EEE 432 Introduction to Data Communications

Asst. Prof. Dr. Mahmut AYKAÇ

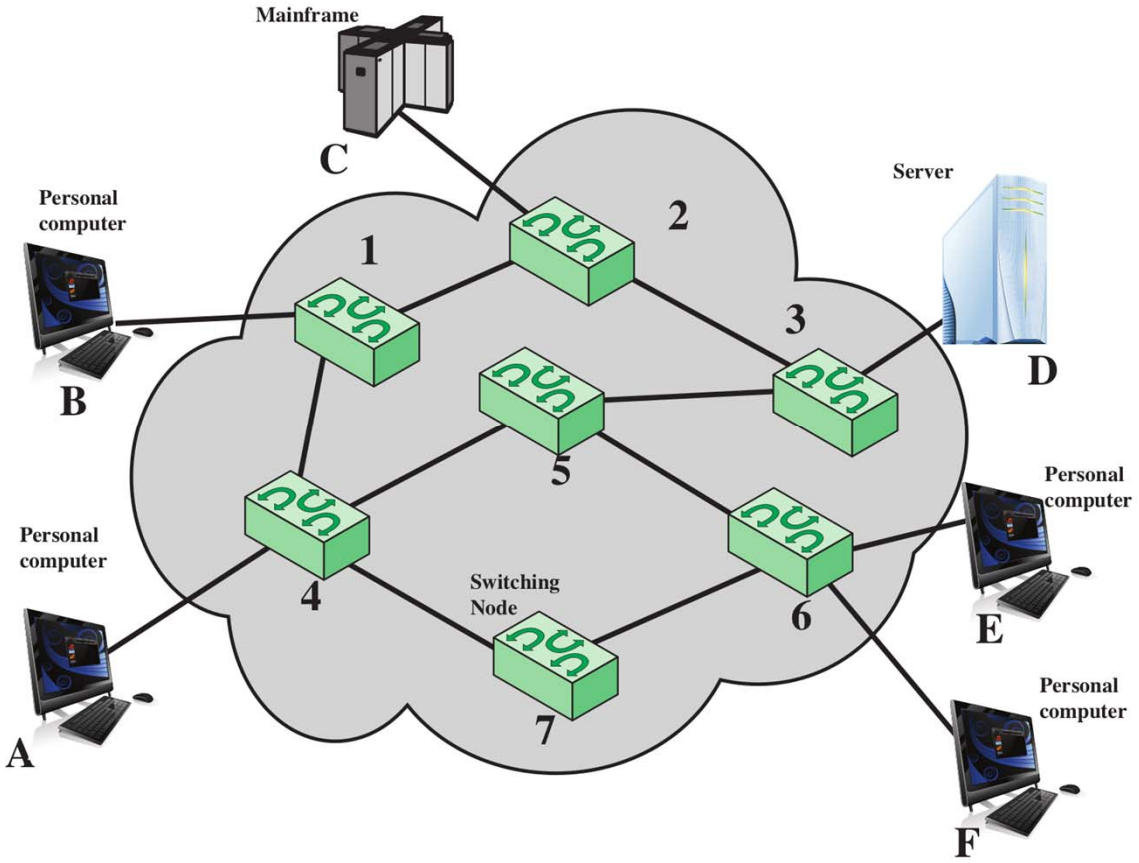
ROUTING

Course Information

1. Data Communications and Networks
2. Data Transmission
3. Transmission Media
4. Signal Encoding Techniques
5. Digital Data Communication Techniques
6. Multiplexing
7. Networking and Protocol Architectures
8. Switching
9. **Routing in Switched Networks**
10. LANs and WANs
11. Ethernet
12. The Internet

Routing in Switched Networks

Which path or route to take from source to destination?



