Computer Networks
The Network Layer

2025/2026, 1st semester

Dr. Szilveszter Kovacs

E-mail: szilveszter.kovacs@uni-miskolc.hu

www.iit.uni-miskolc.hu/~szkovacs

Institute of Information Technology 107/a.

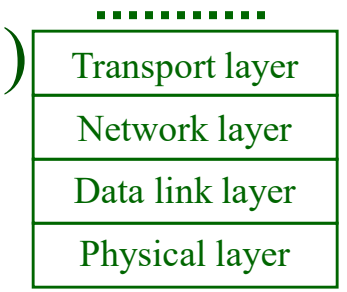
Phone: +36 46 565-111 / 21-07

What will it be about?

- **The network layer**
  - **Its task is to deliver packets from source node to destination node**
  - **above, below end-to-end, here you can see the entire network**
  - **addressing, functions, network organization (connection-based, connectionless)**
- **Traffic control**
- **Congestion control**
- **Inter-network cooperation**

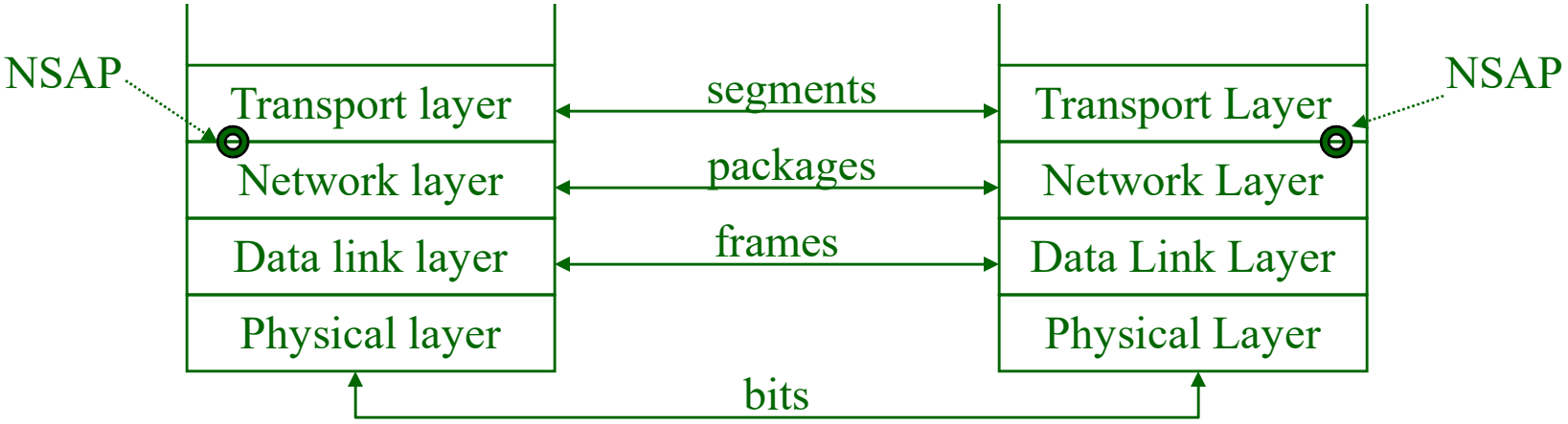
The task of the network layer

- **Delivery packages from source node to the destination node**
  - **The data link layer:**
    only performs frame movement between the two ends of a single "line" (single link), e.g. error correction, queuing, traffic control
  - **The transport layer:**
    true end-to-end (true source-destination connection)
    - **it does not know the topology of the network or the way packets are delivered to their destination.**
- **The network layer deals with how end-to-end transmission occurs**
  (it needs to know the network, topology, etc.)



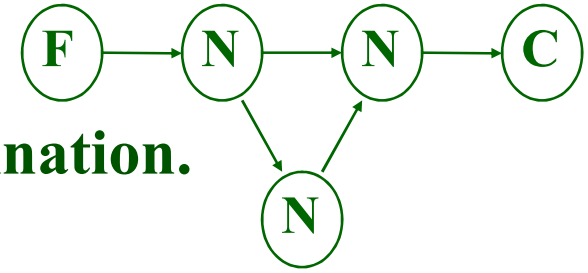
The task of the network layer

- **Generally:**
  - **well-defined services towards the transport layer, i.e.**
  - data **unit (packet)** received **and addressed** (network destination address, source address) **from the transport functional element via NSAP (Network** Services Access Point **) to the addressed NSAP (the functional partner element) (and nowhere else).**



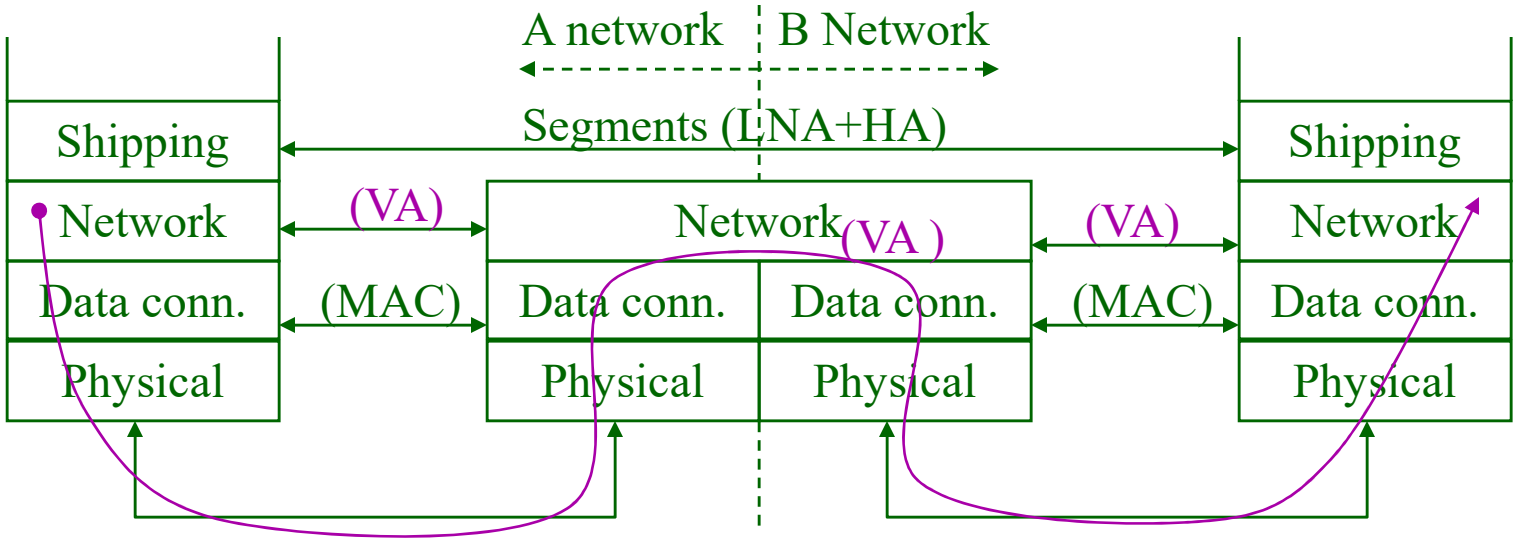
NSAP (Network Service Access Point) **address** : **identifies a network service**

