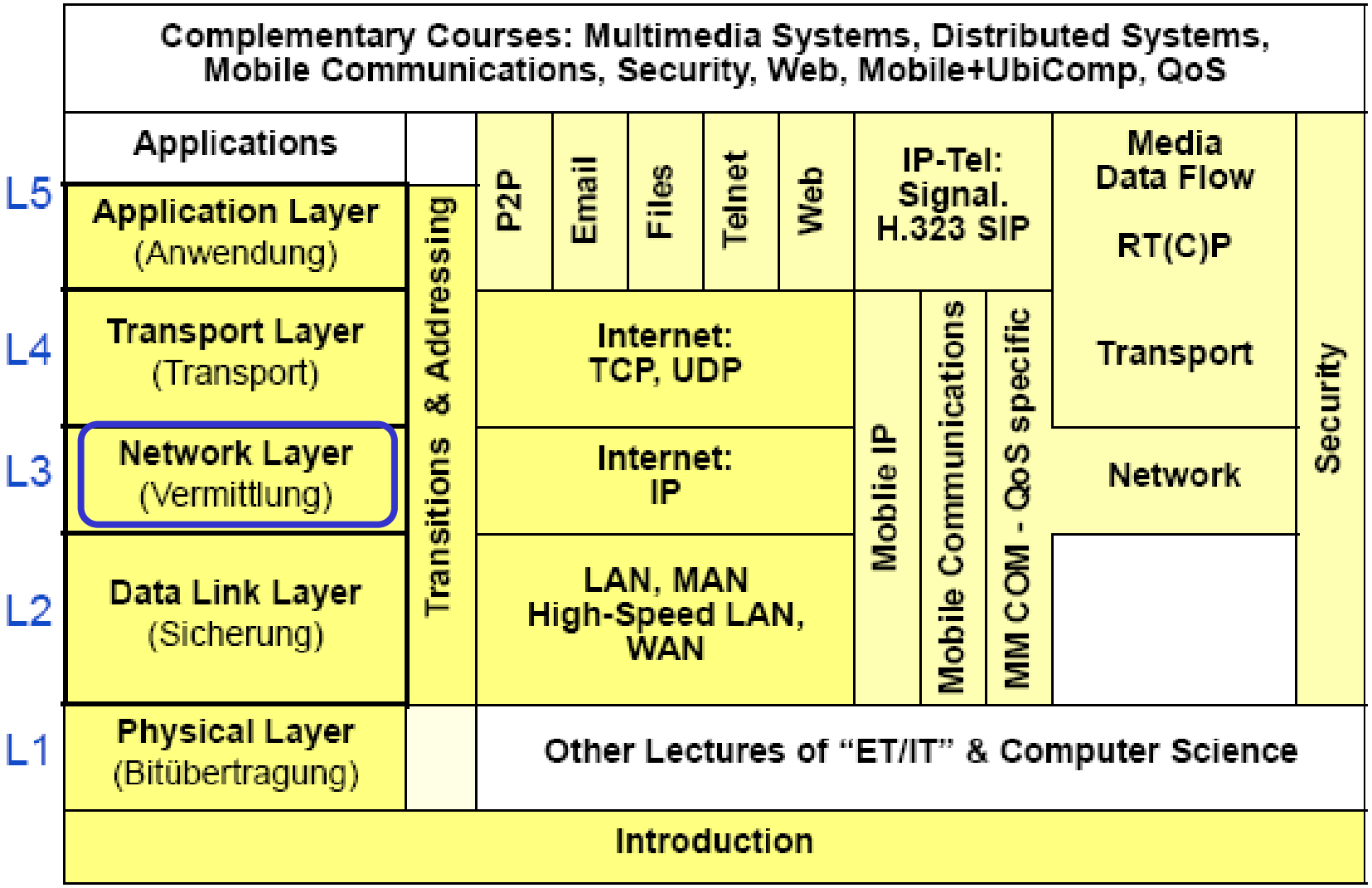
Computer Networks I

Network Layer

Prof. Dr.-Ing. **Lars Wolf**

IBR, TU Braunschweig
Mühlenpfordtstr. 23, D-38106 Braunschweig, Germany,
Email: wolf@ibr.cs.tu-bs.de

Scope


Complementary Courses: Multimedia Systems, Distributed Systems, Mobile Communications, Security, Web, Mobile+UbiComp, QoS
Applications
L5
Application Layer (Anwendung)
L4
Transport Layer (Transport)
L3
Network Layer (Vermittlung)
L2
Data Link Layer (Sicherung)
L1
Physical Layer (Bitübertragung)
Transitions & Addressing
P2P
Email
Files
Telnet
Web
IP-Tel: Signal. H.323 SIP
Media Data Flow
RT(C)P
Internet: TCP, UDP
Transport
Internet: IP
Network
LAN, MAN High-Speed LAN, WAN
Moblie IP
Mobile Communications
MM COM - QoS specific
Security
Other Lectures of "ET/IT" & Computer Science
Introduction

Overview

Overview

4 Routing: Foundations

UNICAST (Point-to-Point) Routing: NON-ADAPTIVE
- 5 Non-Adaptive Shortest Path Routing
- 6 Non-Adaptive Flow-Based Routing (left out due to time constrraints)
- 7 Non-Adaptive Flooding

UNICAST (Point-to-Point) Routing: ADAPTIVE
- 8 Adaptive Centralized Routing
- 9 Adaptive Isolated Routing – Backward Learning
- 10 Adaptive Distributed – Distance-Vector Routing
- 11 Adaptive Distributed – Link State Routing

UNICAST Routing: ENHANCEMENTS

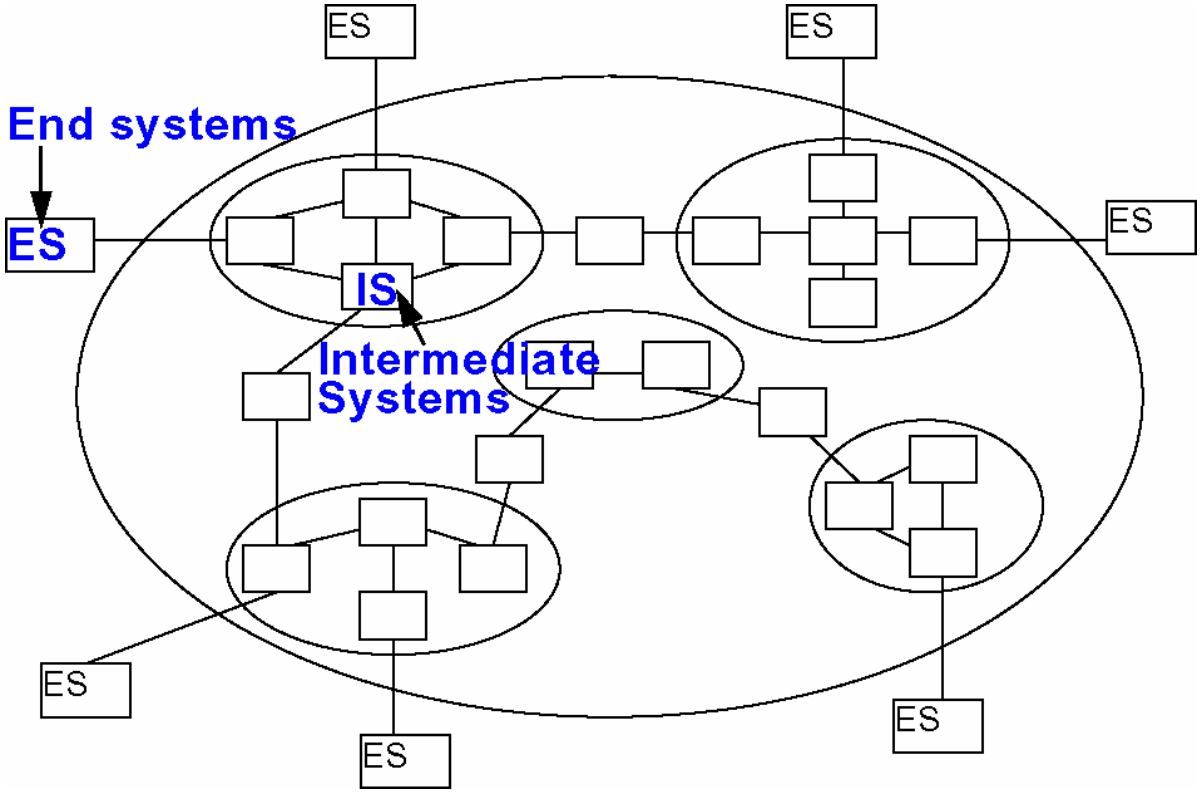
12 Routing: Diverse Enhancements
- 12.1 Multipath Routing
- 12.2 Hierarchical Routing

Overview

In Computer Networks 2:

- 13 Broadcast Routing
  - 13.1 Broadcast Routing: Simple Methods
  - 13.2 Broadcast Routing: Multidestination Routing
  - 13.3 Broadcast Routing: Spanning Tree
  - 13.4 Broadcast Routing: Reverse Path Forwarding (RPF)
  - 13.5 Broadcast Routing: Reverse Path Broadcast (RPB)
- 14 Multicast Routing
  - 14.1 Multicast Routing: Spanning Tree
  - 14.2 Multicast Routing: Core-Based Tree
  - 14.3 Multicast Routing: Truncated Reverse Path Forwarding (TRPB)
  - 14.4 Multicast Routing: Additional Procedures & Topics
- 15 Congestion Control
  - 15.1 Avoidance
  - 15.2 Congestion Correction
- 16 Addressing
  - 16.1 X.121 Addressing
  - 16.2 OSI Addressing
  - 16.3 Internet Addresses (IP)

1 Functions of the Network Layer



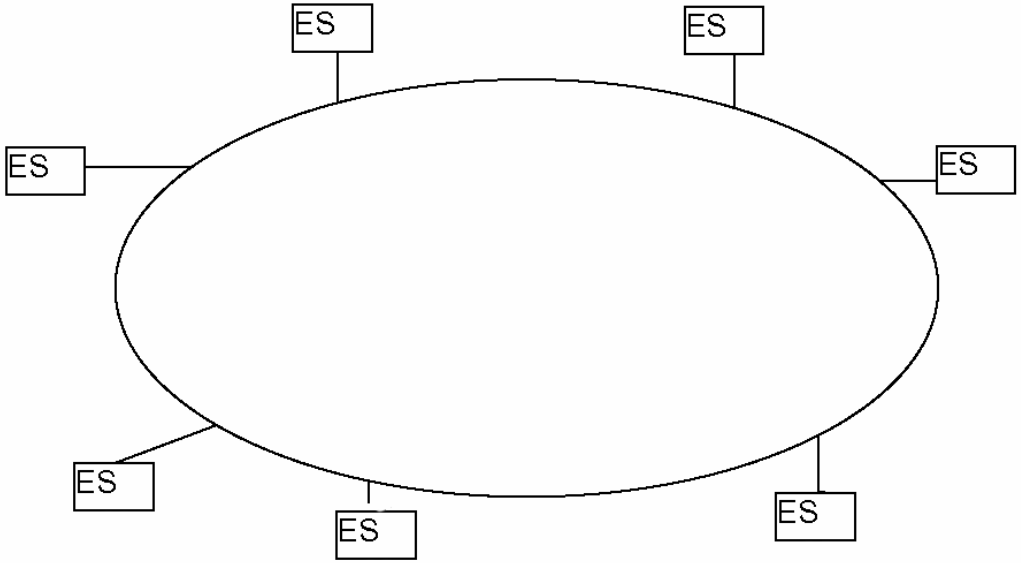
Data transfer from end system to end system

- several hops,
- (heterogeneous) subnetworks
- compensate for differences between end systems during transmission

Relevance of the interface: switching vs. transport service

- L1 up to L1,L2+L3: organization: carrier
- from L4 onward: user/customer/company

Functions of the Network Layer



The provided services are

- standardized for end systems
- independent from network technology
- independent from number, type and topology of the subnetworks

SUBNETWORKS (IS 7498):

A multiple of one or several intermediary systems that

- provide switching functionalities

and

- through which open end systems can establish network connections

Functions of the Network Layer

Primary tasks

- virtual circuits and datagram transmissions
- routing
- congestion control
- Internetworking
  - to provide transitions between networks
- addressing
- Quality of Service (QoS)
  - example: bandwidth, delay, error rate
  - negotiate costs vs. quality of service to be provided

Secondary tasks, based on type service and request:

- multiplexing of network connections
- fragmentation and reassembling
- error detection and correction
- flow control as a means to handle congestion
- maintaining the transmission sequence

Functions of the Network Layer

Required knowledge

- subnetwork topology
- address / localization of the end system
- network status (utilization,...)
- packet / data stream communication requirements (Quality of Service)

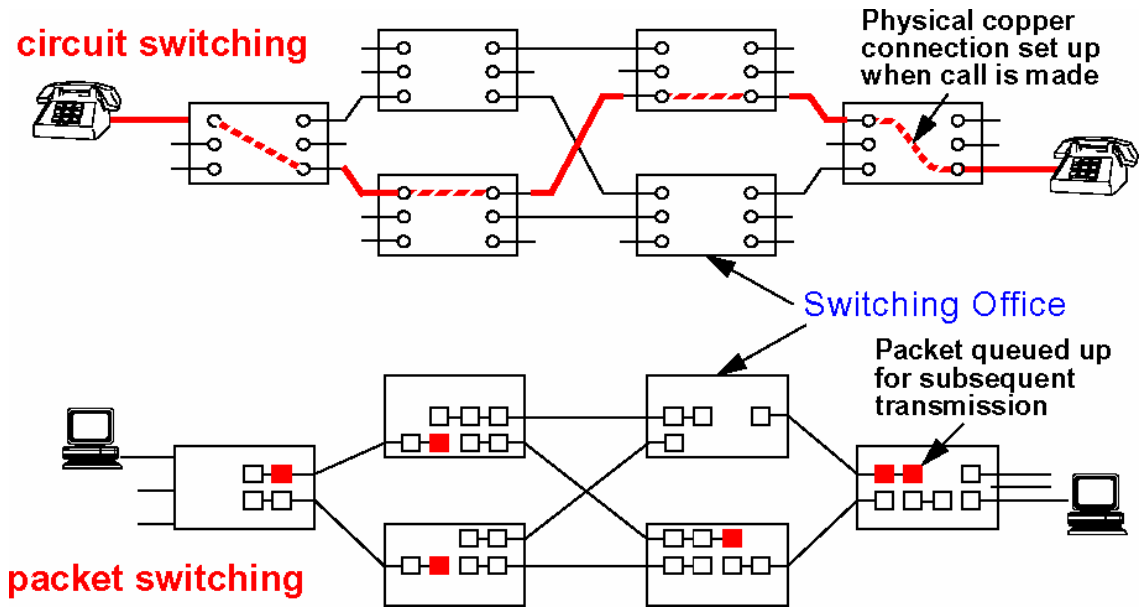
Examples

- X.25 (ISDN, ...)
- Internet protocol IP (TCP/IP,..)

