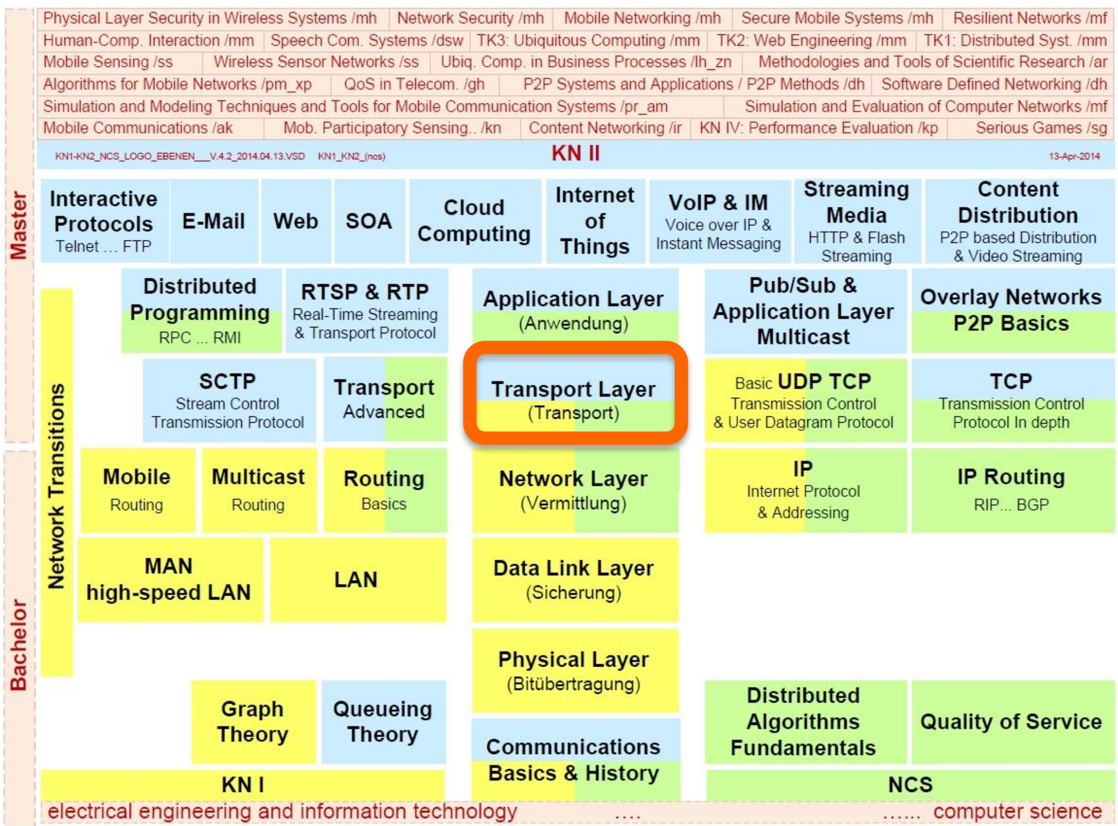
Communication Networks I

L4 Transport Layer - Fundaments



Prof. Dr.-Ing. **Ralf Steinmetz**

KOM - Multimedia Communications Lab

Overview

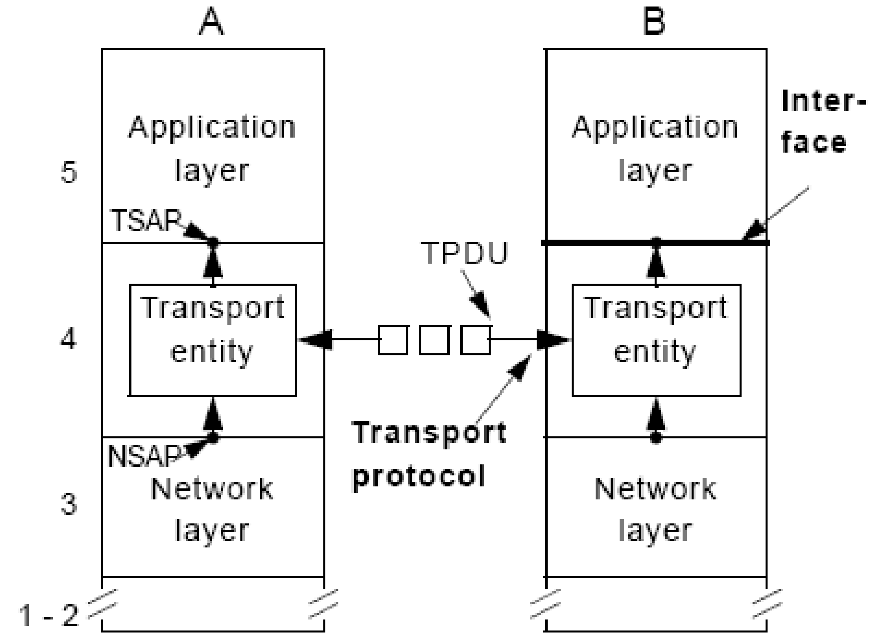
1 Transport Layer Function

To provide data transport

- Reliably
- Efficiently
- At low-cost

For

- Process-to-process (applications)
- I.e., at end system-to-end system



(If possible) independent from

- Particularities of the networks (lower layers) used

1.1 Transport Service

Connection oriented service

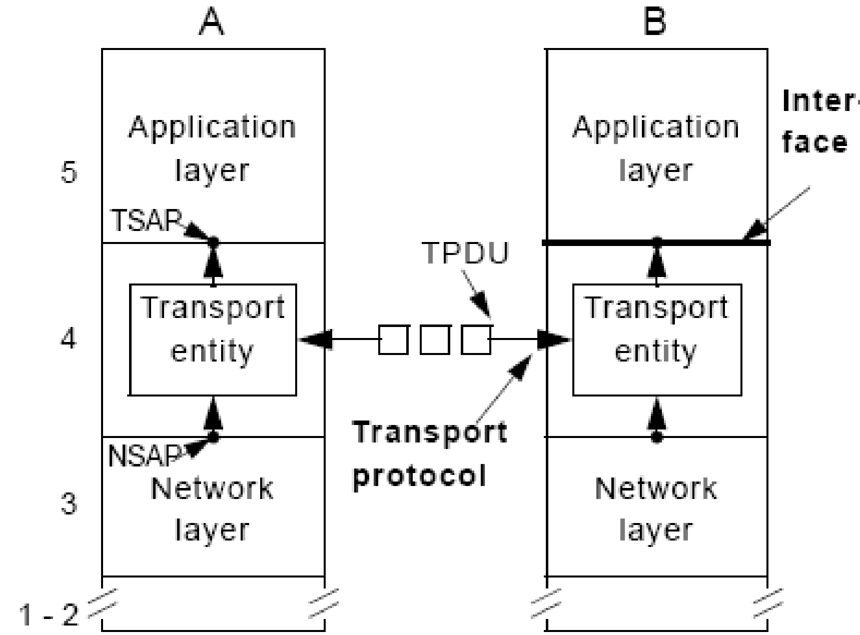
- 3 phases
  1) connection set-up
  2) data transfer
  3) disconnect