Network layer functions



- **Traffic control**
  - **delivering the package to its destination.**
  - **you need to know the topology**
  - **load sharing (alternative routes)**
- **Congestion control**
  - **Do not overload parts of the network**
  - **It is similar to flow control, but it applies not only between two points (transmitter and receiver), but to the entire network.**
- **Inter-network cooperation**
  - **This is the first layer where different networks can be connected (forming heterogeneous networks)**

Inter-network cooperation



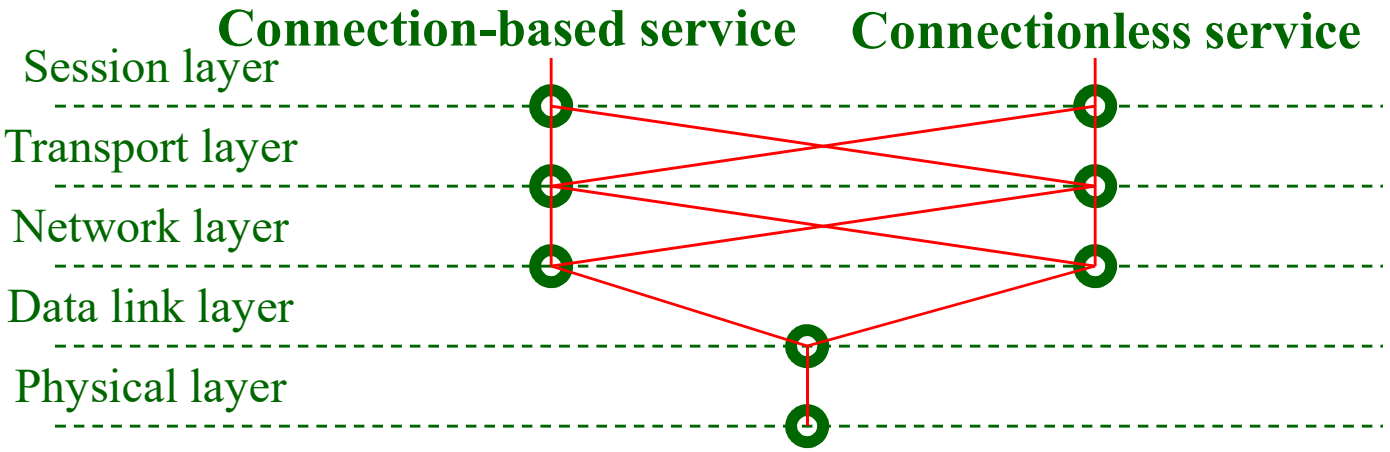
- **The functions of network nodes (traffic routing, congestion control) extend only to the network layer**
- **There may be different data link layers (heterogeneous network) underneath**

Titles:

- **LNA: Logical Network Address, HA: Host Address**
- **VA: virtual circuit address (if any)**
- **MAC: Media Access Control (for broadcast media)**

Services provided to the transport layer

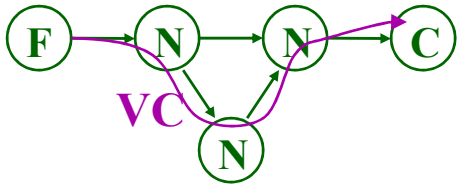
- **They can be:**
  - **Connection-based (virtual circuit)**
  - **Connectionless (datagram)**
- **These types of services occur at multiple levels, and may differ from level to level.**

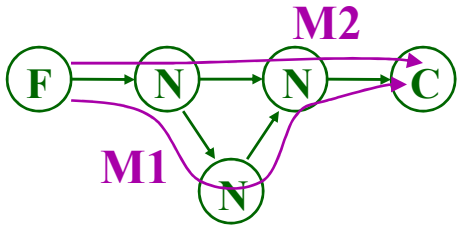


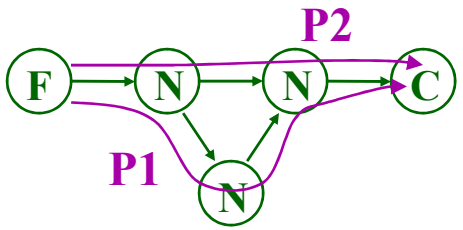
E.g.: Connection-based data connection service can have a connectionless network (e.g.: an Connectionless network on a switched line) – and vice versa

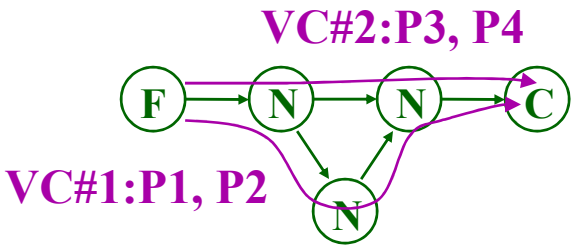
Service types

- **Connection-based service**
  - **(Circuit Switching)**
- **Connectionless service**
  - **Message Switching (Connectionless)**
  - **Packet Switching**
    - **Datagram Packet Switching (Connectionless)**
- **Connection-based service**
  - **Virtual line switching (ÖK based)**
    - **Virtual Circuit Packet Switching**









Internal organization of the network layer

Two different subnetwork organization philosophy:

- **Connections-based – virtual circuits**
- **Connectionless – datagram**
- **Virtual circuit organization is favorable if:**
  - **It primarily provides connection-based services.**
  - **There should be no routing for each packet (there is no predefined route in a datagram subnet, even if the service is connection-based)**

Virtual circuit-based subnet organization

- During call setup, a virtual circuit is established between the source and destination.

