence number field**
    - **A 32-bit sequence number would take 137 years to wrap around, assuming one LSP per second**

Link State Routing Sequence Numbers and Age Fields

- **If a router crashes, it may lose track of its last sequence number**
  - **If it starts again at 0, the packet will be rejected as a duplicate**
- **If a sequence number is corrupted it may make other packets obsolete**
  - **A 1 bit error causes the sequence number 4 to be received as 65,540**
    - **All packets 5 through 65,540 will be considered obsolete**
- **Include an ''Age'' field for each packet which is decremented once per second**
  - **Age field is set to max_value (on the order of an hour) by the router that generates the LSP**

Link State Routing
Disseminating LSP Rules

- **Only newly generated LSP are forwarded to neighbors**
  - **LSPs are not flooded immediately, but queued in a holding area for a short period of time**
- **Discard any LSP packet if its ‘‘Age’’ is 0**
  - **Packet is considered too old**
  - **Prevents initially lost packets from living for an indefinite period of time**
- **Discard any received LSP packet from the same source, if the received packet is identical to the one stored in the memory**
  - **If the two packets are different, discard the older one**

Link State Routing
Disseminating LSP to Routers

- **To guard against errors on the router-to-router link, LSPs are acknowledged**
  - **When a line goes idle, the LSPs holding area is scanned in round robin order to select a packet or an acknowledgement to send**

Disseminating LSP to Routers
Router Data Structure

Link State Packets

	R1 Age:60 #37		R2 Age:60 #10		R3 Age:58 #123		R4 Age:60 #42		R5 Age:59 #35		R6 Age:60 #100	
	S	A	S	A	S	A	S	A	S	A	S	A
Neighbor 1	Y				Y		Y				Y	
Neighbor 2		Y						Y			Y	
Neighbor 3	Y				Y							
.	Y					Y						
.	Y			Y			Y				Y	
.	Y		Y									
.	Y				Y		Y					Y
Neighbor n	Y						Y					

S : Send Flag
A : Ack Flag

Disseminating LSP to Routers
LAN Based Interconnection

- **Consider the LAN itself as a node**
  - **A router is elected to be the ‘‘designated router’’**
    - **The designated router names the LAN and constructs an LSP on behalf of the LAN**
    - **Every router, including the designated router, reports only a single link in its LSP for the LAN itself**

LAN Based Interconnection
Reliable dissemination

- Provide periodic summaries of the Database content
  - A special packet, Sequence Numbers Packet (SNP), which contains the latest sequence numbers of all LSPs is multicast
  - A router can explicitly request any missing LSP from the designated router
  - If a router noticed that the designated router missed its LSP, the router retransmits the LSP
  - This scheme is adopted by ISO in IS-IS and DECnet Phase V

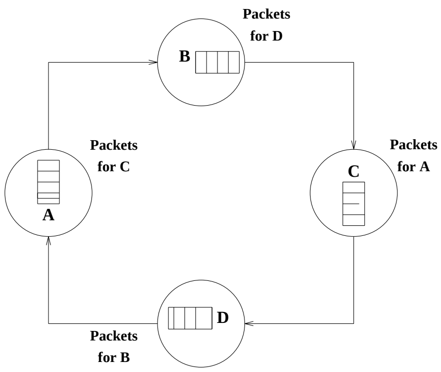
Flow Control

- Flow control regulates the traffic between a source and a destination
  - The main goal of flow control is to prevent overflow of the buffers dedicated to an individual connection.
  - Flow control is perceived as a bipartisan agreement between the source and destination

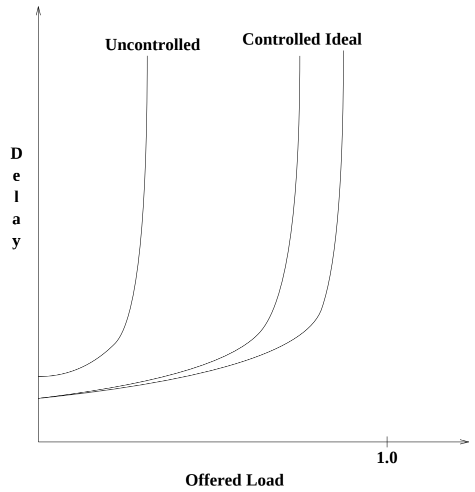
Congestion Control

- The main tasks of congestion control :
  - Regulate the amount of data admitted to the network.
  - Provide efficient utilization of network resources such as logical channels, buffers, transmission bandwidth, processor time.
  - Meet the user performance requirements, when possible.

