EECS, University of Ottawa

Computer Communication Network

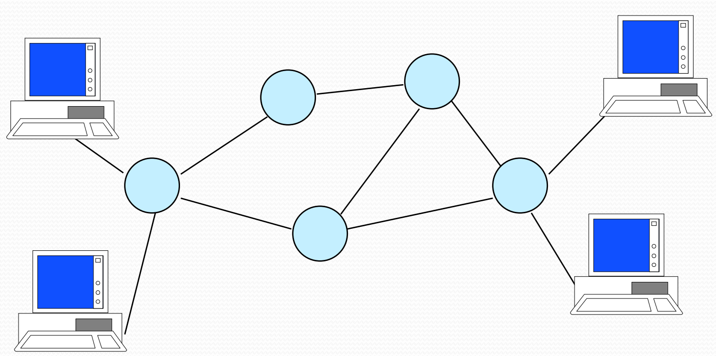
Routing & Routing Protocols

Note: some material in the slides has been taken from various other sources 1-1

1

Switching Networks

Data are transferred from source to destination through a series of intermediate nodes



1-2

2

Routing

- Routing is the selection of paths for packets.
- A routing decision is made based on some performance criterion.
- Routing is the main function of the network/IP Layer.

1-3

3

Routing Strategy

- The procedure the switching nodes follow to select the paths of individual packets or virtual-circuits.
- A good routing strategy has the following properties:
  - correctness
  - simplicity
  - robustness
  - stability
  - fairness
  - optimality
  - efficiency

1-4

4

Properties of Routing Strategies: Clarifications

- *Robust* with respect to changing conditions and equipment failures
- *Stable* if a small change in conditions leads to a small modification in the routing decisions.
- *Fair* if it results in comparable delays and available transmission rates for the different users of the same service class.
- *Optimal* if it maximizes the network designer's objective while satisfying the design constraints.

1-5

5

Routing Strategies

- Static routing
- Flooding
- Random routing
- Adaptive routing
  - isolated adaptive
  - distributed adaptive
  - centralized adaptive

1-6

6

Static Routing

- Use a single permanent route for each source to destination pair
- In many cases, static routes are manually configured by a network administrator by adding in entries into a routing table Route is fixed
  - at least until a change in network topology
  - hence cannot respond to traffic changes
- Advantage is simplicity
- Disadvantage is lack of flexibility

1-7

7

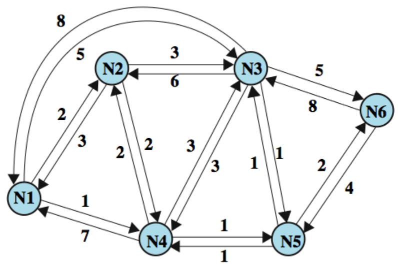
Static Routing

- A route is selected for each source-destination pair of nodes in the network.
- Implementation:
  - a central routing matrix is created, to be stored at a network control centre,
  - from this matrix, a routing table can be developed and stored at each node.

1-8

8

Example



CENTRAL ROUTING DIRECTORY

From Node

To Node	1	2	3	4	5	6
1	—	1	5	2	4	5
2	2	—	5	2	4	5
3	4	3	—	5	3	5
4	4	4	5	—	4	5
5	4	4	5	5	—	5
6	4	4	5	5	6	—

Node 1 Directory

Destination	Next Node
2	2
3	4
4	4
5	4
6	4

Node 2 Directory

Destination	Next Node
1	1
3	3
4	4
5	4
6	4

Node 3 Directory

Destination	Next Node
1	5
2	5
4	5
5	5
6	5

Node 4 Directory

Destination	Next Node
1	2
2	2
3	5
5	5
6	5

Node 5 Directory

Destination	Next Node
1	4
2	4
3	3
4	4
6	6

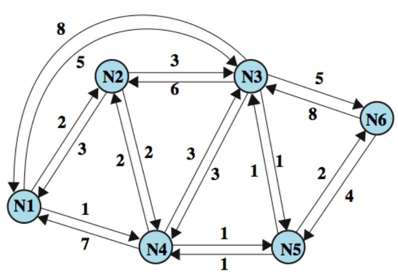
Node 6 Directory

Destination	Next Node
1	5
2	5
3	5
4	5
5	5

1-9

9

Example



CENTRAL ROUTING DIRECTORY

From Node

To Node	1	2	3	4	5	6
1	—	1	5	2	4	5
2	2	—	5	2	4	5
3	4	3	—	5	3	5
4	4	4	5	—	4	5
5	4	4	5	5	—	5
6	4	4	5	5	6	—

Node 1 Directory

Destination	Next Node
2	2
3	4
4	4
5	4
6	4

Node 2 Directory

Destination	Next Node
1	1
3	3
4	4
5	4
6	4

Node 3 Directory

Destination	Next Node
1	5
2	5
4	5
5	5
6	5

Node 4 Directory

Destination	Next Node
1	2
2	2
3	5
5	5
6	5

Node 5 Directory

Destination	Next Node
1	4
2	4
3	3
4	4
6	6

Node 6 Directory

Destination	Next Node
1	5
2	5
3	5
4	5
5	5

1-10

10

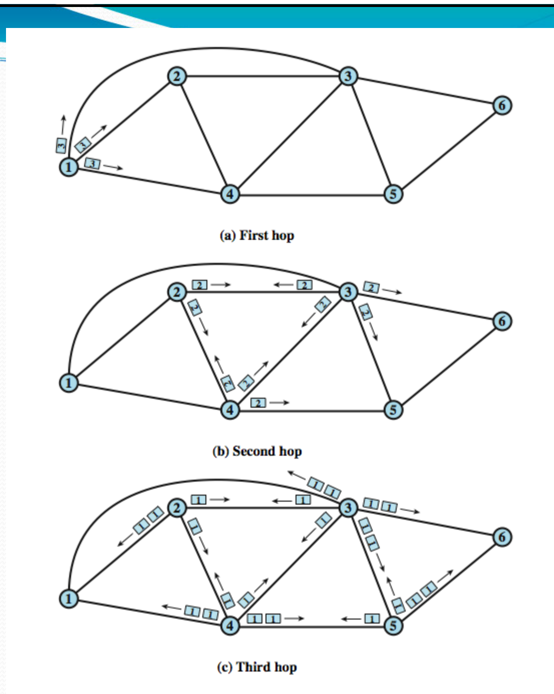
Flooding

- Packet sent by node to every neighbor
- Eventually multiple copies arrive at destination
- No network info required
- Each packet is uniquely numbered so duplicates can be discarded
- Need some way to limit incessant retransmission
  - nodes can remember packets already forwarded to keep network load in bounds
  - or include a hop count in packets

