Routing Algorithms
Unit 8
(LH 4)

Introduction

- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in *every* host, router
- router examines header fields in all IP datagrams passing through it

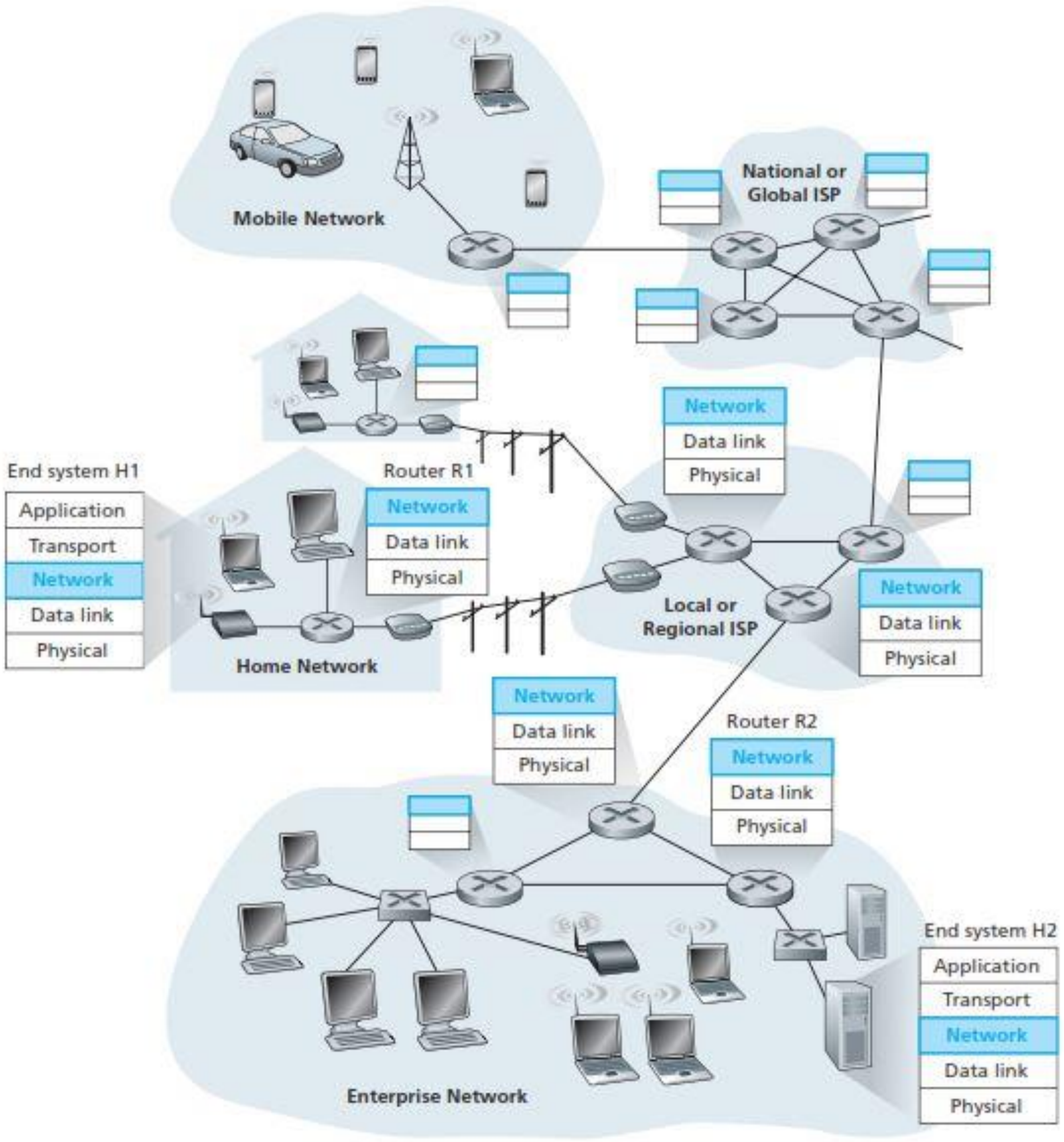


Figure: The Network Layer

Forwarding and Routing

- The role of network layer is simply to move packets form a sending host to a receiving host.
- To do so,, two important network-layer functions can be identified:

1. Forwarding:
   - moves packets from router's input to appropriated router output
   - For example, a packet arriving from Host H1 to Router R1 must be forwarded to the next router on a path to H2.
   - Each router has a **forwarding table**. A router forwards a packet by examining the value of a field in the arriving packet's header, and then using this value to index into the router's forwarding table.
2. Routing:
   - determine route taken by packet from source to destination.
   - the algorithms that calculate these paths are referred to as **routing algorithms**.

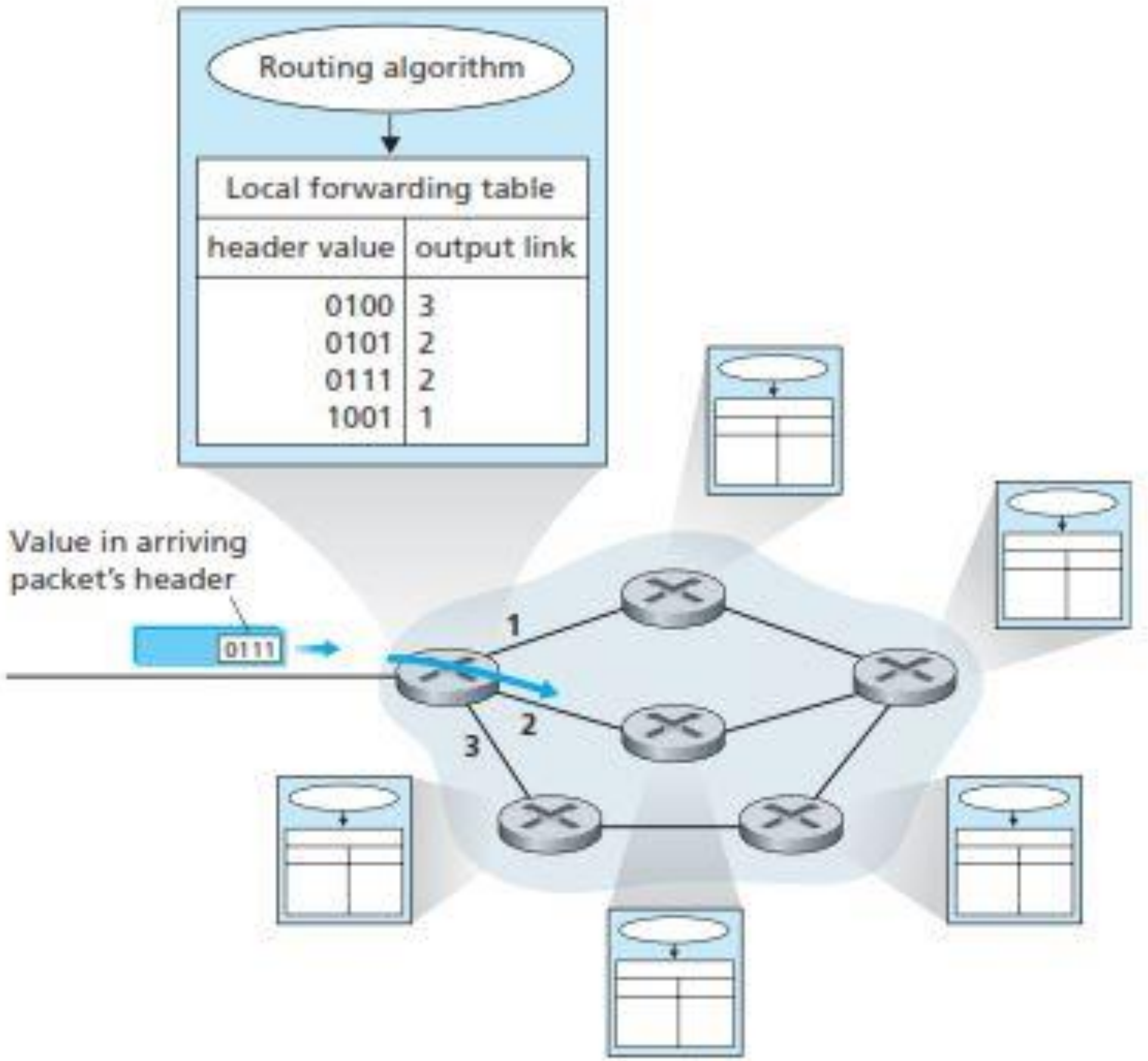


Figure: Routing algorithms determine values in forwarding tables

Network Service Models

- The network service model defines the characteristics of end-to-end transport between sending and receiving end systems.
- In sending host, when the transport layer passes a packet to the network layer, specifies services that could be provided by the network layer include:
- *Guaranteed delivery*: this service guarantees that the packet will eventually arrive at its destination
- *Guaranteed delivery with bounded delay*: this service not only guarantees delivery of the packet, but delivery within a specified host-to-host delay bound (for example, within 100 msec).
- Furthermore, the following service could be provided to a flow of packets between a given source and destination:
  - In order packet delivery
  - Guaranteed minimal bandwidth
  - Guaranteed maximum jitter
  - Security service.
- The Internet’s network layer provides a single service, known as **best-effort-service**.

