Network Principles

Circuit Switching, Packet Switching,
Datagram versus Virtual Call Service
OSI Model

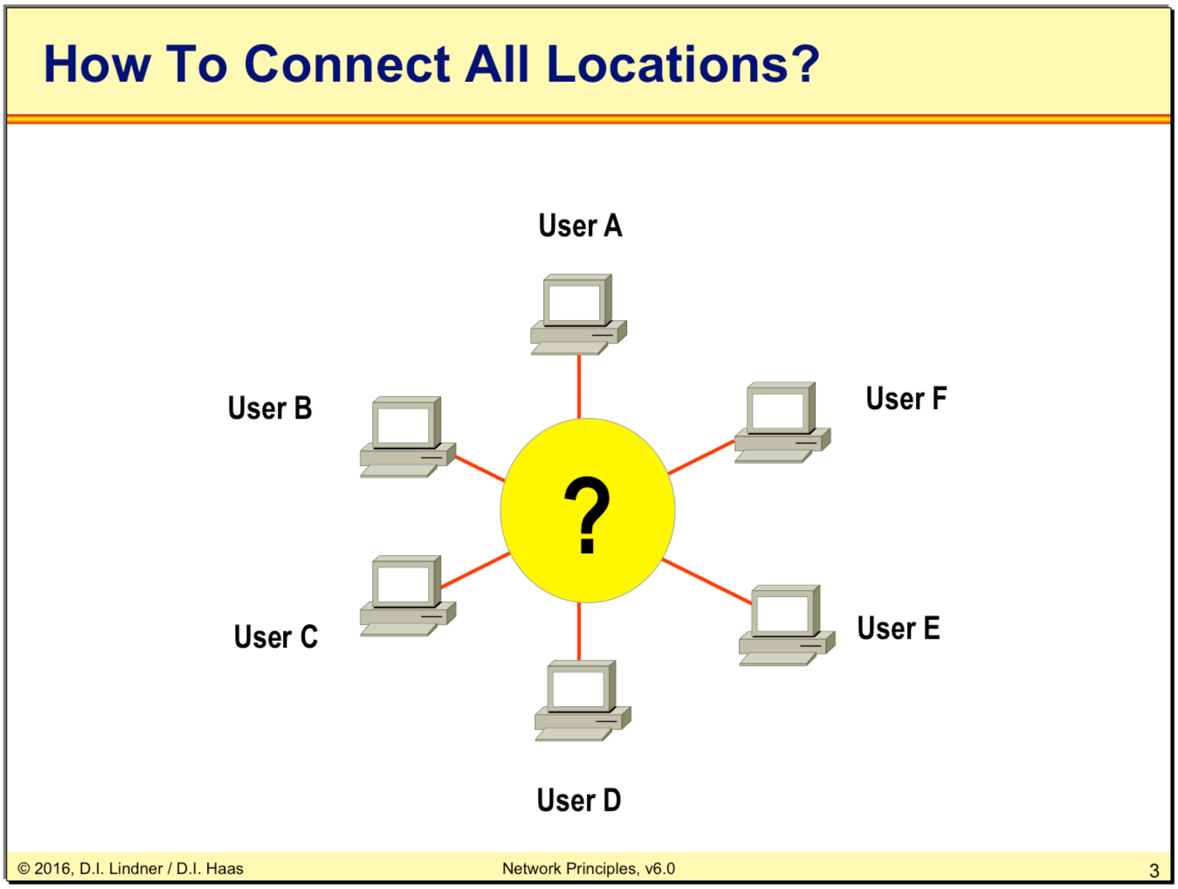
© 2016, D.I. Lindner / D.I. Haas

Agenda

- Introduction
- Circuit Switching
- Packet Switching
  - Principles
  - Datagram Service
  - Virtual Call Service
- OSI Reference Model
- Summary of Network Methods

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 2

© 2016, D.I. Lindner / D.I. Haas



Lecture chapters about line protocols and TDM techniques have explained

1) how communication between two devices can be implemented over a point-to-point physical line using line protocol techniques

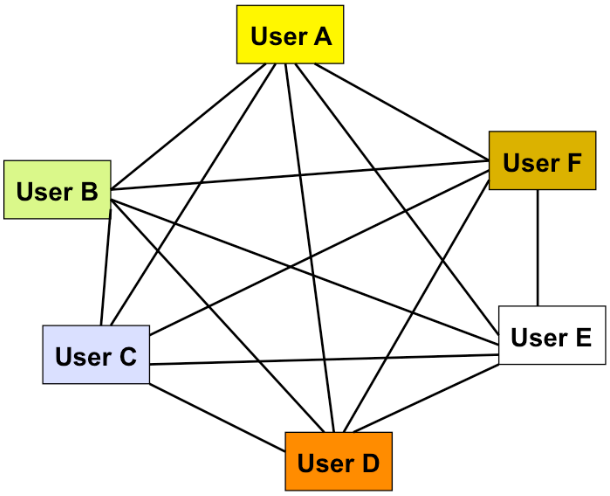
2) how TDM can be used to provide several communication channels between devices located on two locations

Open question:

How should devices to be connected and how should communication between devices be organized, if there are many devices at different locations?

Easy solution would be an any to any topology (fully meshed) establishing multiple point-to-point lines between devices using line protocol techniques on every point-to-point line.

Networking: Fully Meshed



- Metcalfe's Law: n(n-1)/2 links
- Good fault tolerance
- Expensive

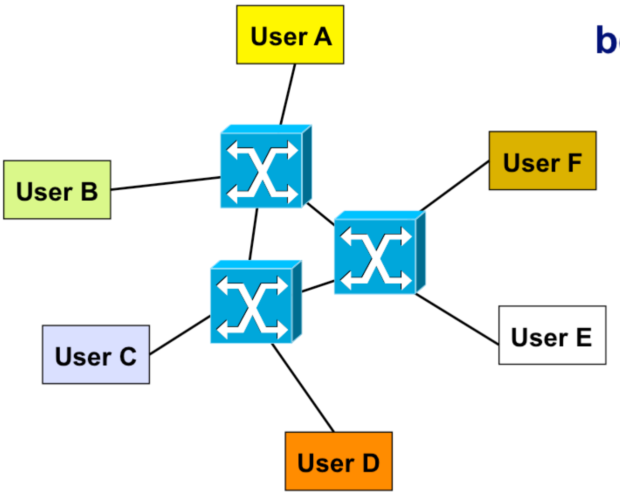
A fully meshed network is a thing that everybody wants, because it gives 100% redundancy and optimized data transport to each destination. But unfortunately only very few can effort it, because the costs of network infrastructure would grow with Metcalfe's law.

Which is expressed by the formula n x (n-1)/2 . This means if you have ten sites you want to connect in an any to any topology you would need 45 connections. If number of sites increases you will get a scalability problem.

Why is any-to-any topology very expensive?

Many lines are required and hence large number of transmission equipment (like modems, DSUs, line repeaters, etc.) is necessary. Also many physical communication ports are required in devices which may lead to a space problem.

Networking: Switching



Network switches could be based on:

Synchronous TDM

- **Circuit Switching**

or

Asynchronous TDM

- **Packet Switching**

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 5

One way to save costs would be the use of network switches, which are responsible for handling the traffic between the different destinations.

The switches may use a technology either based on synchronous (deterministic) or asynchronous (statistical) TDM. In this case we would need only six small range links and three long rate line instead of fifteen links to establish communication between all sites. TDM multiplexers introduced in the TDM techniques chapter are now used in a network environment instead of a point-to-point environment only. Now networking means for the TDM multiplexer to have more than use 1 trunk port. In our example every switch has two trunk ports.