Routing in Switched Networks

- **Routing** is the process of selecting a path for traffic in a network or between or across multiple networks. It is a key design issue in switched networks
- **Question:** What path (route) should be taken from source to destination?
- **Answer:** Choose the "best" path!
- What is "best", and how to choose it?
- Real networks may have 100's to 100,000+ nodes, and many possible paths
- Routing is needed in circuit-switched and packet-switched networks (we focus on packet-switched networks)

Requirements of Routing Algorithms

Correctness, path must be from intended source to intended destination

Simplicity, easy/cheap to implement

Robustness, still deliver in presence of errors or overload

Stability, path changes should not be too frequent

Optimality, choose best paths

Fairness, ensure all stations obtain equal performance

Efficiency, minimise the amount of processing and transmission overhead

Routing Terminology

Link, direct connection between two nodes

Path, a way between two nodes, via one or more links

Hop, to traverse a link

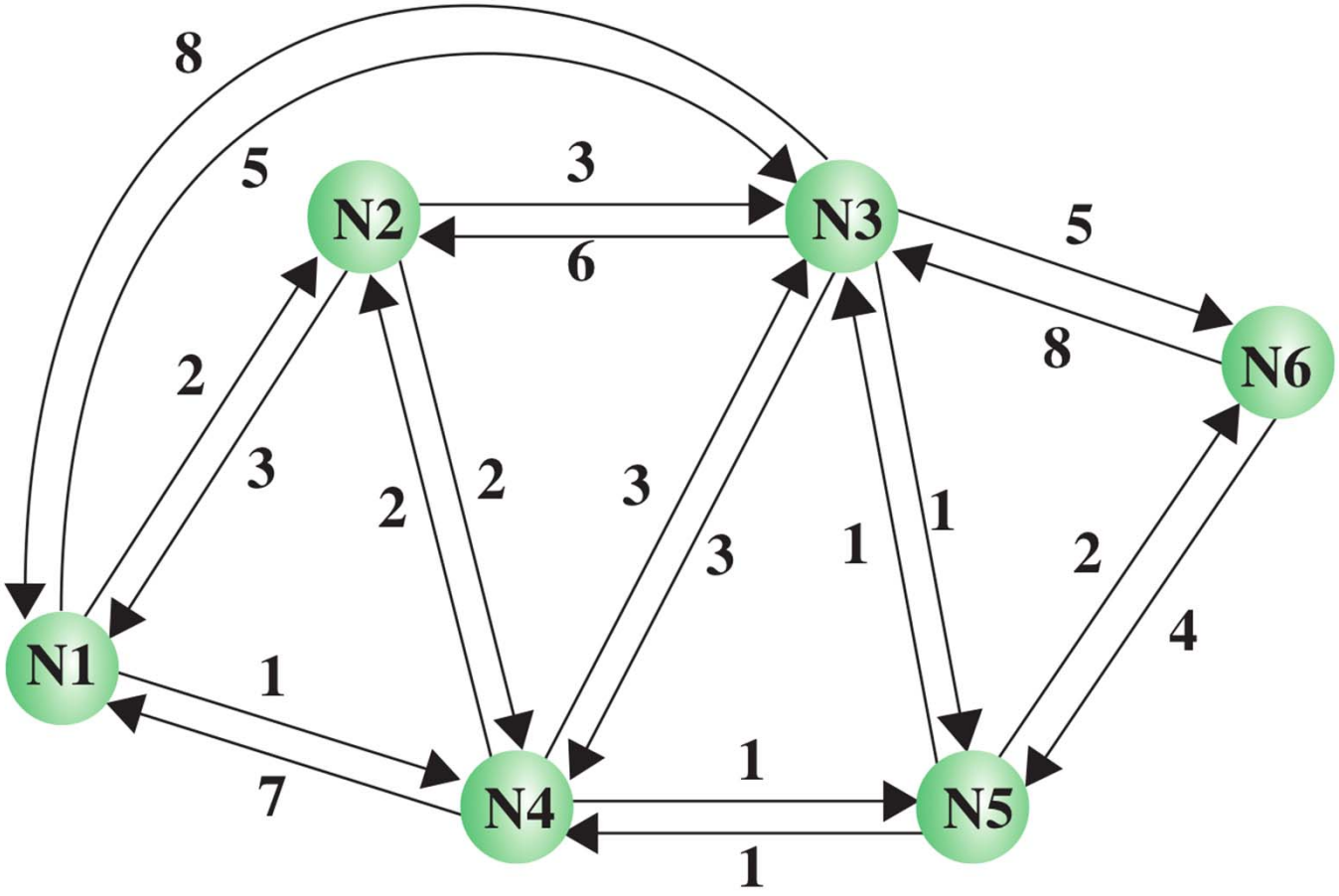
Neighbour, a node at the other end of a link

Cost, value assigned to link to indicate cost of using that link

Topology, arrangement of nodes and links in a network

Least-cost routing is typically used: choose a path with least cost

Example of Network Configuration



N1 to N6 are **nodes**, arrows are **links** with direction and the numbers on the links show the **costs.** Costs of full duplex data transfer may be different.