⇒Traffic routing is done during seting up the call!

- During communication, packets travel on the same path, the open virtual circuit (VC) (in both directions).
- Once the communication is complete, the virtual circuit must be released.
- Each channel can have multiple virtual circuits (their number is maximized).

Virtual circuit-based subnet organization

Address :

- **The full source and destination addresses are only needed during call setup.**
  - **After that, it is enough to indicate the virtual circuit**

  ⇒Each packet contains a field to indicate the virtual circuit
  (There can be multiple virtual circuits on each channel (their number is maximized)).

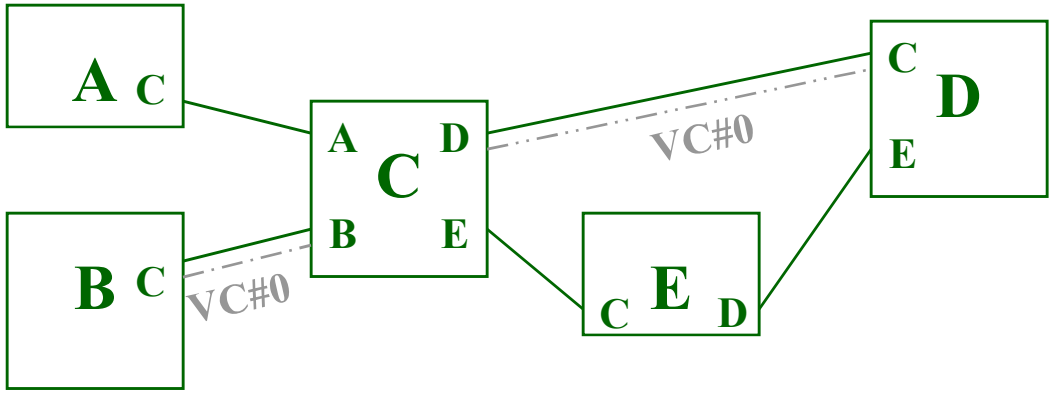
Virtual circuit-based subnet organization

When setting up a call :

- **The node selects the appropriate channel (route) and reserves a virtual circuit on it (e.g. the lowest free sequence number path)**
- **If there is no free circuit, it chooses another route. If there is none, the call setup fails.**
- **This is repeated for all nodes along the path.**

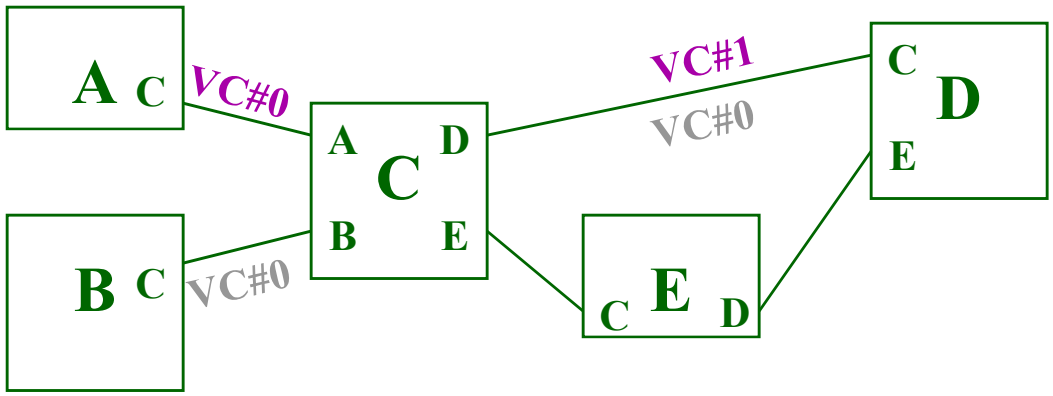
⇒In each node, a table of open virtual circuits: which line's which circuit is connected to – which line's which circuit

Example of a virtual circuit-based subnet organization



- There are 5 nodes: A, B, C, D, E
- In each node, the channel sign is the name of the neighbor.
- At start-up, a virtual circuit should already exist between B-C-D.
- Maximum of 2 VCs can be created on a channel.
- Task: Create two VCs between A and D

The first VC from A to D...

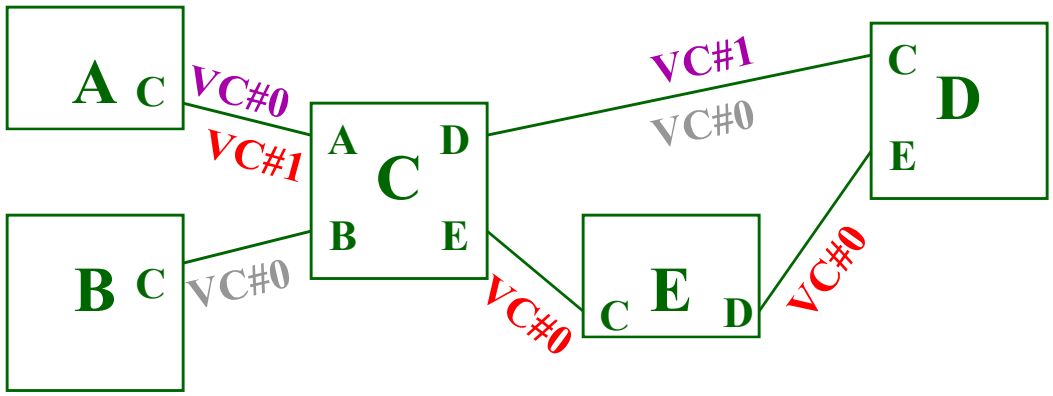


The tables in the nodes

A	B	C	D	E
	C0	B0 - D0	C0	
C0		A0 - D1	C1	

VC-based call setup: first

The other VC from A to D...