By using synchronous or asynchronous time division TDM in a network environment two fundamental network principles were created over time:

Circuit switching based on synchronous (deterministic) TDM

Packet switching based on asynchronous (statistical) TDM

We will cover these fundamental network principles in different levels of details throughout the whole data communication lectures.

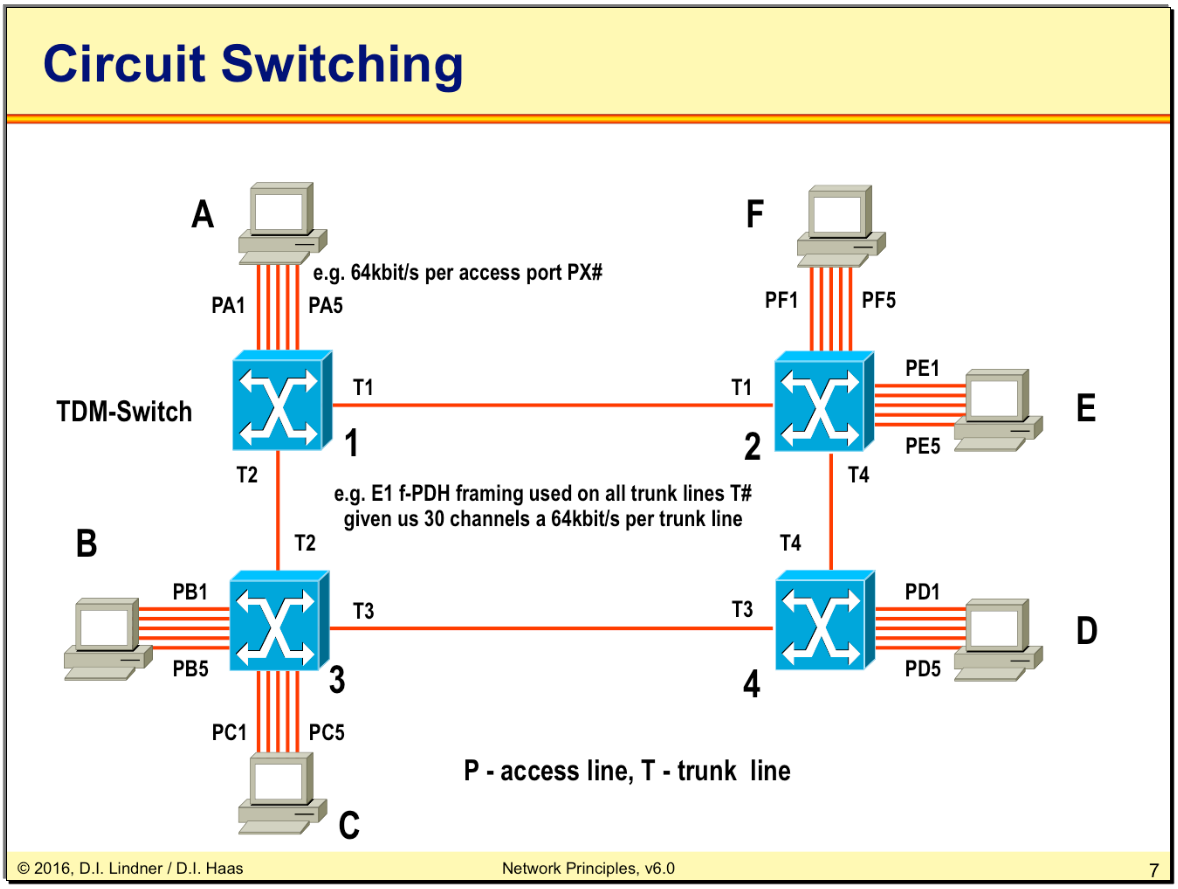
© 2016, D.I. Lindner / D.I. Haas

Agenda

- **Introduction**
- **Circuit Switching**
- **Packet Switching**
  - Principles
  - Datagram Service
  - Virtual Call Service
- **OSI Reference Model**
- **Summary of Network Methods**

© 2016, D.I. Lindner / D.I. Haas

© 2016, D.I. Lindner / D.I. Haas



Circuit switching technology is based on synchronous (deterministic) TDM.

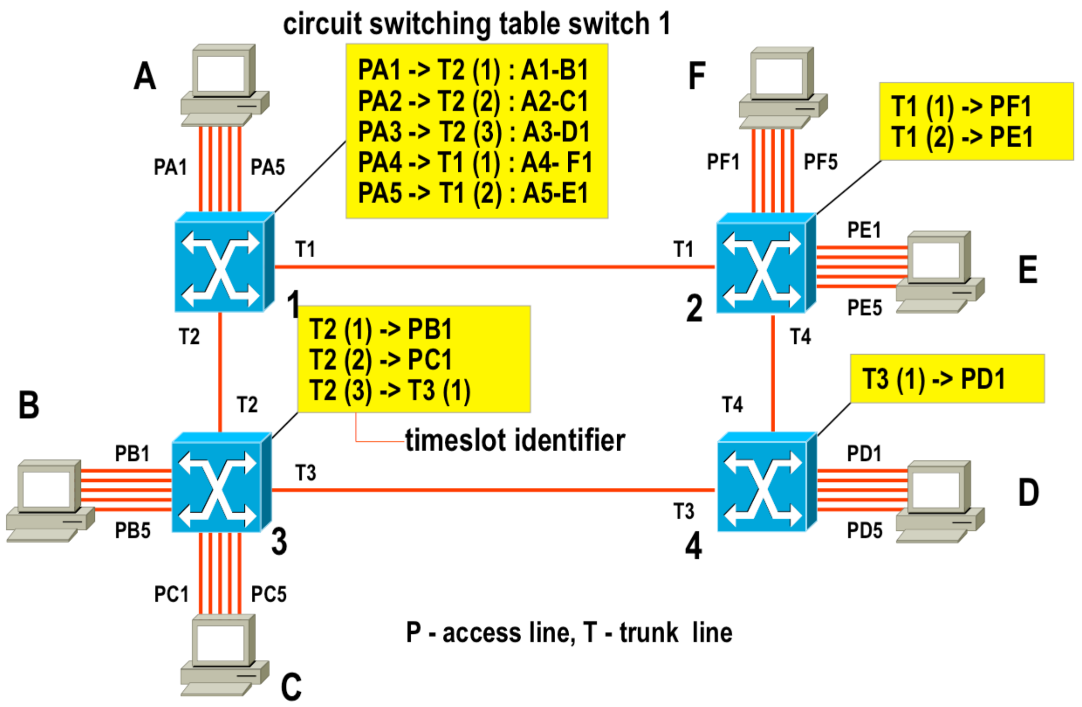
The principle of circuit switching:

Physical communication ports (P.X) of devices are connected locally to synchronous TDM switches. Trunk lines T between switches use synchronous time division multiplexing e.g. standardized E1 (31 timeslots with 64kbit/s each). Each physical port is assigned a timeslot on an outgoing trunk for communication with a remote device. Switches map timeslots on incoming trunks either to local ports or to timeslots on outgoing trunks (in such a case the TDM switch act as transit switch). Mapping information is stored in circuit switching tables.

Circuit switching and synchronous TDM on trunk lines reduce the number of expensive wide area lines required in a fully meshed topology. Synchronous TDM switches - by having more than one trunk link - form a synchronous TDM network.