For ex: Cost of N1 to N4 is 1 but N4 to N1 is 7

Ex: Least hop path from N1 to N6 is **N1→ N3→ N6 (Value is 2)**

Least cost path from N1 to N6 is **N1→N4→N5→N6(Value is 1+1+2=4)**

Elements of Routing Techniques

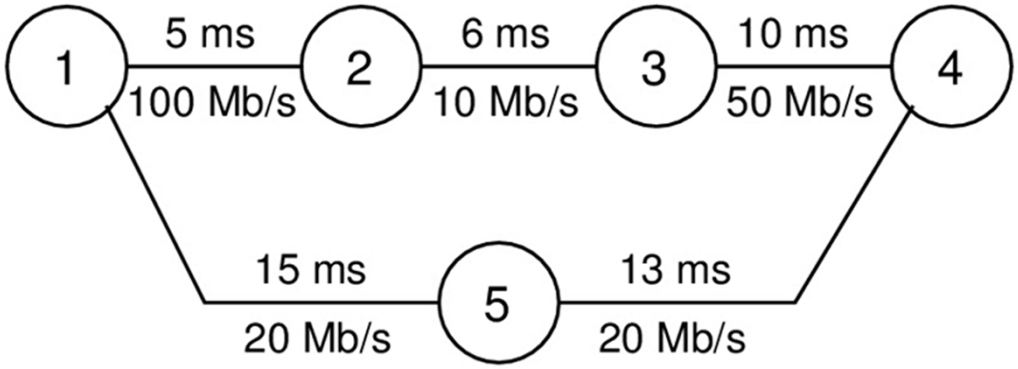
- Which performance criteria are used to select a path?
- When is a path selected?
- Which nodes are responsible for selecting path?
- Which nodes provide information about network status?
  - Topology, link costs, current usage
- How often is network status information updated?

Elements of Routing Techniques

Performance Criteria Number of hops Cost Delay Throughput **Decision Time** Packet (datagram) Session (virtual circuit) **Decision Place** Each node (distributed) Central node (centralized) Originating node (source)	**Network Information Source** None Local Adjacent node Nodes along route All nodes **Network Information Update Timing** Continuous Periodic Major load change Topology change

Example..

➢ **Ex:** Find the least-cost paths (1→4) in terms of hops, delay and throughput metrics.



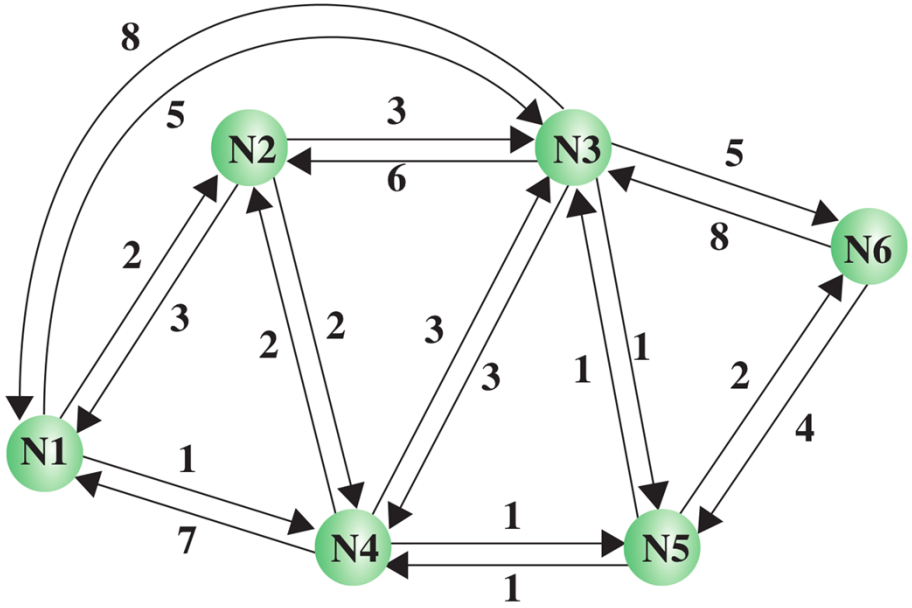
Best = Lowest Delay, Cost = Delay
1→ 2→ 3→ 4 (Top path, Cost= 21ms)
Best = Lowest Hops, Cost = Hops
1→ 5→ 4 (Bottom path, Cost=2)
Best= Highest Throughput, Cost = Max. Throughput/Throughput
Path data rate is limited by the bottleneck link. Bottleneck is the slowest link. The bottleneck link 1→4 is the link between 2 and 3. Therefore we can conclude that the throughput of the top path is 10Mb/s and it is 20Mb/s for the bottom one.
$Cost_{top} = (100/100) + (100/10) + (100/50)$ = **13**
$Cost_{bottom} = (100/20) + (100/20)$ = **10**
Bottom path is the best of both.
We may be able to find a scenario using the approach with particular link rates, the least cost path chosen will be the one with which is not giving the highest throughput. So, it doesn't always work. But in most cases, it works.