Nomenclature:

Layer	Data Entity
Transport	…
Network	Packet
Data Link	Frame
Physical	Bit/Byte (bit stream)

2 Switching Approaches



Circuit switching
- switching a physical connection

Message switching
- message is stored and passed on by one hop

Packet switching
- store-and-forward, but transmission packets limited in size

Switching by virtual circuit
- packets (or cells) over a pre-defined path

2.1 Circuit Switching

Principle
- dedicated path from src to destination for entire duration of call
  - connections between switching centers (frequency spectrum, dedicated ports)

Implementation examples
- historically: on switching boards
- mechanical positioning of the dialers
- setting coupling points in circuits
- early alternative of Broadband-ISDN: STM (Synchronous Transfer Mode)

Properties
- connection has to be setup before transmission
  - establishing a connection takes time
- fixed allocation of bandwidth → no congestion during transfer
- No processing of data at intermediate nodes
  - constant and short delay
- information delivery is sequenced (by nature)
- resource allocation too rigid (possibly waste of resources)
  - No support for transmission of bursty data → potential resource underutilization
- once connection is established it cannot be blocked anymore

2.2 Message Switching

Principle

- all data to be sent is treated as a "message"
- "store and forward" network:
  in each node the message is handled as follows:
  1. accepted
  2. check and treatment of possible errors
  3. stored, and
  4. forwarded (as a whole to the next node)

Example

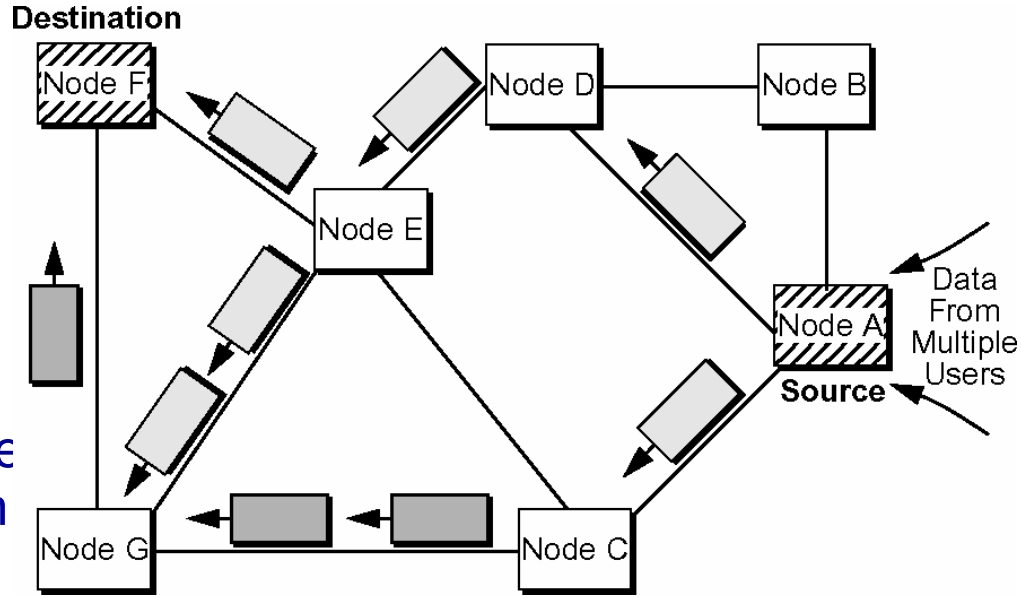
- early telegram service

Properties

- high memory requirements at each node (switching centers),
  - because message may be of any size
  - usually stored on secondary repository (hard disk)
- node may be used to its full capacity over a longer period of time by one message,
  - i. e. better if packets are of limited size (packet switching)

2.3 Packet Switching

Example:
former Datex-P Service, Internet



Principle
- packets of limited size
- dynamic determination of route for every packe
- no dedicated path from source to destination

Properties
- no connect phase
- dynamic allocation of bandwidth
  - suitable for bursty traffic
  - flexible, provides for resource sharing and good utilization
- congestion possible
- bandwidth reservation difficult, QoS provisioning limited
- variable end-to-end delay
  - due to queuing at intermediate nodes (and varying routes)
- information delivery may not be sequenced or reliable

2.4 Virtual Circuit Switching

Principle:

- setup path from source to destination for entire duration of call
- using state information in nodes but no physical connection
- connection setup: defines data path
- messages: as in packet switching
  - follow all ONE path
  - but (may) have only the address of the network entry point
    - not the destination address, e.g., ATM: VPI/VCI

Examples:

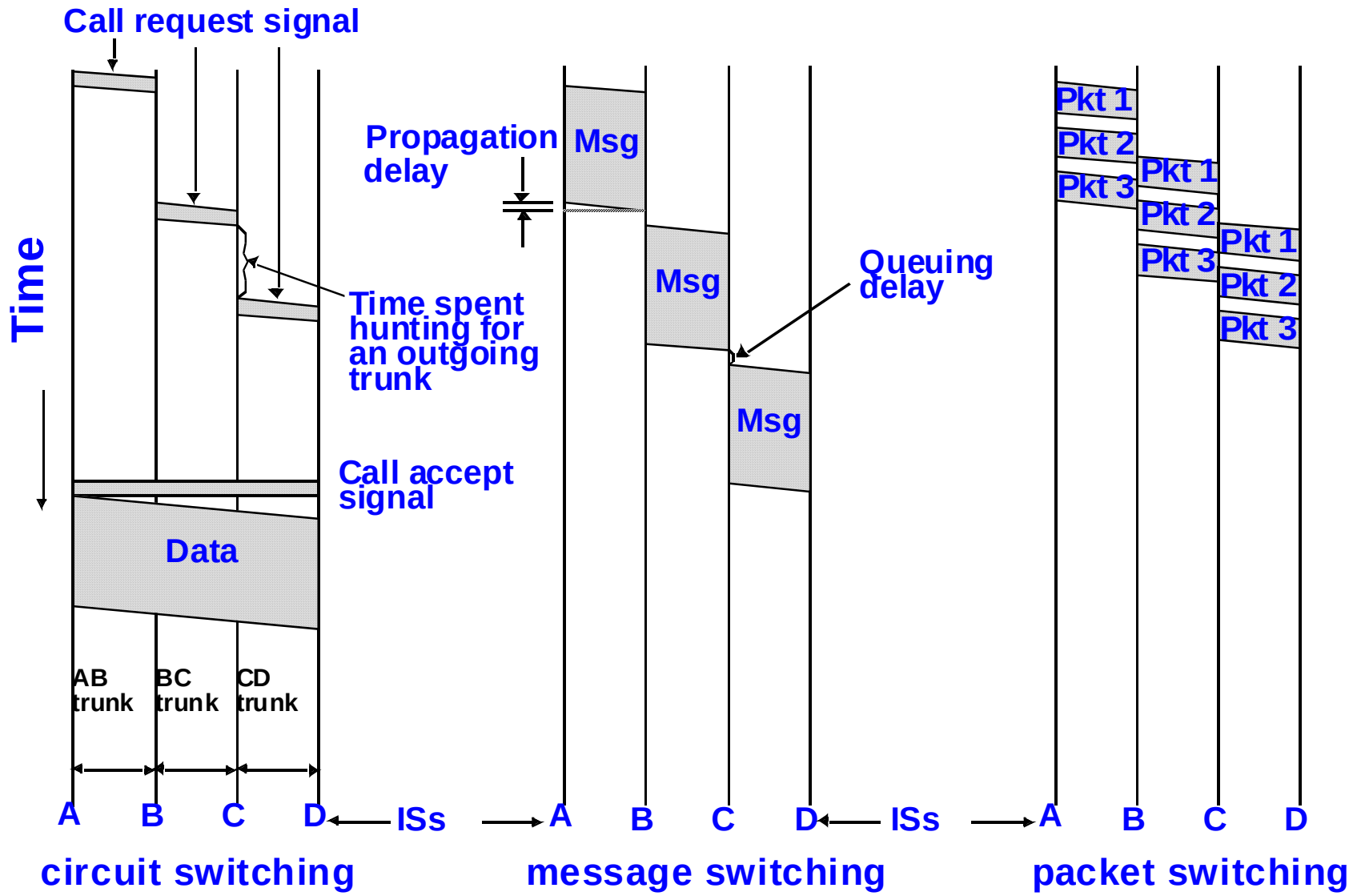
- ATM (Asynchronous Transfer Mode) PVC (permanent virtual circuit)
  - established "manually" (similar to dedicated lines)
- ATM SVC (switched virtual circuit)
  - signaling: connect and disconnect corresponding to telephone netw.
- Internet Integrated Services
  - state established via signaling protocol (RSVP)
  - full addresses are used

Properties

- all messages of a connection are routed over the same pre-defined data path, i.e., sequence is maintained
- it is easier to ensure Quality of Service (see also ATM)

2.5 Comparison: Temporal Performance

Timing of events:



Comparison: Circuit and Packet Switching

Circuit switching:

- connection establishment can take a long time
- bandwidth is reserved
  - no danger of congestion
  - possibly poor bandwidth utilization (bursty traffic)
- continuous transmission time,
  because all data is transmitted over the same path
- price calculation classically based on duration of connection

Packet switching:

- connect phase not (absolutely) necessary
- dynamic allocation of bandwidth
  - danger of congestion
  - optimized bandwidth utilization
- varying transmission time
  - because packets of a connection may use different paths
  - not suitable for isochronous data streams
- price calculation (classically) based on transfer volume

Switching Approaches: Applicability

Circuit switching:

- telephone system
- until now minor usage for computer networks, but various multimedia applications require isochronous data streams

Packet switching:

- used frequently for computer networks
- difficult for voice transmissions but with dominance of Internet (and VoIP) getting importance also here

Message switching:

- seldomly used for computer systems
  - complex storage management (secondary storage)
  - "blockage" because of large messages

Virtual circuit switching

- important for QoS provisioning (perhaps in modified manner)
- integrated services
- voice transmission

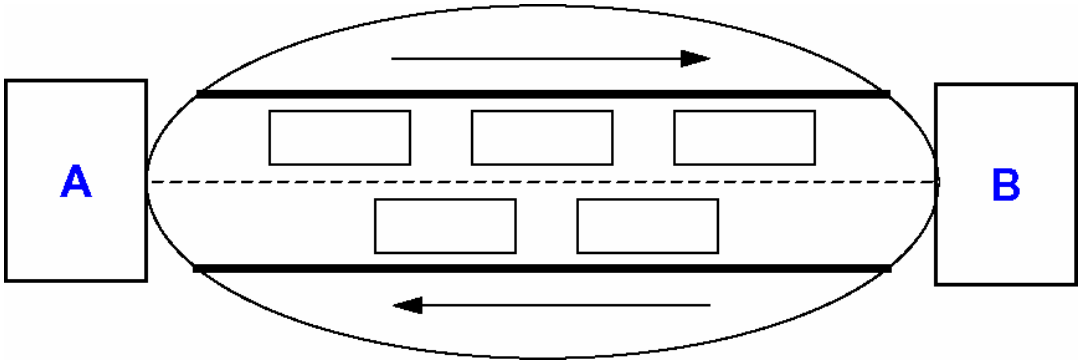
3 Services: Concepts

Concepts
- Connection oriented vs. connectionless communication

Connection oriented
- goal: error free communication channel
- usually error control: L3 (or network)
  - flow control, ...
- usually duplex communication
- more favorable for realtime communications
- typical approach of telephone and telecomm. companies:
  - X.25, ATM, various mobile systems

Connectionless
- unreliable communication
- hardly any error control in L3: left to L4 or higher layers
  - sequence not ensured, ...
- simplex communication
- more favorable for simple data communication:
  - SEND-PACKET, RECEIVE-PACKET
- Internet community: IP

3.1 Service: Connection-Oriented Communication



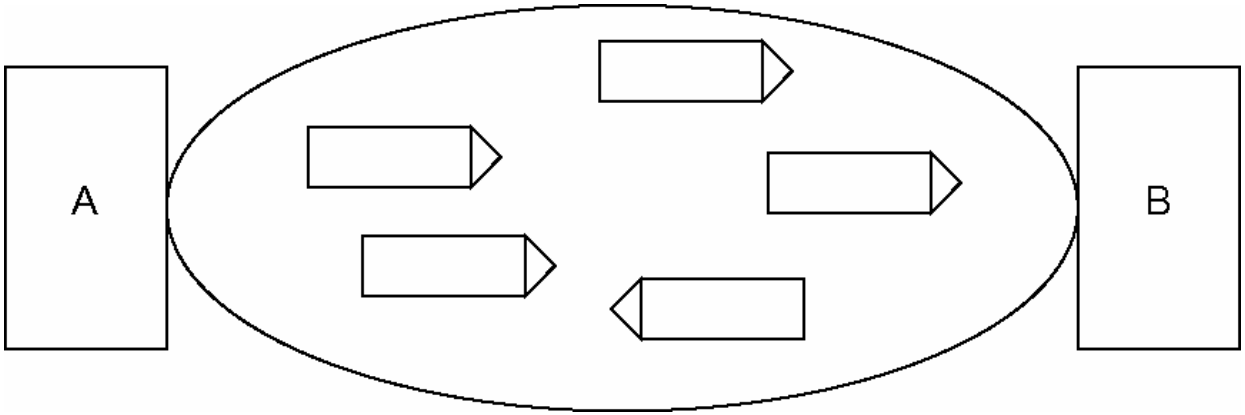
Properties:

- 3-phase interaction
  - connect
  - data transfer
  - disconnect
- (allows for) QUALITY OF SERVICE NEGOTIATION
  - e.g., throughput, error probability, delay
- (typically) RELIABLE COMMUNICATION in both directions
  - no loss, no duplicates, no modification
  - ensures maintenance of the correct sequence of transmitted data
- FLOW CONTROL
- relatively complex protocols

Example:

- telephone service

3.2 Service: Connectionless Communication



Properties:

- network transmits packets as ISOLATED UNITS (datagram)
- UNRELIABLE COMMUNICATION:
  - loss, duplication, modification, sequence errors possible
- no flow control
- comparatively SIMPLE PROTOCOLS

Example:

- mail delivery service

3.3 Services: Comparison of Concepts

Arguments pro connection-oriented service:

- simple, powerful paradigm
- allows for simplification of the upper layers (L4 - L7)
- simplifies task of end systems
- for some applications efficiency in time is more important than error-free transmission
  - e. g. realtime applications, digital voice transmission
- suitable for a wide range of applications

Arguments pro connectionless service:

- high flexibility and low complexity
- avoids high costs for connects and disconnects for transaction-oriented applications
- easier to optimize the network load
- compatibility and costs: IP as common protocol
- "END-TO-END ARGUMENTS" (Saltzer et al.):
  - reliable communication requires error control within the application
  - and: error control in one layer can replace the error control in the layer underneath it.