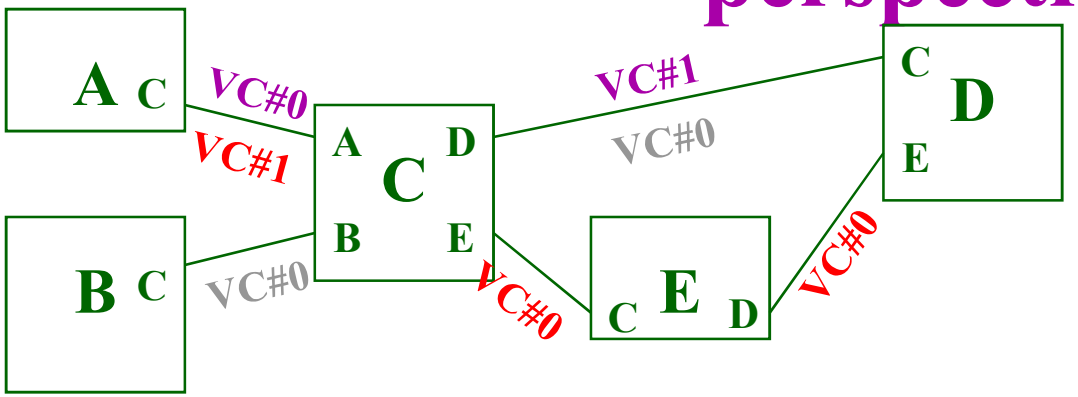


The tables in the nodes

A	B	C	D	E
	C0	B0 - D0	C0	
C0		A0 - D1	C1	
C1		A1 - E0	E0	C0 - D0

VC-based call setup: second

Communication from the C perspective



C		
B0	-	D0
A0	-	D1
A1	-	E0

- If C receives a VC#0 packet from A: it changes the virtual circuit identifier to VC#1 and forwards it to D (according to row 2 of its table)
- If C receives a packet marked #1 from A: it changes the channel identifier to VC#0 and forwards it to E
- If C receives a packet from B with #0: forwards it to D with VC#0

Dismantling the channel: deleting the table entries

VC-based communication

VC-based communication

- **Could there be another technique?**
  - **For example:**
    - **There are no tables on the nodes (but the channel addresses are known!)**
    - **During call setup, the source collects the channel addresses of the route and places them in the address of each packet.**

    eg VC#0 A to D: A-C-D

    VC#1 A to D: A-C-E-D
    - **The router thus knows directly from the address of each incoming packet which channel it should be forwarded on (this is source routing).**

Datagram-based subnet organization

- **Each packet contains a complete destination and source address.**

  Address = host address + NSAP address

  (the station address can be network + host address)

- **Each packet travels independently of the other,**

  ⇒ each packet has its own traffic control

  – **(they can go on different paths)**

Comparison of virtual circuit and datagram-based subnet organization

	Virtual circuit	Datagram
Circuit establishment	Required	Not possible (independent)
Addressing	Each packet contains only a short VC address	Each packet contains a full address (overhead)
State information	Requires a table entry for each open VC in all involved nodes	The subnet is stateless
Traffic control	Only at circuit establishment time	New for each packet
Congestion control	Easy: buffers can be pre-allocated for a known number of VCs	Difficult
Effect of node failures	All VCs passing through the node are dropped	None, at most for some packets
Composition	In the network layer	In the transport layer
Typically suitable for	Connection-based service	Connectionless service

But they are both suitable for providing both connection-based and connectionless services.

Network layer functions

- **Traffic control**
  - **delivery the package to its destination.**
  - **you need to know the topology**
  - **load sharing (alternative routes)**
- **Congestion control**
  - **Do not overload parts of the network**
  - **It is similar to flow control, but it applies not only between two points (transmitter and receiver), but to the entire network.**
- **Inter-network cooperation**
  - **This is the first layer where different networks can be connected (forming heterogeneous networks)**

Traffic control

- **The routing algorithm decides which outgoing line to forward the incoming packet on.**
  - **In case of a datagram-based subnet: for each packet separately,**
  - **In case of virtual circuit based subnet: only when creating the new virtual circuit (call setup)**
    ⇒ session routing

Basic requirements, design considerations

- **Simplicity, reliability**
- **Correctness (do what you have to do**
  → in one copy, to the address provided)
- **Robustness: remain functional (at least to some extent)**
  even in the event of failure
- **Adaptivity: adaptive if it is able to recover**
  independently and adapt to current circumstances
- **Stability:**
  reach a stable state within a finite time from start
- **Optimality:**
  e.g. in terms of cost, delay, min. hop count

Steps for choosing a route

- **Decisions: the (router) nodes decide where to forward the received packet.**
- **Information gathering: obtaining information needed for decisions.**
  For example: creating tables that assign forwarding directions to destination addresses.

Traffic management methods

Hierarchical traffic control:

- **If all destination addresses are collected in a table**
  → too large tables (too many service communications)
  (in the case of large networks)
- **Solution: create a hierarchical network**
  → the entire network is broken down into subnetworks, sub-subnets, etc.
- **Subnetting considerations:**
  - **geographical location;**
  - **functional cohesion (e.g. common goal);**
  - **according to physical medium boundaries, data connection protocols**

  ⇒ An attempt should be made to distribute evenly
- **It is enough for the table to contain the access directions of each subnet and only the directions with the destination address of its own subnet.**

For example: In the address:

: Address

Traffic control decision methods

They can be:

- **One-way or multi-way;**
- **Table-based or table-free methods.**

One-way traffic control

- **Stores a forwarding direction for each address (table)**
- **Advantage:**
  - **simplicity**
  - **can be optimal (if the stored directions are optimal)**
- **Disadvantage:**
  - **not robust (not fault-tolerant).**