Routing Strategies → Strategy 1: Fixed Routing

- Use a single permanent route for each source to destination pair
- Routes are determined using a least cost algorithm, e.g. Dijkstra, Bellman-Ford algorithms
- Route is fixed
  - At least until a change in network topology (node/link added/deleted)
  - Hence cannot respond to traffic changes (e.g. overload in one portion of the network)
- No difference between routing for datagrams and virtual circuits
- Advantage is simplicity
  - You assign the routes at the start, and then nothing to do
- Disadvantage is lack of flexibility
  - When the network is operating, changes in load may mean better routes than initially selected should be used

Fixed Routing Example

- How many least-cost paths are there? What are they?



Dst: 1→
Dst: 2→
.
.

Src: 1					Costs
1	0	0	0		0
1	2	0	0		2
1	4	5	3		3
1	4	0	0		1
1	4	5	0		2
1	4	5	6		4
Src: 2					
2	1	0	0		3
2	0	0	0		0
2	3	0	0		3
2	4	0	0		2
2	4	5	0		3
2	4	5	6		5
Src: 3					
3	5	4	2	1	7
3	5	4	2	0	4
3	0	0	0	0	0
3	5	4	0	0	2
3	5	0	0	0	1
3	5	6	0	0	3

Src: 4					
4	2	1			5
4	2	0			2
4	5	3			2
4	0	0			0
4	5	0			1
4	5	6			3
Src: 5					
5	4	2	1		6
5	4	2	0		3
5	3	0	0		1
5	4	0	0		1
5	0	0	0		0
5	6	0	0		2
Src: 6					
6	5	4	2	1	10
6	5	4	2	0	7
6	5	3	0	0	5
6	5	4	0	0	5
6	5	0	0	0	4
6	0	0	0	0	0

In this case, each node knows to where to send the data according to the destination. This look-up table is fixed and registered in the memory of each node.

Routing Tables

- A node determines least-cost paths to all possible destinations
- No need to store entire path; store only next node in path (and optionally cost of path)
- Path information stored in routing table (or directory)

Destination	*Next*	*Path Cost*
Node_1	Node_x	c_1
Node_2	Node_y	c_2
⋮		

- Routing table may be stored on central node or distributed amongst each node
- Separation of routing and forwarding:

  Routing: strategies, protocols and algorithms are used to create routing table

  Forwarding: routing table used to determine where to send the data to next

Distributed Routing Tables Example

Each node has its own routing table. There is no need to store entire path, only next node.

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

It is mostly used in large networks such as Internet, where the same information is stored but split amongst the different nodes

Centralised Routing Table Example

- Routing table stored on one node. This is just combination of previous 6 tables

To Node \ From 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	—

It is mostly used in small networks when we have one special node that can keep track the routes and stores

Fixed Routing Summary

- When is a decision made for a route? **At network startup**
- Which node chooses the route? **Centralised or distributed**
- Where does the network information come from? **All nodes**
- When is the network information updated? **Never**
- In practice, only used for small, stable networks **(10s of nodes)**

Strategy 2: Flooding

- Instead of choosing a route before sending the data, just send the data to everyone
  - A copy of the original packet is sent to all neighbours of the source
  - Each node that receives the packet, forwards a copy of the packet to all of its neighbours
  - Flooding is very useful for learning information about the network and that is what is used for in practice

- **Advantages:**
  - All possible routes are tried; at least one packet will take minimum hop route, e.g. setup a virtual circuit
  - All nodes are visited, e.g. distributing network status (topology) information
  - Simple