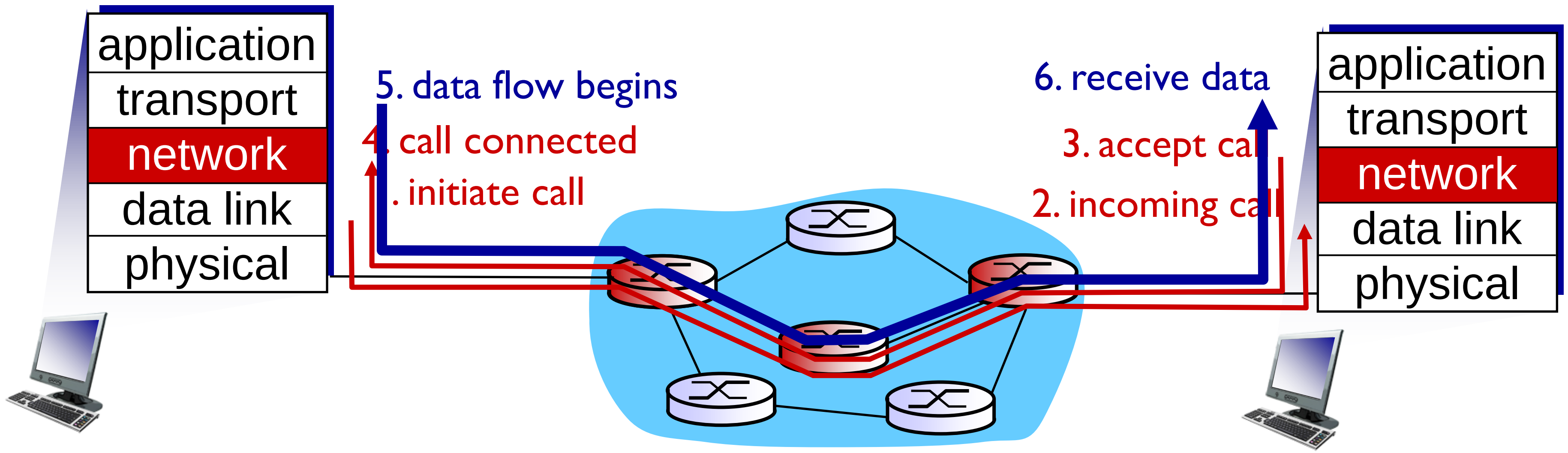
Virtual Circuit and Datagram Networks

- The internet transport layer provides each application a choice between two services: UDP (a connectionless service) or TCP (a connection oriented service).
- In this similar manner, a network layer can also provide connectionless service(datagram networks) or connection service (virtual circuit network).
- Although these transport layer and network layer service models seem parallel, there are some crucial differences:
  I. In transport layer, it is process to process services. But, in network layer, it is host to host service.
  II. In all major computer network architectures to date (Internet, ATM, frame relay and so on), the network layer provides either a host-to-host connectionless service or a host-to-host connection service, but not both.
  III. Connection oriented services in transport layer in implemented at the edge of the network in the end systems; however, the network layer connection service is implemented in the network core as well as the end systems.

Virtual Circuit (VC) Network

- Many network architecture (**not internet**) including those of ATM and frame relay. X.25 are VC network and therefore, use connections at the network layer. These network layer connections are called virtual circuits (VCs).
- Let's now consider how a VC service can be implemented in a computer network.
- A VC consist of
  1. A path (i.e. a series of links and routers) between the source and destination hosts.
  2. VC numbers, one number for each link along the path, and
  3. Entries in the forwarding table in each router in the path.

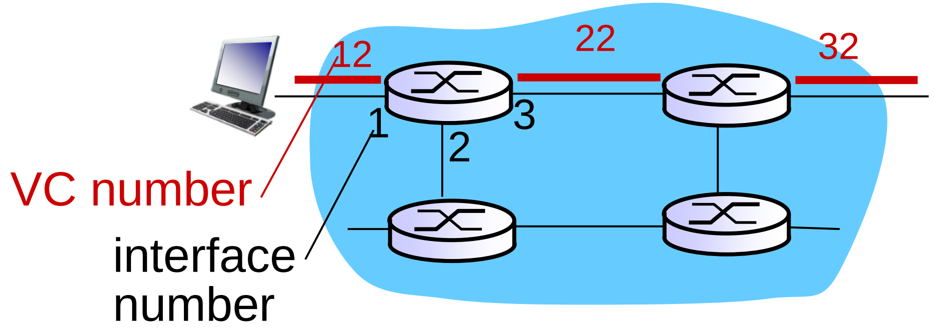
Virtual Circuit



Virtual Circuit (VC) Network

- A packet belonging to a virtual circuit will carry a VC number in its header. Because a virtual circuit may have a different VC number on each link, each intervening router must replace the VC number of each traversing packet with a new VC number. The new VC number is obtained from the forwarding table.
- In a VC network, the network's routers must maintain connection state information for ongoing connections.
  - Each time a new connection is established across a router, a new connection entry must be added to the router's forwarding table; and
  - Each time a connection is released entry must be removed from the table.
- There are three distinct phases in a virtual circuit
  1. VC setup
  2. Data Transfer
  3. VC teardown

VC Forwarding Table



forwarding table in northwest router:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

VC routers maintain connection state information!

Virtual Circuit (VC) Network

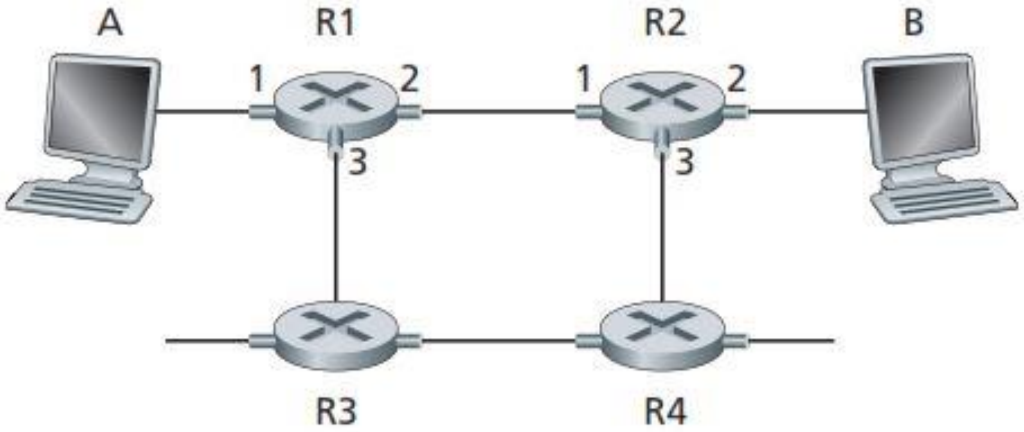


Figure: A virtual circuit network

Incoming Interface	Incoming VC #	Outgoing Interface	Outgoing VC #
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

Figure: Entries in forwarding table

Datagram Networks

- In a datagram network, each time an end system wants to send a packet, it stamps the packet with address of the destination end system and then pops the packet into the network. There is no VC setup and routers do not maintain any VC state information (because there are no VCs!)

- As a packet is transmitted from source to destination, it passes through a series of routers. Each of these routers uses the packets' destination address to forward the packet. Specifically, each router has a routing table that maps destination addresses to link interfaces; when a packet arrives at the router, the router uses the packet’s destination address to look up the appropriate output link interface in the routing table. The router then intentionally forwards the packets to that output link interface.

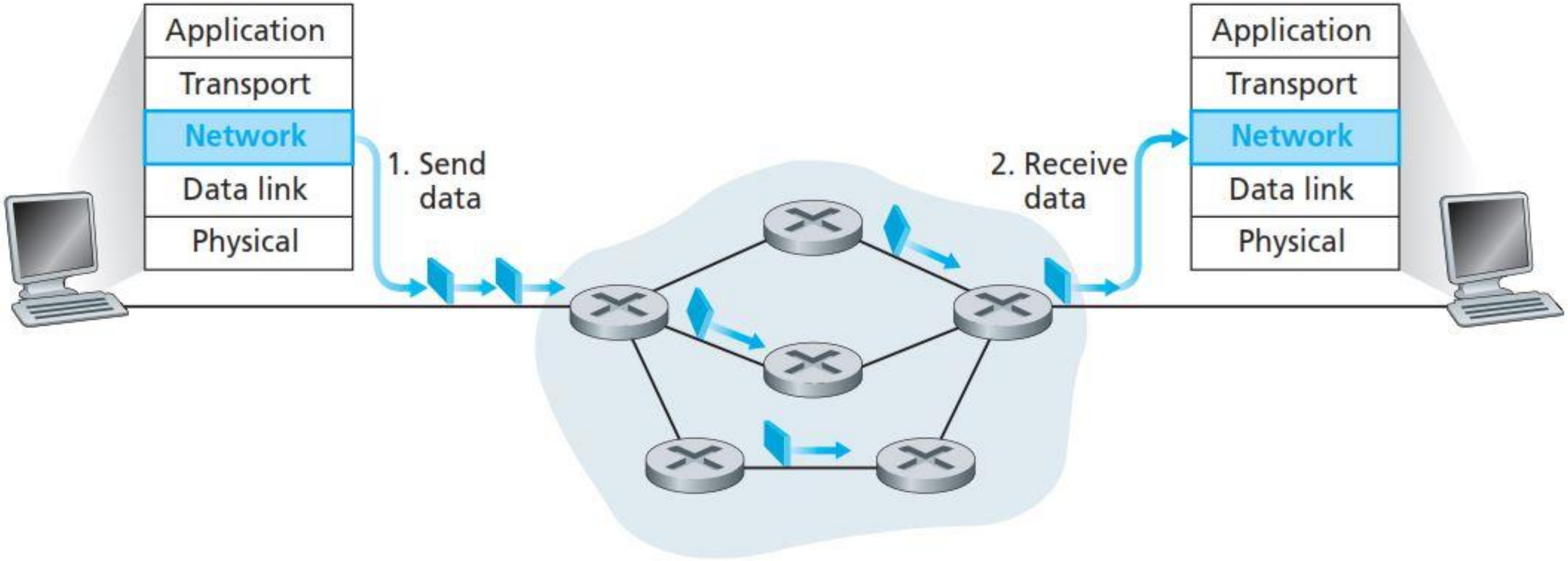
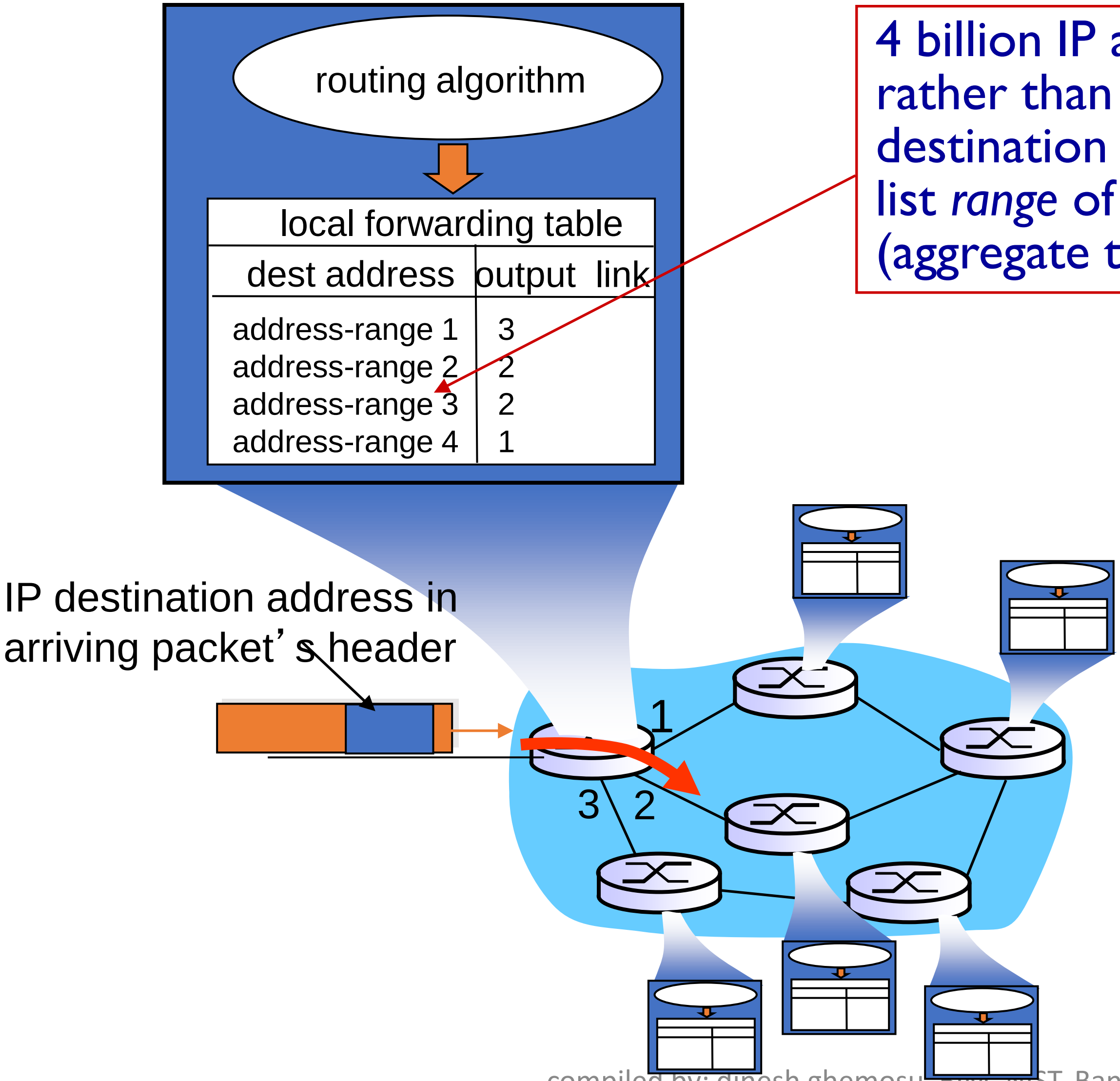