Connectionless service

- Transfer of isolated units

Implementation: transport entity

- Software and/or hardware?
- Software part usually contained within the kernel (process, library)



Transport Service

Similar services of network layer and transport layer:

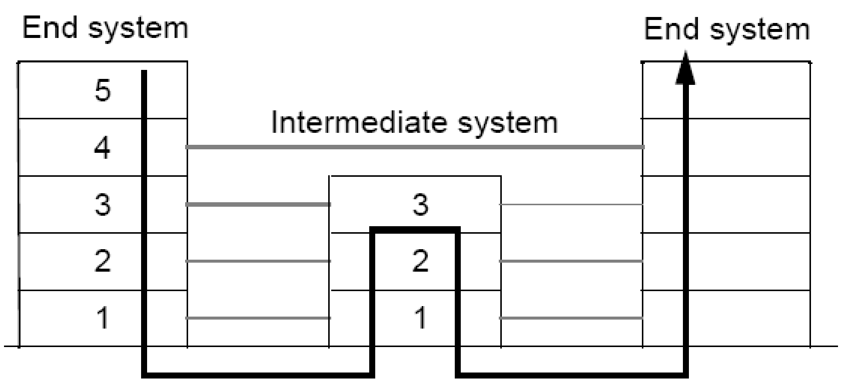
- Why 2 Layers?

Transport service: to Improve the Network Service Quality

- Users and layers want to get from the network layer, e.g.
  - reliable service
  - necessary time guarantees

Network service

- Not to be self-governed or influenced by the user
- Independent from application & user
  - enables compatibility between applications
- Provides for example
  - “only” connection oriented communications
  - or “only” unreliable data transfer



TECHNISCHE UNIVERSITÄT DARMSTADT

Transport layer

- Isolates upper layers from technology, design and imperfections of subnet

Traditionally distinction made between

- Layers 1 - 4
  - transport service provider
- Layers above 4
  - transport service user

Transport layer has key role

- Major boundary between
  - provider and
  - user of reliable data transmission service

Transport Services Primitives

TECHNISCHE UNIVERSITÄT DARMSTADT

Primitives for a simple transport service:

Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ	Actively attempt to establish a connection
SEND	DATA	Send Information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ	Request to release the connection

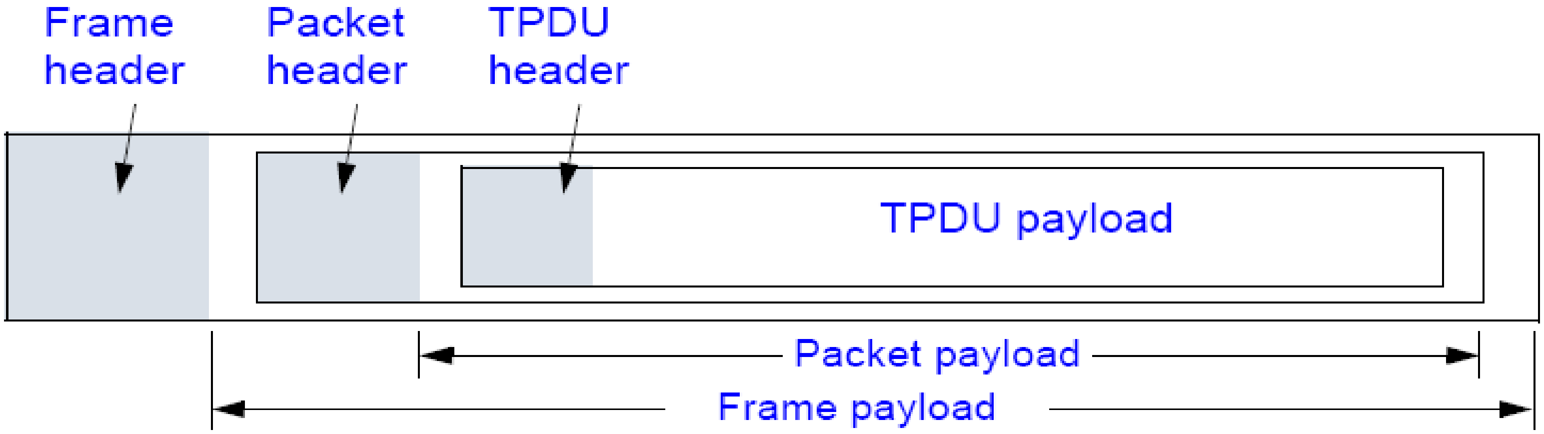
Transport Service: Terminology

Entities exchanged:

Layer	Data Unit
Transport	TPDU / Message
Network	Packet
Data Link	Frame
Physical	Bit/Byte (bit stream)