1-11

11

Flooding Example



1-12

12

Properties of Flooding

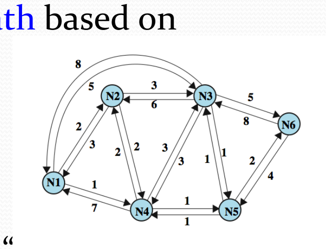
- All possible routes are tried
  - very robust
- At least one packet will have taken minimum hop count route
  - can be used to set up virtual circuit
- All nodes are visited
  - useful to distribute information (e.g. routing information)
- Disadvantage: high traffic load generated

1-13

13

Random Routing

- Simplicity of flooding with much less load
- Node selects one outgoing path for retransmission of incoming packet
- Selection can be random or round robin
- A refinement is to select outgoing path based on probability calculation
- No network info needed



$$P_i = \frac{R_i}{\sum_{j=1}^{N} R_j}$$

P_i: probability of collecting link “

R_j: data rate of link “j”

- Is typically neither least cost nor minimum hop

1-14

14

Adaptive Routing

- Used by almost all-packet switching networks
- Routing decisions change as conditions on the network change due to failure or congestion
- Requires info about network/nodes
- Disadvantages:
  - decisions more complex
  - tradeoff between quality of network info and overhead
  - reacting too quickly can ***cause oscillation***
  - reacting too slowly means info may be irrelevant
- Improved performance
- Aids congestion control (but since it is a complex system, may not realize theoretical benefits)

1-15

15

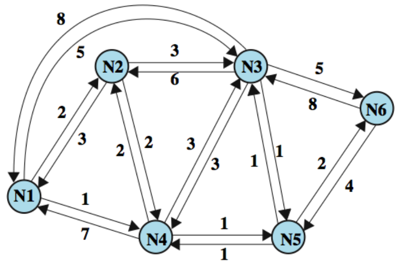
Classification of Adaptive Routing Strategies

- Based on information sources
- Local (isolated)
  - route to outgoing link with shortest queue
  - can include bias for each destination
  - Rarely used - does not make use of available info
- Adjacent nodes
  - takes advantage on delay / outage info
  - distributed or centralized
- All nodes
  - like adjacent

1-16

16

Local (isolated) Adaptive Routing



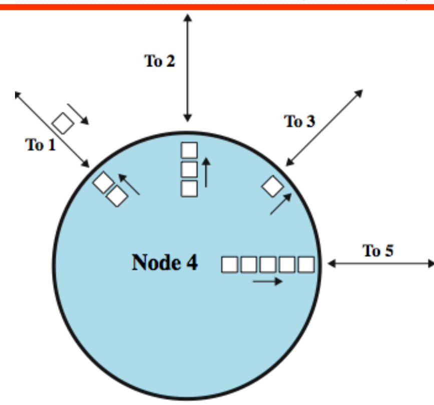
B_i: "bias" of routing a packet through link "i". The **lower the value**, the **higher the preference**.

Q_i: size of outgoing queue for link "i".

Decision rule: $min\ \{B_i + Q_i\}$

Node 4's Bias Table for Destination 6

Next Node	Bias
1	9
2	6
3	3
5	0



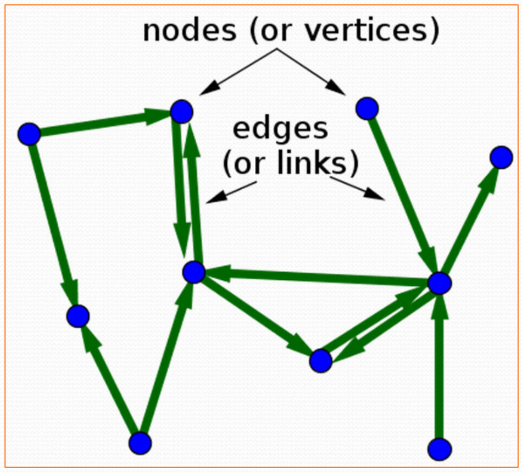
17

Routing Strategies

- Static routing
- Flooding
- Random routing
- Adaptive routing
  - isolated adaptive
  - distributed adaptive
  - centralized adaptive
- What is needed to determine "best" route?
  - Have algorithms to determine it (using math)
  - Have ways to communicate relevant parameters determining the state of the network, needed to determine "best route"

18

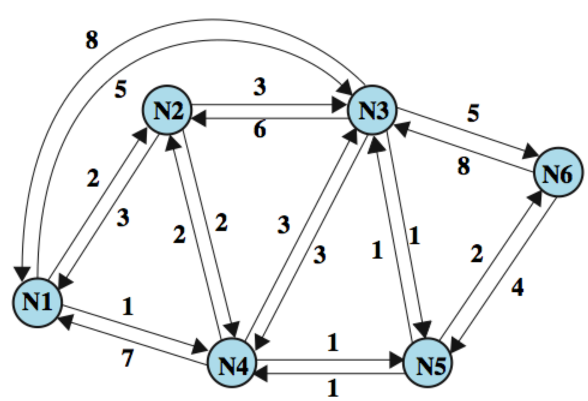
Example of Directed Graph



1-19

19

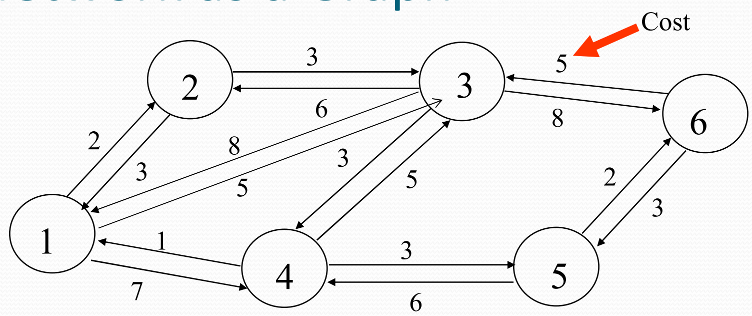
Example of Packet Switched Network



1-20

20

Network as a Graph



The cost could be related:

- the capacity of the link
- the current queuing delay to use the link
- Price
- Security
- ……..