© 2016, D.I. Lindner / D.I. Haas

Connections of Device A



All network switches in circuit switching technology hold a switching table which determines the correlation between

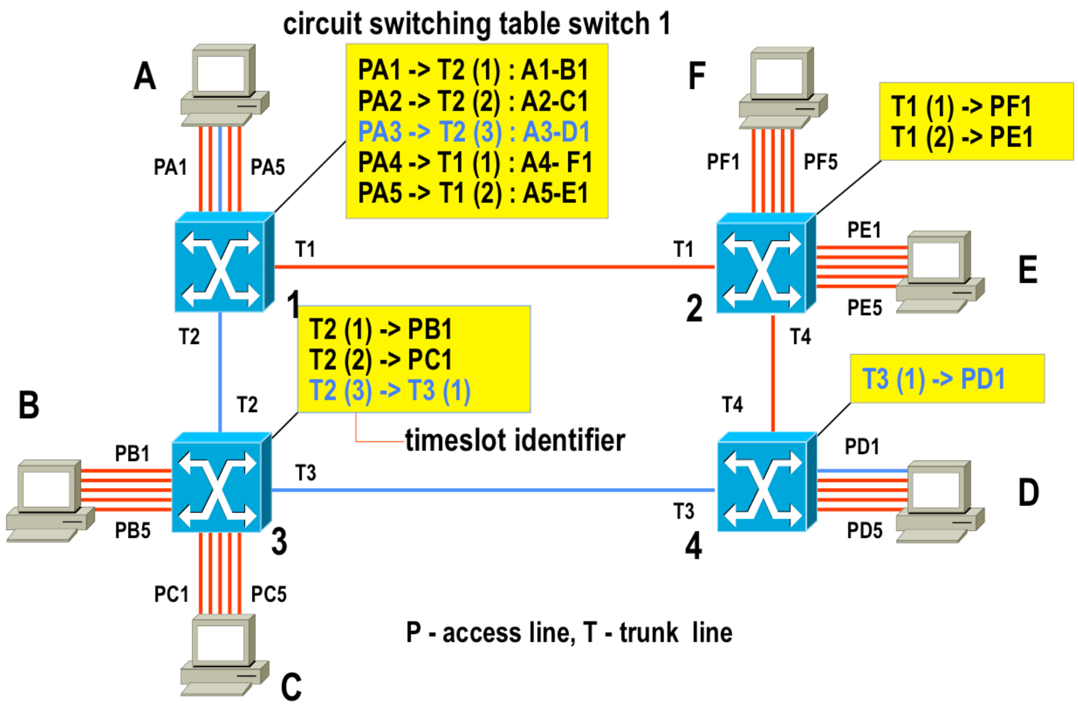
1) incoming access port and outgoing trunk port/timeslot at the source of a communication channel

2) incoming trunk/timeslot and outgoing trunk/timeslot in case of a transit switch

3) incoming trunk/timeslot and outgoing access port at the destination of a channel

Connections of Device A

Circuit Path PA3 (A) - PD1 (D)



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 9

In our example the connection between Port PA3 and PD1 is established by three network switches and their according switching tables. For both users this connection looks like a dedicated point to point link, they are not aware what's going on inside the network cloud.

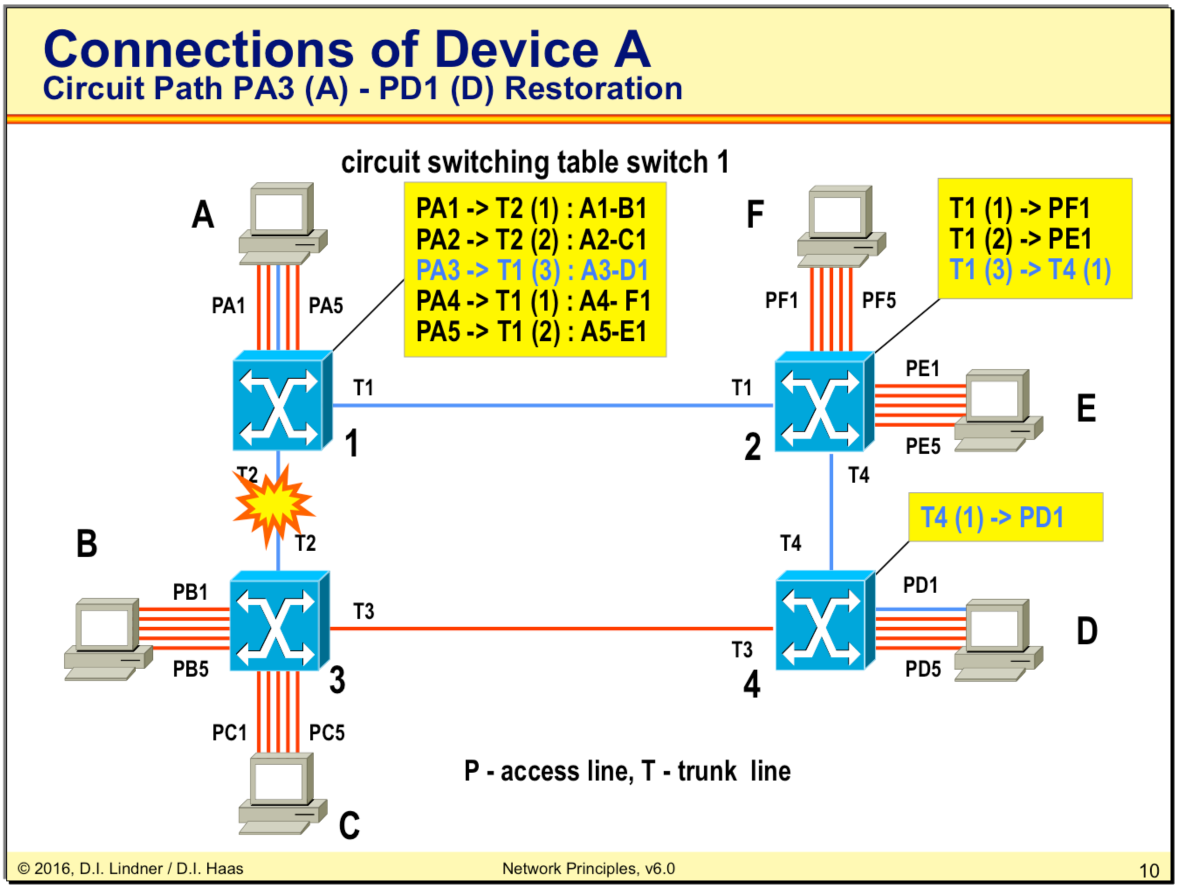
In analogy to the good old patch panel used by the telephone operator at a local telephone exchange to physically connect a local incoming telephone line either to one free outgoing trunk line or to another local user the connection between two devices is called circuit.

The path of a communication channel (circuit) between two devices is marked by corresponding entries in circuit switching tables. In our example shown by the blue lines in the drawing.

In our example redundancy may be used to split channels over separated physical paths to avoid interruption of communication for all channels in case of a single point of failure (e.g. a single trunk line get down). By appropriate changing the maps in case of such failure restoring of broken circuits is possible if not all timeslots along the redundant path are already occupied by other channels.

© 2016, D.I. Lindner / D.I. Haas

L04 - Network Principles (v6.0)



Pictures shows how circuit can be restored by using free timeslots and changed mapping tables along the redundant trunk lines.

Circuit Switching – Facts

- **Based on synchronous (deterministic) TDM**
  - Minimal and constant delay
  - Protocol transparent
  - High bit rate on trunk lines
    - Sum of I access links traversing a given trunk
  - Possibly bad utilization
    - Idle pattern in timeslots if no data present
  - Good for isochronous traffic (voice)
- **Switching table entries**
  - Static (manually configured)
  - Dynamic (signaling protocol)
  - Scales with number of connections!