Figure: Datagram Network

Datagram Forwarding Table



4 billion IP addresses, so rather than list individual destination address list *range* of addresses (aggregate table entries)

Datagram Forwarding Table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Differences:

Datagram Packet Switching	Virtual-circuit Packet Switching
1. Two packets of same user pair can travel along different routes.	1. All packets of the same virtual circuit travel along the same path.
2. The packets can arrive out of sequence.	2. Packet sequencing is guaranteed.
3. Packets contain full source, destination address.	3. Packets contain short VC ID (VCI).
4. Each host occupies routine table entries.	4. Each VC occupies routing table entries.
5. Requires no connection setup.	5. Requires VC setup. First packet has larger delay.
6. Also called Connection less.	6. Also called connection oriented.
7. examples: X.25, Frame Relay, ATM	7. Examples: Internet

Routing

- Once you create an inter network by connecting your WANs and LANs to a router, you'll need to configure local network addresses, such as IP address, to all hosts on the internet work so that they can communicate across that internetwork.
- The term routing *refers to taking a packet from one device and sending it through the network to another device on a different network*. Routers don't really care about host. They only care about the networks and the best path to each network. The logical network address of the destination host is used to get packets to a network through a routed network, and then hardware address of the host is used to deliver the packet from a router to the correct destination host.

- ***If your network has no routers, then it is clear that you are not routing***. Routers route traffic to the entire network in your internetwork. To be able to route packets, a router must know, at minimum, the following:
  - Destination address
  - Neighbor routers from which it can learn about remote network.
  - Possible routers to all remote networks.
  - The best route to each remote network.
  - How to maintain and verify routing information.

Routing

- **Routing** refers to the process of selecting the shortest and the most reliable path intelligently over which the data is send to its ultimate destination.
- Routing protocols are used to continuously update the routing table that are consulted for forwarding and routing.
- Routing algorithm is that part of network layer software responsible for deciding which output line an incoming packet should be transmitted on.
- Two types of routing algorithm
  - Non adaptive/static algorithm
  - Adaptive/Dynamic algorithm

Non-Adaptive/Static Routing

- In this type of routing, the route to be taken in going from one node to the other is computed in advance, off-line, and downloaded to the routers when the network is booted.
- These algorithms do not base their routing decisions on measurements and estimates of the current traffic and topology.

- Static routing means someone must hand-type all network locations into the routing table. If static routing is used, the administrator is responsible for updating all changes by hand onto all routers
- Classified into:
  I. Flooding
  II. Random walk

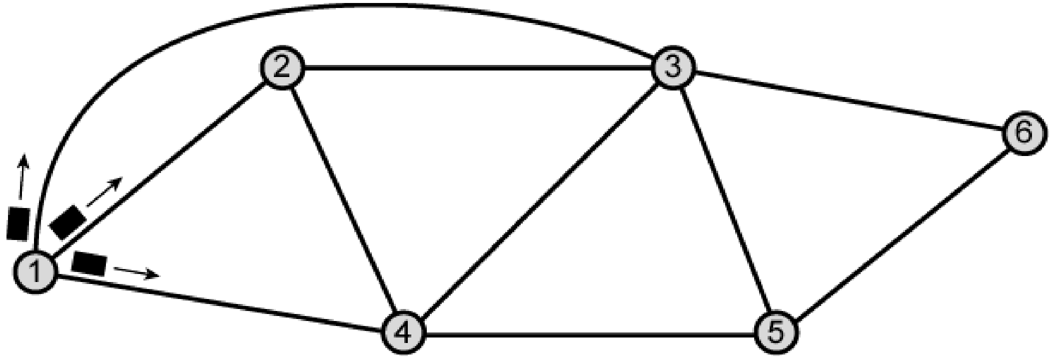
Adaptive Routing/Dynamic Routing

- These algorithms change their routing decisions to reflect changes in the topology and in traffic as well.
- These get their routing information from adjacent routers or from all routers.
- If a change occurs in the network, the dynamic routing protocols automatically inform all routers about the events e.g. RIP V1, RIP V2, OSPF, EIGRP.
- Classified as follow:
  I. Centralized
  II. Isolated
  III. distributed

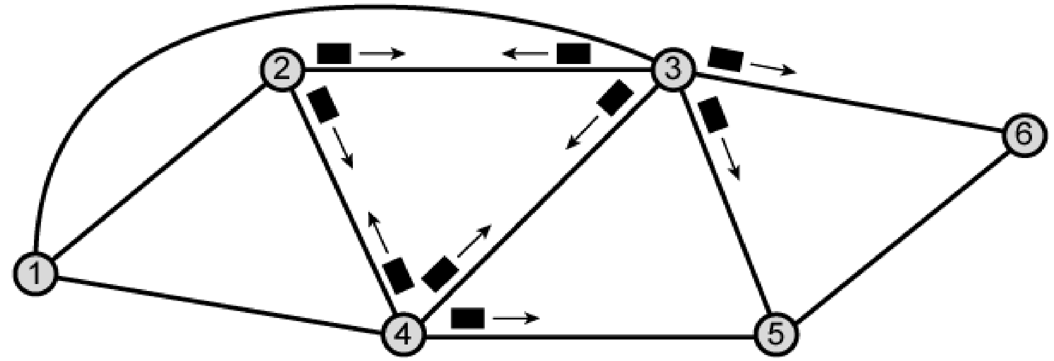
Flooding

- It is static routing technique.
- In this technique, a packet is sent by a source node to every one of its neighbors.
- At each node, an incoming packet is retransmitted on all outgoing links except for the link on which it arrived.
- For example: if above figure, node 1 has to send packet to node 6, then
  - Node 1 send a packet with destination address 6 to nodes 2,3, and 4.
  - Node 2 will send a copy to nodes 3 and 4.
  - Node 4 will send to a copy to nodes 2,3 an 5
  - And it goes on.
- Eventually, a number of copies of packet will arrive at node 6. Node 6 accepts the first copy arrived ,and discards others if arrived.

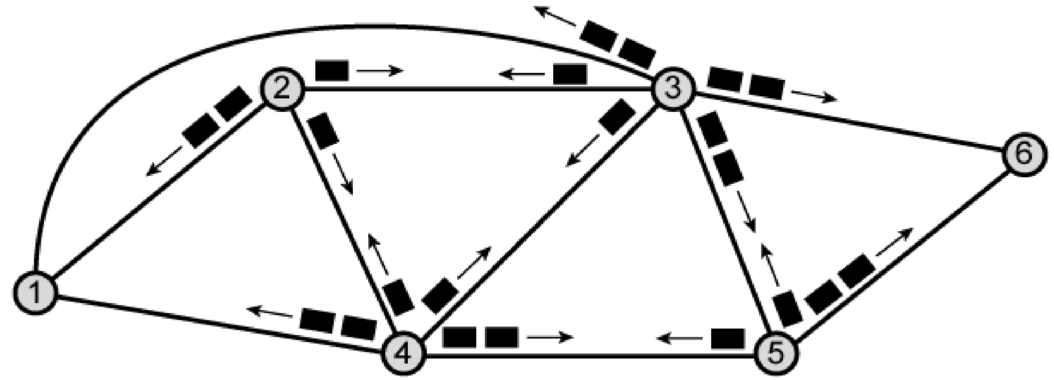
Figure: Flooding Example (hop count=3)



(a) First hop



(b) Second hop



(c) Third hop

Properties of Flooding

- All possible routes form source to destination is tried. → robust technique and used to send emergency message.
- At least one copy of packet arrive at destination with minimum-hop count route. → might use this technique to setup initial virtual circuit.
- All nodes are visited. → can be useful to send important message to all nodes.