TPDU: Transport Protocol Data Unit

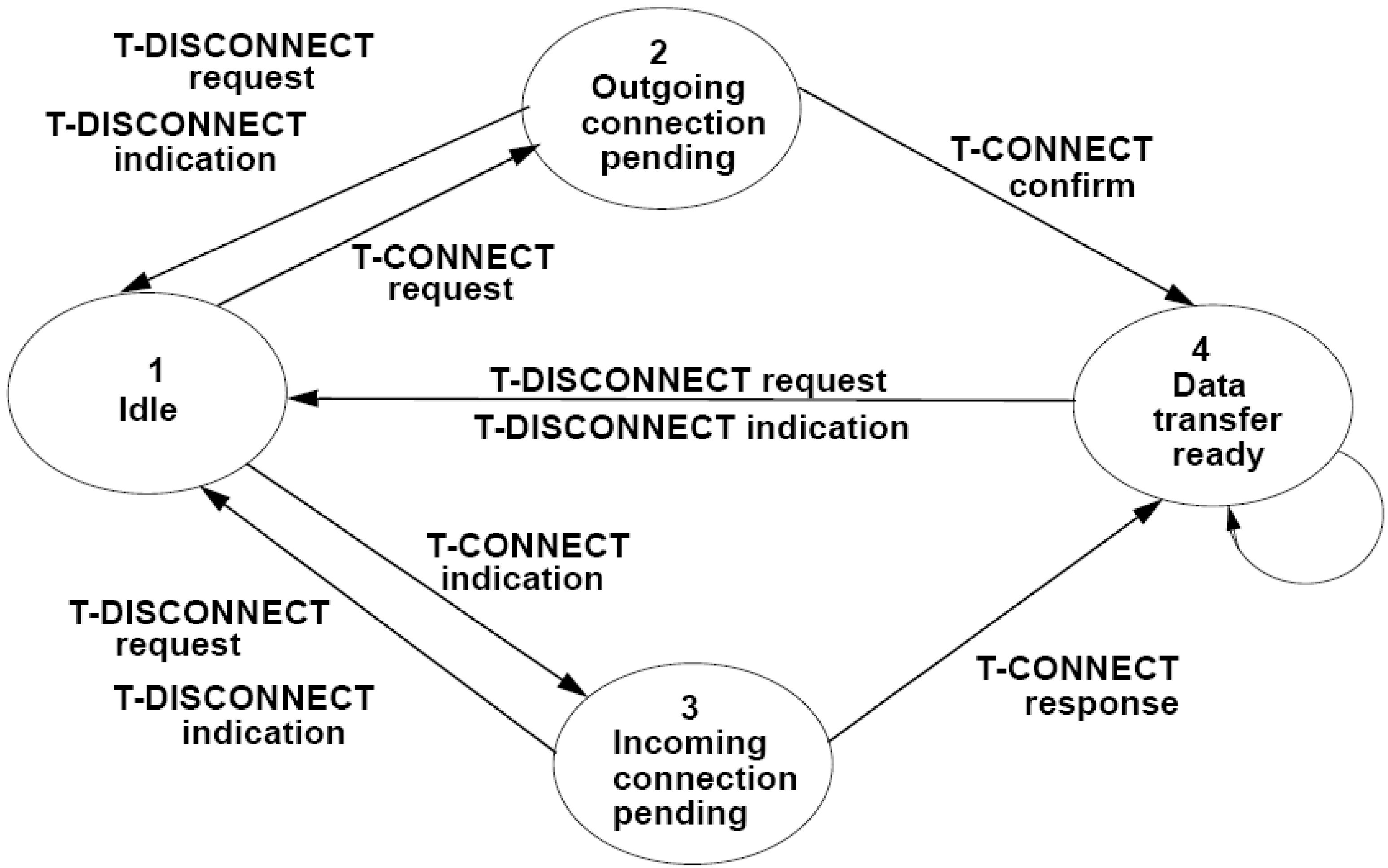
Nesting of TPDUs, packets, and frames:



TECHNISCHE UNIVERSITÄT DARMSTADT

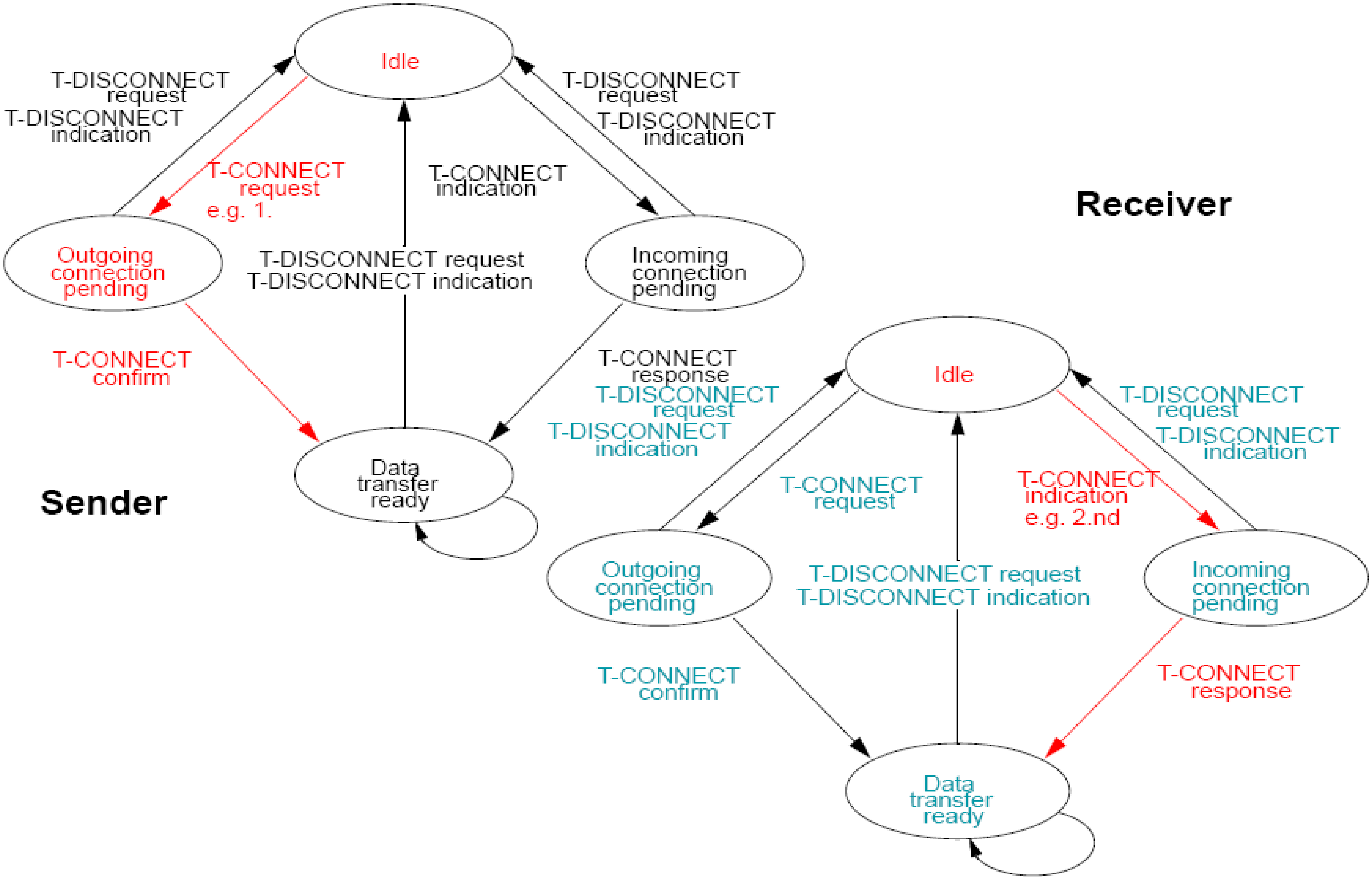
Example: ISO-OSI nomenclature

- State transition diagram



Connection Oriented Service: State Transition Diagram

TECHNISCHE UNIVERSITÄT DARMSTADT



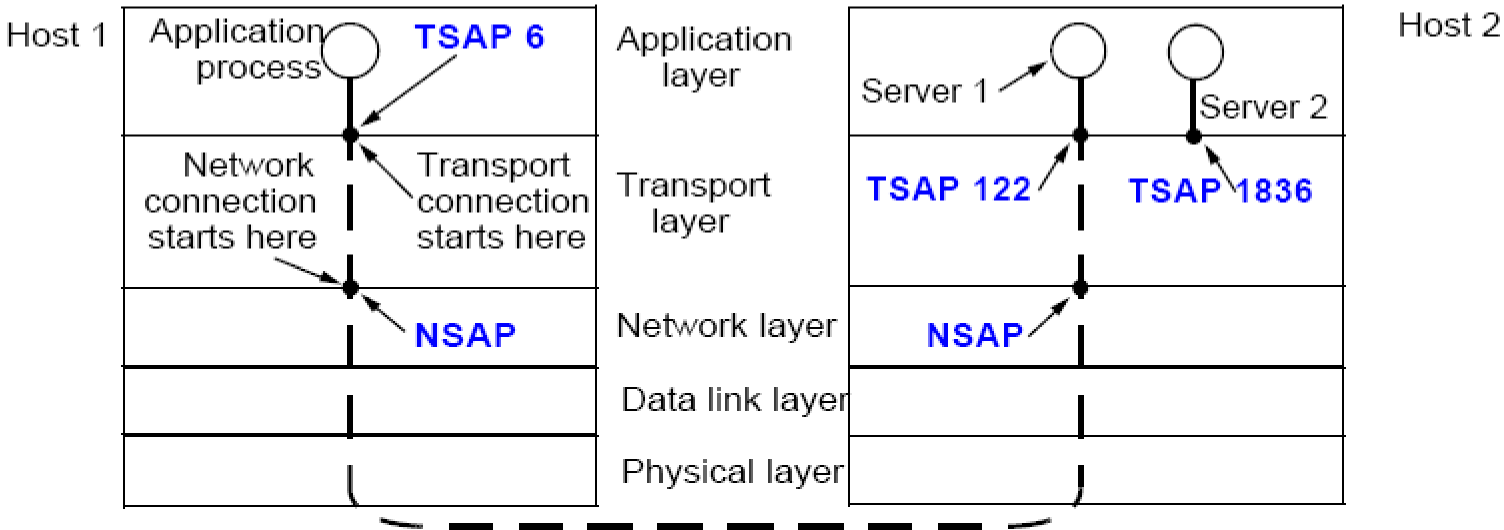
Example: Parameters for Disconnect

Example: ISO-OSI nomenclature, reason for a "T-Disconnect"

Reason	Notes
Normal disconnect initiated by session entity	1, 4
Remote congestion at transport entity during CC	1, 4
Failed connection negotiation	1, 3
Duplicated source reference for same NSAP pairs	1, 4
References are mismatched	1, 4
Protocol error	1, 4
Reference overflow	1, 4
Connection request refused	1, 4
Header or parameter length invalid	1, 4
No reason specified	2, 4
Congestion at TSAP	2, 4
TSAP and session entity not attached	2, 3
Unknown address	2, 3
(Note 1) Used for classes 1 to 4 …. Different "classes" were introduced …. To denote certain Quality of Service	
(Note 2) Used for all classes	
(Note 3) Reported to TS-user as persistent	
(Note 4) Reported to TS-user as transient	

2 Addressing (at Transport Layer)

Model



Why identification?

- Sender (process) wants to address receiver (process)
  - for connection setup or individual message
- Receiver (process) can be approached by the sender (process)

Define transport addresses:

- Generic term: (Transport) Service Access Point (TSAP)
- Internet: port

Reminder: analogous end points in network layer: NSAP

- E.g., IP addresses

2.1 Steps - In General

1. Server (service provider)

- Connects itself to TSAP 122
- Waits for service request (polling, signaling, ..)

2. Client (application)

- Initiates connection via TSAP 6 as source and TSAP 122 as destination
  - i.e. CONNECT REQ

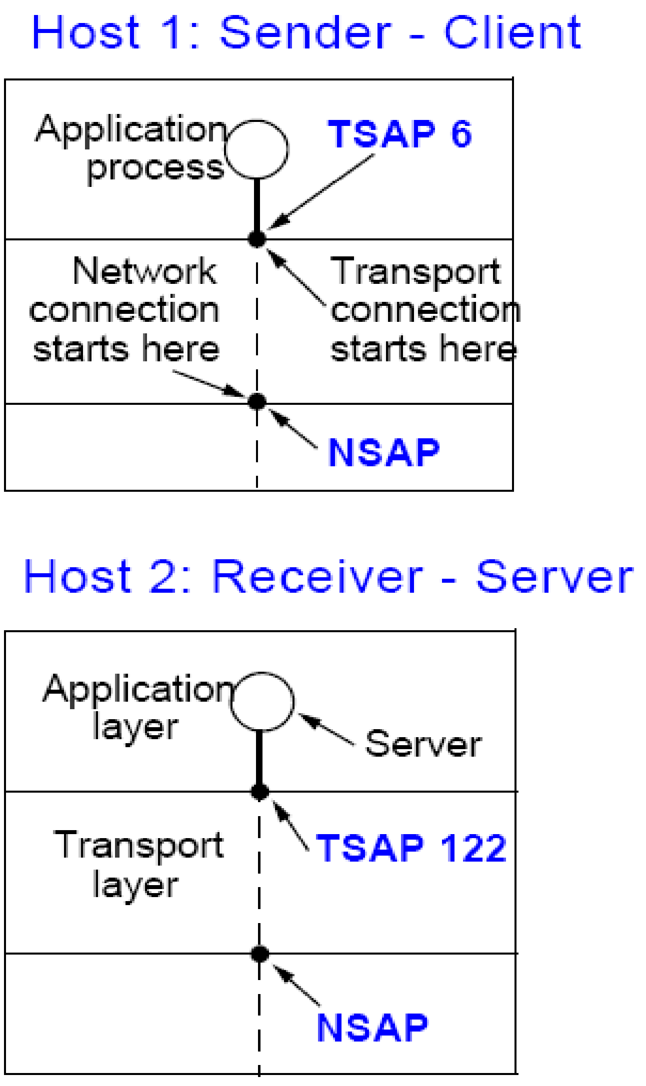
3. Transport system on host 1

- Identifies dedicated NSAP
- Initiates communication at network layer
- Communicates with transport entity on host2
- Informs TSAP 122 about desired connection

4. Transport entity on host 2

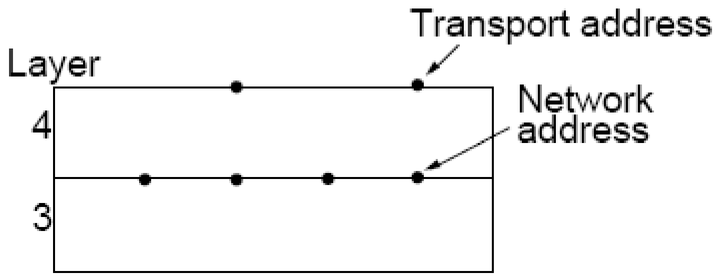
- Addresses the server
- Requests acceptance for the desired connection
  - i.e. CONNECT IND.

5. etc.