3.4 Services of Layer 3 and their Implementations

ISO IS 8348 Network Service Definition

2 Service classes:

- Connection-oriented Network Service (CONS)
- Connectionless-mode Network Service (CLNS)

Implementations:

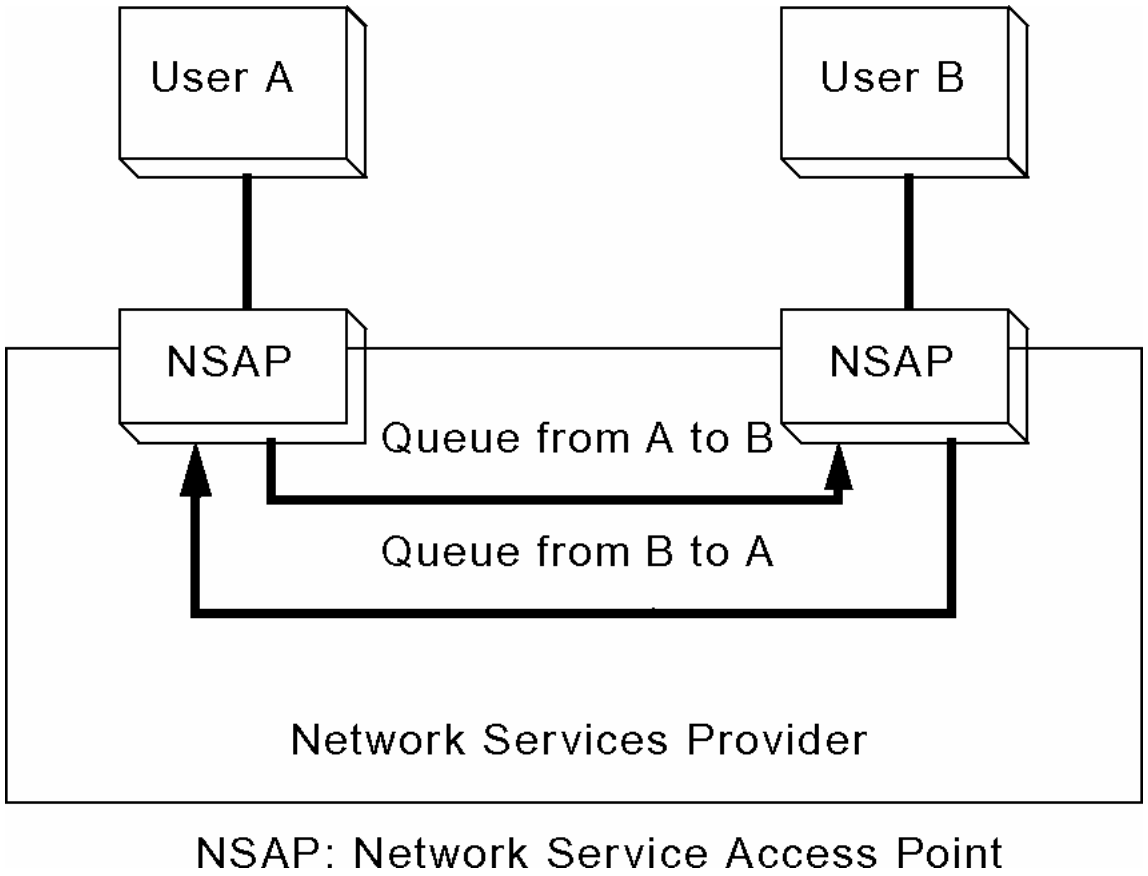
- virtual circuit
- datagram

Comment: service not equal implementation!

Examples for communication architectures:

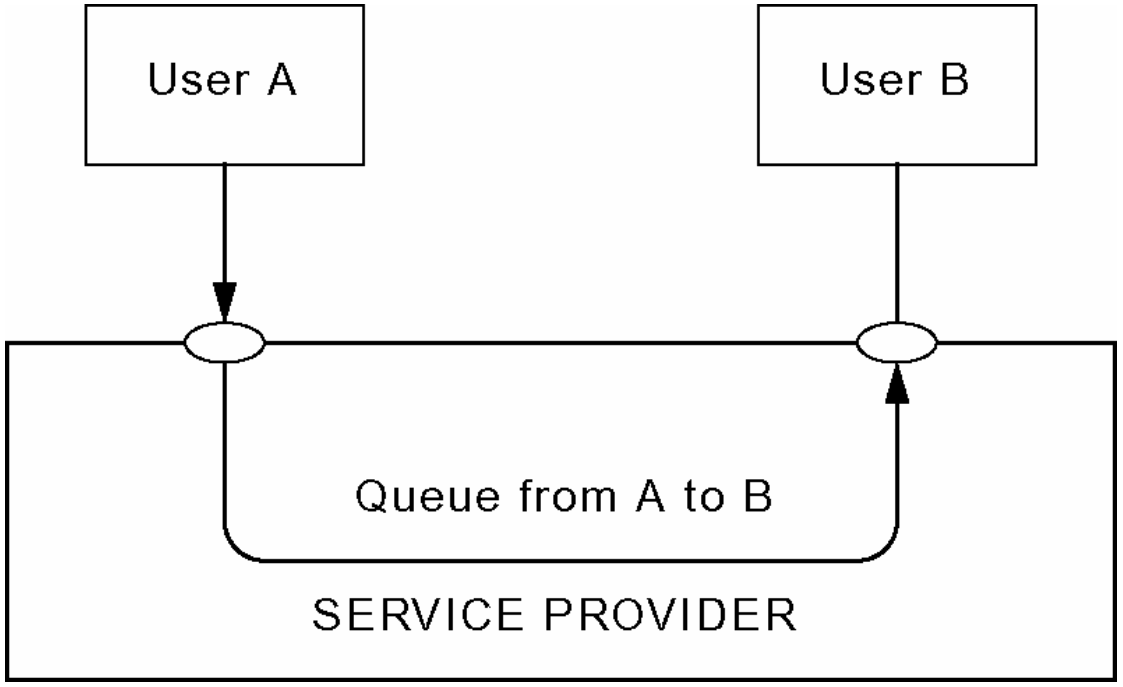
		Service (upper layer/s)	
		connectionless	connection-oriented
L3 Implementation	Datagram	typically: UDP via IP	TCP via IP
	virtual circuit	UDP/IP via ATM	typically: ATM AAL1 via ATM

Service ISO CONS: Model



NSAP: Network Service Access Point

Service ISO CLNS: Model



Service provider can

- delete objects in a queue,
- duplicate objects in a queue and
- change the object sequence within a queue.

Implementation of Virtual Circuit

Connection set-up phase:

- select a path
- Intermediate systems (IS) store path information
- network reserves all resources required for the connection

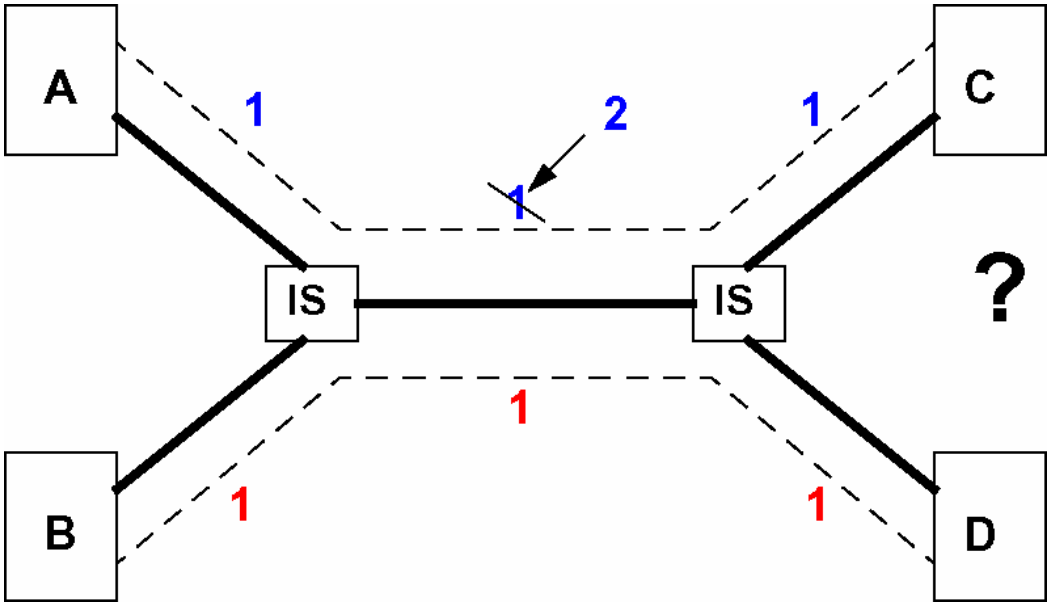
Data transfer phase: all packets follow the selected path

- packet contains VC_number
  - identification of connection, no complete address information
- IS uses the stored path information to determine the successor

Disconnect phase:

- network forgets the path
- releases reserved resources

Implementation of Virtual Circuit



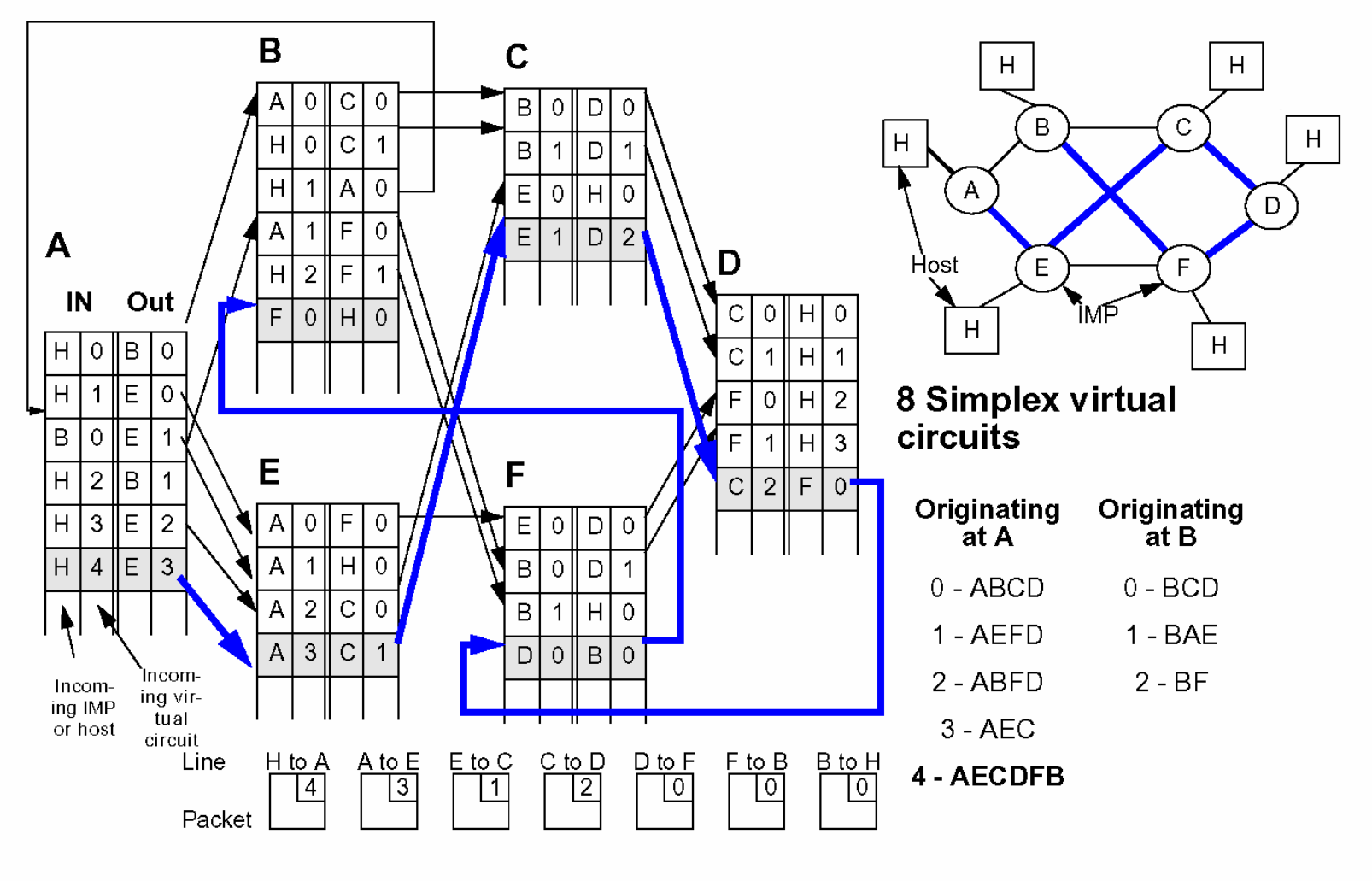
End systems ES assign VC-identifiers (VC-numbers) independently
Problem: the same VC-identifiers may be assigned to different paths