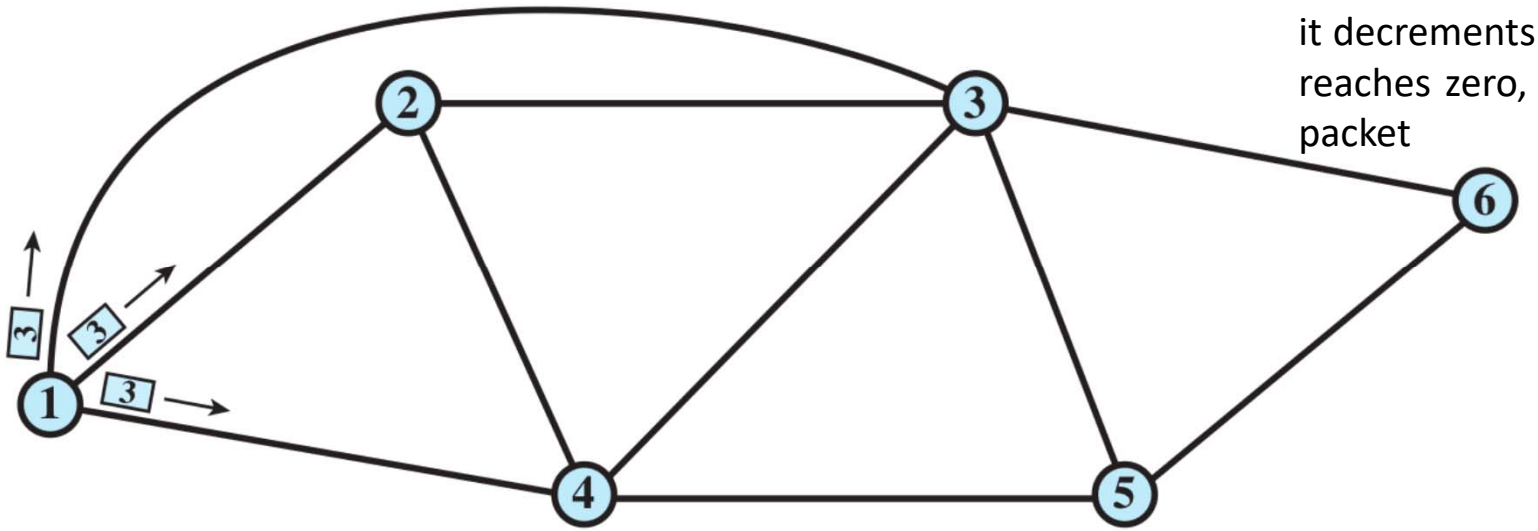
- **Disadvantages:**
  - Inefficient: need to send many copies of packet to get one packet from source to destination
  - Using hop limit and/or selective flooding, packet may not reach destination

Flooding Extensions

- **1.** Don't send back to the node that just sent you the packet
- **2.** Only forward packet once: nodes remember which packets they have forwarded (based on sequence number and source/destination addresses); do not forward a packet if you have previously forwarded that same packet
- **3. Duplicate Detection:** each packet has a sequence number, so if destination receives multiple copies of the same packet, it can discard the duplicates
- **4. Hop Limit:** include a "hop counter" in the packet; decrement the counter each time the packet is forwarded, if it is 0, then discard the packet
- **5. Selective Flooding:** send to selection of neighbours. E.g. random, round-robin, probability-based

Flooding Example

Destination is node 6; Hop limit is 3 (the number in the packet).