2.2 Determination of Appropriate Service Provider TSAP

How does the specific address of a service becomes known?



First Approach: TSAP known implicitly

- Services that are well known and often used have pre-defined TSAPs
  - as "well-known ports" of a transport protocol
- E.g., stored in /etc/services file at UNIX systems

Example: service ’time of day’

Characteristics

- Works well for small number of stable services
- Not suitable for user specific processes
  - existing for short time, no known TSAP address
- Waste of resources; seldom used servers active and listening

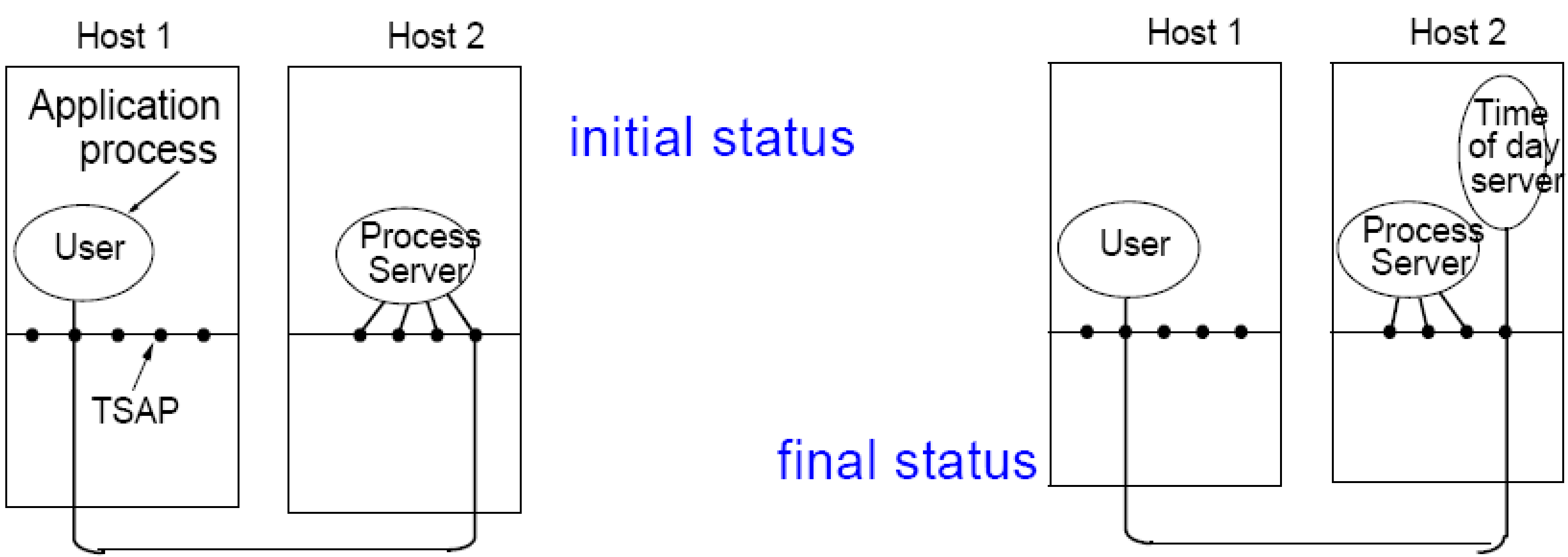
Determination of Appropriate Service Provider TSAP

Second Approach: "initial connection protocol"

- Process server acting as proxy for less often used servers
- Process server listens to a set of ports at same time
- Waits for connection requests
- Creates the appropriate service provider process
- Transfers connection and desired service
- Waits for further requests

Characteristics

- Works well for servers which can be created on demand
- Not suitable if service exists independently of process server at another machine (e.g., file server)



Third Approach: Name Server (directory server)

- Context
  - server process already exists

- Procedure
  - client addresses Name server (establishing connection)
  - client specifies the service as an ASCII data set
    - example “name of day”
  - name server supplies TSAP
  - client disconnects from name server
  - client addresses TSAP provided by name server
  - .....

- Comments
  - new services have to register at the name server
  - name server adds corresponding information at the database

2.3 Determination of Appropriate NSAP

How to localize the respective endsystem NSAP (layer 3!!) ?

- I.e., how to determine the appropriate NSAP?

First approach: hierarchical addressing

- TSAP contains this information

example: <country>.<network>.<port>

Second approach: “flat” addressing

- Dedicated “name server”
  - Entry: TSAP address: address of the endsystem + port
- Request via broadcast
  - e.g., as correlation of Ethernet address and internet address
  - i.e., possible in geographically and topologically limited spaces

Initial Situation:

- network has
  - varying transit times for packets
  - certain loss rate
  - storage capabilities
- packets can be
  - manipulated
  - duplicated
  - resent by the original system after timeout

In the following, uniform term: “Duplicate”

- a duplicate originates due to one of the above mentioned reasons and
- is at a later (undesired) point in time passed to the receiver