Disadvantages

- Generates high traffic load

- One major problem of this algorithm is that it generates a large number of duplicate packets on the network.
- Several measures are takes to stop the duplication of packets. These are:

1. One solution is to include a **hop counter** in the header of each packet. This counter is decremented at each hop along the path. When this counter reaches zero the packet is discarded. Ideally, the hop counter should become zero at the destination hop, indicating that there are no more intermediate hops and destination is reached. This requires the knowledge of exact number of hops from a source to destination.
2. Another technique is to **keep the track of the packed that have been flooded**, to avoid sending them a second time. For this, the source router put a **sequence number** in each packet it receives from its hosts. Each router then needs a list per source router telling which sequence numbers originating at that source have already been seen. If an incoming packet is on the list, it is not flooded.
3. Another solution is to use **selective flooding.** In selective flooding the routers do not send every incoming packet out on every output line. Instead packet is sent only on those lines which are approximately going in the right direction.

Routing Algorithm

Three classes of routing protocols:

1. Distance vector routing
2. Link state routing
3. Hybrid routing

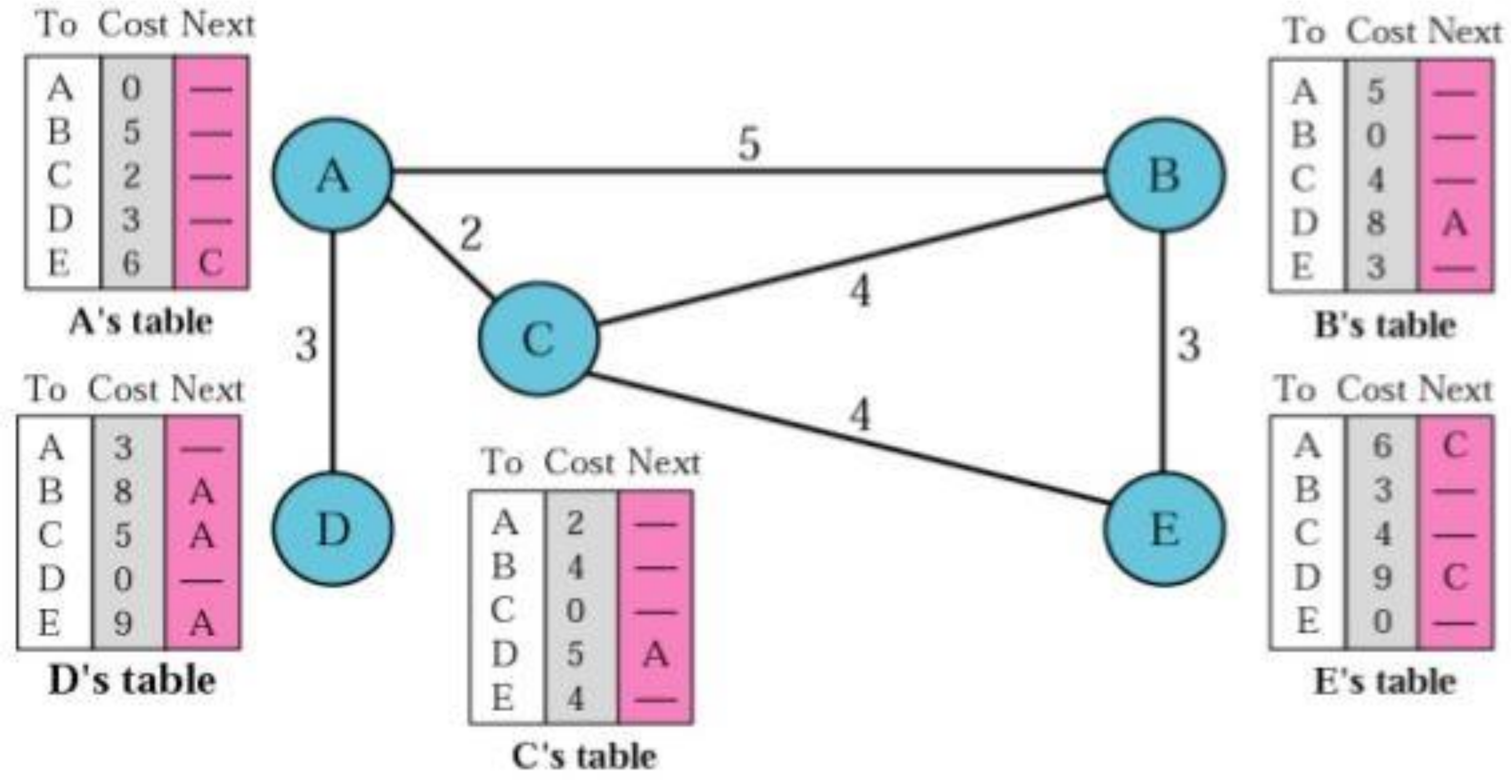
Distance Vector

- The distance-vector protocols are in use today. Find the best path to a remote network by judging distance.
- For e.g., in the case of RIP routing, each time a packet goes through a router, that‘s called a hop. The route with the least number of hops to the network is determined to be the best route.

- In distance vector routing, each node maintain a vector (table) of minimum distance to every node. The table at each node also guide the packet to the desired node by showing the next stop (next hop) in the route.
- Each node shares its routing table with its immediate neighbors periodically and when there is a change.
- E.g. RIP, IGRP

Distance Vector Routing Table

The tables for node A shows how can we reach any node from this node. For example, our least cost to reach node E is 6, through 6.



Disadvantages of Distance Vector Routing

Limited to a hop count of 15

- after a packet travels through 15 routers and still has another router to travel to, it will be discarded.

Bandwidth Consumption

- Periodic updates of routing table

Slow convergence time

- Convergence time is the time needed for all routers within a single routing domain to receive, process and build their routing table.
- So, it is not suitable for large networks that require very small propagation delay.

Routing loops/Count to Infinity

- A routing loop is defined as a condition in which packet is continuously retransmitted among several routers without reaching its destination. This condition give rise to a problem called count to infinity.
- Count to infinity is the condition that occurs when inaccurate routing updates increases the hop number to “infinity” for a network that no longer exist.

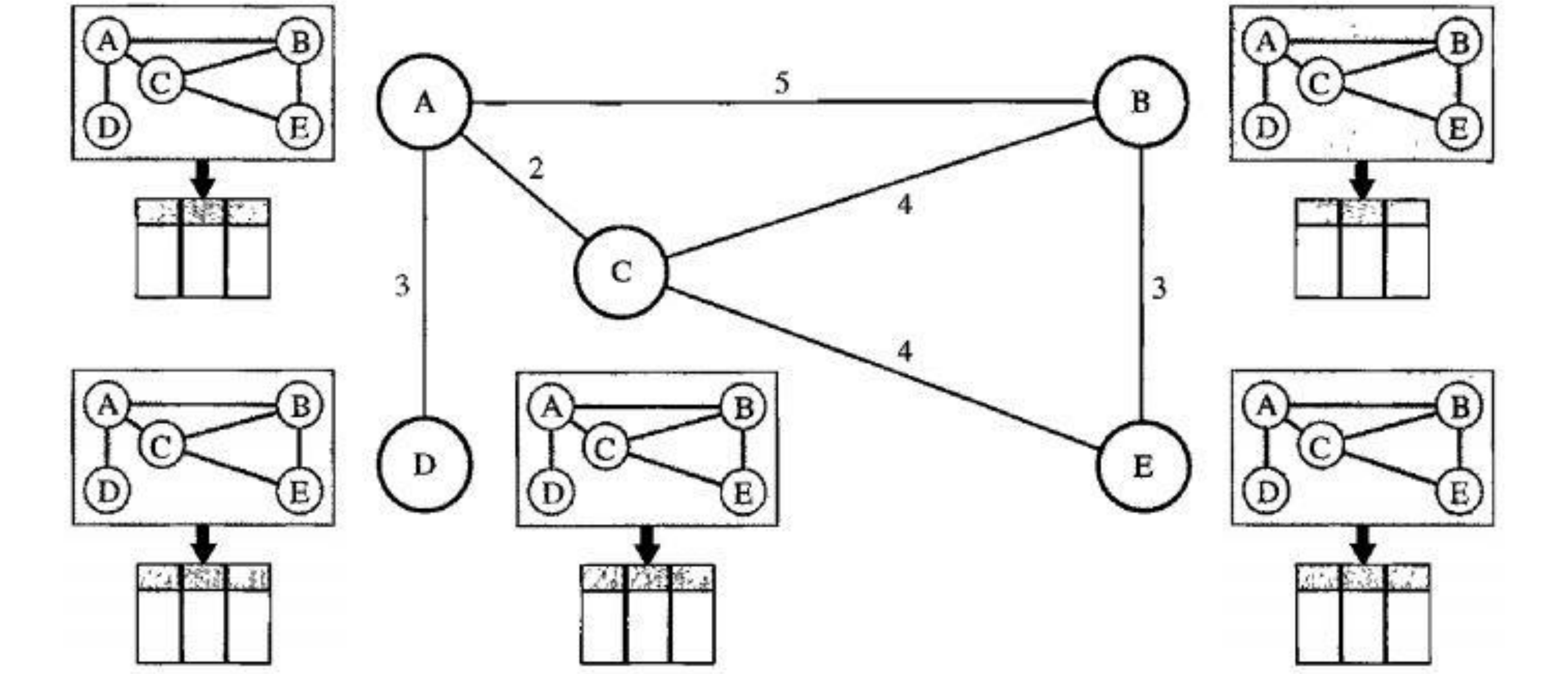
Convergence is part of the **routing** table update process. When a link fails or changes, updates **are** sent across the network that describe changes in the network topology. Each **router** then runs a **routing** algorithm to re-compute **routes** and build new **routing** tables based on this information

Link State Routing

- Dynamic routing protocol developed to overcome the drawbacks of distance vector routing.
- Examples of link-state routing protocols include **open shortest path first** (OSPF) and **intermediate system to intermediate system** (IS-IS).

- In link-state routing is that every node constructs a *map* of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes.
- i.e. in the link state routing, each node in the domain has the entire topology of the domain-the list of nodes and links, how they are connected including the type, cost (metric) , and conditions of the links(up or down).
- Each node then independently calculates the next best logical *path* from it to every possible destination in the network. The collection of best paths will then form the node's routing table.
- Requires more memory and large computations.

Link State Routing



The figure shows a simple domain with five nodes. Each node uses the same topology to create a routing table, but the routing table for each node is unique because the calculations are based on different interpretations of the topology. This is analogous to a city map. While each person may have the same map, each needs to take a different route to reach her specific destination.