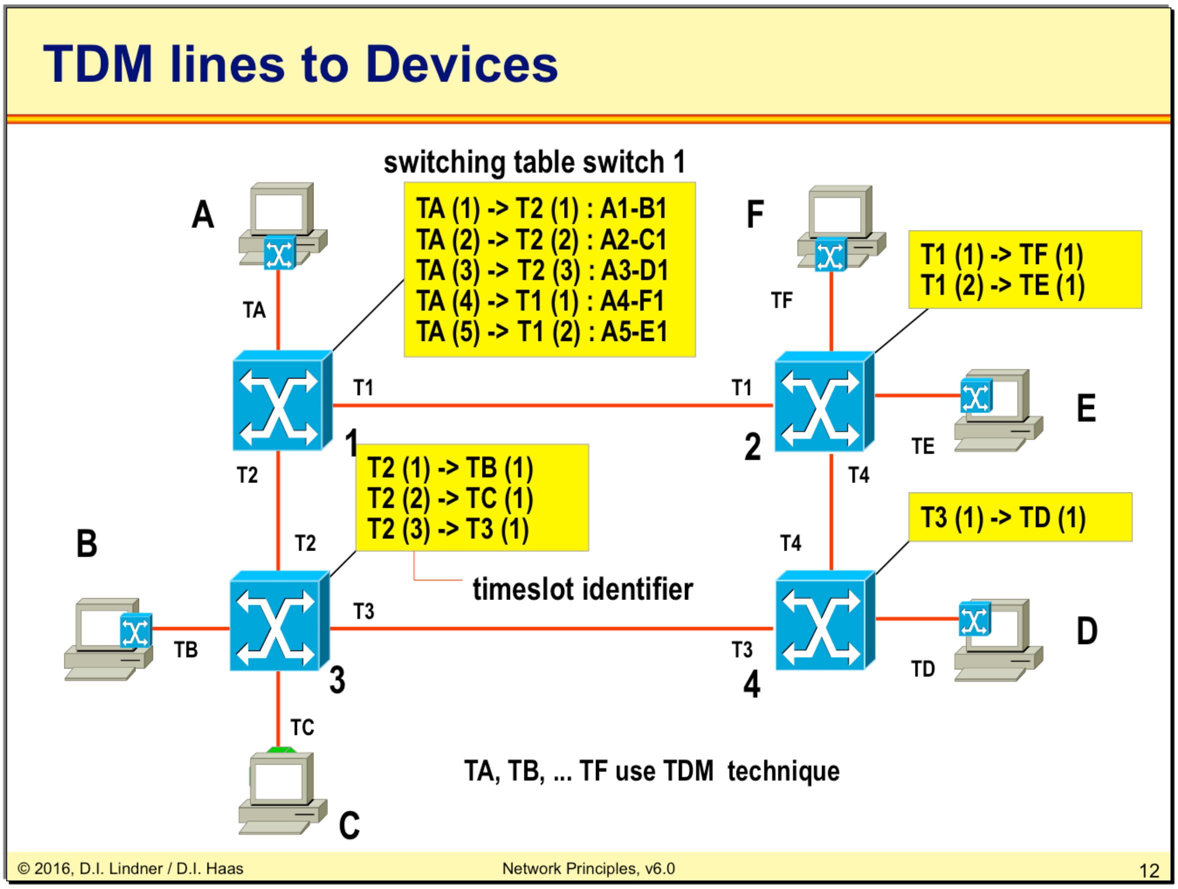
© 2016, D.I. Lindner / D.I. Haas

Circuit switching based on deterministic TDM has minimal fixed delay, is protocol transparent, but may have bad network utilization due to currently unused connections.

So circuit switching is very well suited for isochronous traffic like voice communication or video conferencing. Circuit switching is the typical technology that is used by Telco's.

The switching table entries which are needed for proper data forwarding might be generates manually by the help of some network management software or dynamically by some signaling protocol.

© 2016, D.I. Lindner / D.I. Haas



The number of local physical ports can be further reduced by using synchronous TDM between a device and the local switch too. One physical access line may carry many logical channels in corresponding timeslots and hence a mapping between these timeslots can be done in the same way as was already shown for trunk lines.

Handling Of Circuit Switching Table

- **Static**
  - Entries are configured by TDM network administrator
  - **Permanent circuit service**
- **Dynamic (fail-safe)**
  - Entries are changed automatically by TDM network management protocol to switch over to a redundant path in case a trunk line breaks
  - **Soft permanent circuit service**
- **Dynamic (on demand)**
  - End-systems use a signaling protocol to local TDM switch in order to transport setup or tear down requests
  - TDM switches establish path (corresponding entries in circuit switching tables) using their own signaling protocols
  - **Switched circuit service**

© 2016, D.I. Lindner / D.I. Haas

© 2016, D.I. Lindner / D.I. Haas

Circuit Switching Data Networks

- **Network providers offer permanent circuit services**
  - With permanent entries in circuit switching tables
  - Optional with fast automatic switchover (50ms) in case of trunk failure
  - **Leased line**
- **Network providers offer switched circuit services**
  - With dynamic entries in circuit switching tables generated on demand
  - Today implementations are based on **ISDN** only
  - **Integrated Services Digital Network**
  - Outband signaling via D-channel) between ISDN end system (ISDN-TE) and ISDN-LE (Local Exchange) = that is the local TDM switch
  - Communication between ISDN-LEs is based on Signaling System 7 (SS7)
- **Base is PDH or SDH transmission infrastructure**

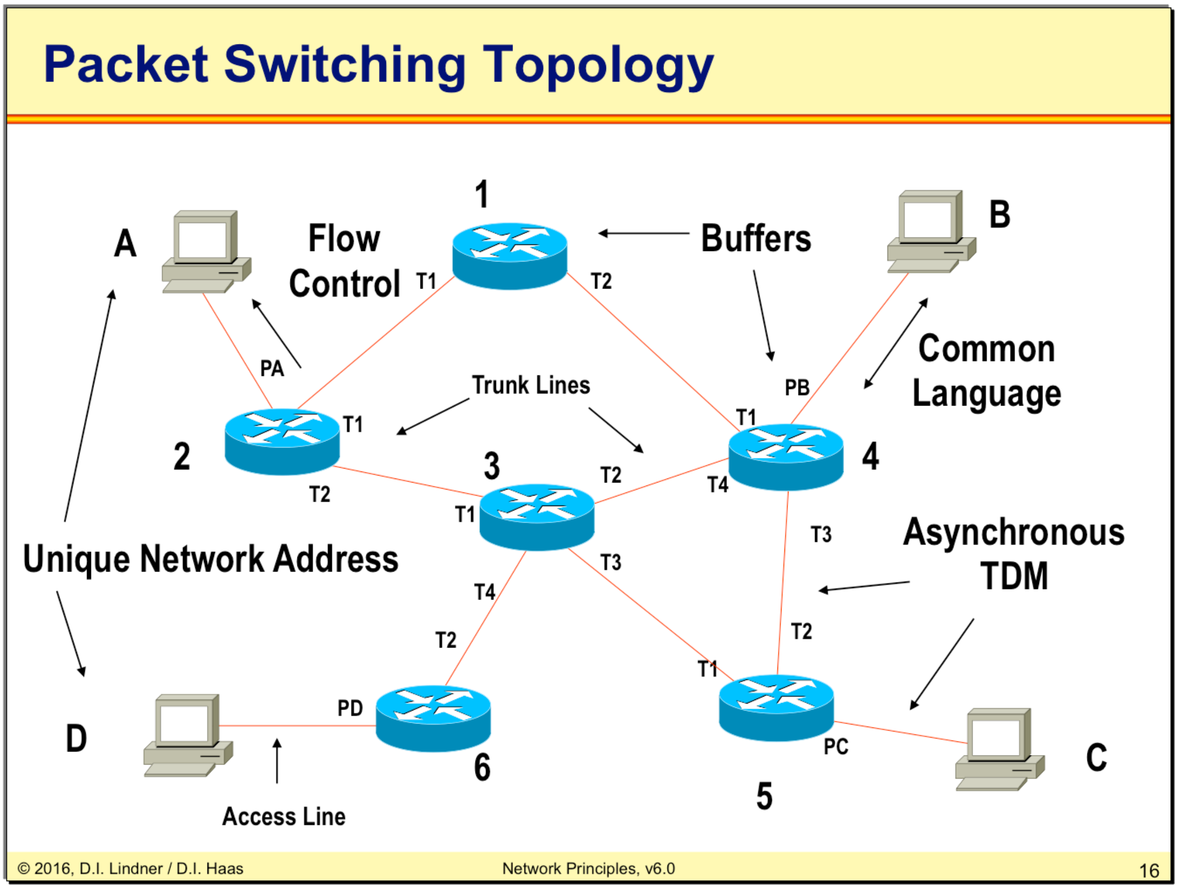
© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 14