Solution: assign VC-numbers for virtual circuit segments

- IS differentiates between incoming and outgoing VC-number
  1. IS receives incoming VC-number in CONNECT.ind
  2. IS creates outgoing VC-number (unique between IS and successor(IS))
  3. IS sends outgoing VC-number in CONNECT.req

Implementation of Virtual Circuit

Example:



Implementation of Datagram

Every datagram passes through network as isolated unit

- has complete source and destination addresses
- individual route selection for each datagram
- generally no resource reservation
- correct sequence not guaranteed

3.5 Datagram vs. Virtual Circuit: A Comparison

Virtual circuit: destination address defined by connection

+ packets contain short VC-number only
+ low overhead during transfer phase
+ "perfect" channel throughout the net
+ resource reservation: "Quality of Service" guarantees possible

but:

- overhead for connection setup
- memory for VC tables and state information needed in every IS
- sensible to IS and link failures
- resource reservation: potentially poor utilization

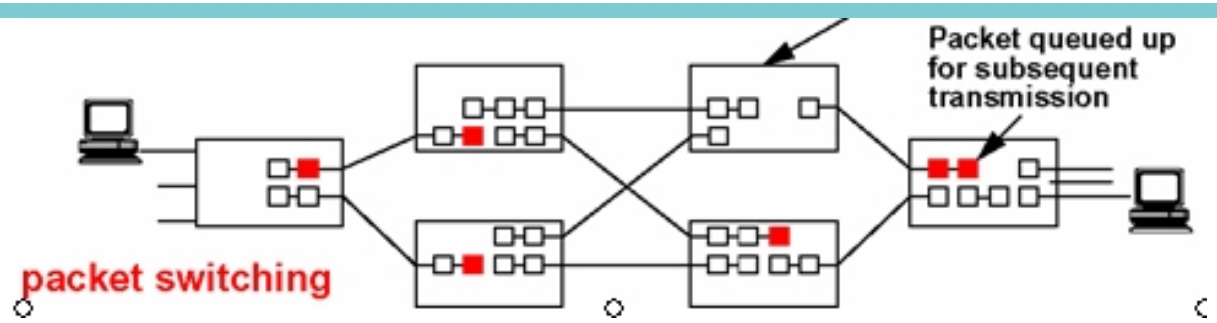
Datagram: IS routing table specifies possible path(s)

+ no connection setup delay
+ less sensible to IS and link failures
+ route selection for each datagram: quick reaction to failures

but:

- each packet contains the full destination and source address
- route selection for each datagram: overhead
- QoS guarantees hardly possible

4 Routing: Foundations



Task:

- to define the route of packets through the network
  - from source
  - to destination

ROUTING ALGORITHM:

- to define on which outgoing line
  an incoming packet will be transmitted on

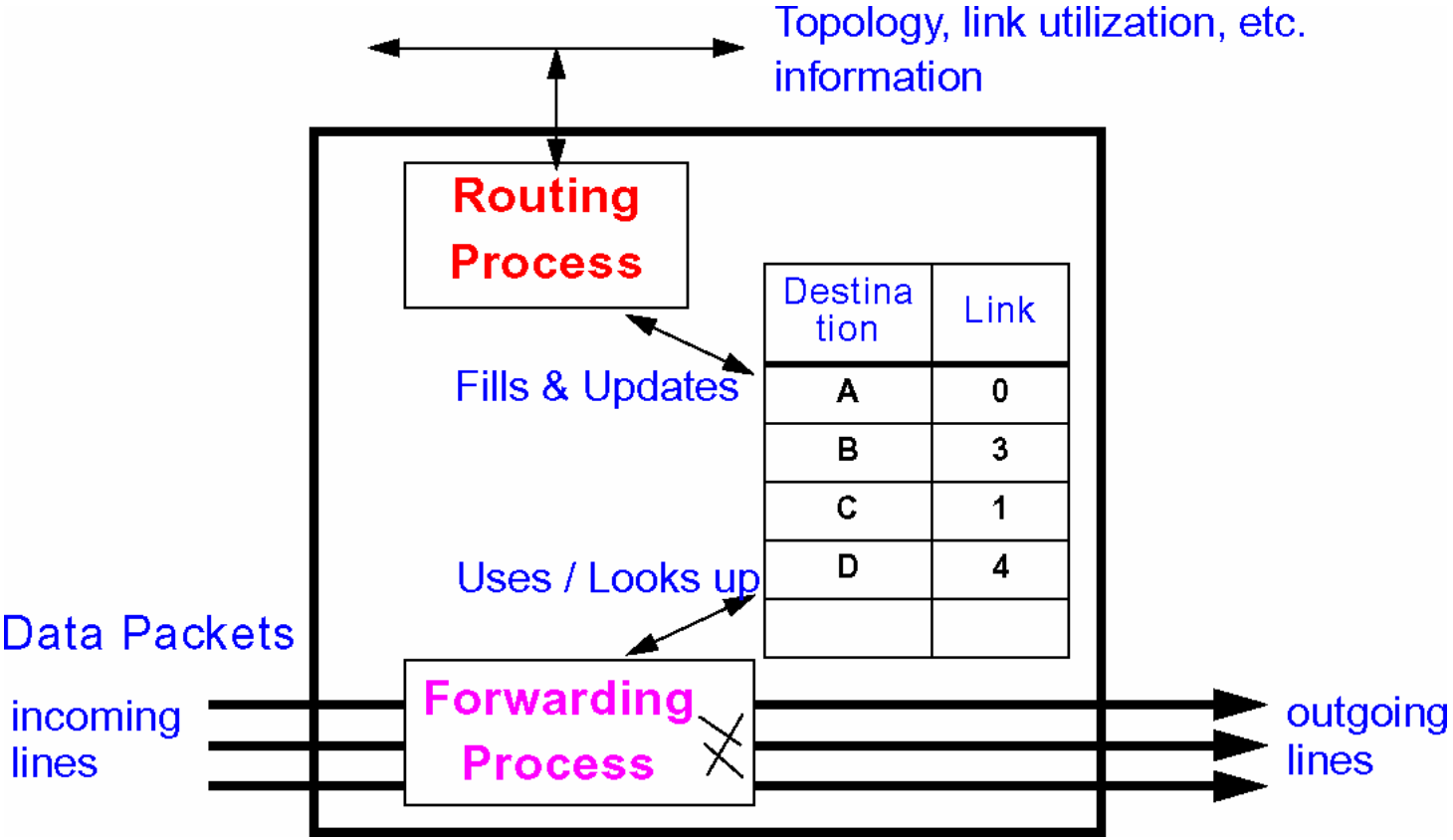
Route determination:

- datagram
  - individual decision for each packet
- virtual circuit
  - one decision for all packets of the same flow
  - routing only during connect (session routing)

Routing & Forwarding

Distinction can be made

- Routing: to take a decision which route to use
- Forwarding: to define what happens when a packet arrives



Desirable Properties of a Routing Algorithm

correctness

simplicity

robustness
- compensation for IS and link failures
- handling of topology and traffic changes

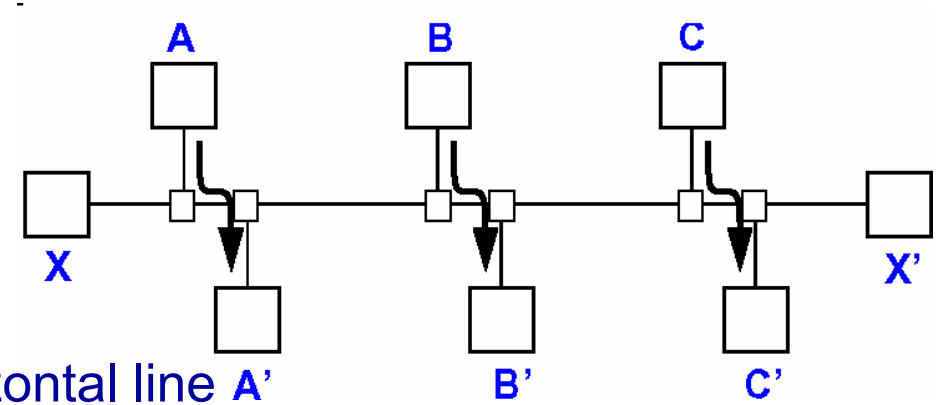
stability
- consistent results
- no volatile adaptations to new conditions

fairness
- among different sources compared to each other

optimality

Routing Algorithms: Conflicting Properties

Often conflicting:
fairness and optimization



Example:

- Communication among
  A → A', B → B', C → C'
  uses full capacity of horizontal line
- optimized throughput, but
- no fairness for X and X'
  - ➔ tradeoff between fairness and optimization

some different optimization criteria

- average packet delay
- total throughput
- individual delay
  - ➔ conflict

therefore often

- hop minimization per packet
  - it tends to reduce delays and decreases required bandwidth
  - also tends to increase throughput

Classes of Routing Algorithms

NON-ADAPTIVE ALGORITHMS
- current network state not taken into consideration
  - assume average values
  - all routes are defined off-line before the network is put into operation
  - no change during operation (static routing)
- WITH knowledge of the overall topology
  - spanning tree
  - flow-based routing
- WITHOUT knowledge of the overall topology
  - flooding

ADAPTIVE ALGORITHMS
- decisions are based on current network state
  - measurements / estimates of the topology and the traffic volume
- further sub-classification into
  - centralized algorithms
  - isolated algorithms
  - distributed algorithms

Enhancements (adaptive and non-adaptive algorithms)
- multiple routing and hierarchical routing definition

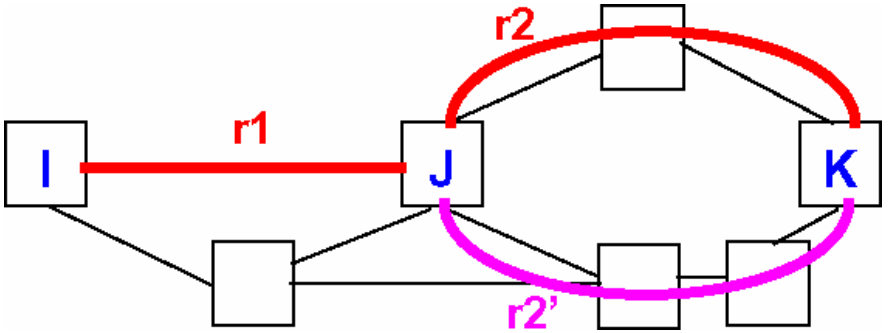
Optimality Principle and Sink Tree

General statement about optimal routes:

if
- ROUTER J IS ON OPTIMAL PATH FROM ROUTER I TO ROUTER K

then
- THE OPTIMAL PATH FROM ROUTER J TO ROUTER K USES THE SAME ROUTE

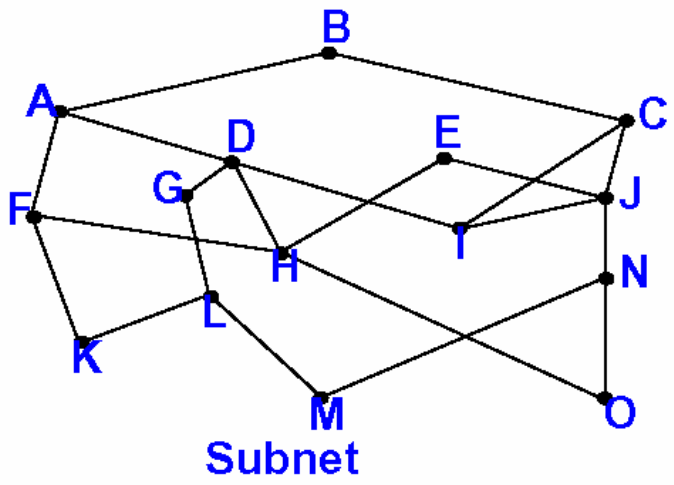
Example:
- r1: route from I to J
- r2: route from J to K
- if better route r2' from J to K would exist
  - then concatenation of r1 and r2' would improve route from I to K (contradiction)



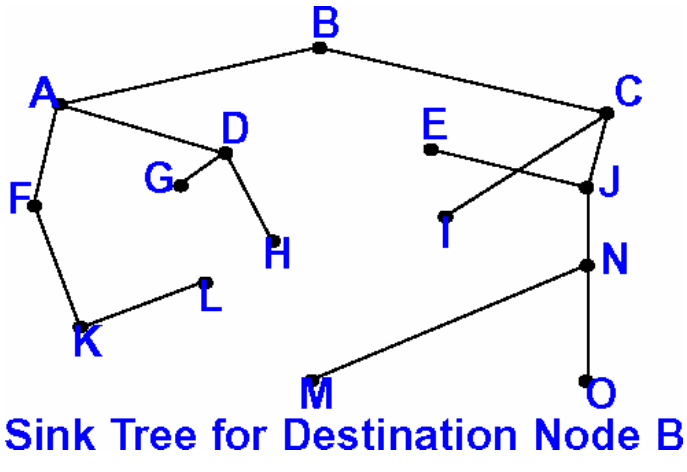
➔ set of optimal routes
- from all sources
- to a given destination

form a tree rooted at the destination: SINK TREE

Sink Tree: Example



Subnet



Sink Tree for Destination Node B

Comments:

- tree: no loops
  - each packet reaches its destination within finite and bounded number of hops
- not necessarily unique
  - other trees with same path lengths may exist

Goal of all routing algorithms

- discover and use the sink trees for all routers

Further comments:

- information about network topology necessary for sink tree computation
  - yet, sink tree provides benchmark for comparison of routing algorithms

Methodology & Metrics

Networks represented as graphs:
- node represents a router
- arc represents a communication line (link)

Compute the SHORTEST PATH between a given pair of routers

Different metrics for path lengths can be used
- can lead to different results
- sometimes even combined
  - (but this leads to computational problems)

Metrics for the "ideal" route, e.g., a "short" route
- number of hops
- geographical distance
- bandwidth
- average data volume
- cost of communication
- delay in queues
- ...