1-21

21

How to assign cost to a link: Examples

- Assign to all links same cost (e.g., 1), enforces the shortest path criterion
- By considering the capacity of the link
- According to bandwidth availability in the link connecting node "i" to node "j". It can be assessed by using
  - queue size
  - ….. what else?
- Nature of the link and associated cost per unit (e.g., "light over fiber" or wireless in congested regulated wireless frequencies, e.g. cellular radio)

1-22

22

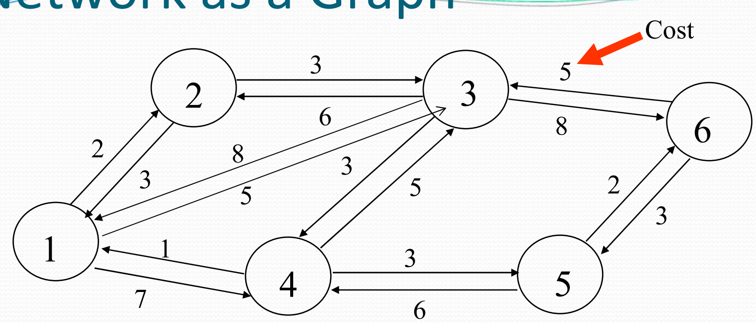
Least Cost Algorithms

- Basis for routing decisions varies. Examples:
  - minimize number of hops by using the same cost value for all links (e.g. cost= 1)
  - make link value inversely proportional to capacity (higher capacity availability gives lower cost)
- Define total cost of a path between two nodes as the sum of costs of links traversed
- For each pair of nodes, find path with least cost
  - link costs in different directions may be different
- Alternatives: Dijkstra or Bellman-Ford algorithms

1-23

23

Network as a Graph



The cost could be related:

- the capacity of the link
- the current queuing delay to use the link
- Price
- Security
- ……..

1-24

24

Dijkstra's Algorithm

- Finds "shortest paths" from given source node ***s*** to all other nodes (meaning the algorithm needs to be run at each network node)
- Does so by developing paths in order of increasing path length
- Algorithm runs in stages
  - in following stage it adds node with next shortest path
- Algorithm terminates when all nodes have been processed by the algorithm

1-25

25

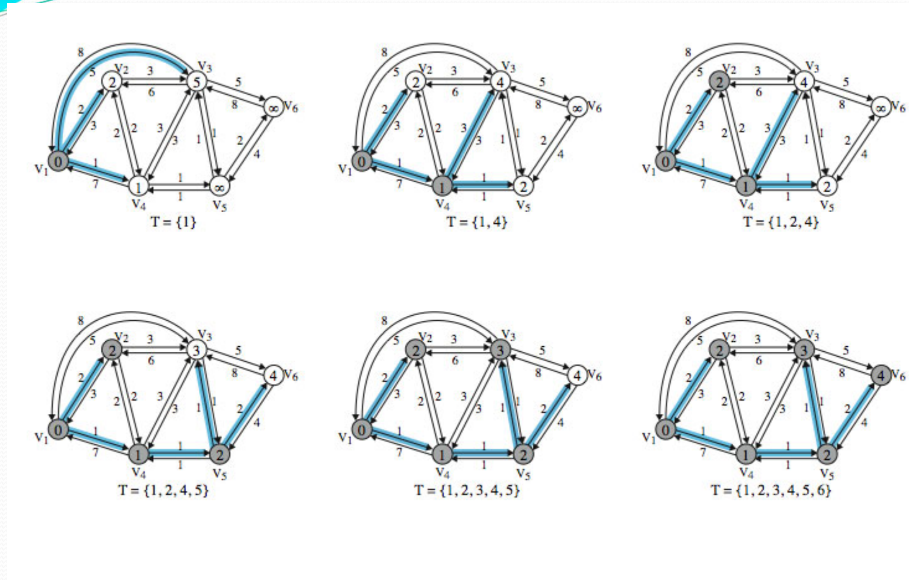
Dijkstra's Algorithm:

- Step 1: Initialization
  - ***T*** = {***s***}: *Set of nodes so far incorporated*
  - L(n) = w(***s***, n) for n $\notin$ ***s***
  - w(i,j): aggregate link cost from node "i" to node "j"
  - initial path costs to neighboring nodes are simply link costs
- Step 2: Get Next Node
  - find neighboring node not in ***T*** with least-cost path from ***S***
  - incorporate node into ***T***
  - also incorporate the edge that is incident on that node and a node in ***T*** that contributes to the path
- Step 3: Update Least-Cost Paths
  - L(n) = min[L(n), L(x) + w(x, n)] for all n $\notin$ T
  - if latter term is minimum, path from s to n is path from s to x concatenated with edge from x to n

1-26

26

Dijkstra's Algorithm Example



1-27

27

Dijkstra's Algorithm Example

Iteration	T	L(2)	Path	L(3)	Path	L(4)	Path	L(5)	Path	L(6)	Path
1	{1}	2	1–2	5	1-3	1	1–4	∞	-	∞	-
2	{1,4}	2	1–2	4	1-4-3	1	1–4	2	1-4–5	∞	-
3	{1, 2, 4}	2	1–2	4	1-4-3	1	1–4	2	1-4–5	∞	-
4	{1, 2, 4, 5}	2	1–2	3	1-4-5–3	1	1–4	2	1-4–5	4	1-4-5–6
5	{1, 2, 3, 4, 5}	2	1–2	3	1-4-5–3	1	1–4	2	1-4–5	4	1-4-5–6
6	{1, 2, 3, 4, 5, 6}	2	1-2	3	1-4-5-3	1	1-4	2	1-4–5	4	1-4-5-6

28

28

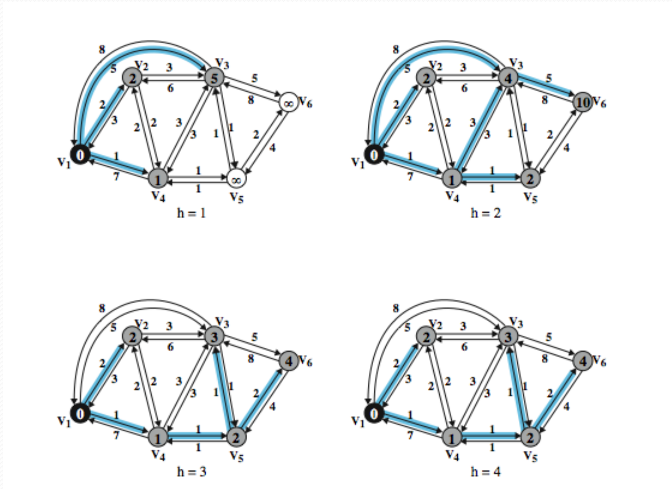
Bellman-Ford Algorithm

- Find shortest paths from a given node subject to constraint that paths contain at most one link
- Find the shortest paths with a constraint of paths of at most two links
- and so on ....