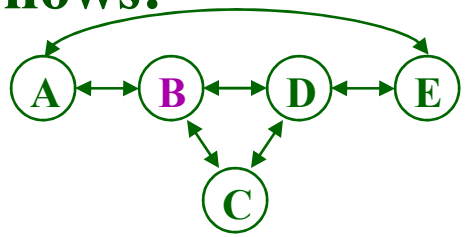
Multi-way traffic control

- **There are multiple weighted forwarding directions for each address, which you can choose from, for example, by weighted drawing.**

For example, a table entry for B is as follows:

E	**A: 0.75; C: 0.2; D: 0.05**

address access 3 possible directions, 3 weights



Multi-way traffic control

- **Criteria for choosing between directions (drawing weights):**
  - **predefined (fixed) weights;**
  - **priority (determined by the priority of the packets);**
  - **type of communication: according to traffic classes (e.g. requires fast response or high bandwidth)**
  - **according to the size of local lines (load on outgoing lines) (load sharing)**
- **Preference:**
  - **suitable for taking into account multiple aspects;**
  - **robust;**
  - **adaptive.**
- **Disadvantage:**
  - **More complicated (requires more processing time).**

Tableless methods

The “hot potato” method:

- We will forward the package to the place with the shortest queue (the sooner you can get rid of it)
- Preference:
  - No need to collect information,
  - simple, robust.
- Disadvantage:
  - Poor line utilization,
  - the delay time is not limited.

Tableless methods

The "flooding" method:

- **forwards all packets in all directions except where they came from.**
  ⇒ Resulting in a large number of duplicate packets.
- **Braking mechanisms:**
  - **With jump counting** (field in packet header, set by every node)
    - **After a certain number of hops (at least the max. diameter of the network), all nodes discard them.**
  - **Package numbering :**
    - **The sender serializes the packages.**
    - **If a node receives a packet from the same sender with the same sequence number as one it has already received (and the timing has not yet expired) ⇒ it discards it as a duplicate**
  - **Selective flooding :**
    - **Based on the topology, it makes traffic routing decisions in advance (roughly - therefore it can be said to be tableless) → and accordingly floods.**

Tableless methods

The “flooding” method:

- **Preference:**
  - **Simple, robust,**
  - **optimal delay.**
- **Disadvantage:**
  - **Poor line utilization.**

The “random walk” method:

- **Forwards the incoming packet in a random direction**
- **Poor line utilization, but simple and robust**

Information collection methods

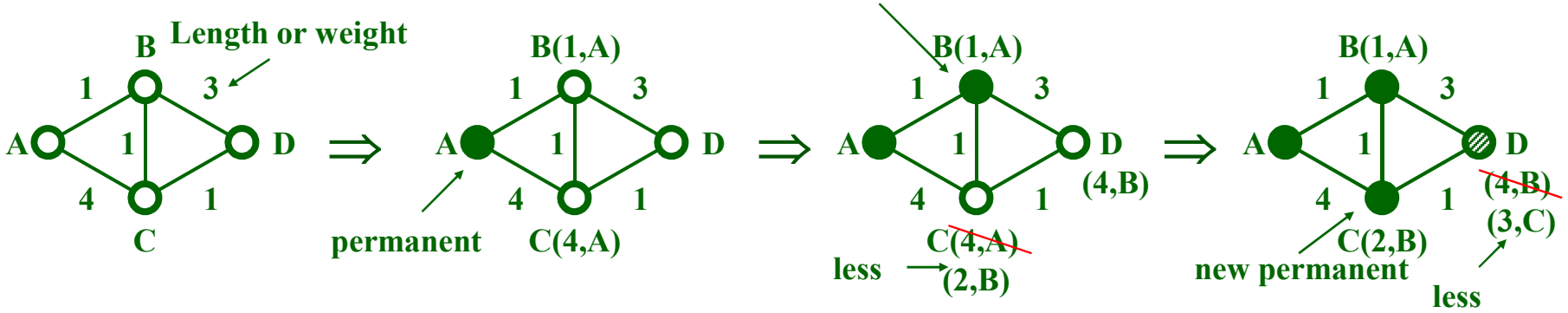
- **They can be:**
  - **Static or dynamic ;**
  - **Centralized or distributed methods**

- **Static traffic control**
  - **The network operator fills in the node tables** (e.g. X.25 public packet switched network)
  - **Input parameters (for the person filling in)**
    - **the topology and**
    - **other considerations (e.g. cost, delay, shortest path, etc.)**

Static traffic control

- **E.g. Shortest path algorithm (Dijkstra 1959):**
  - **We make the starting point permanent → (1)**
  - **(1) Temporarily label neighboring nodes – with the length of the route and the address of the permanent station**
  - **We make the shortest path length label permanent → (1)**

Eg: A → D shortest path



The goal has become permanent → done

Adaptive centralized traffic control

- **How it works:**
  - **center (high capacity) collects all information about the nodes (topology, traffic directions, load)**
  - **from this it calculates the optimal paths and**
  - **downloads them to the nodes' tables.**
- **Advantage: adaptive and optimal.**
- **Disadvantages:**
  - **Vulnerable → unprotected against control panel errors → in case of an error, it may lose its adaptability and optimality.**
  - **The roads leading to the center can be overloaded with information gathering and sign download data.**
  - **A large amount of service information.**
  - **Can be unstable at times (due to delays)**

Adaptive isolated methods

- (Dynamic, distributed information gathering)
- These include tableless methods and
- the “backward learning” method

Backward learning

- At the beginning, no one knows anything ⇒ flooding
- Each packet has a hop counter that is incremented by each node it passes through.