5 Non-Adaptive Shortest Path Routing
Non-Adaptive Routing

Static Procedure

- network operator generates tables
- tables
  - are loaded when IS operation is initiated and
  - will not be changed any more

Characteristics

+ simple
+ good results with relatively consistent topology and traffic

- but:
  - poor performance if traffic volume or topologies change over time

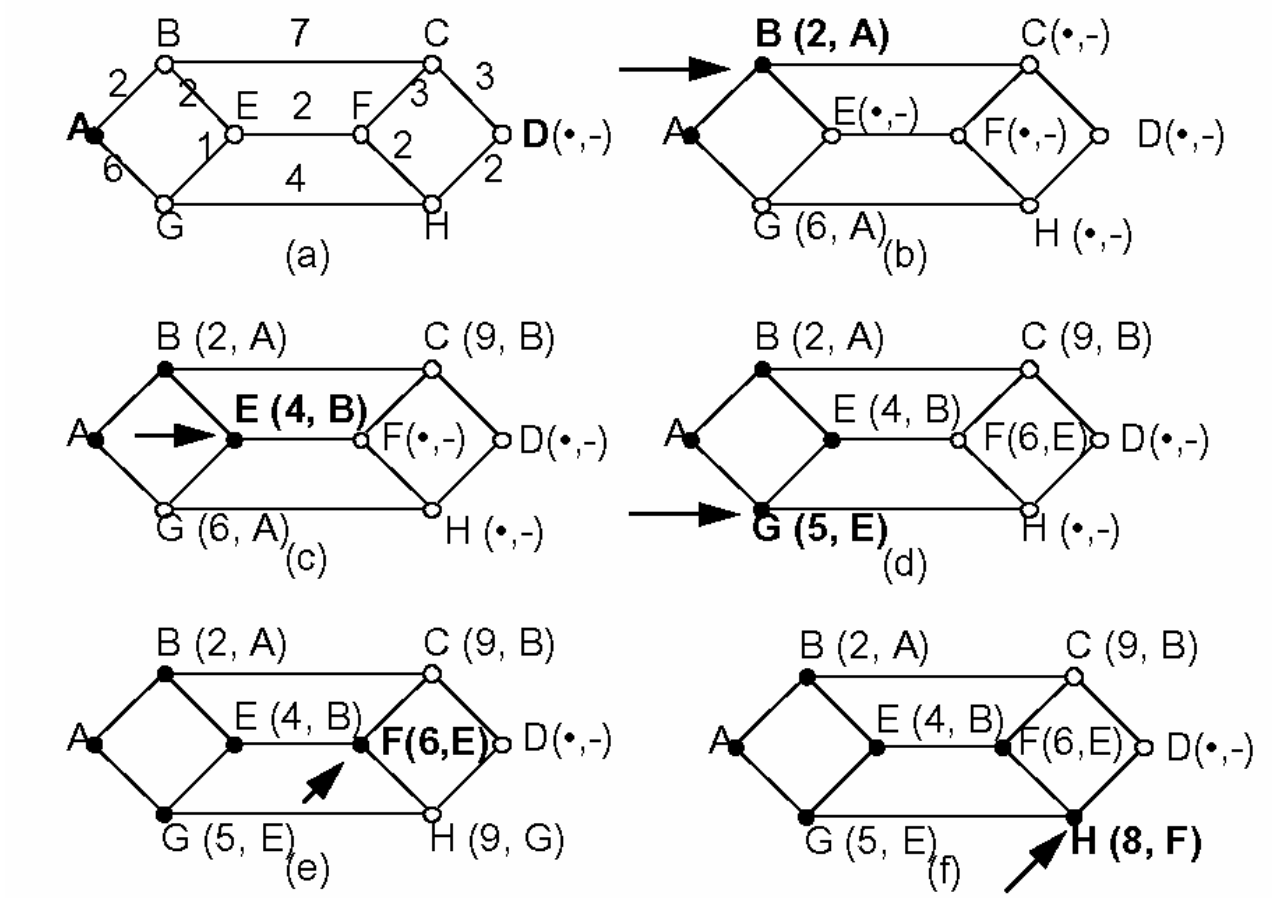
Non-Adaptive Shortest Path Routing

Spanning Tree and Optimized Route

- information about the entire network has to be available
  - i. e. can be used for comparison purposes / as a benchmark

Example:

- link is labeled with distance / weight
- node is labeled with distance from source node along best known path (in parentheses)



Non-Adaptive Shortest Path Routing

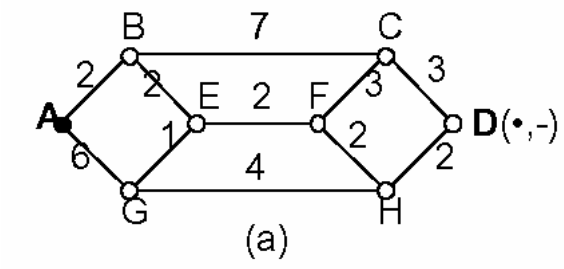
Procedure: e. g. according to Dijkstra

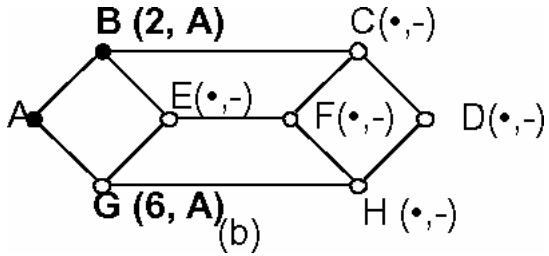
find the shortest path from A to D:

- labels may be permanent or tentative
- initially, no paths are known
  ➔ all nodes are labeled with infinity (TENTATIVE)
- discovery that label represents shortest possible path from source to node:
  ➔ label is made PERMANENT

1. Node A labeled as permanent (full black mark)
2. relabel all directly adjacent nodes with the distance to A (path length, nodes adjacent to source):
   - e.g. B(2,A) and G(6,A)
3. examine all tentatively labeled nodes; make the node with the smallest label permanent
   - e.g. B(2,A)
4. this node will be the new working node for the iterative procedure (i.e., continue with step 2.)

Non-Adaptive Shortest Path Routing (worksheet 1)





Example:

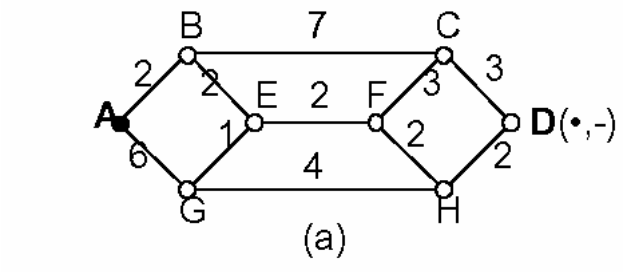
- link is labeled with distance
- node is labeled with distance from source along best known path

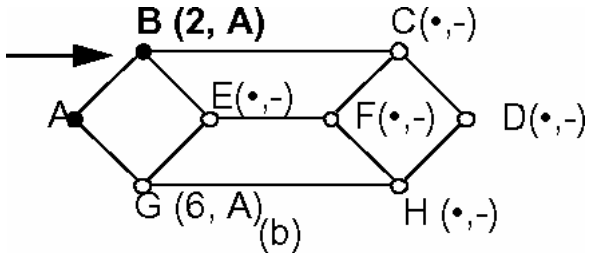
Procedure: e. g. according to Dijkstra

find: the shortest path from A to D:

1. Node A labeled as permanent (black mark)
2. relabel all directly adjacent nodes with the distance to A
   - (path length, IS adjacent to the source):
   - e. g. B(2,A) and G(6,A)

Non-Adaptive Shortest Path Routing (worksheet 2)





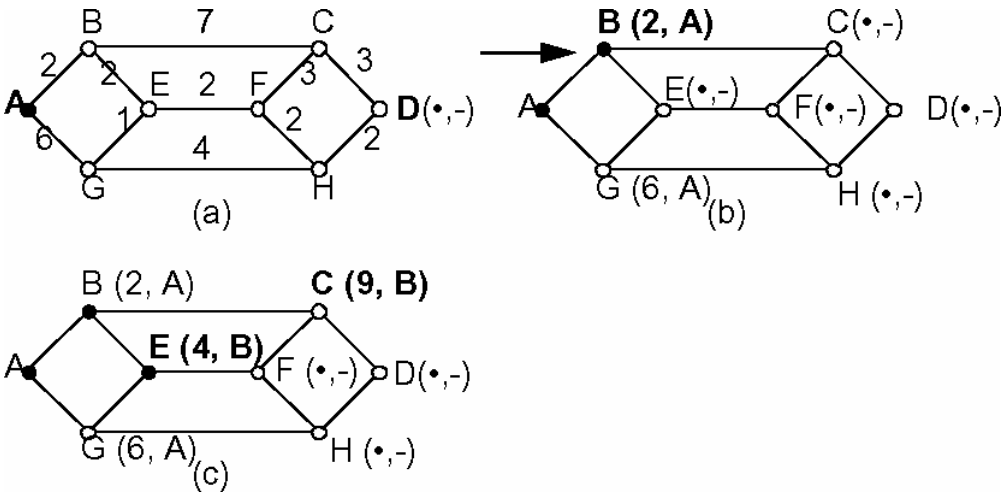
Example:

- link is labeled with distance
- node is labeled with distance from source along best known path

Procedure: e. g. according Dijkstra
find: the shortest path from A to D:
...

3. examine all tentatively labeled nodes
   - make the node with the smallest label permanent:
   - B(2,A)
4. this node will be the new working node for the iterative procedure
   - (i. e. continue with step 2)

Non-Adaptive Shortest Path Routing (worksheet 3)



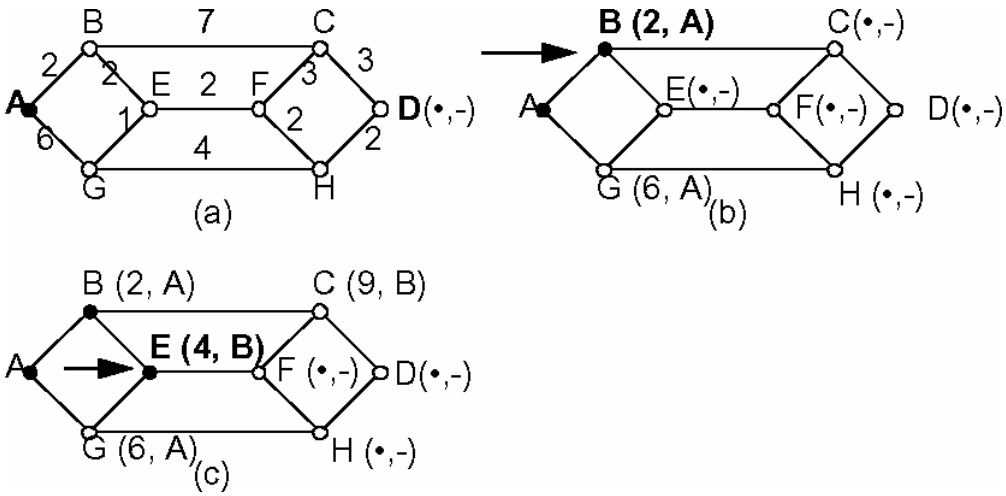
Example:

- link is labeled with distance
- node is labeled with distance from source along best known path

Procedure: e.g., according to Dijkstra
find the shortest path from A to D:

1. Node B has been labeled as permanent (black mark)
2. relabel all directly adjacent nodes with the distance to B
   (path length, nodes adjacent to source):
   - A (does not apply, because it is the origin),
   - i.e. E (4,B), C (9,B)

Non-Adaptive Shortest Path Routing (worksheet 4)



Example:

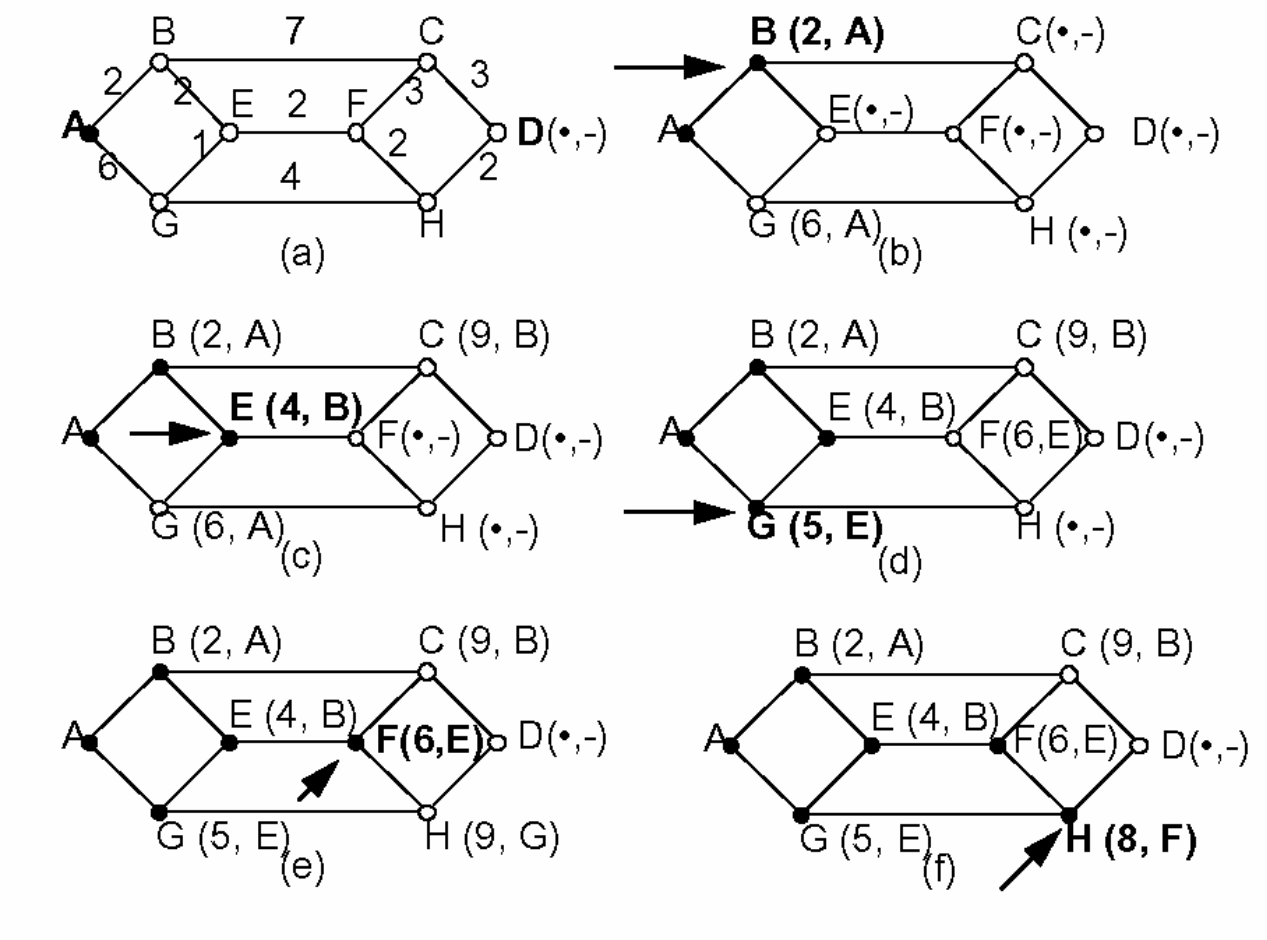
- link is labeled with distance
- node is labeled with distance from source along best known path

Procedure: e.g., according to Dijkstra
find the shortest path from A to D:

1. ....
2. ....
3. examine all tentatively labeled nodes;
   - make the node with the smallest label permanent: e.g. E(4,B)
4. this node will be the new working node for the iterative procedure ...