Bellman-Ford Algorithm

- Step 1 [Initialization]
  - $L_0(n) = \infty$, for all $n \neq s$
  - $L_h(s) = 0$, for all h
  - "h" corresponds to length of path
- Step 2 [Update]
  - for each successive $h \geq 0$
    - for each $n \neq s$, compute: $L_{h+1}(n) = \min_j [L_h(j) + w(j,n)]$
  - connect n with predecessor node j that gives min
  - eliminate any connection of n with different predecessor node formed during an earlier iteration
  - path from s to n terminates with link from j to n

Example of Bellman-Ford Algorithm



1-31

31

Results of Bellman-Ford: Example

h	$L_h(2)$	Path	$L_h(3)$	Path	$L_h(4)$	Path	$L_h(5)$	Path	$L_h(6)$	Path
0	∞	-	∞	-	∞	-	∞	-	∞	-
1	2	1-2	5	1-3	1	1-4	∞	-	∞	-
2	2	1-2	4	1-4-3	1	1-4	2	1-4-5	10	1-3-6
3	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6
4	2	1-2	3	1-4-5-3	1	1-4	2	1-4-5	4	1-4-5-6

32

32

Comparison

- Results from the two algorithms are identical
- Bellman-Ford
  - calculation for node n needs link cost to neighbouring nodes plus total cost to each neighbour from s
  - each node can maintain set of costs and paths for every other node
  - can exchange information with direct neighbours
  - can update costs and paths based on information from neighbours and knowledge of link costs
- Dijkstra
  - each node needs complete topology
  - must know link costs of **all** links in network
  - must exchange information with all other nodes

1-33

33

Evaluation

- Dependent on
  - processing time of algorithms
  - amount of information required from other nodes
- Implementation specific
- Both converge under static topology and costs
- Both converge to same solution
- If link costs change, algorithms attempt to catch up
- If link costs depend on traffic, which depends on routes chosen, may have feedback instability

1-34

34

Characteristics of Routing Strategies

- Static routing:
  - requires network information to define routes
  - no change unless network topology changes force action of network administration
  - independent of traffic
  - simple but may cause congestion
- Flooding:
  - no network information needed
  - send to all neighbors (exclude the one packet came from)
  - requires unique identification of every passing through packet
  - requires recognition of, and maintains in memory (for a certain period) every packet already transmitted

Characteristics of Routing Strategies

- Random routing:
  - network information may be needed
  - choose one outgoing path of the switching node
  - unpredictable delay
- Adaptive routing:
  - uses network information to select a route at any particular time
  - three kinds
    - **isolated adaptive**: uses local information
    - **distributed adaptive**: uses information from adjacent nodes
    - **centralized adaptive**: uses information from all network nodes

Network Information Source and Update Timing

- Routing decisions are usually based on knowledge of network (not always)
  - distributed routing
    - using local knowledge, info from adjacent nodes, info from all nodes on a potential route
  - central routing
    - collect info from all nodes
- Issue of update timing
  - when is network info held by nodes updated
  - fixed - never updated
  - adaptive - regular updates

1-37

37

Approaches to Routing – **Distance-vector**

- Each node (router or host) exchange information with neighboring nodes
- First generation routing algorithm for ARPANET
  - used by **Routing Information Protocol (RIP)**
- Based on Bellman-Ford Algorithm
- The term *distance vector* refers to the fact that the protocol manipulates *vectors* (arrays) of distances to other nodes in the network.
- Routers that use distance-vector protocol determine the distance between themselves and a destination.
- **Each node** maintains vector of link costs for **each directly attached network (node)** and distance and next-hop vectors for **each destination**
- Requires **transmission of too much info** by routers
  - distance vector & estimated path costs
- Changes take **long time to propagate**

1-38

38

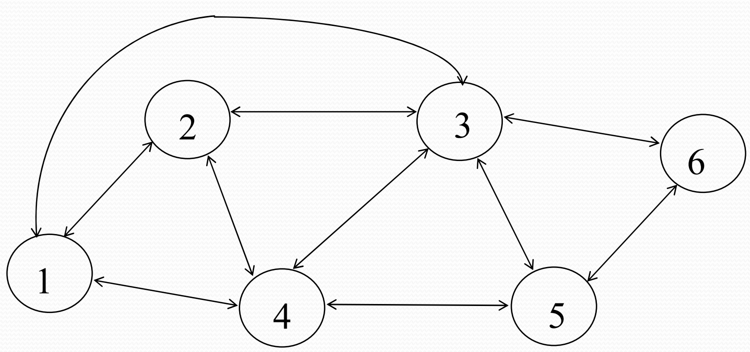
How it works

- Each node (source) needs to know the least cost route leading to each other node (destination) within its network' s domain, and the next hop node of this path.
- This requires that information is exchanged among ALL nodes. Each node sends:
  - Node "***k***" sends Distance Vector D_k containing estimated cost of paths to each of all other nodes.
- An algorithm is needed to:
  - determine the costs for each source/destination pair
  - make the needed modifications to the cost values when changes occur (e.g. link failure, node failure, congestion between two nodes etc.)
- The Bellman-Ford algorithm is used.