Base for all these services is an underlying PDH or SDH infrastructure for transporting channels over geographic wide areas.

Agenda

- Introduction
- Circuit Switching
- Packet Switching
  - Principles
  - Datagram Service
  - Virtual Call Service
- OSI Reference Model
- Summary of Network Methods

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 15



Packet switching technology is based on asynchronous (statistical) TDM usage on every single link of a network topology consisting of packet switches as intermediate systems und user devices as end-systems. Packet switches are interconnected by trunk lines. End-systems are connected via access links to their local packet switch. Quite a lot of different transmission technologies were used on access and trunk links like V.24/V.28, X.21, PDH, SDH, ATM and LAN. Today most links are implemented by using Ethernet technology.

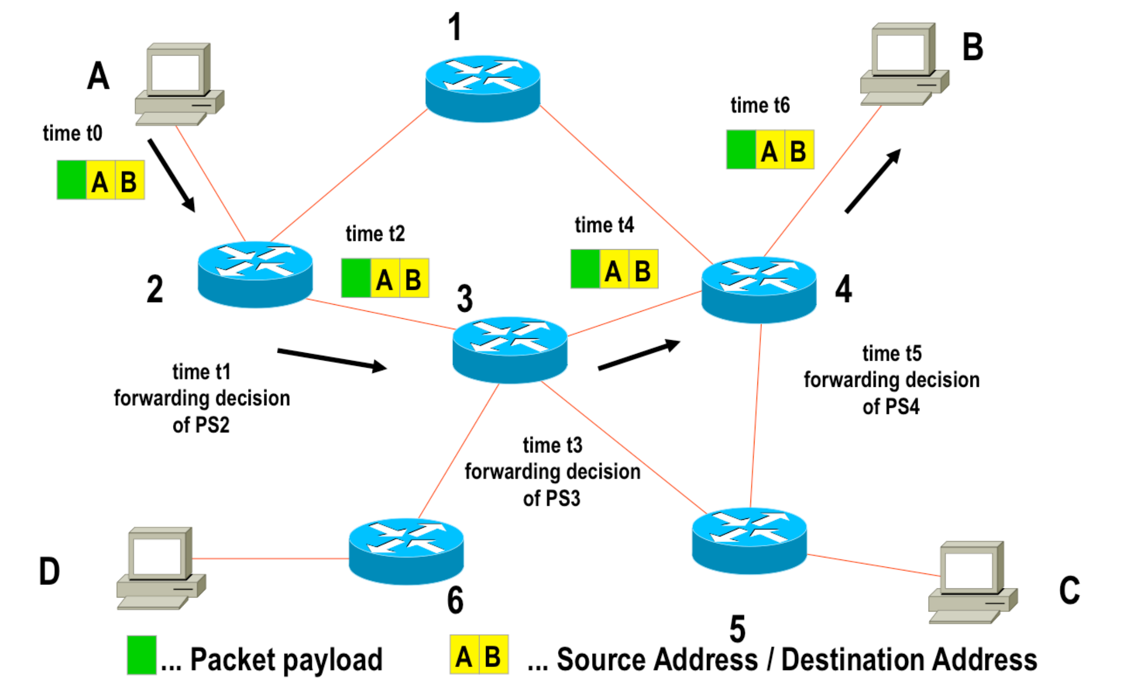
Usage of statistical TDM on trunks and access line allows many end systems to communicate without exclusively reserving capacity on a trunk or access line. There is no correlation between timeslot and a destination device like in circuit switching. Therefore we need explicit addressing information in such a network. We will find source and destination address in every packet which should be transported over the network. Each packet switch must analyze the destination address of every packet to be able to forward it according to some forwarding table.

Of course statistical TDM on trunks and access lines avoids again a large number of physical point-to-point lines which would be required in a pure any-to-any topology. Also the bit rate on trunk lines is not the sum of the access links as in circuit switching but should be calculated in such a way to carry the statistical average traffic between all end-systems. Statistical TDM requires a protocol between end-systems and packet switches because of addressing and optional flow control between end-system and packet-switches. Therefore the method is <u>not</u> protocol-transparent, end-system must speak the language of the packet switch. You see that packet switching inherits all the features of asynchronous TDM including variable delay, buffering and so on.

Redundant trunk lines can provide redundant paths in case of failure or can be used for load

© 2016, D.I. Lindner / D.I. Haas

Packet Forwarding Principle (Store and Forward)



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 17

End-systems break information in small pieces called packets and deliver these packets to their corresponding local packet switch.

As you can see in the picture packets contain addressing information (A is the source address, B is the destination address in the given protocol).

Packet switches buffer incoming packets, use the address information of the packet to decide where to forward them, put packets in outgoing queues after the decision is done. Finally they transmit all packets waiting in queues - packet by packet - on access and trunk links. We call that behavior **"store and forward"**.

Packet Forwarding is based on Tables

- Tables contains
  - Information how to reach destinations
  - Mapping between destination address or local connection identifier and outgoing trunk or access port
  - "**Signposts**"

- Two types of tables
  - Depending on the actual implementation of packet switching technology
  - **Routing tables**
  - **Switching tables**

Routing - Addressing

- **Routing in packet switched networks**
  - Process of path selection in order to forward a packet to a given destination
  - Selection based on (destination) address
- **Address specifies the location of end system**
  - Contains topology information
  - Address must be unique within the network in order to enable routing based on signposts
- **Protocol using unique and structured addresses**
  - Is called **routed** or **routable protocol**

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 19

Protocols which use unique but unstructured addresses are non-routable. We will see later an example for such protocols in the Ethernet Transparent Bridging (which is connectionless packet switching on OSI layer 2) and MAC address world.

© 2016, D.I. Lindner / D.I. Haas

Routing Types

- **Static routing**
  - Routing table entries are static
  - Based on preconfigured routing tables
  - Configuration done by the network administrator
  - Non-responsive to network topology changes
- **Dynamic routing**
  - Routing table entries are variable (dynamic)
  - Changes done by **routing protocols**
    - Communication protocol used between packet switches to find out the network topology and to calculate the best path to any given destination
  - Responsive to any network topology changes

© 2016, D.I. Lindner / D.I. Haas

Dynamic routing is based on a distributed routing processes and communication between theses processes which run on packet switches as administrative task.

The communication is implemented by so called routing protocol.

The task of a routing protocol is used

1.) to find out the network topology

2.) to calculate all possible paths to a given location

3.) to select one path (best path) in case of redundancy

4.) and finally to store this best path as a signpost used for packet forwarding into the routing table

The basic problem of routing is to keep the routing tables (distributed database) consistent.