⇒ Backward learning

- If a station receives a packet on a line with hop number j, it knows that the station with the sender address is at most j steps away in the receiving direction.
- It collects the received data in its table ("learns"),
  - determines which station can be reached in which direction with the fewest number of hops
- Sometimes you have to "forget" old values (to adapt to changes)
- Advantages:
  - does not require service communication, adaptive, robust.
- Disadvantages:
  - there is unnecessary communication (initial and then occasional flooding),
  - computationally intensive,
  - does not ensure optimality (forgetting, flooding) (e.g. bridge).
- Note: backward learning is selective flooding

(flows along the generally good directions)

Distributed traffic control

Operation:

- **Neighbors periodically share their current tables** (their knowledge about the network) with each other
- **The tables contain**
  - **the directions for achieving each goal**
  - **the estimated value of the distance (hop count or reach time) to the target**
- **The station receiving the tables adds the estimated distance of the sender of the table to the distance values and updates its own table based on this.**
- **Preference:**
  - **can be close to optimum, (e.g. IP routing)**
  - **adaptive, robust.**
- **Disadvantage:**
  - **service communication**
  - **Computational processing**

Distributed traffic control: Distance vector based traffic control (Bellman-Ford 1957, 1962)

A , who receives tables from *B* and *C* :

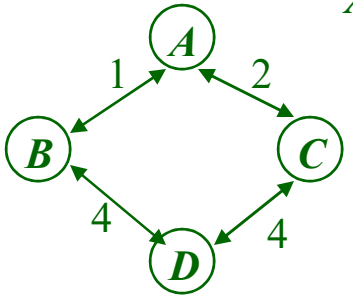


Table *B*	
A	A:1
C	A:3
D	D:4

Table *C*	
A	A:2
B	A:3
D	D:4

"Corrects" the received tables with the distance from the sender: +1 +2

Modified	
B	B:1
C	B : 4
D	B : 5

Modified	
C	C : 2
B	C : 5
D	C : 6

A selects the best:

A creates a new table from the best:

Table *A*	
B	B:1
C	C:2
D	B:5

Bellman-Ford algorithm: count-to-infinity

The count-to-infinity problem :

- **The good news (*A* recovered) quickly,**
- **bad news (*A* is broken) spreads slowly**
  (the speed of propagation depends on the representation of ∞ e.g. 16)
  - **The problem is that a node cannot decide whether it is on a path proposed by another node.**

A recovered

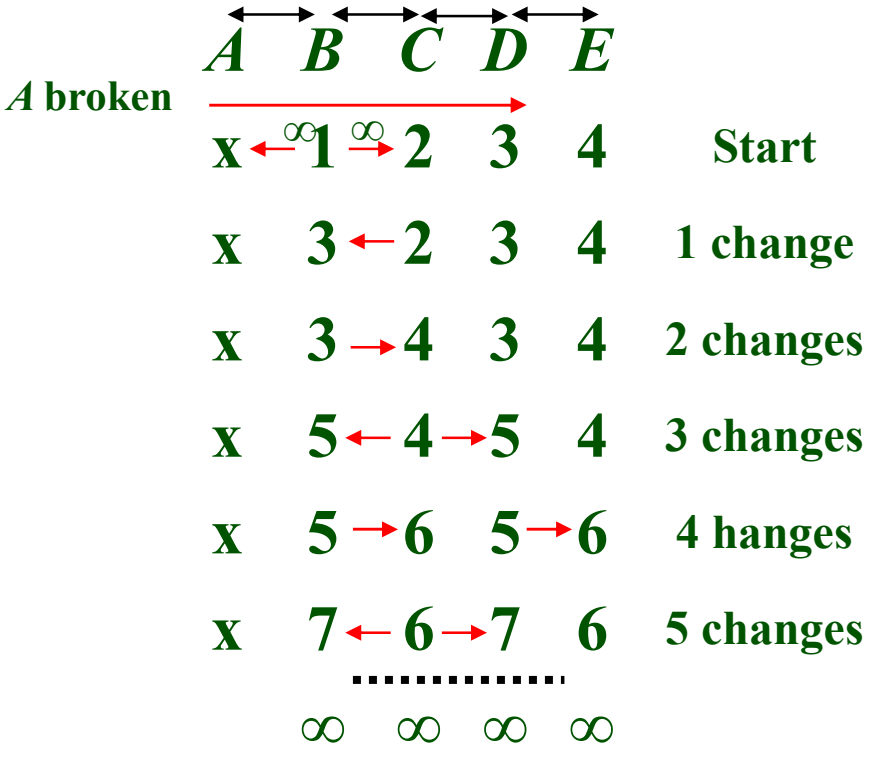
A	*B*	*C*	*D*	*E*	
0	∞	∞	∞	∞	Start
0 → 1	∞	∞	∞	1 change
0	1 → 2	∞	∞	2 changes
0	1	2 → 3	∞	3 changes
0	1	2	3 → 4	4 changes

A broken

A	*B*	*C*	*D*	*E*	
x	1	2	3	4	Start
x	3 ← 2	3	4	1 change
x	3 → 4	3	4	2 changes
x	5 ← 4 → 5	4	3 changes
x	5 → 6	5 → 6	4 changes
x	7 ← 6 → 7	6	5 changes
∞	∞	∞	∞	∞	

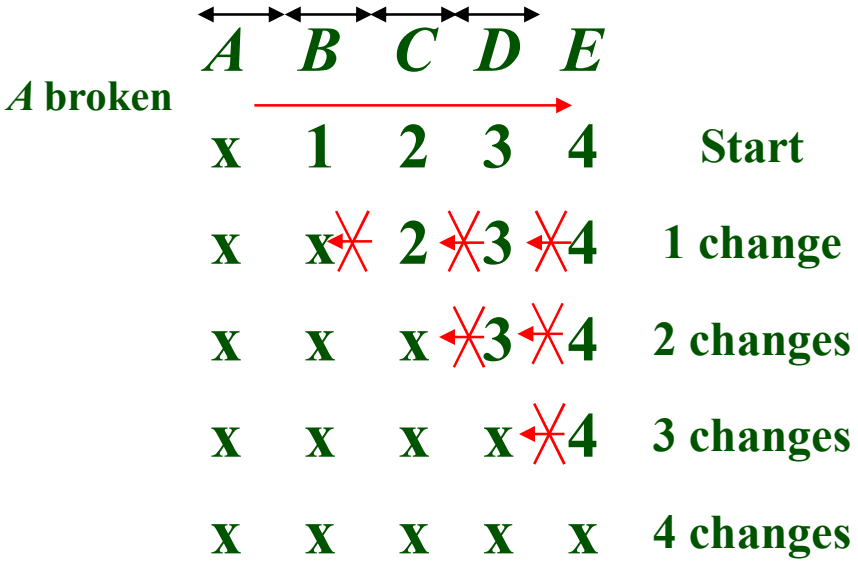
Route poisoning: route poisoning

- **If a network is unreachable, the router changes the rate for that route to infinity and advertises it.**