1-39

39

Example of a Network

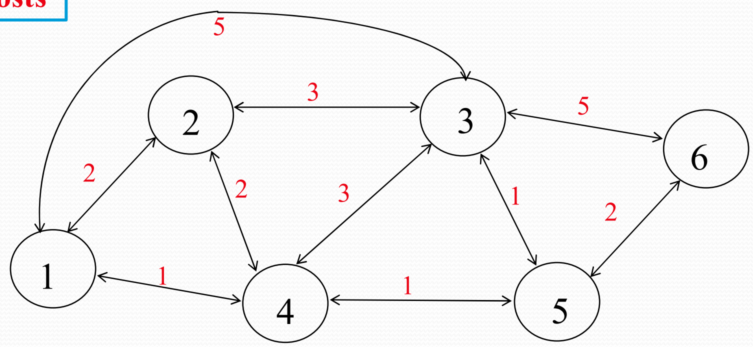


1-40

40

Example of a Network

Link Costs



NOTE: For simplicity we use equal “cost” values for both directions

1-41

41

Approaches to Routing– **Link-state**

- Designed to overcome drawbacks of distance-vector
- Each router determines **link cost on each interface**
- Advertises set of link costs **to all other routers** in topology
- If link costs change, router advertises new values
- Each router constructs ***topology of entire configuration***
  - can calculate shortest path to each destination
  - used to construct routing table with first hop to each destination
- **Does not use** distributed routing algorithm, but any suitable Algorithm to determine shortest paths, e.g. Dijkstra’ s algorithm
- Open Shortest Path First **(OSPF) is a link-state protocol**

1-42

42

Open Shortest Path First (RFC2328)

- It is the IRP of Internet
- replaced Routing Information Protocol (RIP)
- uses Link State Routing Algorithm
  - each router keeps list of state of local links to network
  - transmits update state info
  - generates light traffic as messages are small and not sent often
- uses least cost based on user cost metric
- topology stored as directed graph
  - vertices or nodes (router, transit or stub network)
  - edges (between routers or router to network)

1-43

43

ARPANET Routing Strategies: 1st Generation

- 1969
- Distributed adaptive using estimated delay
  - queue length used as estimate of delay
- Uses Bellman-Ford algorithm
- Node exchanges delay vector with neighbors
- Updates routing table based on incoming info
- Problems:
  - doesn't consider line speed, just queue length
  - queue length **not a good measurement** of delay (Why?)
  - responds slowly to congestion

1-44

44

ARPANET Routing Strategies 2nd Generation

- 1979
- Distributed adaptive using measured delay
  - using timestamps of arrival, departure & ACK times
- Re-computes average delays every 10 secs
- Any changes are flooded to all other nodes
- Re-computes routing using Dijkstra' s algorithm
- Good under light and medium loads
- Under heavy loads, little correlation between reported delays and those experienced (Why?)

1-45

45

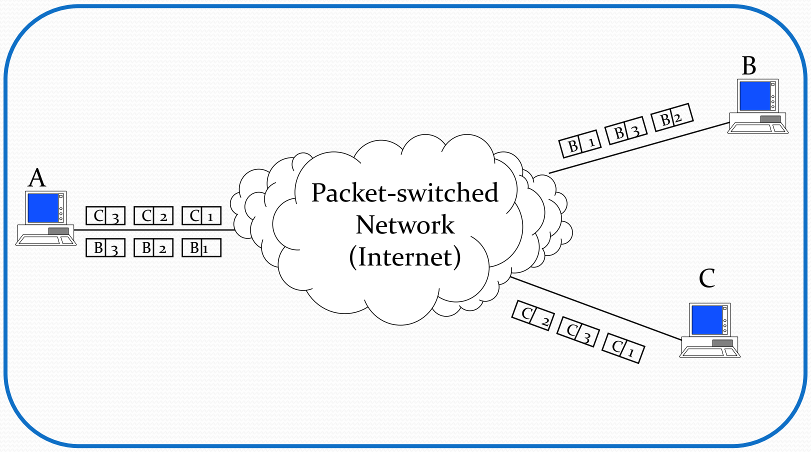
ARPANET Routing Strategies 3rd Generation

- 1987
- link cost calculations changed to:
  - damp routing oscillations
  - reduce routing overhead
- Measure average delay over last 10 secs and transform into link utilization estimate
- normalize this based on current value and previous results
- set link cost as function of average utilization

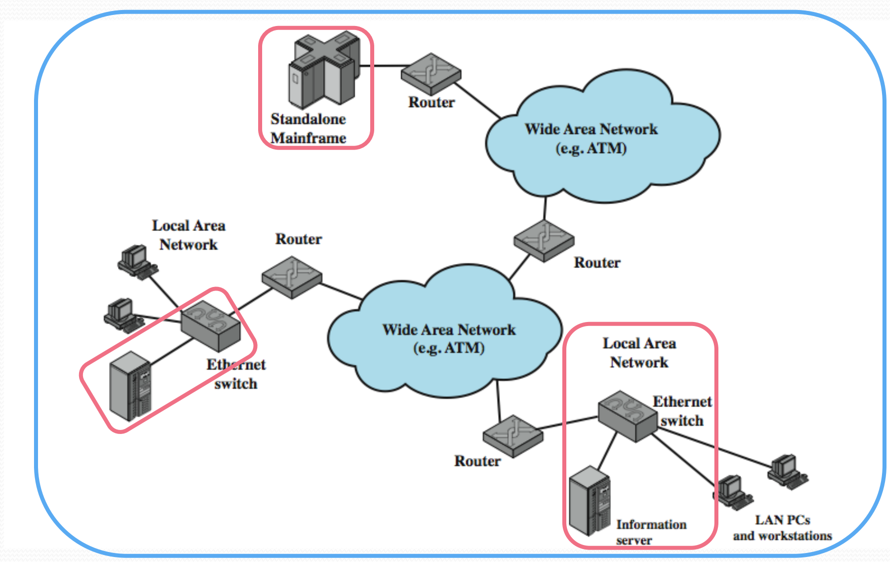
1-46

46

User' s view of Internet



"Insider's" view of Internet



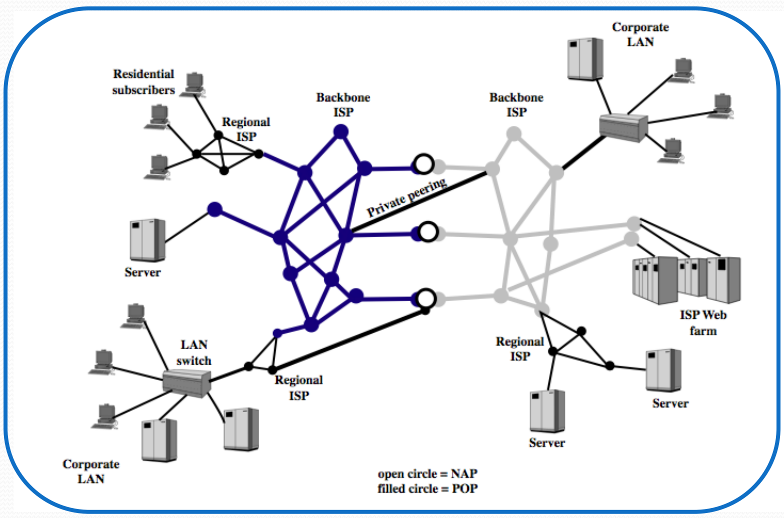
Autonomous System (AS)

- AS is a set of routers and networks managed by a single organization.
- AS consists of a group of routers exchanging information via a common routing protocol.
- Claiming that an AS is "connected", means that (excluding times of failures) there is always a "path" between any pair of nodes.