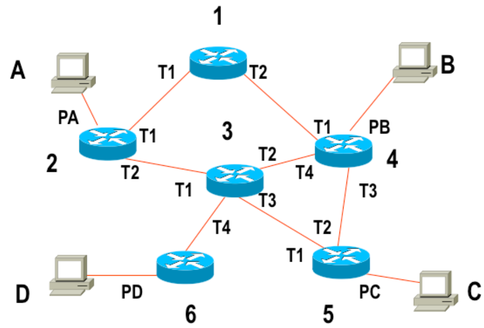
With static routing it is the task of network administrator.

With dynamic routing it is the task of routing protocol algorithm although there may exist some inconsistency during times of network convergence (time which is needed to implement network changes by the routing techniques into all routing tables).

© 2016, D.I. Lindner / D.I. Haas

Routing Table

- **Example of a static routing table of packet switch 2**



Address of destination	*incoming line*	outgoing line	next PS
B	*PA, T1*	T2	PS 3
D	*PA,T1*	T2	PS 3
C	*PA, T1*	T2	PS 3
A	*T1 or T2*	PA	local

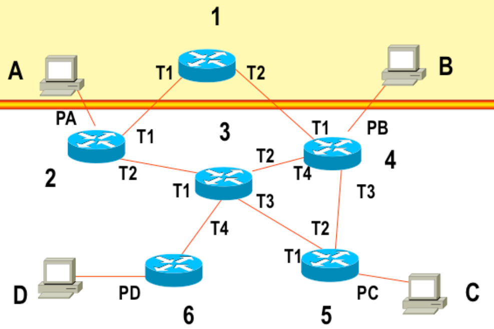
© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 21

Here you can see the signpost principle of a routing table: Based on the destination address packet switch 2 forwards a packet to an outgoing line leading to a next hop device (either packet switch or end-system).

© 2016, D.I. Lindner / D.I. Haas

Routing Table

- **Example of a static routing table of packet switch 3**



Address of destination	*incoming line*	outgoing line	next PS
B	*T1*	T2	PS 4
C	*T1*	T3	PS 5
D	*T1*	T4	PS 6
B	*T2*	kill	- - -
B	*T3*	T2	- - -
B	*T4*	T2	- - -
C	*T2*	T3	PS 5
C	*T3*	kill	- - -
.....	*.....*	.....	.....

Here you can learn that sometimes it may be necessary for a packet switch to “kill” (silently discard) a packet if it comes from the wrong direction.

For example, if packet destined for B arrives on packet switch 3 ob incoming trunk line T2, it should be killed - otherwise a loop could occur. Reason for that: The packet arrives on a trunk which is used by switch 3 for forwarding packets to destination B in the opposite direction.

© 2016, D.I. Lindner / D.I. Haas

Routing Table Usage / Type of Service

- **Routing tables are differently used**
  - Depending on the type of service of the packet switching network
- **Packet switched networks based on**
  - Connectionless Service (CL) - Datagram service
    - Routing tables are used to forward all kind of packets
  - Connection-oriented Service (CO) – Virtual Call service
    - Routing tables are used to forward control packets for connection establishment
    - These control packets generate entries in switching tables
    - After connection establishment, only the switching tables are used to forward data packets

© 2016, D.I. Lindner / D.I. Haas

Technology Differences - Summary

- **Datagram Principle**
  - Global and routable addresses
  - Connectionless
  - Routing table for forwarding of packets
- **Virtual Call Principle**
  - Local addresses
  - Connection-oriented
  - Routing table for setup of connections
  - Switching table for forwarding of packets

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 24

There are two major technologies that make use of the statistical TDM principle.

The datagram principle which is using global unique and routable addresses. Data forwarding decisions are made by statically or dynamically generated routing tables and the data transport is connectionless. Examples for the Datagram principle are IP, IPX, Appletalk, etc.

The Virtual Call principle uses locally significant address well known under the term virtual circuit identifier. The data transport is done connection-oriented and the forwarding decisions are made by switching tables. The switching tables hold the information about incoming trunk/circuit identifier and the corresponding outgoing trunk/circuit identifier. Examples for Virtual Call services are X25, Frame-relay, ATM, etc.

© 2016, D.I. Lindner / D.I. Haas

Agenda

- **Introduction**
- **Circuit Switching**
- **Packet Switching**
  - Principles
  - Datagram Service
  - Virtual Call Service
- **OSI Reference Model**
- **Summary of Network Methods**

© 2016, D.I. Lindner / D.I. Haas

Datagram Service Principles

- **Connectionless service**
  - Packets can be sent without establishing a logical connection between end systems in advance
  - Packets have no sequence numbers
  - They are called **“Datagrams”**
- **Every incoming datagram**
  - Is processed independently regarding to all other datagrams by the packet switches
- **The forwarding decision for incoming packets**
  - Depends on the current state of the routing table
- **Each packet contains**
  - Complete address information (source and destination)

© 2016, D.I. Lindner / D.I. Haas

The addresses used in datagram service technologies need to be globally unique and structured. They contain topological information. Structured means a part of the address is reserved for the user identification while another part of the address is used for topology information (describes network where the user is located).

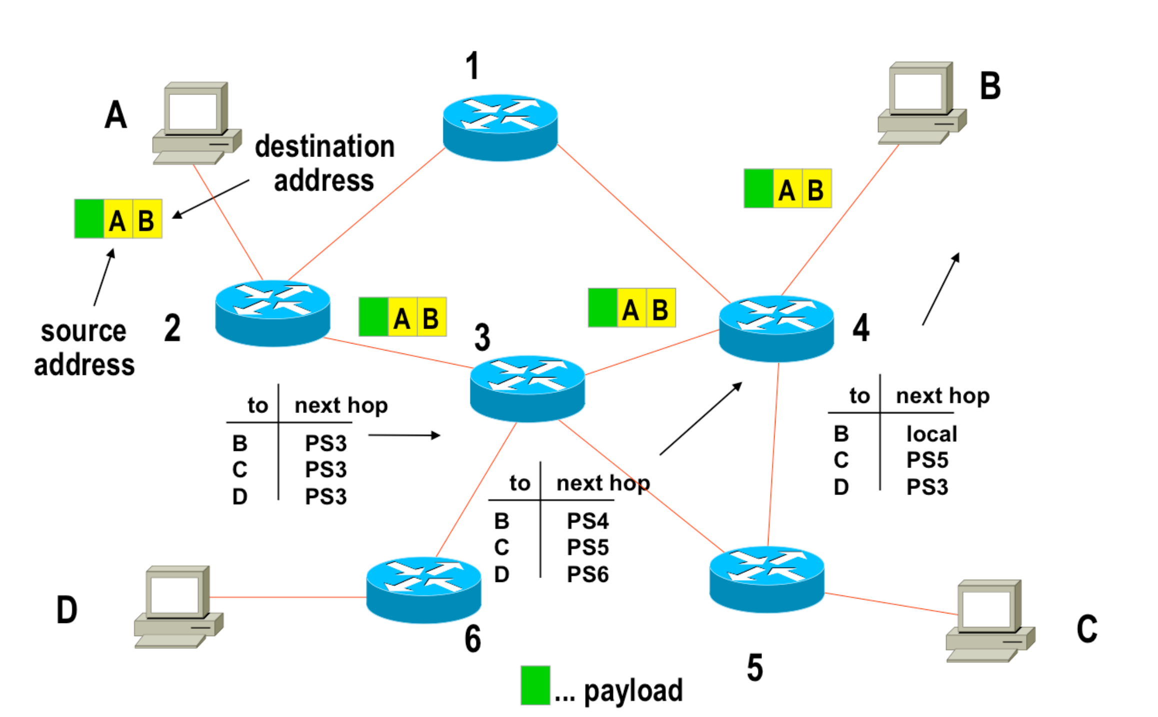
As already mentioned routing table can be based on a static configuration or on dynamic routing protocols.