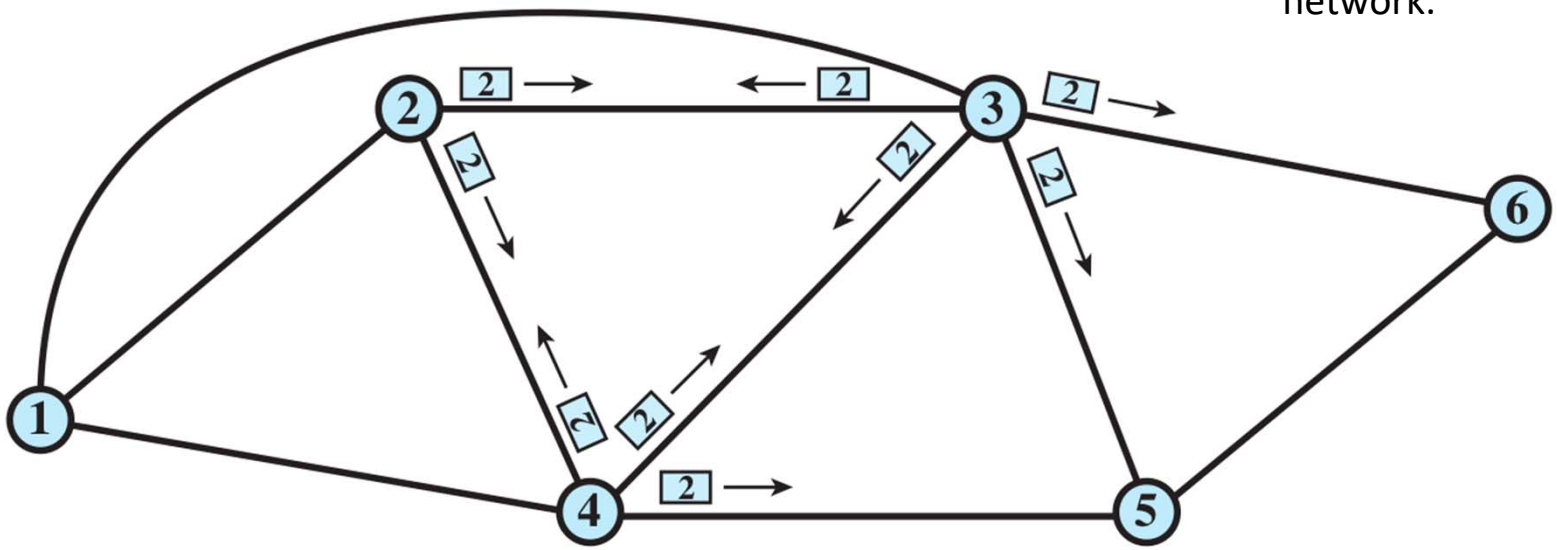
Step 1: 3 packets

* Source determines the hop limit
* Every time a node receives a packet, there is also “hop counter” info inside.
* When a node receives a packet, it decrements the hop counter, if it reaches zero, it does not send the packet

Flooding Example

- Destination is node 6; Hop limit is 3.

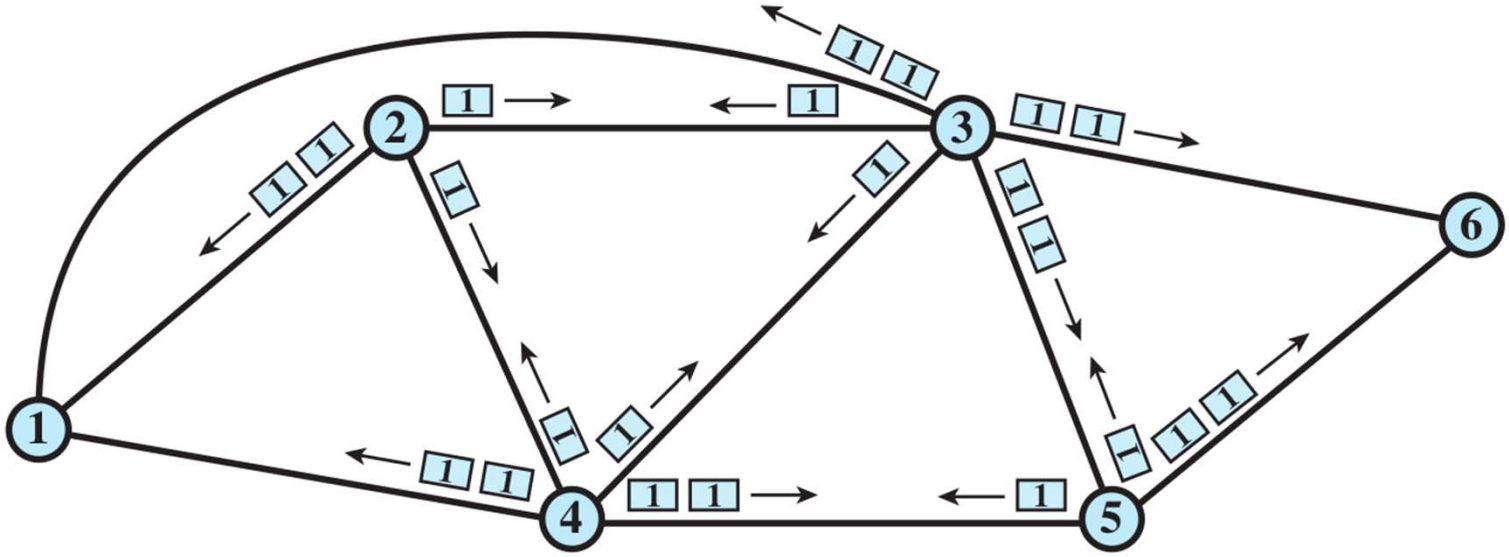
* If the number of hops are not limited, transmissions of the packets may last forever, which will cause a big amount of load on the network.