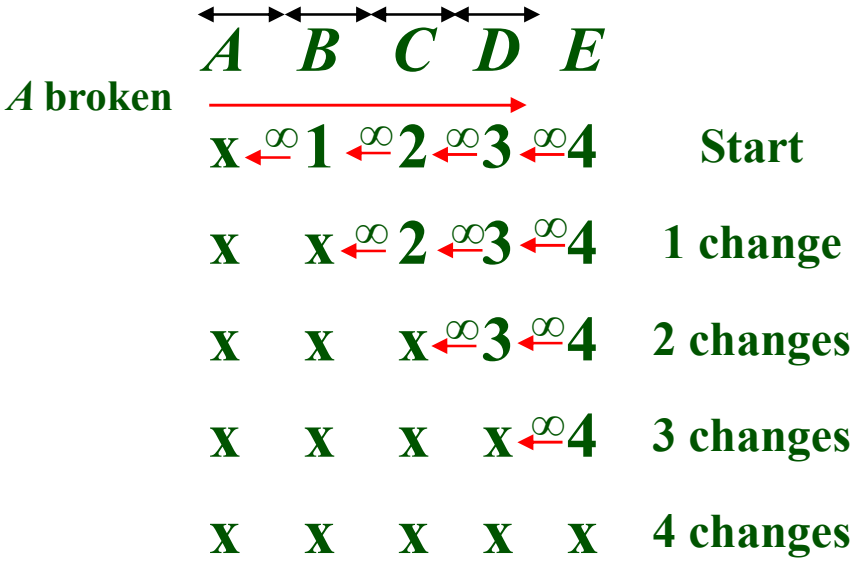
Split horizon: split horizon

- **Prevents it from advertising a route on the interface on which it learned it.**



Split-horizon routing with poison reverse

- **It advertises infinite distance on the interface where it learned about the route.**



Retention timer

- They prevent an update for a route that has been stopped from being accepted if the update arrives within a specified time interval after the stoppage and contains a larger metric than before.
- If the original route is restored before the timer expires, or route information is received that has a smaller metric, it will start using it immediately.

Broadcasting

- **Sends a message to all stations.**
  - **E.g.: updating shared databases, scheduling calls, etc.**

Possible implementations:

- **The source sends a separate package to everyone. (a list of those participating in the broadcasting is required)**
- **Multi-destination routing.**

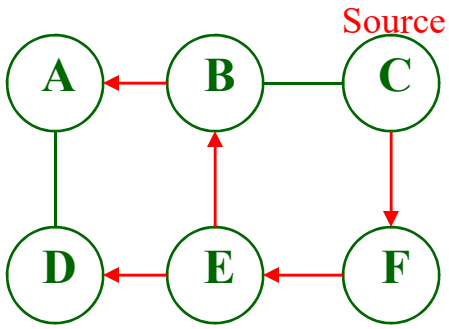
⇒ Multi-Destination Routing

- **The package contains all the target addresses** (in list or bitmap form)
- **The node examines the "all destination addresses" structure to determine the outgoing line(s).**
- **It creates a new packet for each outgoing line, creates a new "all destination addresses" structure in it, and sends these.**

Solutions using a Sink Tree

- **Sink tree: a set of optimal paths that lead from all sources to a given destination.**
- **The nodes know the source's sink tree and forward the packets along it.**
- **Note:** If the optimal path C $\rightarrow$ A includes B, then the path B $\rightarrow$ A is also optimal.
- **Note: the Sink Tree is also a Spanning Tree**
  Spanning Tree: a set of paths in a network that do not contain loops (simply connected),
  but touch all nodes in the network

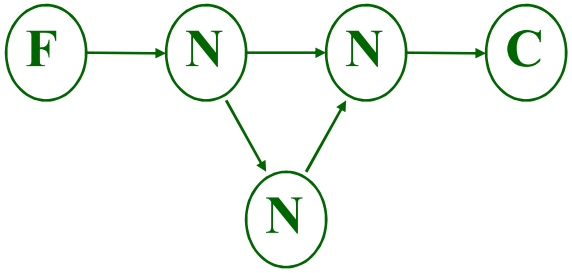
Example of C's Sink Tree



- C sends to F
- F sends to E
- E sends to B and D
- B sends it to A

On the C's Sink Tree: if A → C is an optimal path, and B is on the optimal path, then B → C is also optimal