Non-Adaptive Shortest Path Routing (worksheet 5)

And continue with source E …



7 Non-Adaptive Flooding

Principle:

IS transmits the received packet to all adjacent IS

- (except over the path it came in)
- but generates "an infinite amount" of packets

Methods to limit packets

1. HOP COUNTER in the packet header
   - each IS decrements this hop counter
   - when the hop counter = 0, the packet is discarded
   - initialization for maximum path length (if known);
     - worst case: subnet diameter
2. each STATION REMEMBERS THE PACKETS THAT HAVE ALREADY BEEN TRANSFERRED and deletes them upon recurrence
   - source router inserts sequence number into packets received from hosts
   - each router needs an "already seen sequence number" list per source router
   - packets with sequence number on list is dropped
   - sequence number list must be prevented from growing without bounds
     - store only upper-counter / highest sequence number(s)

Variation: Selective Flooding

Approach:

- do not send out on every line
- IS transmits received packet to adjacent stations, LOCATED IN THE DIRECTION OF THE DESTINATION
- with 'regular' topologies this makes sense and is an optimization
- but some topologies do not fit well to this approach

Comment:

- geographically-oriented routing got recent interest for mobile scenarios

Flooding: Evaluation and use

- overhead: not practical in most applications
- extremely robust: military use
- reaches all IS: e.g., the exchange of control data between nodes
- initialization phase: does not need information about the topology
- always finds shortest path: can be used as benchmark

Class ADAPTIVE ALGORITHMS

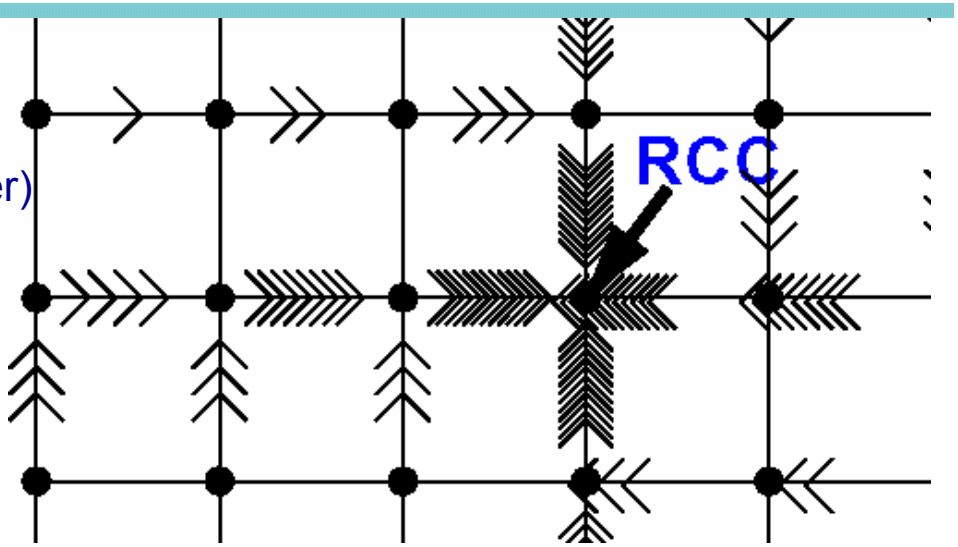
- decisions are based on current network state
  - measurements / estimates of the topology and the traffic volume
- further sub-classification into
  - centralized algorithms
  - isolated algorithms

  and
  - distributed algorithms

Adaptive Centralized Routing

Principle:

- in the network:
  - RCC (Routing Control Center)
- each IS sends periodically information on the current status to the RCC
  - list of all available neighbors
  - actual queue lengths
  - line utilization, etc.
- Routing Control Center RCC
  - collects information
  - calculates the optimal path for each IS pair
  - generates routing tables and distributes these to the ISs



Example: TYMNET

- packet exchanging network
- 1000 nodes/IS
- virtual circuits

Adaptive Centralized Routing

Characteristics:

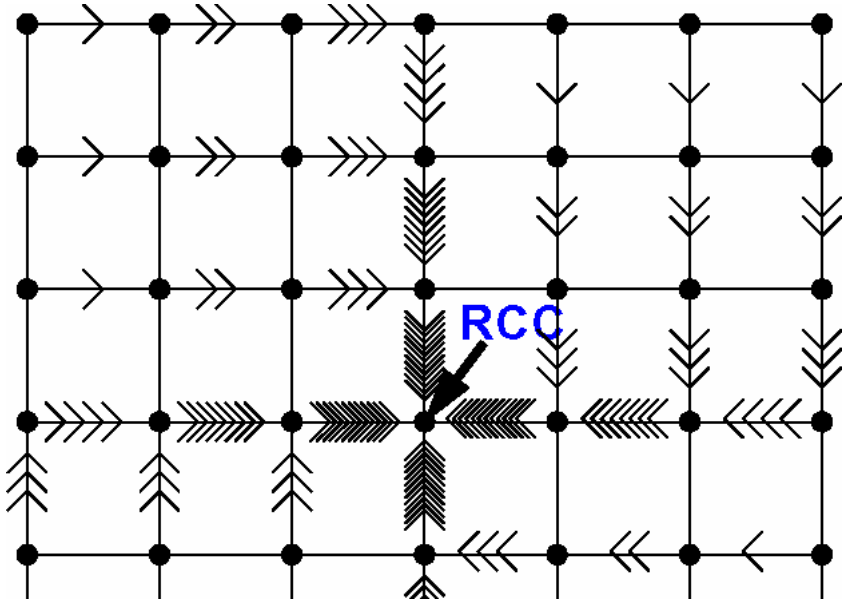
- Routing Control Center RCC has complete information

➔ perfect decisions

- and IS is free of routing calculations

but

- re-calculations quite often necessary (approx. once/min or more often)
- low robustness
- no correct decisions if network is partitioned
- IS receive tables at different times
- traffic concentration in RCC proximity



9 Adaptive Isolated Routing – Backward Learning

Isolated routing

- every IS makes decision based on locally gathered information only
  - no exchange of routing information among nodes
  - only limited adaptation possibility to changed traffic / topology

IS "learns" from received packets (..., S, C, ...)

- S ... source - IS
- C ... hop counter

Packet of source S is received on line L after C hops

➔ S is reachable on L within C hops

Routing table in IS

- per line: L-table (destination - IS, outgoing line, C_{min})
- update of the routing table

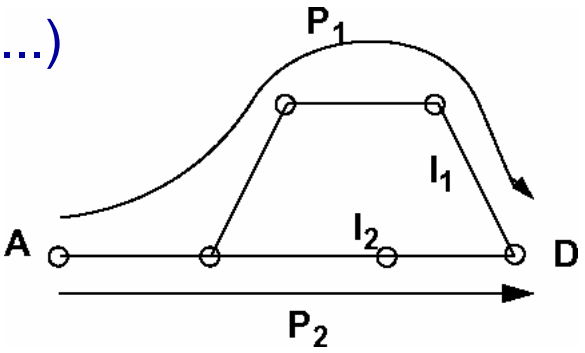
IS receives packet (..., S, C, ...) on L

```
if not (S in L-Table)
   then Add(S,L,C)
   else if C < Cmin
      then Update(S,L,C)
```

Adaptive Isolated Routing – Backward Learning

Example:

- packet (..., source - IS, section counter, ...) at node D:
- P1 (..., A, 4, ...) → Add (A, l_1, 4)
- P2 (..., A, 3, ...) → Update (A, l_2, 3)



Problem:

- packets use a different route, e. g. because of failures, high load
- algorithm retains only the old value (because it was "better"),
  - i.e. algorithm does not react to deteriorations

Solution:

- periodic deletion of routing tables
  - (new learning period)
- table deletion
  - too often: mainly during the learning phase
  - not often enough: reaction to deteriorations too slow

10 Adaptive Distributed – Distance-Vector Routing

Distance-Vector Routing

Group of DISTANCE VECTOR ROUTING ALGORITHMS

- also known as
  - distributed Bellman-Ford algorithm, Ford-Fulkerson algorithm

Usage

- was the original ARPANET routing algorithm
- has been used in the Internet as RIP –(ROUTING INFORMATION PROTOCOL)

Basic principle

- IS maintains table (i.e., vector) stating
  - best known distance to destinations
  - and line to be used
- ISs update tables
  - by exchanging routing information with their neighbors

Distance-Vector Routing - Foundations

Each IS maintains routing table with one entry per router in the subnet
- estimate of the distance (hops, delay, packets queued, ...) to destination
- outgoing line to be used for that destination

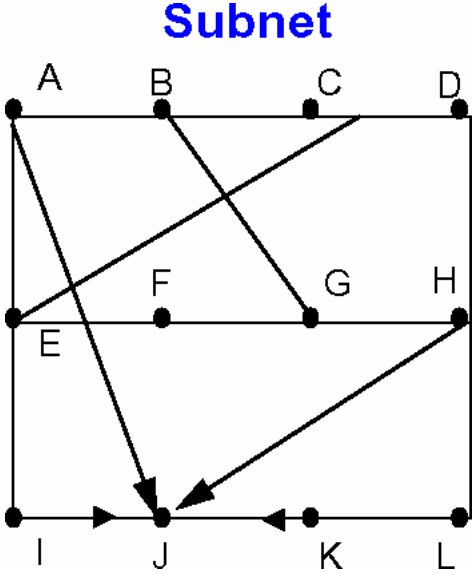
Each IS is assumed to know the "distance(s)" to each neighbor
- number of hops (= 1)
- delay (echo packets)
- queue length (e.g., used in the ARPANET),...

IS sends lists with estimated distances to each destination periodically to its neighbors Y
- e.g., Internet RIP every 30 sec, maximum distance 15 hops

X receives list D(Z) from neighbor Y
- distance X to Y: d
- distance Y to Z: D(Z)
- i.e. distance X to Z (via Y): D(Z) + d

IS calculates a new routing table from the received lists, containing
- destination IS, preferred outgoing path, "distance"

Distance-Vector Routing – Example

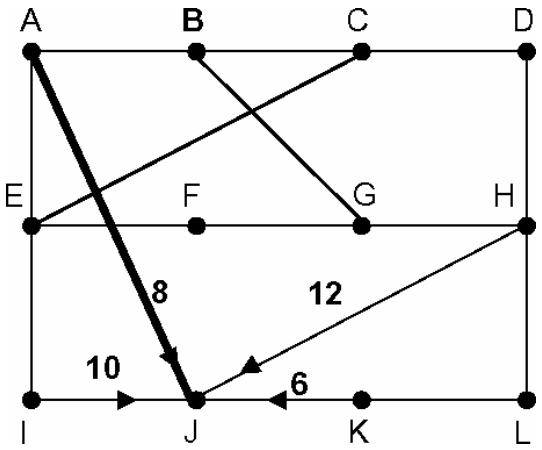


delays at and of nodes A/I/H/K/. (column).
to nodes A;B;C;D… (row)

To	routing table of A	routing table of I	routing table of H	routing table of K	new estimated delay from J	line
A	0	24	20	21	**8**	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	**12**	H
I	21	0	14	22	**10**	I
J	9	11	7	10	0	-
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA Delay=8	JI Delay=10	JH Delay=12	JK Delay=6	new routing table for J	

Previous routing table will not be taken into consideration
➔ Reaction to deteriorations

Distance-Vector Routing – Example



Example: defining a section

.B.sends information to node J

	A	I	H	K	newly estimated delay starting at J	line
B	12	36	31	28	20	A

JA Delay =8 JI delay =10 JH delay =12 JK delay =6

from B via A: costs (JA) + costs path (AB) = 8 + 12 = 20

from B via I: costs (JI) + costs path (IB) = 10 + 36 = 46

from B via H: costs (JH) + costs path (HB) = 12 +31 = 43

from B via K: costs (JK) + costs path (KB) = 6 + 28 = 34

seek for minimum: Min (JAB, JIB, JHB, JKB) = JAB = 20

Distance Vector Routing – “Count to Infinity”

Information distribution over new
- short paths (with few hops): fast
- long paths with many hops: SLOW

Example: route improvement
- previously: A unknown
- later: A connected with distance 1 to B, this will be announced
- Note: Synchronous update used here for simplification
- distribution proportional to topological spread

Example: deterioration,
(here: connection destroyed)
- A previously known, but now detached
- the values are derived from (incorrect) connections of distant IS

Comment
- limit “infinite” to a finite value, depending on the metrics
  - example:
    "infinite = maximum path length + 1"

A	B	C	D	E	
	∞	∞	∞	∞	Initially
	1	∞	∞	∞	After 1 exchange
	1	2	∞	∞	After 2 exchanges
	1	2	3	∞	After 3 exchanges
	1	2	3	4	After 4 exchanges

A_1	B_1	C_1	D_1	E_1	
	1	2	3	4	Initially

B: no connection directly to A,
but C reports distance CA=2
i. e. BA = BC+ CA = 1 + 2 = 3
actually wrong!