Step 2: 9 packets

Flooding Example

- Destination is node 6; Hop limit is 3.



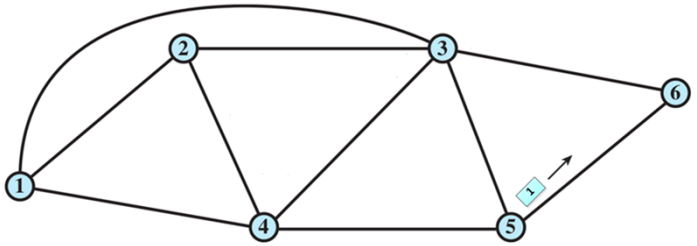
Step 3: 22 packets

Flooding Example

- How many packets are transmitted in the example?
- **1st Step:** 3 packets
- **2nd Step:** 9 packets
- **3rd Step:** 22 packets
- **Total:** 34 packets
- In this example, each node unconditionally distributes packets to each of its neighbours. Without conditional logic to prevent indefinite recirculation of the same packet. It is called **uncontrolled flooding**.
- **In controlled flooding, sequence number** keeps the flooding in check. Every router will flood a packet indicating the same source and sequence number only once.

Flooding Example

- How many packets transmitted in previous example with sequence number (controlled flooding)?
- **1st Step:** 3 packets
- **2nd Step:** 9 packets
- **3rd Step:** 1 packet (from 5 to 6)
- **Total:** 13 packets (significant reduction in packet sending)