1-49

49

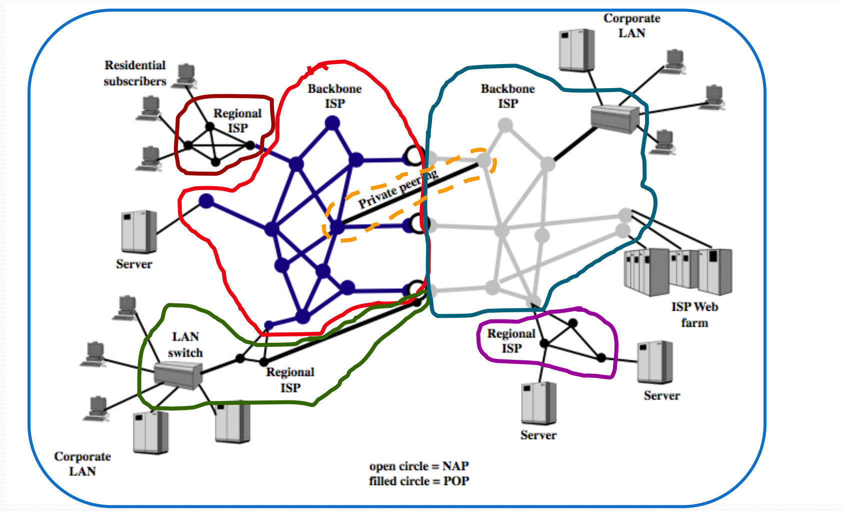
A "possible" Internet Architecture(1)



1-50

50

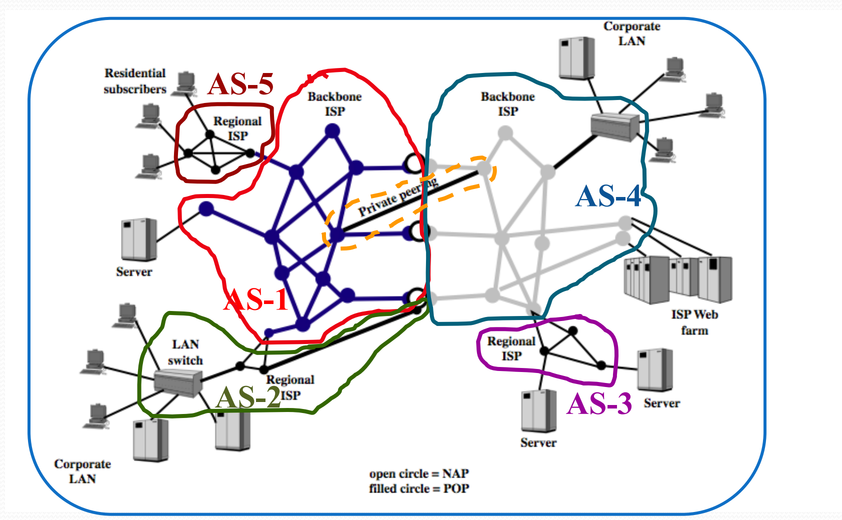
A "possible" Internet Architecture(2)



1-51

51

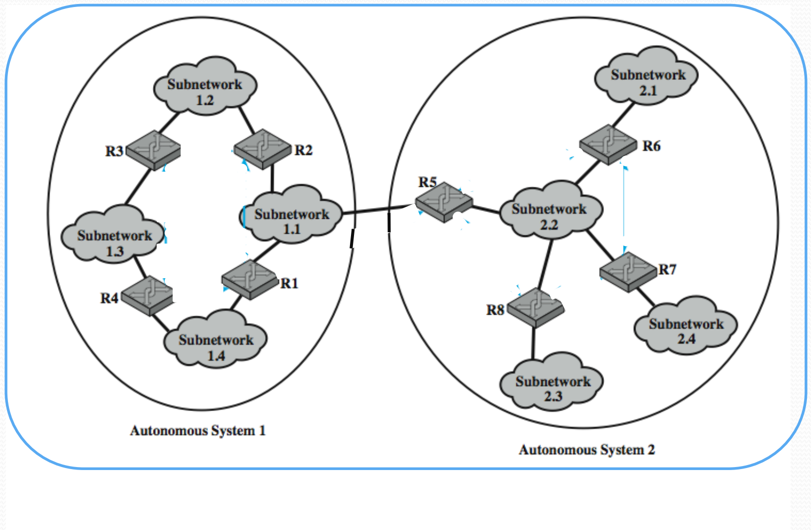
A "possible" Internet Architecture(3)



1-52

52

Passing data from source to destination: How?