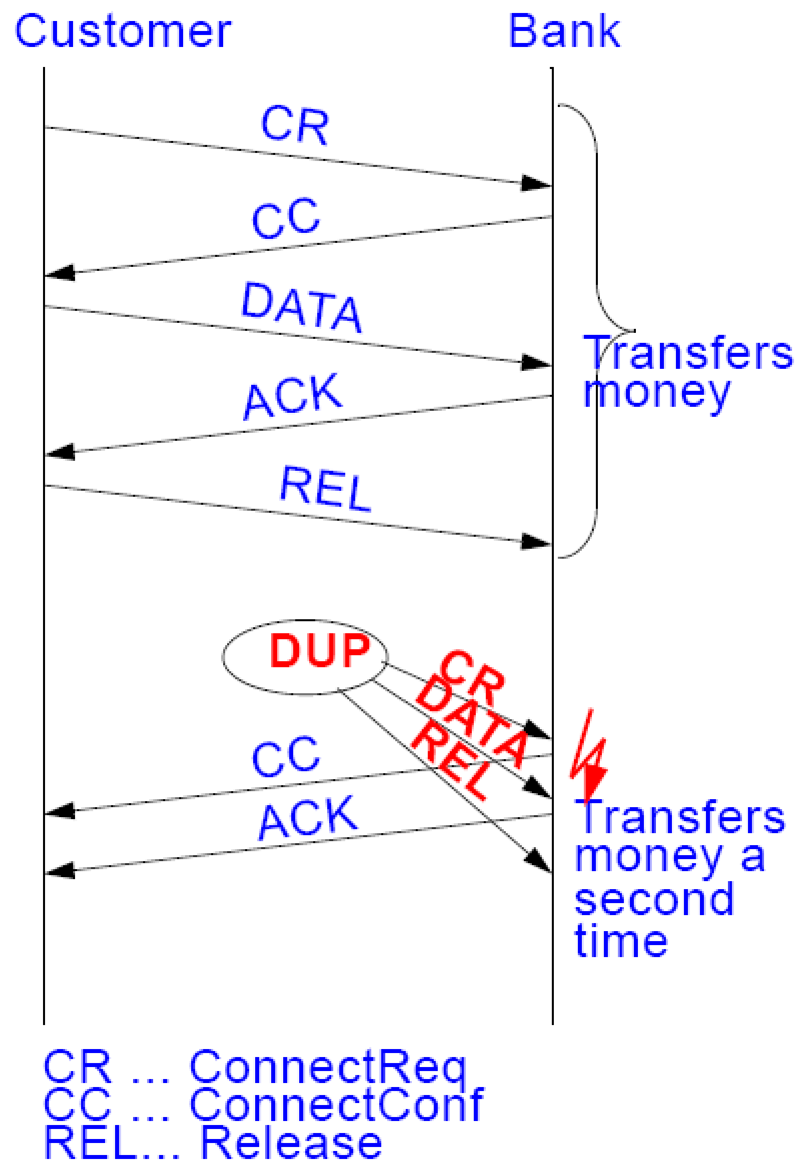
3.1 Basic Challenges - Example

E.g. description of possible error causes and their possible consequences (5 steps)

- due to network capabilities
  - duplication of sender's packets
  - subsequent to the first 5 packets duplicates are transferred in correct order to the receiver
  - also conceivable is that an old delayed DATA packet (with faulty contents) from a previous session may appear; this packet might be processed instead of or even in addition to the correct packet

Result:

- without additional means the receiver cannot differentiate between correct data and duplicated data
- would re-execute the transaction

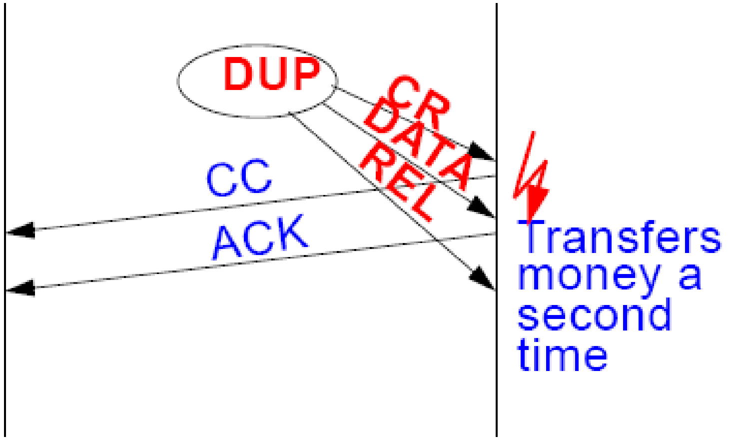


3 somehow disjoint problems

1. how to handle duplicates WITHIN a connection?

2. what characteristics have to be taken into account regarding
   - consecutive connections or
   - connections which are being re-established after a crash?

3. what can be done to ensure that a connection that has been established …
   - has actually been initiated by and with the knowledge of both communicating parties?
   - see also the lower part of the previous illustration



3.2 Basic Methods of Resolution

1. to use temporarily valid TSAPs

- method:
  - TSAP valid for one connection only
  - generate always new TSAPs
- evaluation
  - in general not always applicable:
  - process server addressing method not possible, because
    - server is reached via a designated/known TSAP
    - some TSAPs always exist as “well-known”

2. to identify connections individually

- method
  - each individual connection is assigned a new SeqNo and
  - endsystems remember already assigned SeqNo
- evaluation
  - endsystems must be capable of storing this information
  - prerequisite:
    - connection oriented system (what if connection-less?)
  - endsystems, however, will be switched off and it is necessary that the information is reliably available whenever needed

Duplicates – Methods of Resolution

TECHNISCHE UNIVERSITÄT DARMSTADT

3. to identify PDUs individually:

individual sequential numbers for each PDU

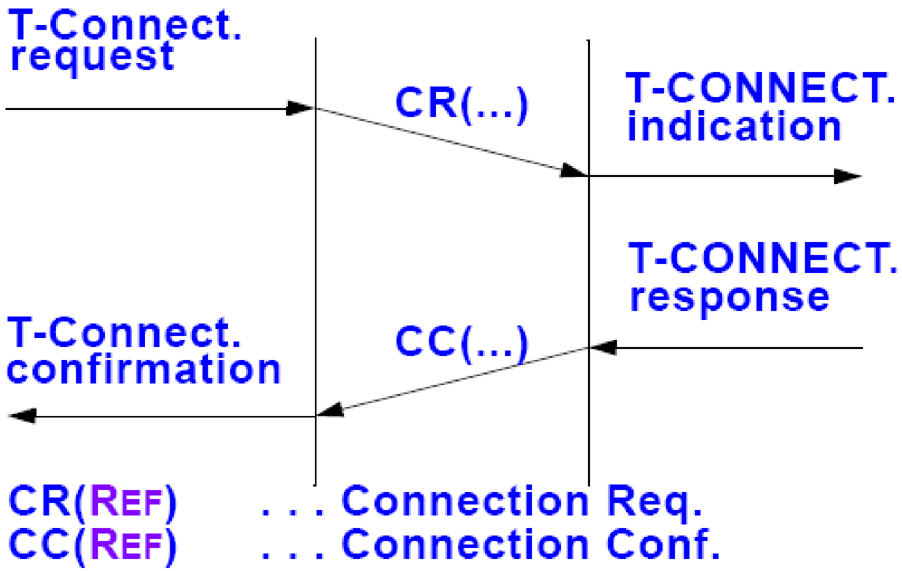
- method
  - SeqNo basically never gets reset
  - e.g. 48 bit at 1000 msg/sec: reiteration after 8000 years

- evaluation
  - higher usage of bandwidth and memory
  - sensible choice of the sequential number range depends on
    - the packet rate
    - a packet's probable "lifetime" within the network

Connection

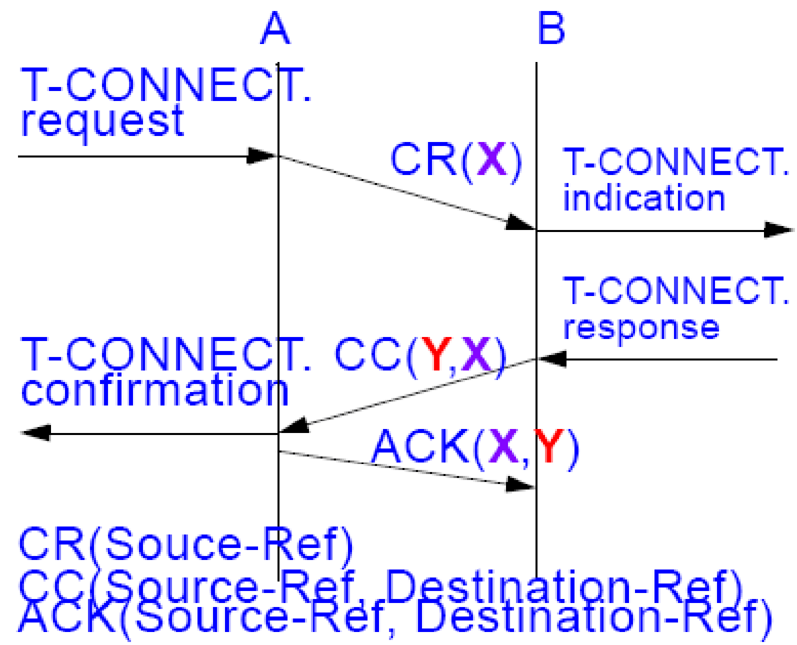
- see also
  Connection Oriented Service:
  State Transition Diagram