Differences Between

Distance vector	Link state
sends the entire routing table	sends only link state information
slow convergence	fast convergence
susceptible to routing loops	less susceptible to routing loops
updates are sometimes sent using broadcast	always uses multicast for the routing updates
doesn't know the network topology	knows the entire network topology
simpler to configure	can be harder to configure
examples: RIP, IGRP	examples: OSPF, IS-IS

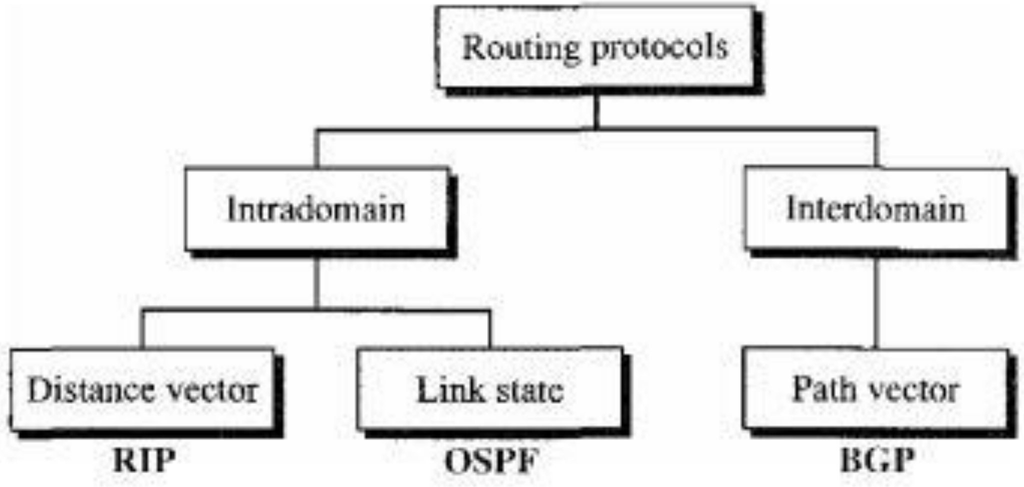
Hybrid

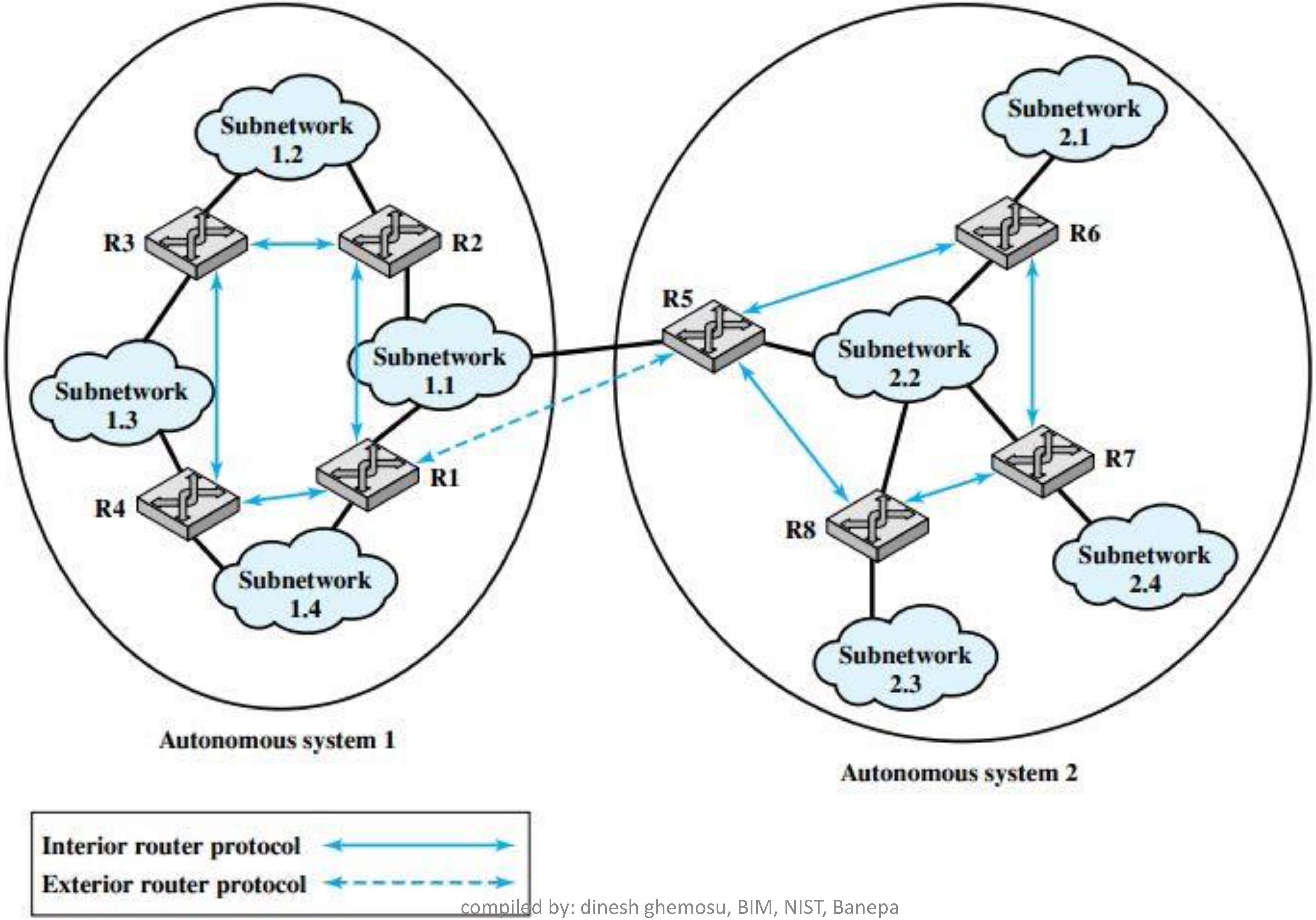
- Hybrid protocols use aspects of both distance vector and link state. E.g. : EIGRP. (Enhanced Interior Gateway Routing Protocol)

Autonomous System (AS)

- Autonomous system (AS) is a collection of networks under a common administrative domain, which basically means that all routers sharing the same routing table information are in the same AS
- According to AS, there are two types of routing protocols:

1. Intra-AS routing/interior Gateway protocol (IGP) e.g.: RIP, OSPF.
2. Inter-AS routing/exterior Gateway protocol (EGP) e.g.: Border gateway protocol (BGP)




Subnetwork 1.2
R3
R2
Subnetwork 1.1
Subnetwork 1.3
R4
R1
Subnetwork 1.4
Autonomous system 1
Subnetwork 2.1
R6
R5
Subnetwork 2.2
R7
R8
Subnetwork 2.4
Subnetwork 2.3
Autonomous system 2
Interior router protocol
Exterior router protocol

Routing in Internet

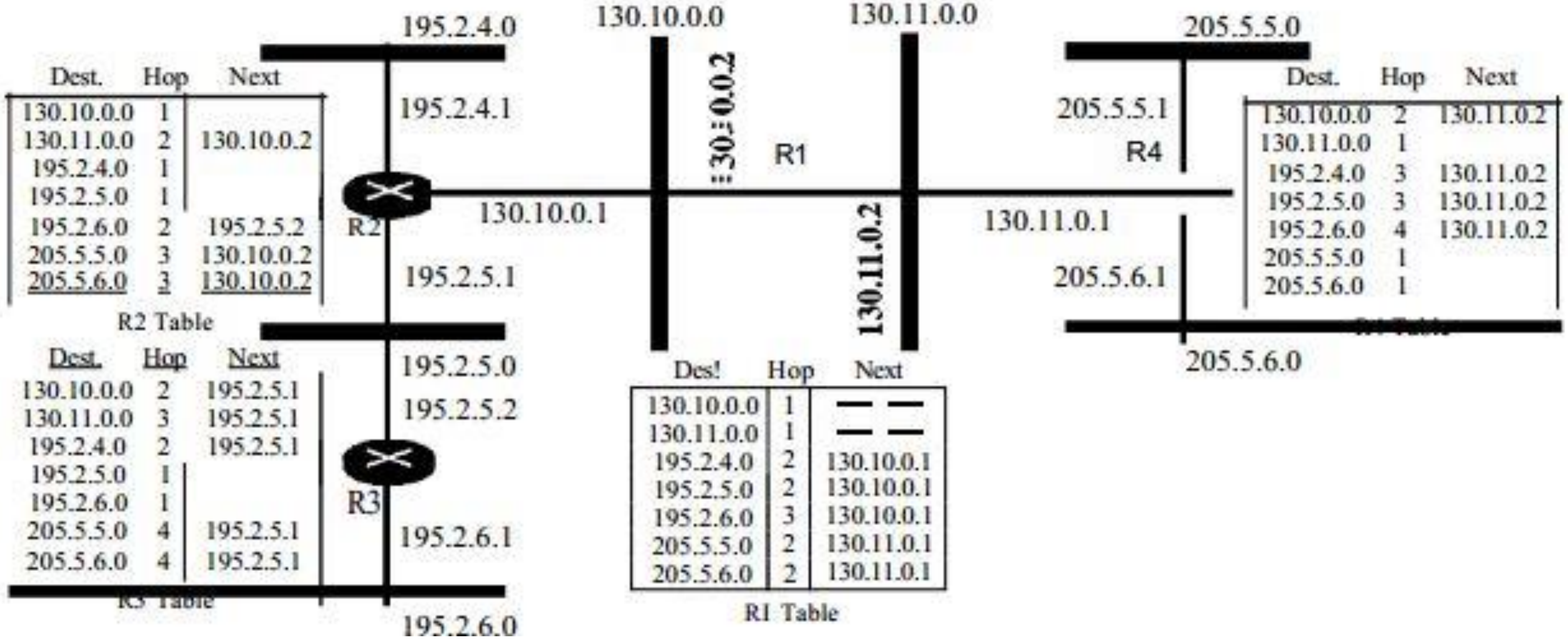
- RIP (Routing Information Protocol)
- OSPF (Open Shortest Path First)
- BGP (Border Gateway Protocol)

RIP (Routing Information Protocol)

- Distance-vector routing protocol.
- Intra AS routing Protocol.
- Employs the hop count as a routing metric.
- RIP prevents routing loops by implementing a limit on the number of hops allowed in a path from the source to a destination.
- The maximum number of hops allowed for RIP is 15.