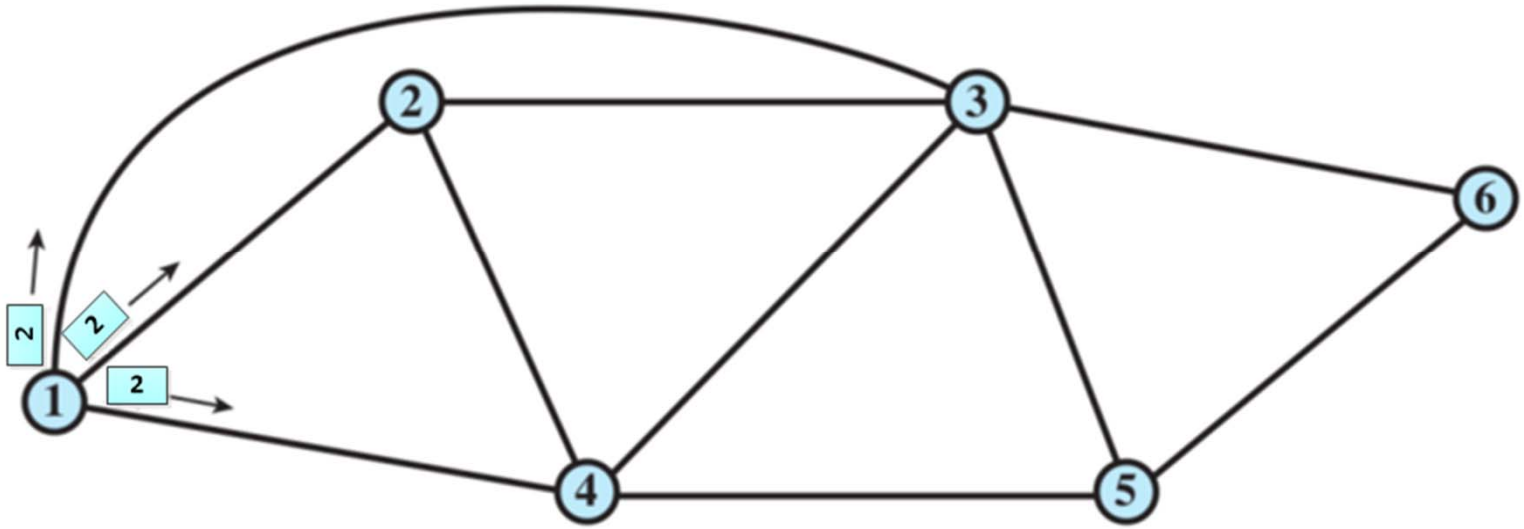


Step 3: 1 packet

- In this example, steps 1 and 2 are identical with previous one, but in the last step, node 5 receives the same packet from 3 and 4 (step 2), so it will pass this packet to its different neighbours except 3 and 4, which is only 6.
- If you receive a packet, don't send it again. Because your neighbours have already received it! Nodes remember the packets that they have already sent.

Flooding Example

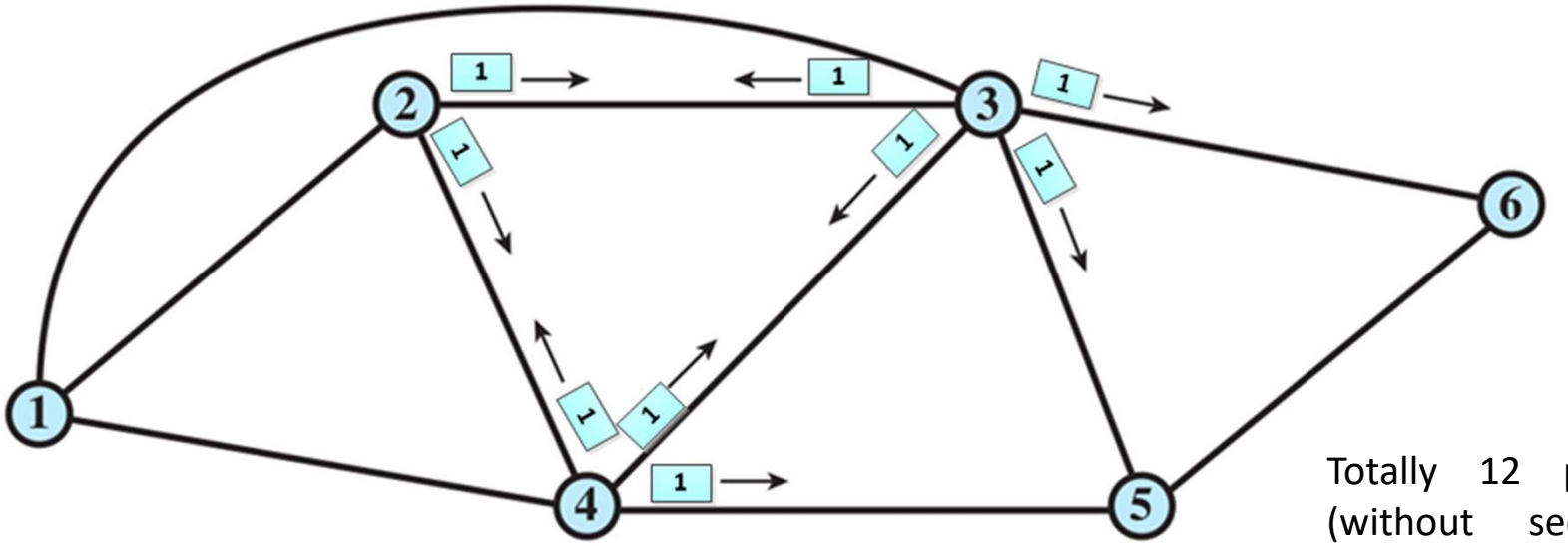
- How many packets transmitted in previous example if hop limit was 2?



Step 1: 3 packets

Flooding Example

- How many packets transmitted in previous example if hop limit was 2?



Step 2: 9 packets

Totally 12 packets (without sequence number) and all nodes receive the data from node 1

Selective Flooding

- Nodes send data to its one or some of its neighbours.
- For our example, node 1 sends data to 2 and 4 (excluding 1) and imagine the hop limit is 2. Since our destination is 6, the data may reach the destination but it helps us to reduce the number of transmissions.
- That is why while it reduces the load on the network, it may cause transmission failures
- Because we don't send data to some node(s), we take risk and the desired data may not reach the destination. Therefore, there is a trade-off here.