1-53

53

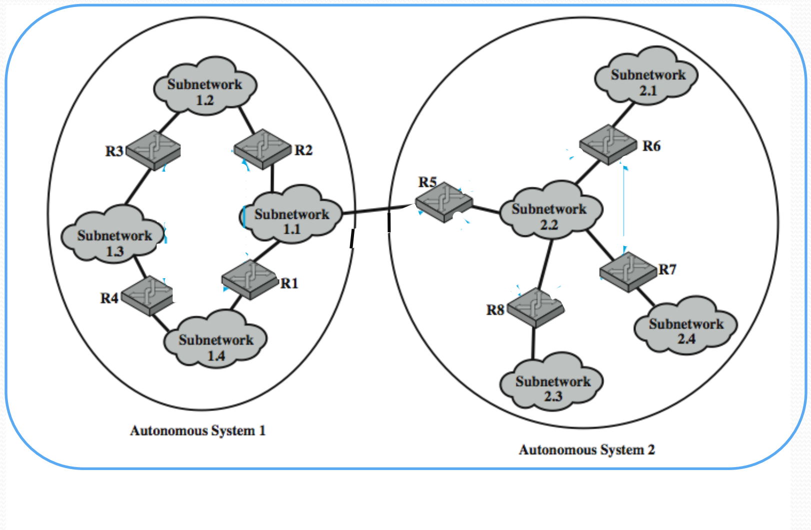
Interior Routing Protocol & Exterior Routing Protocol

- **Interior routing protocol (IRP)**
  - passes routing information between routers belonging to the same AS
  - can be tailored to specific applications (e.g. support QoS mechanisms to service high quality videoconferencing, DIVE based distributed video--gaming – highly delay sensitive – etc.)
  - needs detailed knowledge of network' s topology to function
- We have more than one AS in internet
  - routing algorithms & formatting of tables may differ between them
  - Should all routers within internet were treated as a single AS domain, **the size of routing tables and routing control related traffic would be overwhelmingly large (scalability problem)**
  - Different approach needed for routing between/across different AS
- Routers need **info on networks outside their own AS**
- Use an **Exterior router protocol (ERP)** for this
  - Support provision of summary information on AS reachability