- This hop limit, however, also limits the size of networks that RIP can support.
- A hop count of 16 is considered an infinite distance and used to deprecate inaccessible, inoperable, or otherwise undesirable routes in the selection process.
- Periodic updates every 30 seconds, even the topology not changed.
- In the early deployments, routing tables were small enough that the traffic was not significant. As networks grew in size, however, it became evident there could be a massive traffic burst every 30 seconds, even if the routers had been initialized at random times
- RIP uses the User Datagram Protocol (UDP) as its transport protocol, and is assigned the reserved port number 520.

Example of a domain using RIP



Dest.	Hop	Next
130.10.0.0	1	
130.11.0.0	2	130.10.0.2
195.2.4.0	1	
195.2.5.0	1	
195.2.6.0	2	195.2.5.2
205.5.5.0	3	130.10.0.2
205.5.6.0	3	130.10.0.2

R2 Table

Dest.	Hop	Next
130.10.0.0	2	195.2.5.1
130.11.0.0	3	195.2.5.1
195.2.4.0	2	195.2.5.1
195.2.5.0	1	
195.2.6.0	1	
205.5.5.0	4	195.2.5.1
205.5.6.0	4	195.2.5.1

R3 Table

Des!	Hop	Next
130.10.0.0	1	— —
130.11.0.0	1	— —
195.2.4.0	2	130.10.0.1
195.2.5.0	2	130.10.0.1
195.2.6.0	3	130.10.0.1
205.5.5.0	2	130.11.0.1
205.5.6.0	2	130.11.0.1

R1 Table

Dest.	Hop	Next
130.10.0.0	2	130.11.0.2
130.11.0.0	1	
195.2.4.0	3	130.11.0.2
195.2.5.0	3	130.11.0.2
195.2.6.0	4	130.11.0.2
205.5.5.0	1	
205.5.6.0	1	

Types of RIP

- RIPv1 (version 1)
- RIPv2 (version 2)
- RIPng (next generation)

RIPv1 (version 1)

- Uses classful routing.
- The periodic routing updates do not carry subnet information,
- Lacking support for variable length subnet masks (VLSM).
- There is also no support for router authentication, making RIP vulnerable to various attacks.
- Broadcast is used for database update

RIPv2 (version 2)

- Includes the ability to carry subnet information, thus supporting Classless Inter-Domain Routing (CIDR).
- In an effort to avoid unnecessary load on hosts that do not participate in routing,
- RIPv2 multicasts the entire routing table to all adjacent routers at the address 224.0.0.9, as opposed to RIPv1 which uses broadcast.
- Support Authentication

RIPng (next generation)

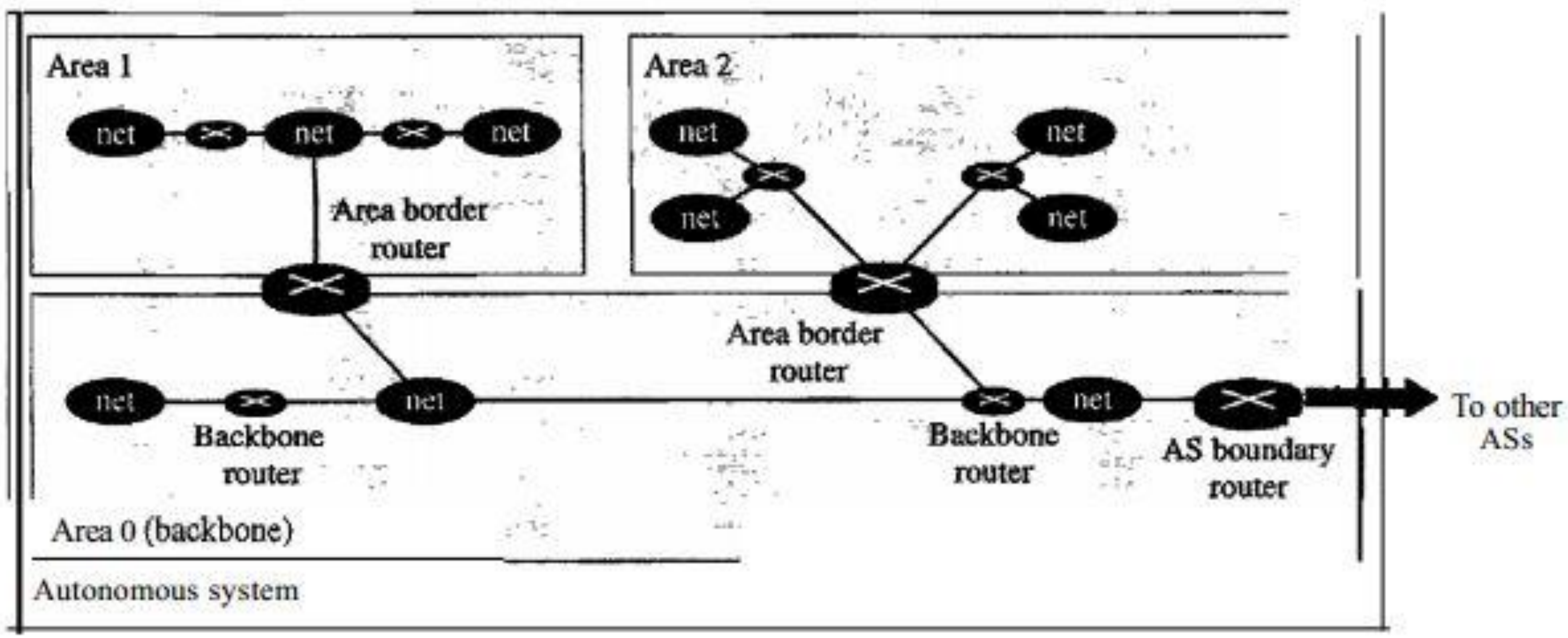
- Support of IPv6 networking.
- RIPng sends updates on UDP port 521 using the multicast group FF02::9.

OSPF (contd..)

- Link State Routing Algorithm
- Cost/Metric = Link Bandwidth
- Shortest Path Algorithm to calculate best path from source to destination.
- Open Shortest Path First (OSPF) is an adaptive routing protocol for Internet Protocol (IP) networks.
- It uses a link state routing algorithm and falls into the group of interior routing protocols, operating within a single autonomous system (AS).
- OSPF is perhaps the most widely-used interior gateway protocol (IGP) in large enterprise networks
- It included the ability to carry subnet information, thus supporting Classless Inter- Domain Routing (CIDR).
- Supports Authentication

OSPF: Open Shortest Path First

- OSPF is an interior routing protocol based on the link state routing, operating within a single autonomous system.
- To handle routing efficiently, OSPF divides an autonomous system into areas. An area is a collection of networks, hosts, and routers all contained within an autonomous system.
- Each area has an identification number.



Routers in OSPF

Internal Router

- Routers with all the interfaces within the same area.

Border Router

- Router with interfaces connected to multiple area.
- Summarize the information about the area and send to other area

Backbone Router

- Backbone: special area → primary area
- Routers with at least one interface connected to Area 0. (Identification no.)
- Areas inside autonomous system must be connected to the backbone.

Autonomous System Boundary Area

- Router that has at least one interface connected to another autonomous system

Metric

- The OSPF protocol allows the administrator to assign a cost, called metric, to each route.
- The metric can be bases on a type of service (minimum delay, maximum throughput, and so on).
- A routing tables have multiple routing tables based on different types of service.

Types of Links in OSPF

- A connection is called a *link*.
- Four types of links in OSPF protocol:

1. **Point-to-point link**: connects two routers without any host or router in between.
2. **Transient Link**: a network with several routers attached to it.
3. **Stud Link**: a network connected to only one router.
4. **Virtual Link**: when the link between two routers is broken, the administrator may create a virtual link between them, using a longer path the probably goes through several routers.

BGP (Border Gateway Protocol)

- Exterior Gateway protocol
- It is an interdomain routing protocol using path vector protocol.
- Routing protocol of the internet
- Slowest routing protocol.

Multicast Routing

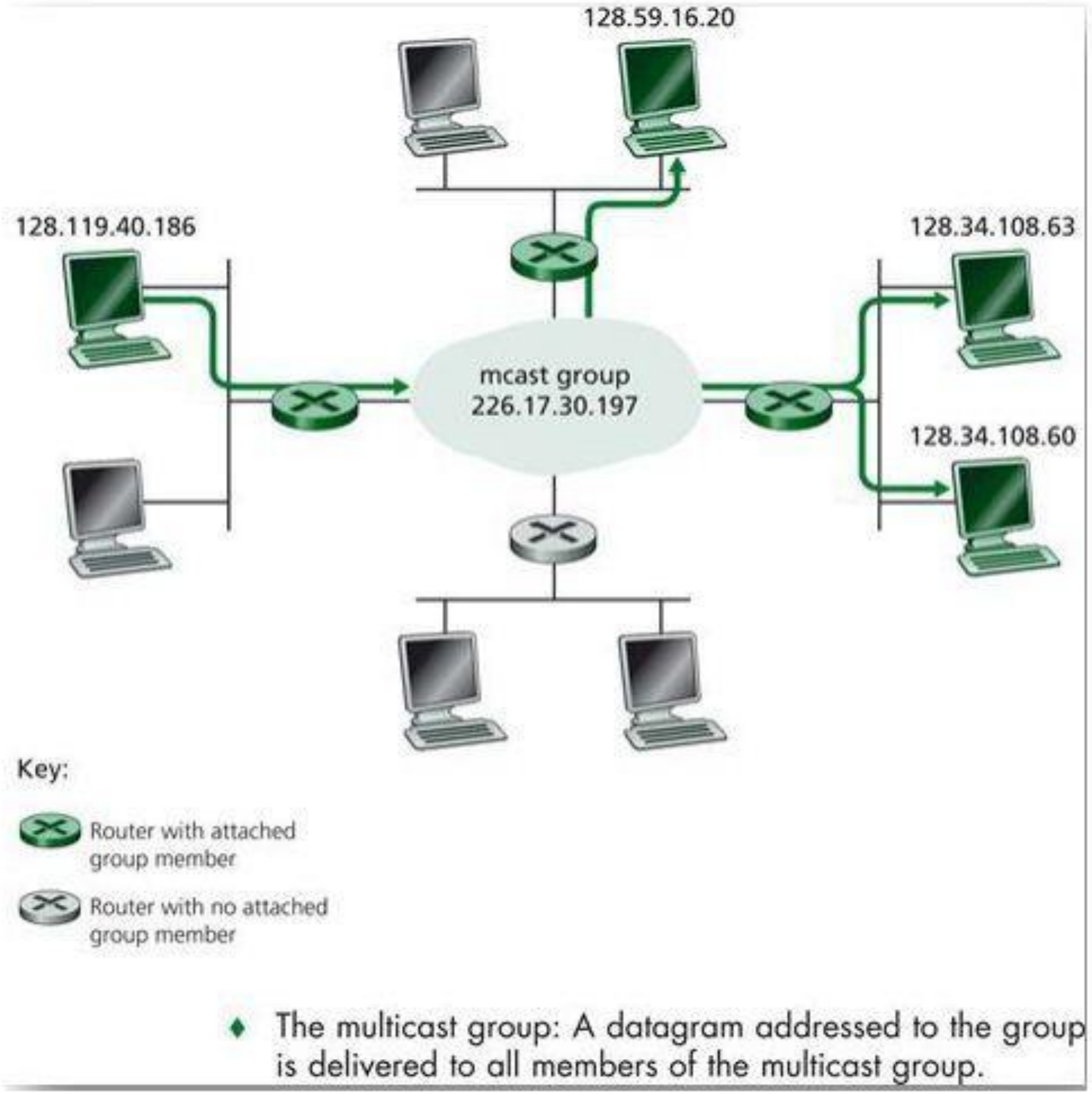
- In computer networking, multicast is the delivery of a message or information to a group of destination computers simultaneously in a single transmission from the source.
- Multicast is most commonly implemented in IP multicast, which is often employed in Internet Protocol (IP) applications of streaming media and Internet television.
- In IP multicast the implementation of the multicast concept occurs at the IP routing level, where routers create optimal distribution paths for datagrams sent to a multicast destination address.
- IGMP(Internet Group Management Protocol) runs between hosts and the nearest multicast routers.
- A local host can use it to inform the multicast router that, which multicast group it wants to be join, while the multicast routers can use it to poll the LAN periodically, thus determine if known group members are still active.