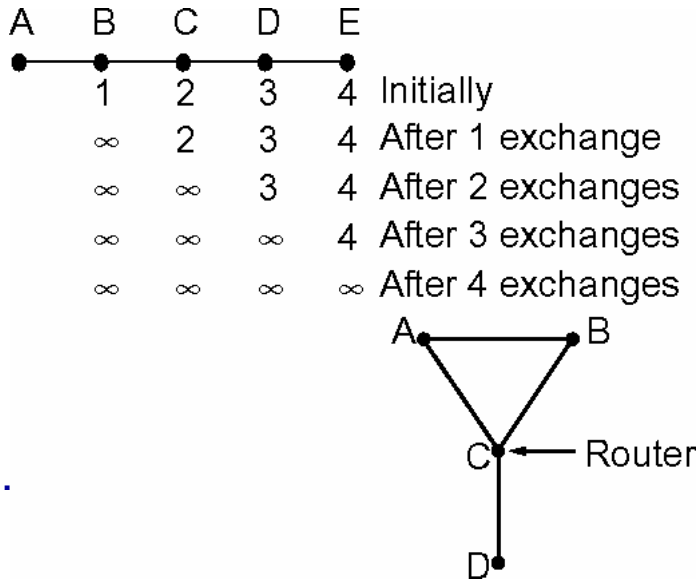
3	2	3	4	After 1 change
3	4	3	4	After 2 changes
5	4	5	4	After 3 changes
5	6	5	6	After 4 changes
7	6	7	6	After 5 changes
7	8	7	8	After 6 changes
…				
∞	∞	∞	∞	

Distance Vector Routing Variant “Split Horizon Algorithm”

Objective - based on the Distance Vector principle
- to improve the "count to infinity" property

Principle
- in general, to announce the "distance" to each neighbour
- special case:
  - if neighbour Y exists on the reported route,
  - then X reports the response "false" to Y

➔ distance X (via Y) according to arbitrary i: ∞



Example: deterioration,
e.g. connection destroyed
- B to C: A = ∞ (real),
- C to B: A = ∞ (because A is on path), ...

Note:
still poor, depending on topology, example:
- connection CD is removed
- A receives "false information" via B
- B receives "false information" via A

➔ slow distribution (just as before)

11 Adaptive Distributed – Link State Routing

also "distributed routing"

Basic principle

- IS measures the "distance" to the directly adjacent IS, distributes information, calculates the ideal route

Procedure

1. determine the address of adjacent IS
2. measure the "distance" (delay, ...) to neighbor IS
3. organize the local link state information in a packet
4. distribute the information to all IS
5. calculate the route based on the information of all IS

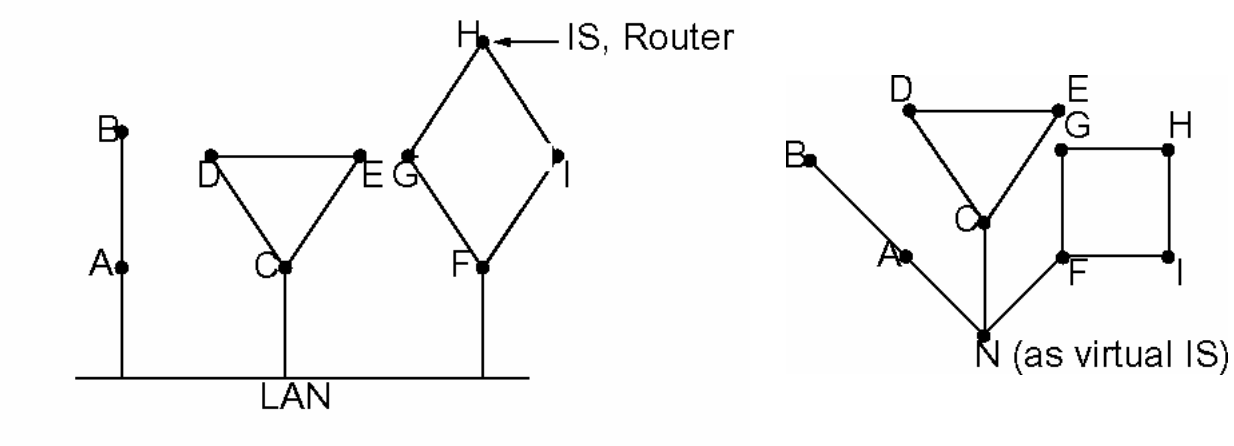
Usage

- introduced into ARPANET in 1979, nowadays most prevalent
- IS-IS (Intermediate System-Intermediate System)
  - developed by DECNET
  - also used as ISO CLNP in NSFNET
  - Novell Netware developed its own variation from this (NLSP)
- OSPF (Open Shortest Path First)
  - since 1990 Internet RFC 1247

Link State Routing

1. Phase:
   gather information about the adjacent intermediate systems
   - initialization procedure:
     - new IS:
       - sends a HELLO message over each L2 channel
     - adjacent IS:
       - responds with its own address, unique within the network

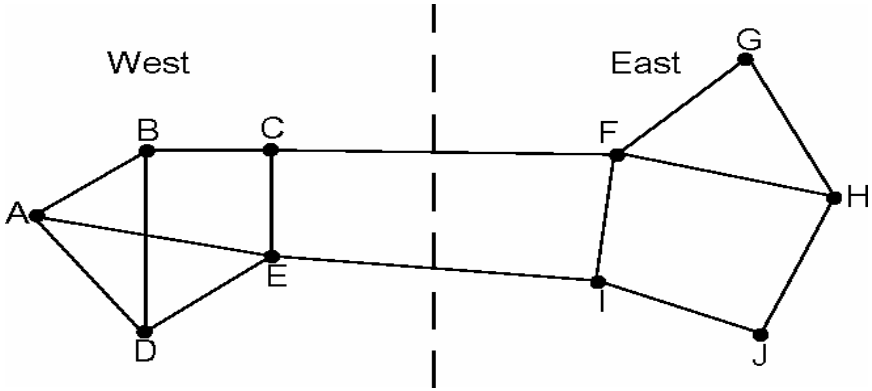
Example:
- with LAN (as virtual IS)



Link State Routing

2. Phase:
define the "distance"

- distance is generally defined as delay
- detection via transmission of ECHO messages, which are reflected at receiver
- multiple transmission:
  - improved average value
  - with or without payload:
    - with payload is usually better,
  - but "with load" may lead to an "oscillation" of the load:

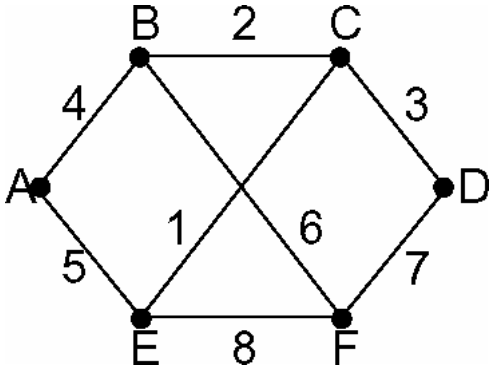


  - after each new routing table the other link CF or EI is charged

Link State Routing

3. Phase:
organizing the information as link state packet

- including own address, sequence number, age, "distance"
- timing problems: validity and time of sending
  - periodically
  - in case of major changes



Link State Packets:

A	
Seq.	
Age	
B	4
E	5

B	
Seq.	
Age	
A	4
C	2
F	6

C	
Seq.	
Age	
B	2
D	3
E	1

D	
Seq.	
Age	
C	3
F	7

E	
Seq.	
Age	
A	5
C	1
F	8

F	
Seq.	
Age	
B	6
D	7
E	8

Link State Routing

4. Phase:
distribute the local information to all IS

- by applying the flooding procedure (very robust)
  - therefore sequence number in packets
- problem: inconsistency
  - varying states simultaneously available in the network
  - indicate and limit the age of packet,
    i.e. IS removes packets that are too old

5. Phase:
compute new routes

- each IS for itself
- possibly larger amount of data available

12 Routing: Diverse Enhancements

12.1 Multipath Routing

Principle:

- using alternative routes between the IS pairs
- usage frequency depends on the quality of the alternative
- higher throughput due to the data traffic being distributed to various paths
- increased reliability

Implementation:

- each IS contains a rating table including
  - one row for each possible destination IS

D	A_1	G_1	A_2	G_2	…	A_n	G_n

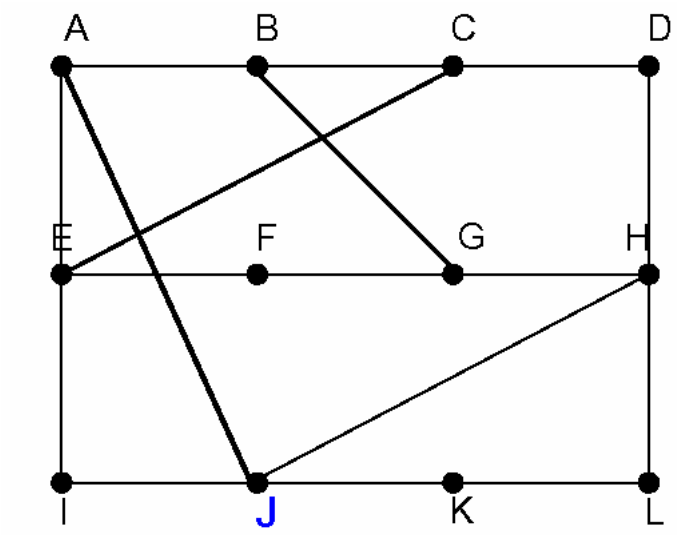
D … destination

A_i … i-best outgoing line

G_i … weight for A_i

G_i determines the probability with which A_i will be used: $\left(\sum_{i=1}^{n} G_i = 1\right)$

Multipath Routing: Example



dest.	1st choice		2nd choice		3rd choice	
A	A	0.63	I	0.21	H	0.16
B	A	0.46	H	0.31	I	0.23
C	A	0.34	I	0.33	H	0.33
D	H	0.50	A	0.25	I	0.25
E	A	0.40	I	0.40	H	0.20
F	A	0.34	H	0.33	I	0.33
G	H	0.46	A	0.31	K	0.23
H	H	0.63	K	0.21	A	0.16
I	I	0.65	A	0.22	H	0.13
K	K	0.67	H	0.22	A	0.11

Example: Table from J →

Selecting the alternatives: i.e., generating a random number z ($0 \le z < 1$)

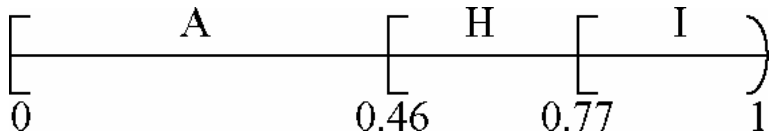
$A_1: 0 \le z < G_1$

$A_2: G_1 \le z < G_1 + G_2$

....

$A_n: G_1 + G_2 + ... + G_n\text{-}1 \le z < 1$

Example: destination B

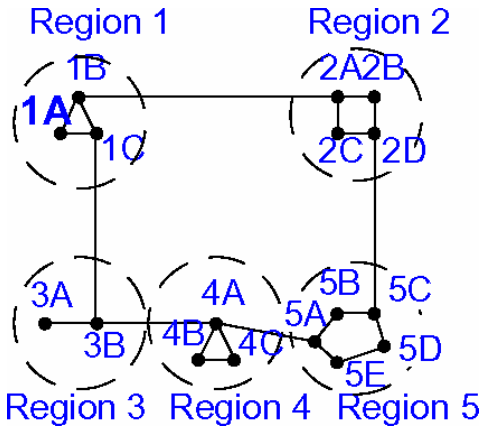


12.2 Hierarchical Routing

Motivation

- a large number of IS means
  - time-consuming dynamic routing calculation
  - storage of very large routing tables

➔ hierarchical structure reduces individually treated IS



Example (of 2 tables)

Comparison

- the best path is not always calculated
- design issue: number of layers

Hierarchical table for 1A		
Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Full table for 1A:

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

16 Addressing

3 types of identifiers: names, addresses and routes [Shoch 78]

"The NAME of a resource indicates WHAT we seek,
an ADDRESS indicates WHERE it is, and
a ROUTE says HOW TO GET THERE."