- by simple protocol
  - approach using 2 messages
    (2 phases)
    - problems may occur due to delayed duplicates
    - compare with previous example (bank transaction)



Connect: Three-way Handshake Protocol

Principle



1. CR: Connect Request

- initiator (A) sends request with
  - SequenceNo (X)
    - selected by sender

2. CC: Connect Confirmation

- receiver (B) responds with
  - sequence number transmitted by the
    - initiator (X) and
  - (randomly) selected sequence
    - number (Y) by receiver
      - while observing the previously
        - discussed criteria for selection,
        - in order to avoid a collision with delayed duplicates

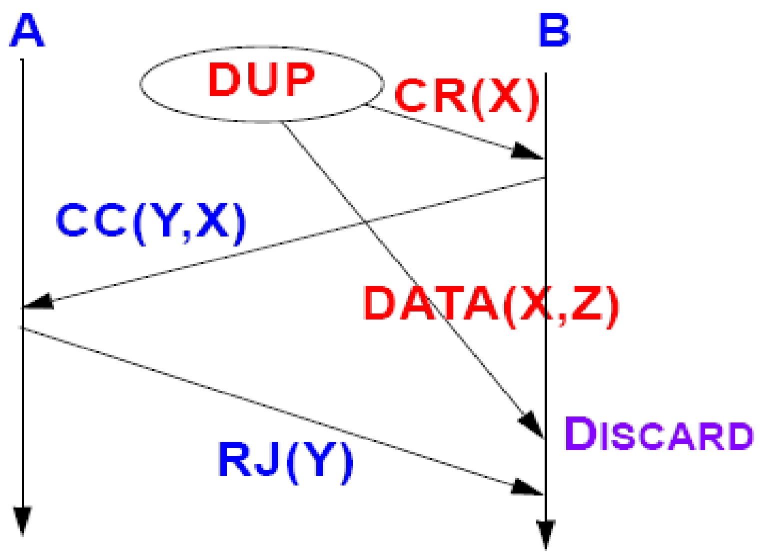
3. Acknowledgment

- initiator (A) acknowledges
  - sequence numbers X, Y (as received before)
- after receiving a valid ACK, receiver (B) accepts data

Note:

- some protocols (including TCP) acknowledge the next byte expected
  - (ACK X+1,Y+1), not the last byte received

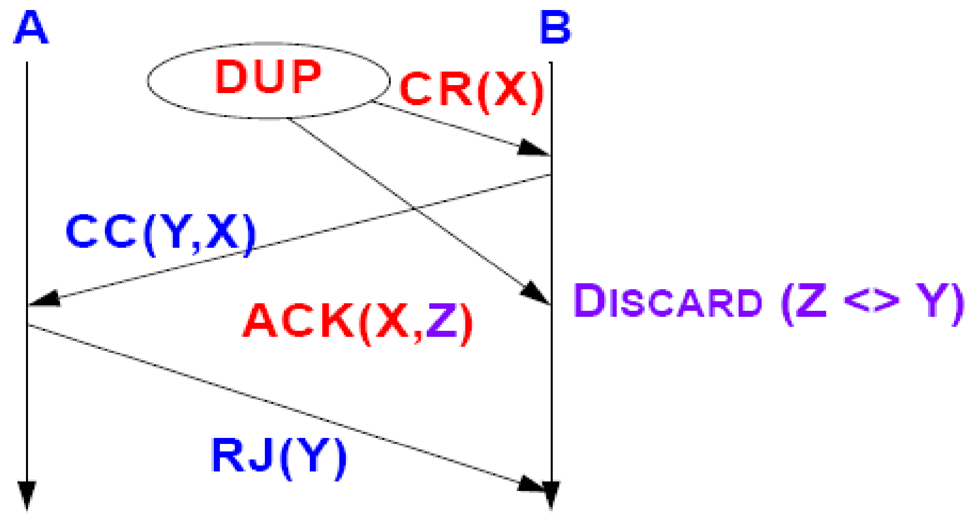
Three Way Handshake Protocol: Results



CR and data duplicate

- duplicated data is discarded
  - for success
    - should have occurred an ACK (X,Z) before

Three Way Handshake Protocol: Results

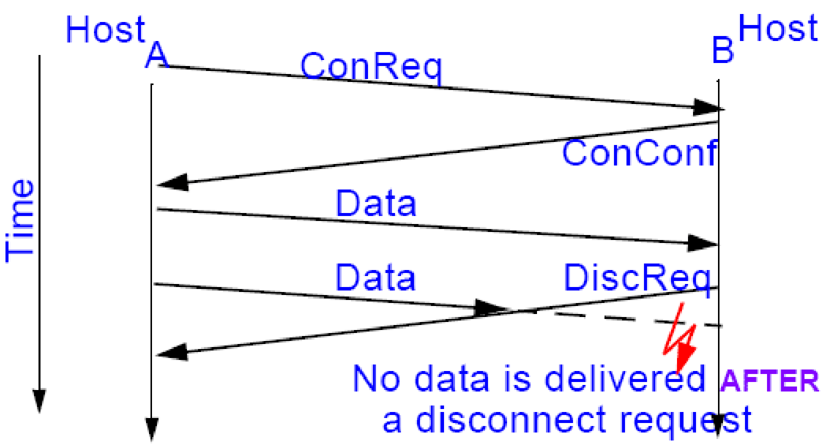


Connect Request CR Duplicate and Acknowledgment ACK Duplicates

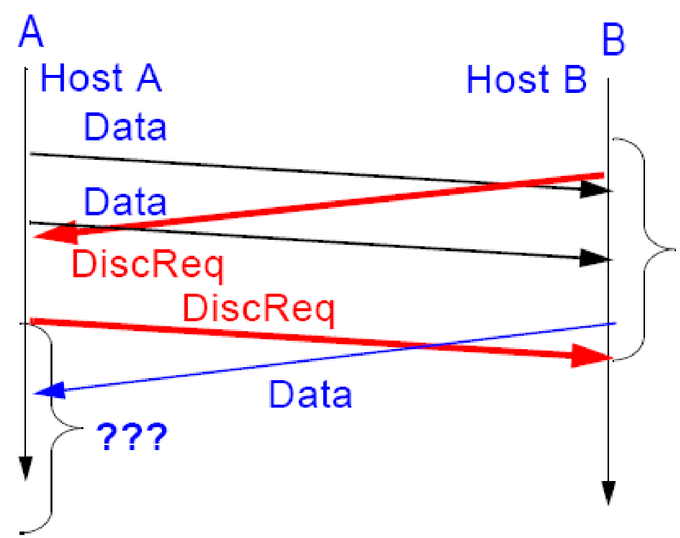
- AK (X,Z) discarded because
  - AK (X,Y) expected
  - AK (X,Z) received, Z <> Y
    - B will be ensured by a premise of a maximum packet lifetime by selecting the initial sequence number according to the described algorithms