Applications of Multicast

- Video/audio conference
- IP TV, Video on Demand
- Advertisement, Stock, Distance learning
- Distributed interactive gaming or simulations
- Voice-over-IP
- Synchronizing of distributed database, websites

How Multicast

- Using Class D in IP v 4 (224-239) or addresses that begin with 1111 1111 (FF) in IP v 6
- e.g. 224.0.0.1, FF5B:2D9D:DC28:0000:0000:FC57:D4C8:1FFF
- Rather than sending a separate copy of the data for each recipient, the source sends the data only once using the multicast group, and routers along the way to the destinations make copies as needed.



Shortest Path Routing

- In this technique, the shortest path is choose to route the packet form source to destination.
- One way of measuring path length is the number of hops. Another metric is the geographical distance in kilometers.
- However, the shortest path is the fastest path rather than the path with the fewer hops. Shortest path is calculated as a function of different factors like distance, bandwidth, communication cost, delay, average traffic etc.
- Two common algorithms are used.
  - I. Dijkstra’s Algorithm
  - II. Bellman-Ford Algorithm

Dijkstra’s Algorithm

- **Dijkstra's algorithm** is an algorithm for finding the shortest paths between nodes in a graph.
- It is an iteration algorithm. After k^{th} iteration, the least cost path is found; these nodes are in the set T.
- At stage (K+1), the node not in T that has the shortest path from source node is added to T.
- As each node is added to T, its path from the source is defined.

Dijkstra's Algorithm

Let,

N= set of nodes in the network

s=source node

T= set of nodes so far incorporated by the algorithm

$w(i,j)$= cost from node i to node j

$w(i,i) = 0$;

$w(i,j) = \infty$ if nodes are not directly connected.

$w(i,j) \geq 0$, if nodes are directly connected

L(n) = cost of the least-cost from node s to node n that is currently known to the algorithm; at termination, this is the least-cost path in the graph from s to n.

Step 1: Initialization

- T = {s} i.e. the set of nodes so far incorporated ; in this case consists of only the source node.
- L(n) = *w(s,n)* for n≠ s i.e. the initial path costs to neighbor nodes are simply the link cost

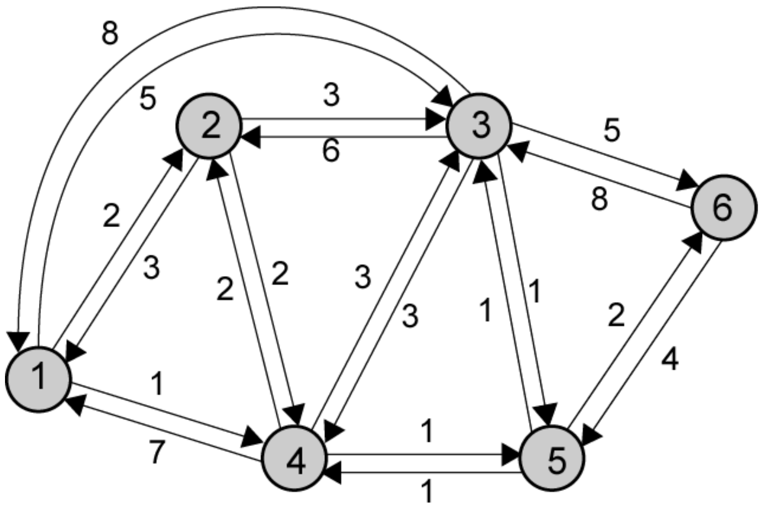
Step 2: Get Next Node

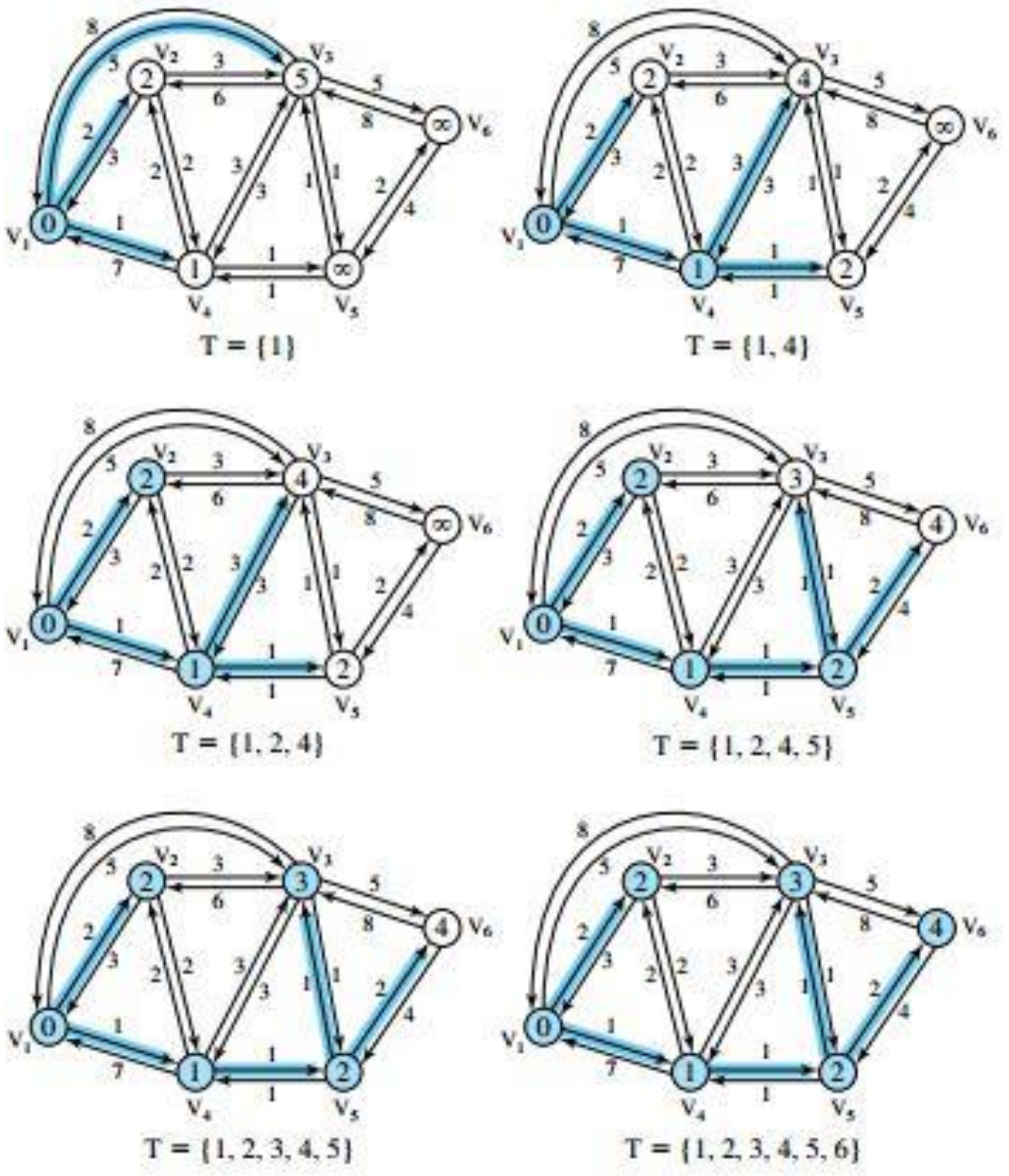
- Find the neighboring node not in T that has the least-cost path from node s and incorporate that node into T.
- Also incorporate the edge contributes to the path. Mathematically expressed as:
- Find x such that it does not belong to T and such that L(x) = min L(j), j≠T
- Add x to T; and find the least cost of edge for nodes in updated T.

Stet 3: Update Least-Cost Path

- L(n) =min [(*L(n), L(x) + w(x,n)*] for all n not belongs to T

- Repeat step 2 and 3 until T = N
- The algorithm terminates when all nodes have been added to T.





- The values in each circle are the current estimates of L(x) for each node x.
- A node is shaded when it is added to T.
- At each step the path to each node plus the total cost of that path is generated.
- After the final iteration, the least-cost path to each node and the cost of that path have been developed.