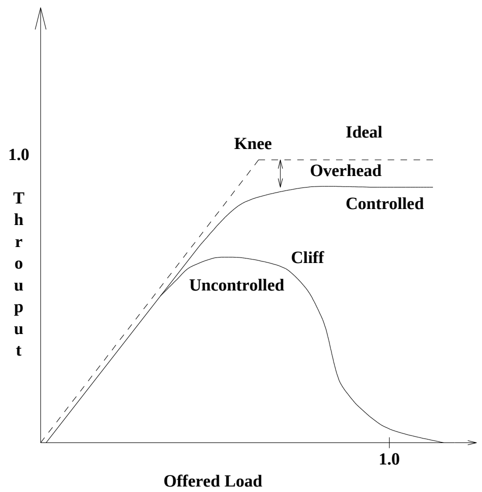
Congestion Effect



Congestion Effect



Congestion Effect



Congestion Management

- **The controlling action may be applied to many source-destination pairs indiscriminately or simultaneously.**
- **The term congestion management incorporates**
  - **congestion control, and**
  - **congestion avoidance techniques.**

Congestion Management

- **Congestion control in communications networks is a difficult problem.**
  - **Dramatic bandwidth mismatch between links.**
  - **Unbalanced load among different networks.**
  - **Sharp contrast between the very high speed of LANs and packet processing rate of routers.**
  - **State control is hard to achieve in distributed environment.**

Congestion Control

- The objective of flow and congestion control schemes includes:
  - Striking a balance between gain in performance due to controls and global state information availability, and loss in efficiency owing to overhead in gathering the information
  - Fairness in resource allocation among users of the network
    - It is a non-trivial task in connectionless networks, where resources are allocated on demand.

Congestion Management Techniques

- Packet dropping
  - When the limit is exceeded, input packets are dropped
  - The technique is often employed in connectionless networks
    - Three techniques are commonly used tail drop, random drop and fair queueing

Congestion Management Techniques
Packet dropping

- The techniques are simple and consistent with the ‘‘best effort’’ philosophy
- The drawback is that dropping packets may be counterproductive because it triggers end-to-end retransmission
  - It does not provide feedback to slow down the sources that generate the network traffic

Congestion Management
End-to-End Techniques

- Choke Packets
  - The technique uses choke packets containing the header of the packet traveling in the congestion direction to indicate congestion to the end points.
  - Upon receiving the choke packet, the source declares the destination congested and slows down traffic to that destination

Congestion Management
Hop-by-Hop Techniques

- **Backpressure**
  - **The scheme is used in conventional networks (e.g., X.25) to regulate the flow along a virtual circuit**
  - **The node upstream of a congested link can throttle the flow by closing the window of permissible outstanding packets**
  - **This starts a chain reaction which eventually cuts off the supply of new packets to the virtual circuit**
  - **Credits are returned when the congestion clears**

Reservation Based Control
Open Loop

- **Source specifies its performance requirements at the connection set up time**
  - **Based on the current status of the network, the connection is either accepted or rejected**
  - **Reservation provides bandwidth and delay guarantees**
- **Reservation is only efficient if the traffic is steady and predictable (Voice/Video)**
  - **Resources that are reserved but not used are wasted (bursty traffic)**
- **Reservation based scheme are easy to manage**
  - **Walk-in scheme require dynamic resource management which may be complex**

Congestion Control
Common Misconceptions

- **Congestion is caused by shortage of buffer space**
  - **The decreasing cost of memory will eliminate the problem**
- **Congestion is caused by slow links**
  - **The availability of high speed links will eliminate the problem**
- **Congestion is caused by slow processors**
  - **The availability of high speed processors will eliminate the problem**
- **If not one, all of the above will eliminate the problem**

Congestion Control
Common Misconceptions

- **The advent of the new technology and proliferation of high speed networks have actually aggravated the congestion problem**
  - **Congestion management and avoidance mechanisms are more important than ever before**

Congestion Control
High Speed Networks

- The key issues in the design of congestion control for high speed networks is the characterization of the traffic
  - The scheme should accommodate all sorts of traffic
    - Delay sensitive, but loss insensitive, voice and video
    - Delay insensitive, but loss sensitive, data
    - Delay sensitive and loss sensitive, interactive computing and graphics

Congestion Control
High Speed Networks

- The congestion techniques need to be reassessed
  - Closed Loop or Open Loop?
  - Window or Rate based Control?
  - Router-Based or Source-Based Control?
  - Is Backpressure still applicable?
  - Prior Reservation or Walk-in?
  - One scheme is enough, or a combination of many?

Conclusion

- Discussed Internetworking Main Issues
  - Internetworking with Bridges
- Discussed Network Layer Main Issues
  - Addressing
  - Routing
  - Flow Control