Objectives:

- global addressing concept for ES
- simplified address allocation
- addresses independent from
  - type and topology of the subnetworks
  - number and type of the subnetworks to which the ES have been connected
  - location of a source ES

16.1 X.121 Addressing

CCITT/ITU "numbering scheme"

- addressing concept for public data networks
- a.o., used by X.25

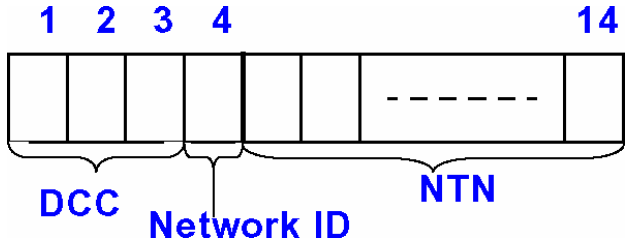
X.121 address:

- a maximum of 14 digits
- consisting of
  - Data Network Identification Code (4 digits)
    - Data Country Code (digits 1 - 3)
    - Network Identification (digit 4)
  - Network Terminal Number (max. 10 digits)

Example:

DCC for USA: 310 - 329, i. e. max. 200 networks

DCC for Tonga: 539, i. e. max. 10 networks



16.2 OSI Addressing

Objective:

- global addressing concept for both existing and new subnetworks

Situation: different concepts exist for

- public networks:
  - X.121: data networks
  - F.69: telex
  - E.163: telephone network
  - E.164: ISDN, ...
- private networks

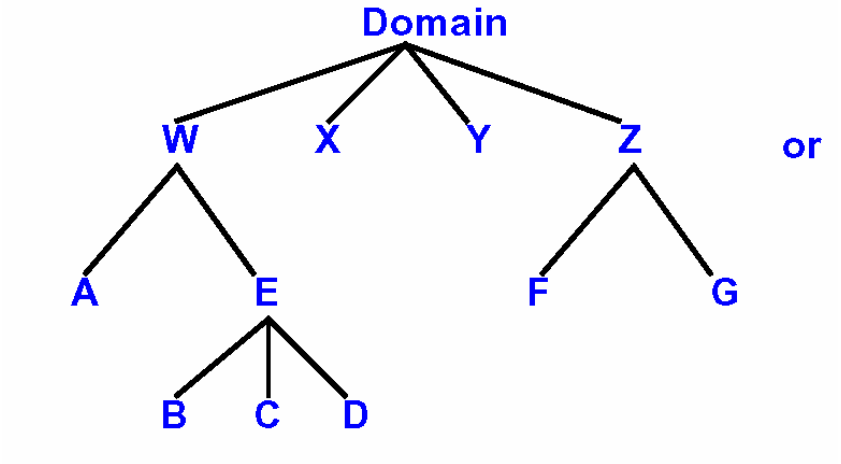
➔ i.e., a flexible and expandable concept is necessary

OSI method: unique NSAP identification

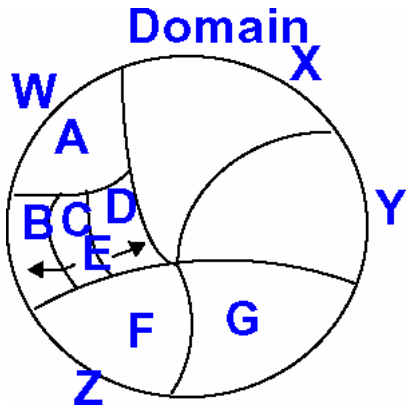
OSI method: hierarchic addresses

- OSI defines the ADDRESSING DOMAINS
- the domain contains the ADDRESSING AUTHORITY
- Addressing Authority
  - allocates addresses
  - creates new domains and delegates authority

OSI Addressing



or



graphic representation of the domain hierarchy:

A domain may be

- networks of one type
- networks of a geographical region
- networks of an organization
- ...

OSI Addressing: Structure

Address length: 20 bytes (binary) or 40 digits

Address structure:

IDP	DSP

- Initial Domain Part (IDP) with
  - AUTHORITY AND FORMAT IDENTIFIER (AFI)
    - specifies how to interpret the IDI (syntax and semantics)
    - e.g. the format of the DSP (binary or digits)

IDI Format	DSP SYNTAX	
	Decimal	Binary
X.121	36	37
ISO DCC	38	39
F.69	40	41

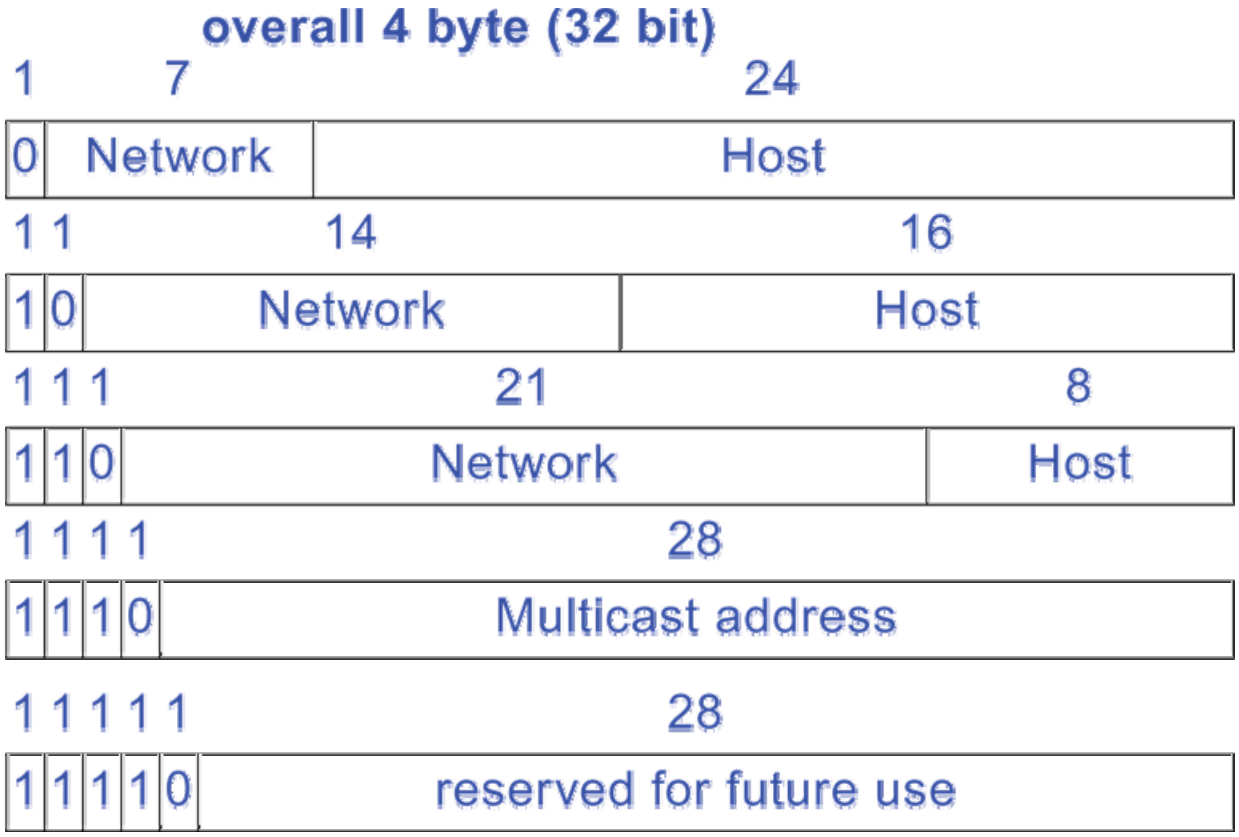
Character	National Character
50	51

- INITIAL DOMAIN PART (IDI)
  - identifies the Addressing Authority (AA), responsible for ALLOCATING THE NSAP ADDRESSES
  - identifies the domain
- Domain Specific Part (DSP)
  - contains the address clearly identifying the ES within the domain

16.3 Internet Addresses (IP)

Global addressing concept for ES (and IS) in the Internet

- 32 bit address (amount is limited!)
- each address is unique worldwide
- structure: Net-ID (Subnet-ID), ES-ID



Internet Addresses (IP)

Notation

- decimal value for each byte (0...255)
- subdivided by dots
- value range: 0.0.0.0 ... 255.255.255.255

Formats: 5 classes

A:	1.0.0.0	up to	127.255.255.255
B:	128.0.0.0	up to	191.255.255.255
C:	192.0.0.0	up to	223.255.255.255
D:	224.0.0.0	up to	239.255.255.255 (Multicast)
E:	240.0.0.0	up to	247.255.255.255

Broadcast addresses: (convention: 11...1 for Host-ID)

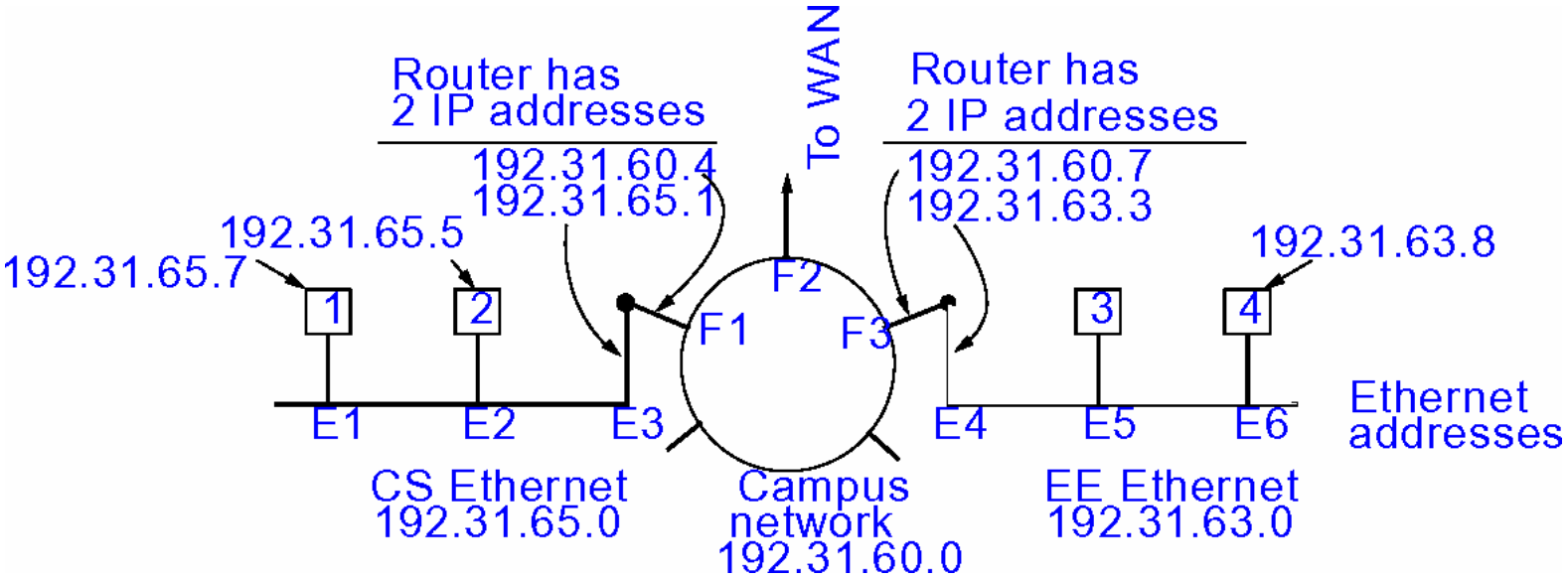
Internet Addresses (IP)

Address allocation

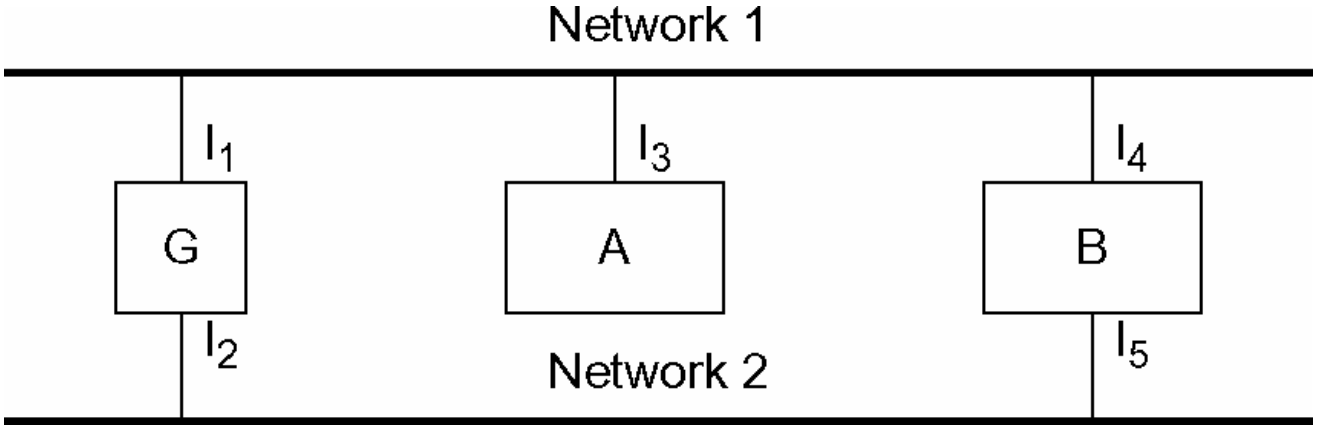
- class allocation and network range:
  - by a central authority
  - Network Information Center NIC
- end system
  - local
  - possibly forming a subnetwork

Example

- network



Internet Addresses (IP): A Critical Review



Addresses IDENTIFIY "NETWORK CONNECTIONS", not the ES

- "multi-homed" ES have more than one address
- a change of the connection forces the modification of the address
- the address has an impact on the chosen route (constitutes a problem in the mobile area)

Example: A cannot reach B via address I5 if G fails

- comment: is also valid for X.121

Amount of addresses

- limited

Internet Addresses (IP): The Future

IP Version 6 (IPv6)

- 16 byte length (instead of 4 byte length, i. e. approx. 3 x 10^{38})

Distribution

- provider-based: approx. 16 mio. companies distribute addresses
- geographic-based: distribution as it is today
- link, site-used: address relevant only locally (security, Firewall concept)

e. g. new: Anycast

- sending data to an individual of a group
- e. g. the one who is geograhically the closest

Internet Addresses (IP): The Future

Prefix (binary)	Usage	Fraction
0000 0000	Reserved (including IPv4)	1/256
0000 0001	Unassigned	1/256
0000 001	OSI NSAP addresses	1/128
0000 010	Novell Netware IPX addresses	1/128
0000 011	Unassigned	1/128
0000 1	Unassigned	1/32
0001	Unassigned	1/16
001	Unassigned	1/8
010	Provider-based addresses	1/8
011	Unassigned	1/8
100	Geographic-based addresses	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1/16
1111 0	Unassigned	1/32
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
1111 11100	Unassigned	1/512
1111 111010	Link local use addresses	1/1024
1111 111011	Site local use addresses	1/1024
1111 1111	Multicast	1/256