Figure: Dijkstra’s Algorithm Applied to graph, using s=1

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

Figure: Dijkstar’s Algoirthm (s=1)

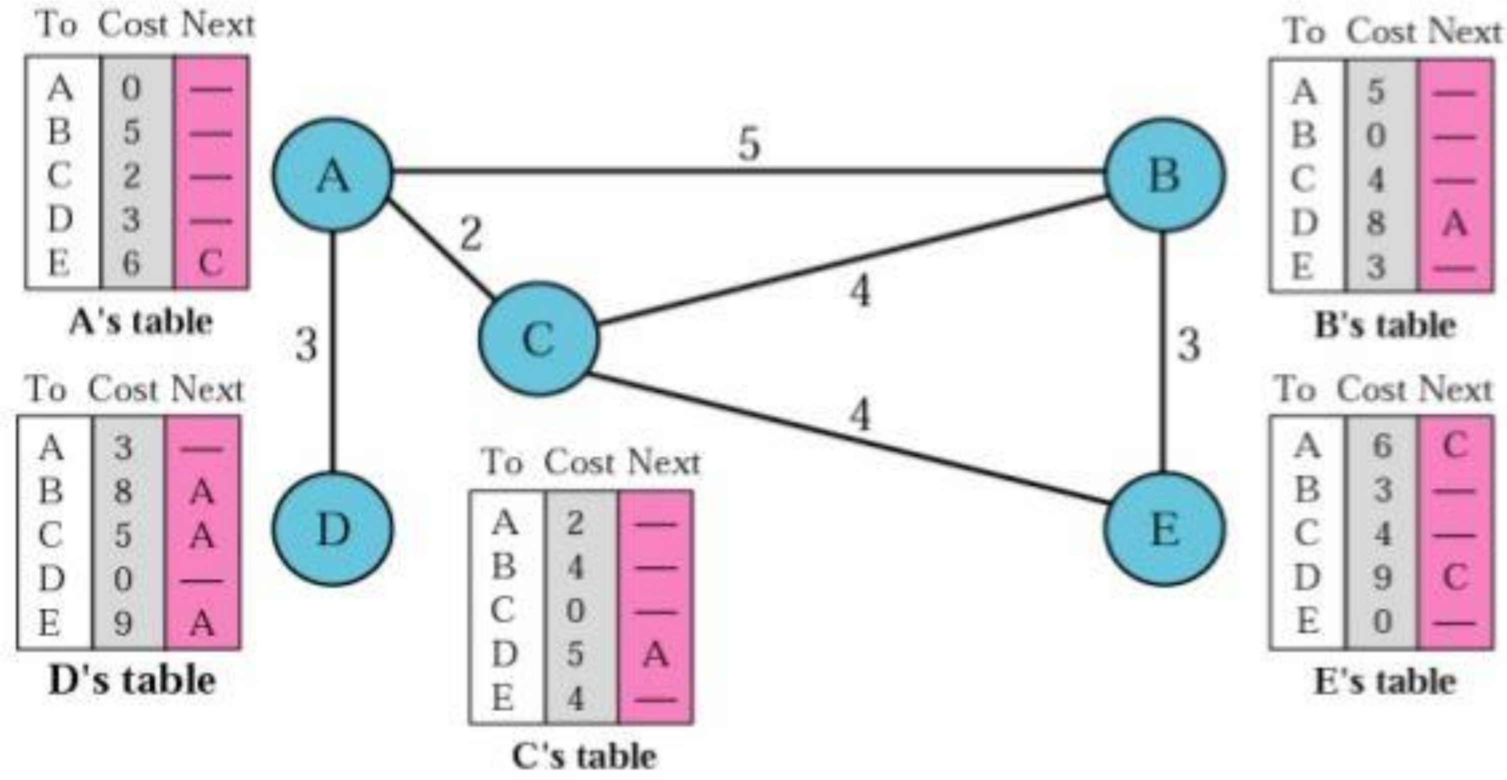
Distance Vector Routing

- Dynamic vector routing algorithm
- Also called as Bellman-Ford Routing Algorithm
- Used in ARPANET and RIP

- In distance vector routing, each node maintain a vector (table) of minimum distance to every node. The table at each node also guide the packet to the desired node by showing the next stop (next hop) in the route.
- Each node shares its routing table with its immediate neighbors periodically and when there is a change.

Distance Vector Routing Table

The tables for node A shows how can we reach any node from this node.
For example, our least cost to reach node E is 6, through 6.



- Each node maintains a vector of distances (and next hops) to all destinations

1. Initialize vector with 0 (zero) cost to self, ∞ (infinity) to other destinations
2. Periodically send vector to neighbors (every 30s)
3. Update vector for each destination by selecting the shortest distance heard, after adding cost of neighbor link
    - Use the best neighbor for forwarding

Bellman-Ford Algorithm

s = source node

$w(i, j)$ = link cost from node i to node j; $w(i, i) = 0$; $w(i, j) = \infty$ if the two nodes are not directly connected; $w(i, j) \geq 0$ if the two nodes are directly connected

h = maximum number of links in a path at the current stage of the algorithm

$L_h(n)$ = cost of the least-cost path from node s to node n under the constraint of no more than h links

1. **[Initialization]**

$$L_0(n) = \infty, \text{ for all } n \neq s$$

$$L_h(s) = 0, \text{ for all } h$$

2. **[Update]**

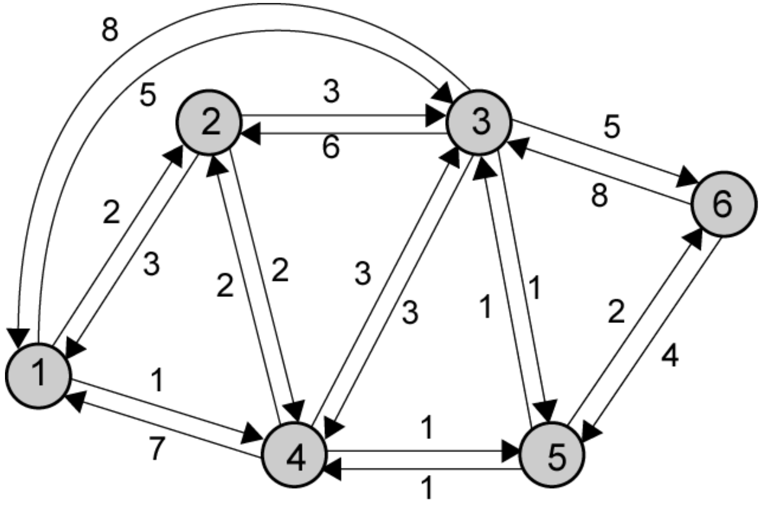
For each successive $h \geq 0$:

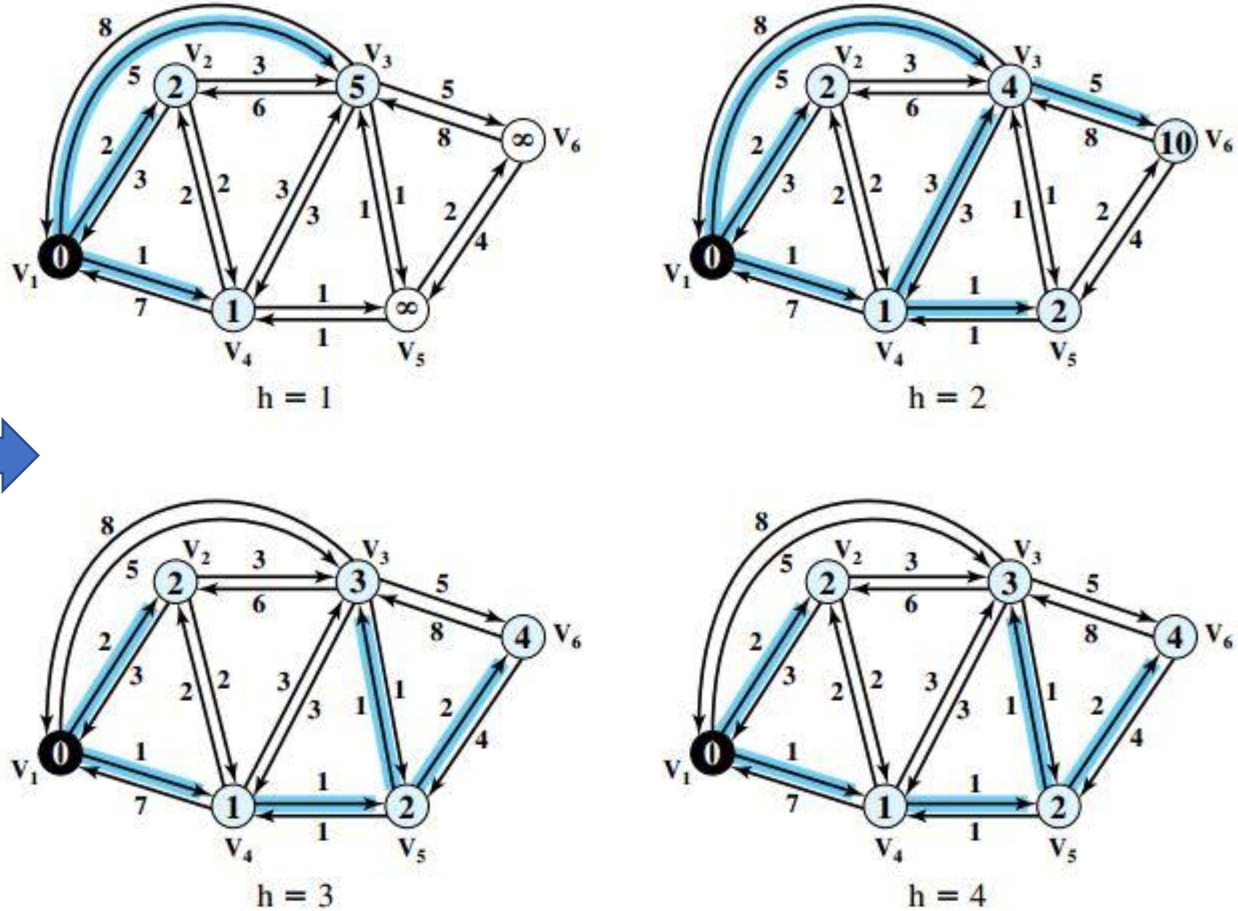
For each $n \neq s$, compute

$$L_{h+1}(n) = \min_j [L_h(j) + w(j, n)]$$

Connect n with the predecessor node j that achieves the minimum, and eliminate any connection of n with a different predecessor node formed during an earlier iteration. The path from s to n terminates with the link from j to n.

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





compiled