Networks which are build on the datagram service technology typically need two different types of protocols: routed protocols which are used by the end user and routing protocols between routers to build up the routing tables.

© 2016, D.I. Lindner / D.I. Haas

L04 - Network Principles (v6.0)

Packet forwarding (CL) is based on routing tables only
(Datagram Service, Best-Effort Service)



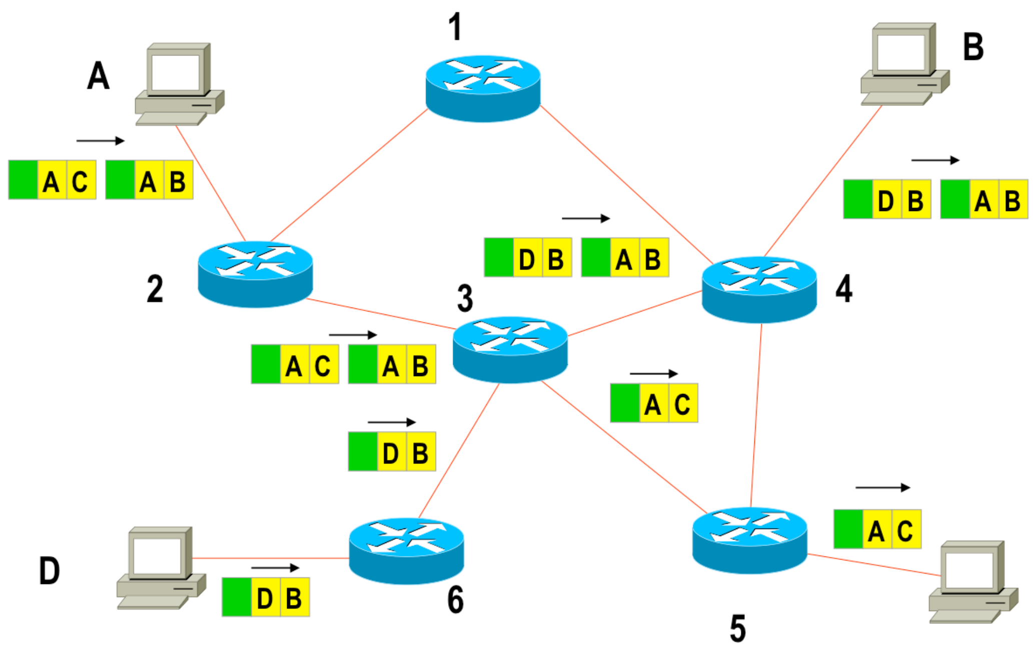
In the datagram technology device A sends out data packets destined for the device B. Each single datagram holds the information about sender and receiver address.

The datagram forwarding devices hold a routing table in memory. In the routing table we find a correlation between the destination address of a data packet and the corresponding outgoing interface as well as the next hop. So data packets are forwarded through the network on a hop by hop basis.

© 2016, D.I. Lindner / D.I. Haas

L04 - Network Principles (v6.0)

Datagrams are forwarded completely independent from each other based on current state of routing tables

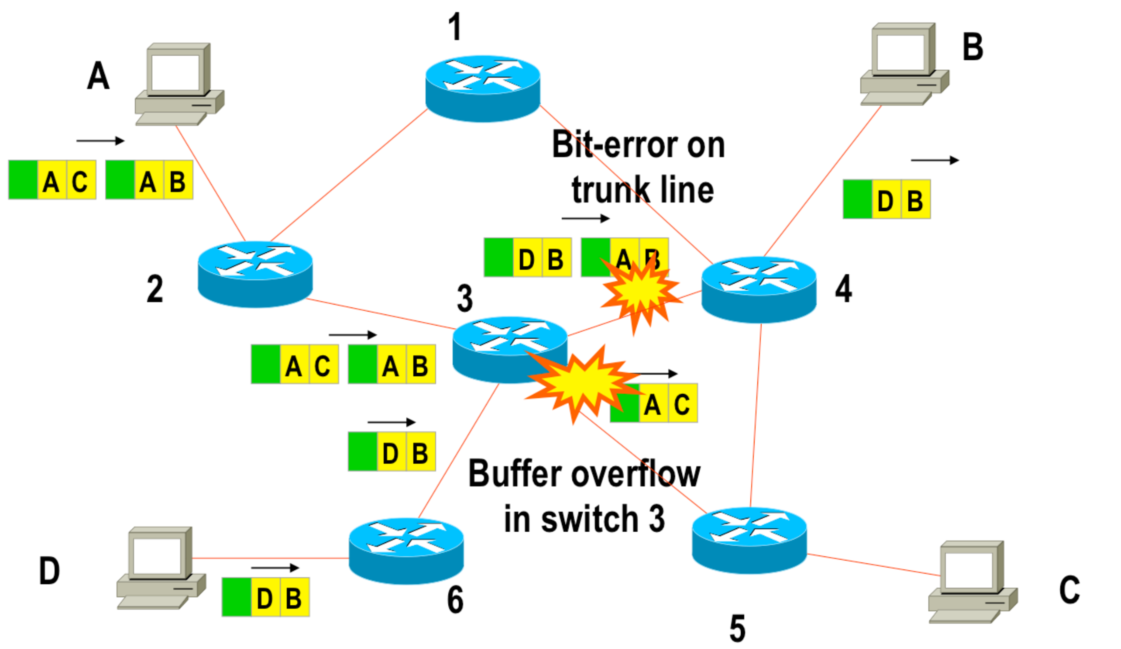


© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 28

The routing tables can be set up either by manual configuration of the administrator or by the help of dynamic routing protocols (in case of IP that are protocols like RIP, OSPF, IS-IS, etc). The use of dynamic routing protocols may lead to rerouting decisions in case of network failure and so packet overtaking may happen in these systems.

© 2016, D.I. Lindner / D.I. Haas

Best-Effort Service



© 2016, D.I. Lindner / D.I. Haas

Datagram Service Facts (1)

- **Packets may be discarded / dropped by packet switches**
  - In case of network congestion
  - In case of transmission errors
- **Best effort service**
  - Transport of packets depends on available resources
- **The end systems may take responsibility**
  - For error recovery (retransmission of dropped or corrupted packets)
  - For sequencing and handling of duplicates
- **Reliable data transport requires good transport layer**
  - "Dumb network, smart hosts"