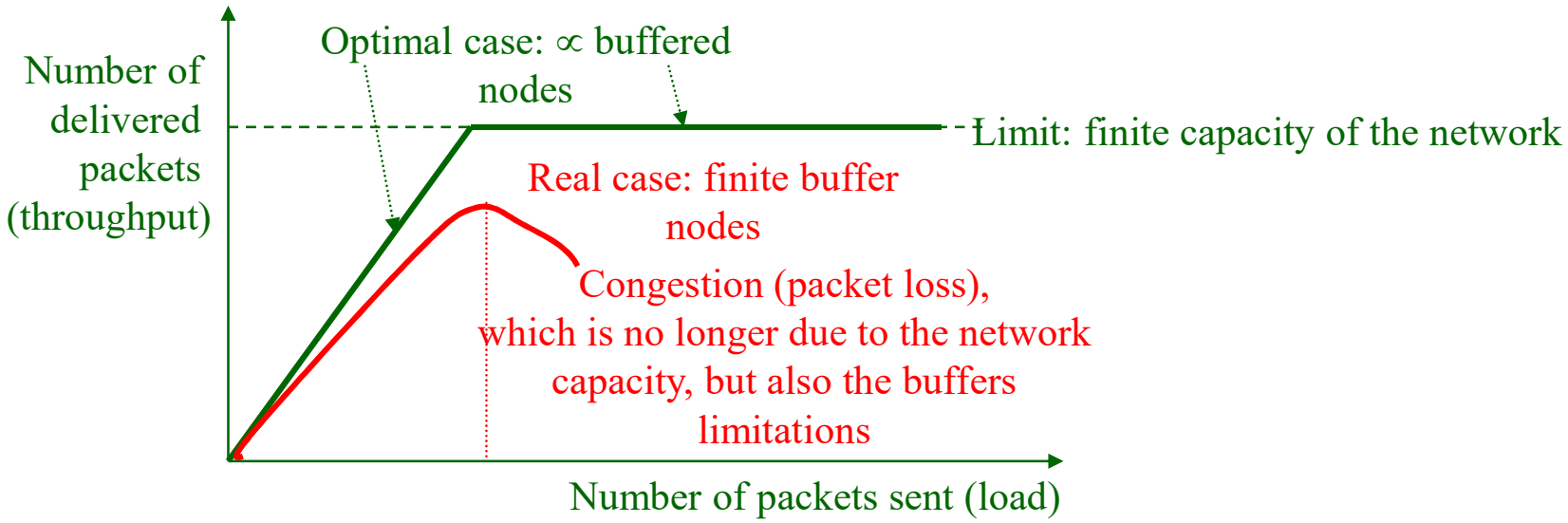
Network layer functions



- Traffic control
  - the package.
  - you need to know the topology
  - load sharing (alternative routes)
- **Congestion control**
  - **Do not overload parts of the network**
  - **It is similar to traffic flow, but it applies not only between two points (transmitter and receiver), but to the entire network.**
- Inter-network cooperation
  - This is the first layer where different networks can be connected (forming heterogeneous networks)

The purpose of congestion control

- **Prevent and/or resolve situations in which a connection or node becomes overloaded.**



Congestion control algorithms

- **Pre-allocating buffers**
- **Package dropping**
  (with different disposal considerations)
- **Choke packet method**
- **Isarithmic congestion control**
- **Traffic congestion control**

Pre-allocating buffers

- **Reserves a buffer in advance for each packet to be transmitted**
- **It can be used in virtual circuit-based networks, it allocates buffer space to the virtual circuit during call setup - the call setup package not only generates table entries, but also allocates buffers.**
- **Rejection is possible if the requested resource is not available. E.g., it reserves a buffer corresponding to the transmission window size.**
- **Disadvantage: not economical (occupies unnecessary buffer capacity) (and may even reject other call structures because of this). Solution e.g.: frees buffers that have been idle for a long time. (this can be risky)**

Package dropping

Uses buffers in a First-Come-First-Served manner ⇒ when they are full, discards new ones

(No buffers are reserved, but some are on each line.)
(Other protocols may provide retransmission.)

Changes:

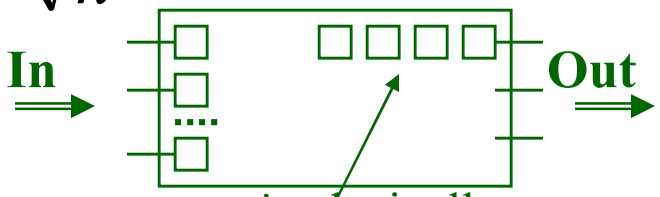
- **At least 1 buffer must be reserved for the inputs and left free (do not become deaf).**
  E.g.: service messages: if other, discard, if special message ⇒ process ⇒ it will not become "deaf".

⇒ Package dropping

Modifications:

- **Allocation of buffers between outputs**
  - **limit the maximum lengths of output buffer lines**
  - **at the same time, it also requires a minimum buffer number for the outputs** (against "starvation")
  - **The maximum buffer count depends on traffic.**
  - **Ex: “Rule of thumb” for output buffers max : where**
    - **p: total number of buffers;**
    - **k: the number of outputs;**
    - **m: the max length for an output.**

$$m = \frac{p}{\sqrt{k}}$$



one can't take it all
(there wouldn't be anything left for the others)

Package dropping

Dropping considerations (priorities)

- **e.g. by priority classes, or**
- **e.g. they look at the number of hops and discard the one that traveled the least**
  (it probably requires less resources to resend)

Choke - packet method

Choke packets :

- **Throttling resources (even before congestion occurs)**
- **A node monitors the saturation of its outgoing lines and if it exceeds a threshold value ⇒ sends a choke packet to the sender**
  (reduce traffic in this direction with a report),
  but forwards the original packet.
  You can also mark the original (forwarded) packet:
  ⇒ this packet has already triggered a throttle message
  (subsequent nodes do not need to send it anymore)

Choke - packet method

- **Sender adaptivity:**
  - **reduces its traffic after receiving the first choking packet, then**
  - **(does not further reduce its traffic) for a time period (there may be duplicate chokings - e.g. multiple elements of the same sequence).**
  - **After this, another timing:**
    - **if another choking packet arrives during this time ⇒ it further reduces its traffic,**
    - **if he doesn't arrive ⇒ increases the traffic to the given destination.**

Isarithmic congestion control

- **Limit the number of packets in the network at the same time**
- **Uses permit packages**
  ⇒ can only send if it has received an authorization packet
  ⇒ then generates another authorization packet
- **Authorization packets travel around the network.**
- **Change:**
  Authorization center from which permission can be requested.
  - **it involves service overhead (although not a large one), and**
  - **sensitive to center failure**
- **Problem:**
  Destruction of permiting packages
  (Difficult to manage (replace))
  - **except for the authorization-centric solution.**

Congestion control with flow control

- **Flow control:**
  - **transmitter should not flood receiver (for two stations)**
- **Receiving stations do not apply traffic control based on their capacity, but rather take into account some absolute limitation.**
  ⇒ It can also be installed on the transmitter, so it can be validated directly at the emission.
- **If the limits are appropriate ⇒ it is definitely good**
  → if they are a little larger ⇒ it is possible that congestion will develop at some points in the event of uneven load.
- **Problem: low limits ⇒ high delays.**

Network layer functions

- **Traffic control**
  - **delivering the package to its destination.**
  - **you need to know the topology**
  - **load sharing (alternative routes)**
- **Congestion control**
  - **Do not overload parts of the network**
  - **It is similar to flow control, but it applies not only between two nodes (transmitter and receiver), but to the entire network.**
- **Inter-network cooperation**
  - **This is the first layer where different networks can be connected (forming heterogeneous networks)**