1-54

54

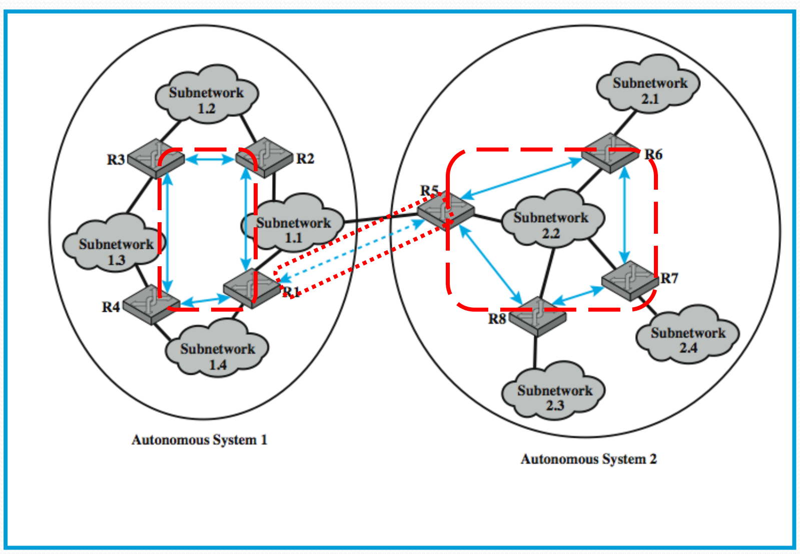
Example: A 2-AS formed Internet



1-55

55

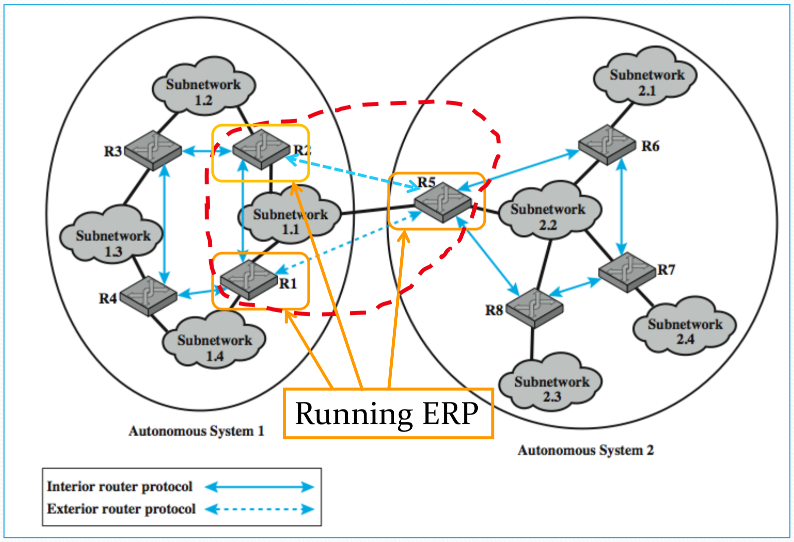
Application of IRP and ERP



1-56

56

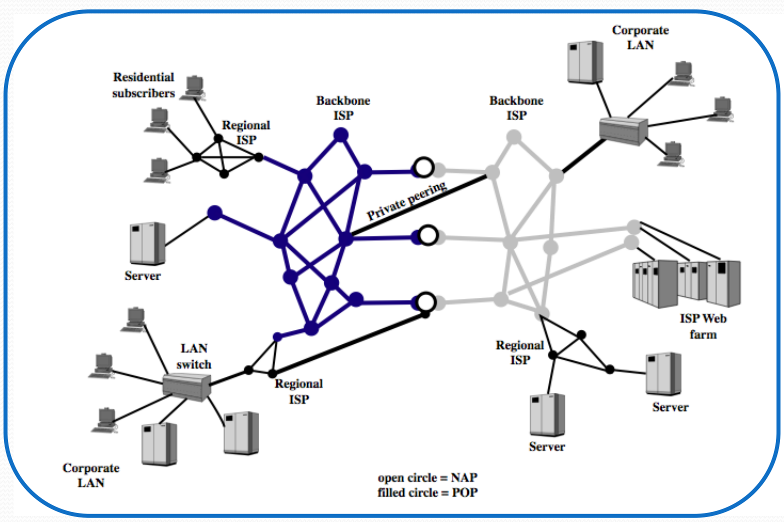
Application of IRP and ERP



1-57

57

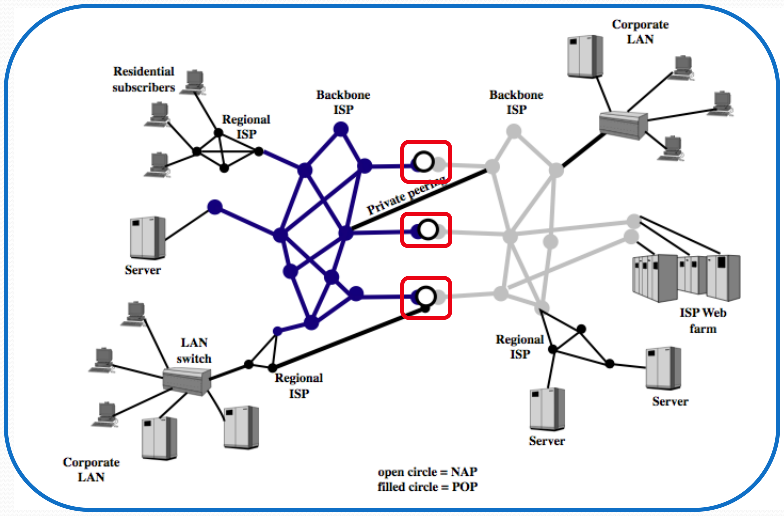
Example: Identify routers running ERP



1-58

58

Example: Identify routers running ERP



1-59

59

What Exterior Routing Protocols are **not**

- Link-state and distance-vector are not effective for exterior router protocol
- Distance-vector
  - assumes routers **share common distance metric**
  - but different ASs may **have different priorities & needs**
  - but have no info on AS's visited along route
- Link-state
  - different ASs may **use different metrics** and **have different restrictions**
  - flooding of link state information to **all routers** is **unmanageable**

1-60

60

Exterior Router Protocols – Path-vector

- Alternative **path-vector** routing protocol
  - provides info about **which networks can be reached** by a given router and ASs crossed to get there
  - does **not include** distance or cost estimate
  - hence dispenses with concept of routing metrics
- Have **list of all ASs visited on a route**
- Enables router to **perform policy routing**
  - e.g. avoid path to avoid transiting particular AS
  - e.g. link speed, capacity, tendency to become congested, and overall quality of operation, security
  - e.g. minimizing number of transit ASs

1-61

61

Border Gateway Protocol (BGP)

- Developed for use with TCP/IP internets
- Is the preferred EGP of the Internet
- Uses messages sent **over TCP connection**
- BGP-4 (RFC1771)
- Functional procedures:
  - neighbor acquisition - when agree to exchange info
  - neighbor reachability - to maintain relationship
    - the two routers periodically issue ***"keepalive"*** messages
  - network reachability - to update database of routes
    - router maintains database of networks it can reach and preferred route for reaching each network.
    - when a change is made to database, router broadcasts ***"update"*** message to all other BGP routers.

1-62

62

BGP Messages

Marker:

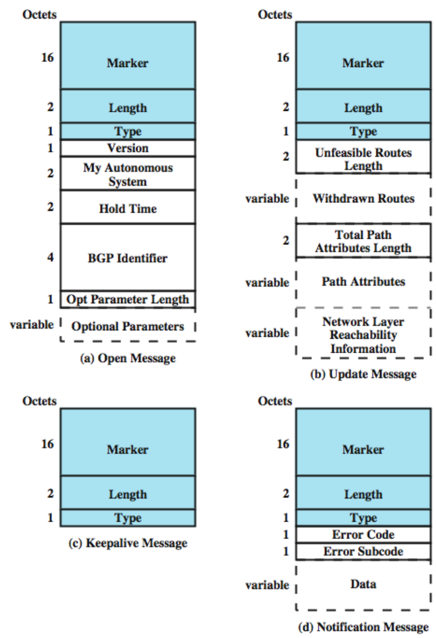
- Reserved for authentication.
- sender may insert a value in this field that would be used as part of an authentication mechanism to enable the recipient to verify the identity of the sender.

Length:

- Length of message in octets.

Type:

- Type of message: *Open, Update, Notification, Keepalive.*



1-63

63

BGP Messages

Open:

- Open a neighbor relationship with another router.

Update:

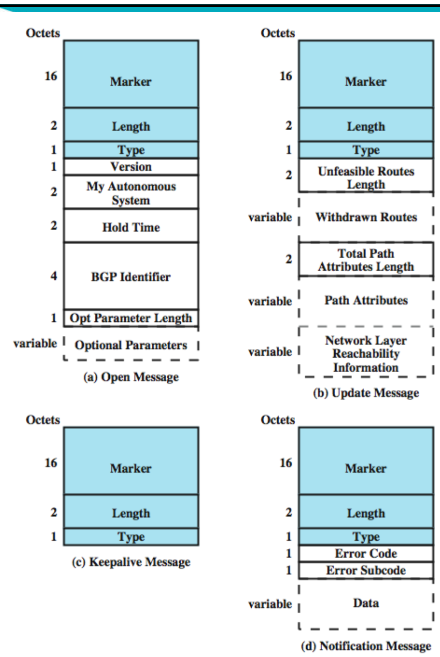
- Transmit information about a single route.
- List multiple routes to be withdrawn.

Keep alive:

- Acknowledge an ***Open*** message;
- Periodically confirm the neighbor relationship.

Notification:

- Sent when an error condition is detected.



1-64

64

Message Types - **Update**

- Update message conveys two info types:
  - Info about single routes through internet
  - List of routes being withdrawn
- Info on a route uses 3 fields:
  - Network Layer Reachability Information (NLRI)
  - Total Path Attributes Length
  - Path Attributes
- Withdraw route identified by dest IP address

AS_Path and Next_Hop Use

- AS_Path
  - used to implement routing policies
    - e.g. to avoid a particular AS, security, performance, quality, number of AS crossed
- Next_Hop
  - only a few routers implement BGP
  - responsible for informing outside routers of routes to other networks in AS

Notification Message

- Sent when some error condition detected:
- Message header error
- Open message error
- Update message error
- Hold time expired
- Finite state machine error
- Cease

BGP Routing Information Exchange

- Within AS a router builds topology picture using IGP
- Router issues Update message to other routers outside AS using BGP
- These routers exchange info with other routers in other AS
  - **AS_Path** field used to prevent loops
- Routers must then decide best routes

Routing methodology: Summary

- **Distance Vector Routing:** each node exchanges information with its neighboring nodes, e.g. Routing Information Protocol (RIP).
- **Link-state Routing:** sends link costs of each of its network interfaces to all routers (not just neighboring). Typically used with a Dijkstera-based algorithm, e.g. Open Shortest Path First (OSPF).
- **Path-vector Routing**: router provides information about which networks can be reached by a given router and the ASs that must be crossed; e.g. Border Gateway Protocol (BGP).

1-69

69

END

1-70

70