Two alternatives

asymmetric disconnect



symmetric disconnect



Joint characteristics (flow control on data link layer)

- fast sender shall not flood slow receiver
- sender shall not have to store all not acknowledged packets

Differences (flow control on data link layer)

- L2-DLL: router serves few connections to other routers
- L4-TL: endsystem contains a multitude of
  - connections
  - data transfer sequences
- L4-TL: receiver may (but does not always have to) store packets

Strategies

- E.g.
- credit mechanism / dynamic buffer allocation

Credit Mechanism

Flow control

- credit mechanism

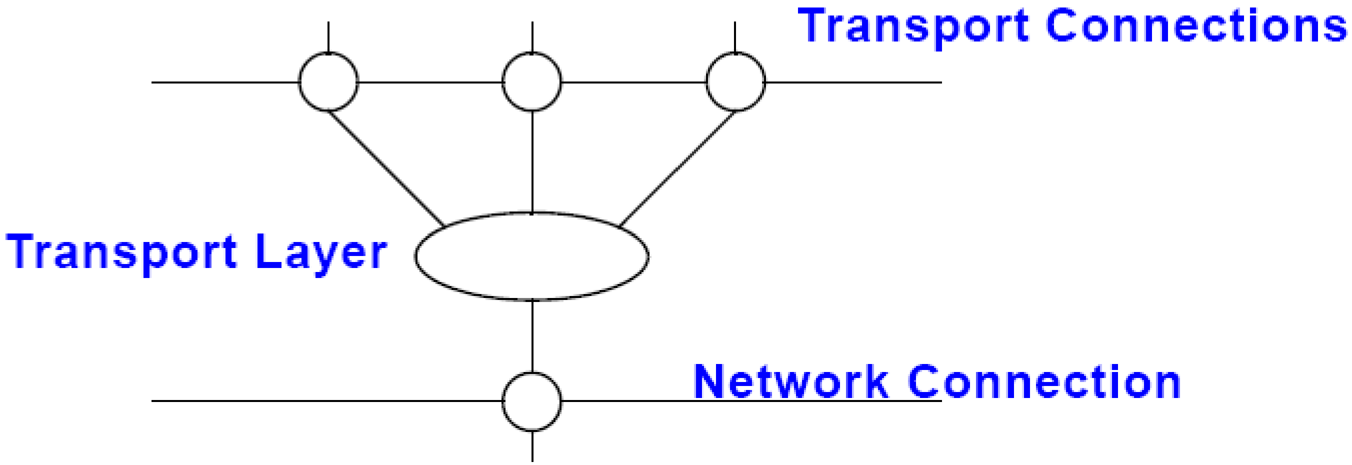
Buffer reservation

- receiver allocates buffers dynamically for the connections
- allocation depends on the current situation

Principle

- sender requests required buffer amount
- receiver reserves as many buffers as the current situation permits
- receiver returns ACKs and buffer-credits separately
  - ACK: confirmation only (does not imply buffer release)
  - CREDIT: buffer allocation
- sender will be blocked, when all credits have been used up

7 Multiplexing / Demultiplexing



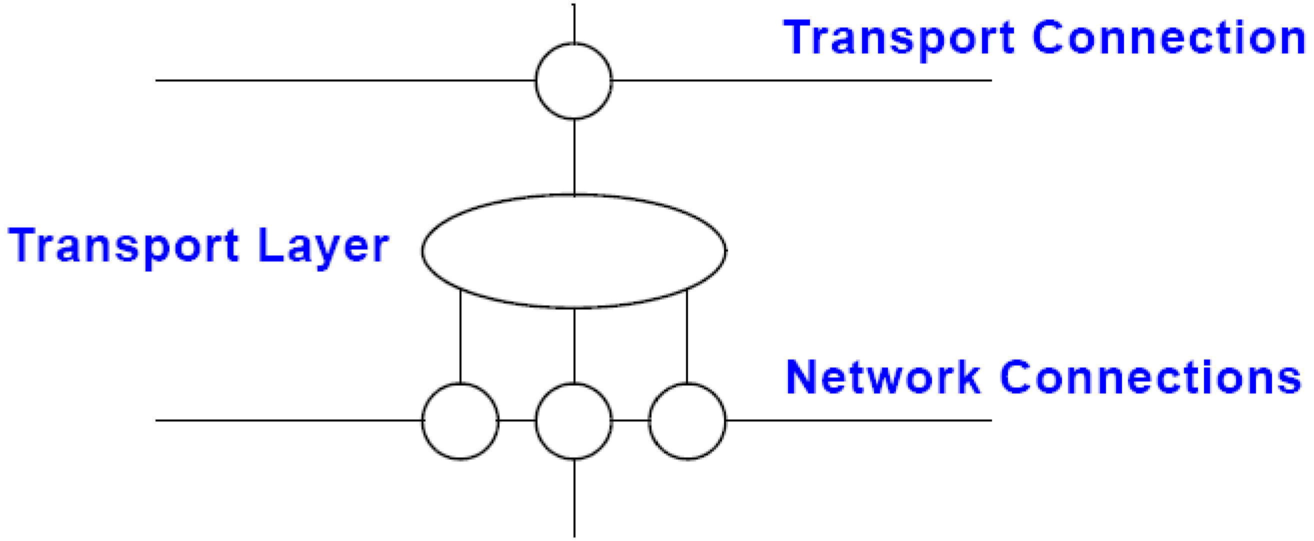
Application

- minimizing costs when num. of connections/ connection time represents the main cost factor

Multiplexing function

- grouping of T connections by destination address
- each group is mapped to the minimum number of network connections
  - too many L4-T connections per L3-N connection
    - ➔ possibly poor throughput
  - too few T connections per N connection
    - ➔ possibly transfer costs too high

Splitting / Recombination



Application:

- implementation of T connections with high bandwidth

Splitting function

- distributing the TPDUs onto the various network connections
- usual algorithm: Round Robin

Comment

- also known as “upward” multiplexing