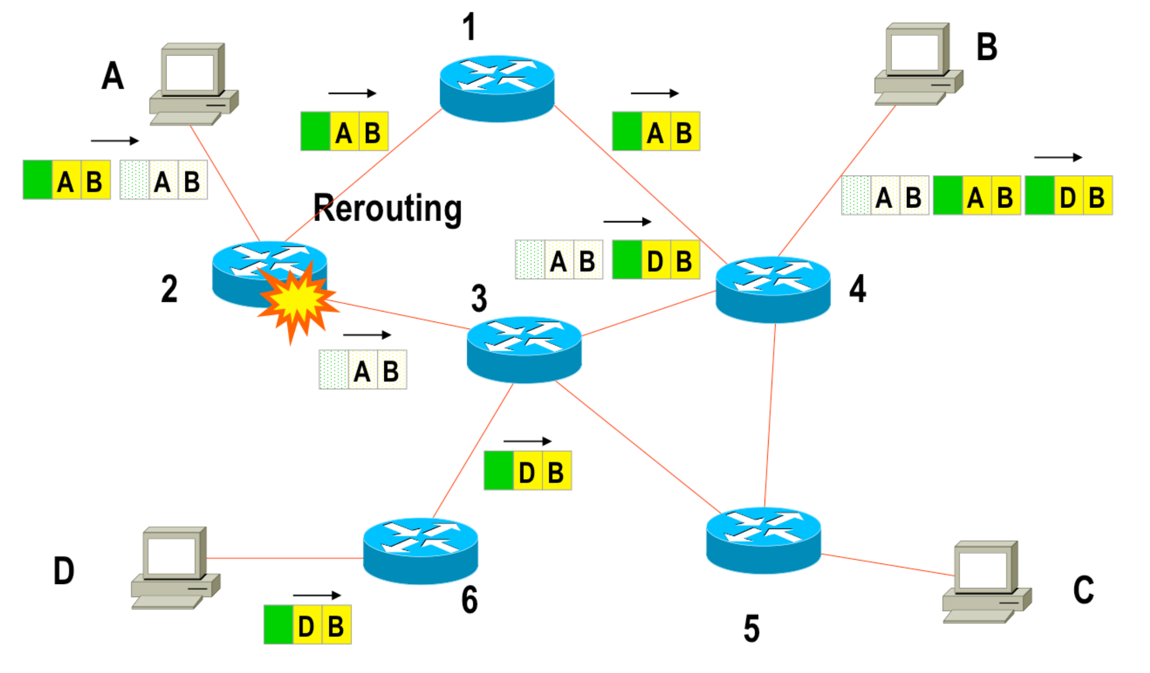
© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 30

Networks based on datagram technology support only best effort service, this means as good as it gets.

Routers that drop data packets because of buffer overflow or other problems don t care about error recovery. Error recovery is a task that needs to be performed by the end stations of a network. They have to take care for retransmissions in case of packet loss or transmission errors. This is typically done by layer 4 protocols like TCP which uses an connection-oriented mode.

L04 - Network Principles (v6.0)

Rerouting – Sequencing Not Guaranteed !



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 31

© 2016, D.I. Lindner / D.I. Haas

Datagram Service Facts (2)

- **Rerouting in case of topology changes or load balancing means**
  - Packets with the same address information can take different paths to destination
  - Packets may arrive out of sequence

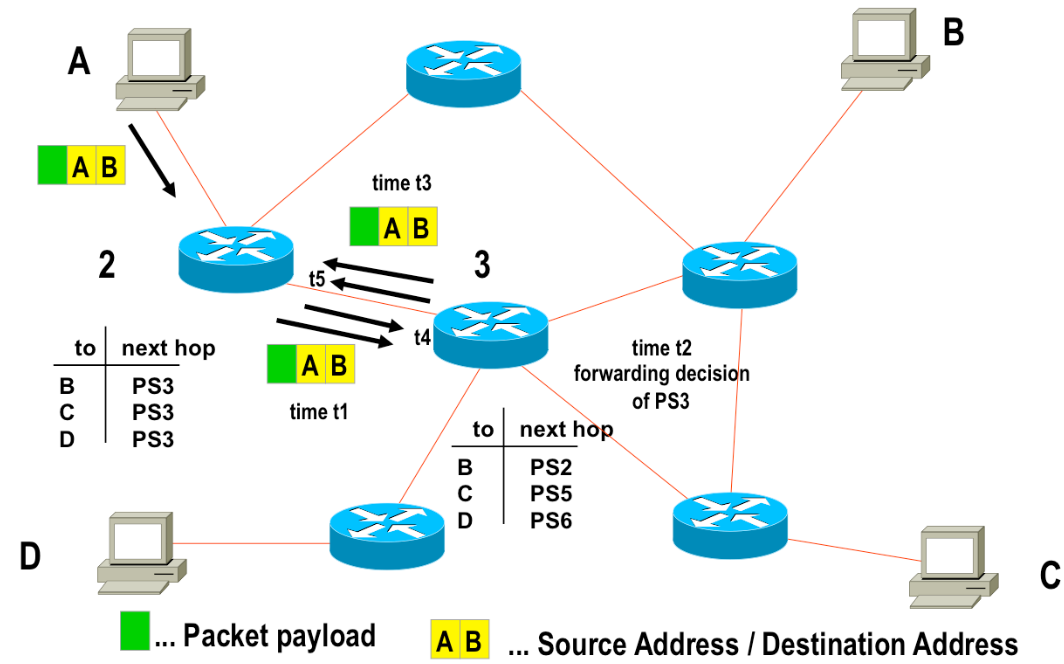
- **Sequence not guaranteed**
  - Rerouting on topology change
  - Load sharing on redundant paths
  - End stations must care

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 32

Topology changes cause rerouting when dynamic routing protocols are used and load sharing is practiced in the case of two or more paths with identical distance towards the destination. Rerouting and load balancing may also lead to packet overtaking, so the correct order of data packet arrival is not guaranteed.

© 2016, D.I. Lindner / D.I. Haas

Datagram forwarding needs a kill-mechanism to overcome inconsistent routing tables



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 33

In case of inconsistent information held in routing tables routing loops may occur which would lead to endless circling packets. Endless circulation means blocking of buffer memory in a packet switch. If there are two many endless circling packets in a network then all the buffers will be used up and hence other well-behaving traffic will be discarded because of lack of buffers. Special methods (kill mechanism) are necessary for avoiding or dampening that situation. Some protocols like IP use a maximum Time to Live field in their header to get rid of the endless cycling data packets.

That is a very important issue for all packet switching networks relying on forwarding of packets based on routing tables only.

© 2016, D.I. Lindner / D.I. Haas

Datagram Service Facts (3)

- **Connectionless behavior**
  - Faster delivery of first data
  - No resource reservation is possible
    - e.g. bandwidth on trunk line for a certain communication between end systems
    - e.g. buffer memory on packet switches
  - Bad "**Quality of Service**" (QoS)
  - Flow control between packet-switch and end-system
    - Would help in case of congestion if proactively performed
    - But lack of trust makes it impossible to implement it

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 34

Datagram services are typically driven in an connectionless mode, this guaranties a slightly faster delivery of datagrams because the time to establish a connection is saved.

The reservation of resources for QoS support is very difficult because the path of the data packets through the network may change during one session.

Proactive flow control is also very difficult to establish because there is no connection establishment phase between end-system and the packet switch / the network hence a trusted relationship can neither be established nor controlled.

© 2016, D.I. Lindner / D.I. Haas

Datagram Service Facts (4)

- **Advantages**
  - Small protocol overhead (easy to implement in end systems and packet switches)
  - Fastest delivery of data between end systems because no connection must be established in advance
- **Disadvantage**
  - Delivery of packets is not guaranteed by the network, must be handled by end systems using higher layer protocol
  - Proactive flow control between end-system and packet switch is not possible

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 35

Due to this behavior of datagram networks, the protocols to drive this kind of network can be kept simple and hence easy to implement.

© 2016, D.I. Lindner / D.I. Haas

Network Technologies based on Datagram Method

- **IP**
  - Packet is called IP datagram
  - End system is called IP host
  - Packet switch is called IP router
- **IPX (Novell)**
- **XNS**
- **Appletalk**
- **Decnet Phase IV**
- **OSI CNLP**

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 36

Remember typical examples of datagram networks are IP, IPX, Appletalk and the quite unknown OSI CLNP protocol stack.

Agenda

- **Introduction**
- **Circuit Switching**
- **Packet Switching**
  - Principles
  - Datagram Service
  - Virtual Call Service
- **OSI Reference Model**
- **Summary of Network Methods**

© 2016, D.I. Lindner / D.I. Haas

© 2016, D.I. Lindner / D.I. Haas

Virtual Call Principles

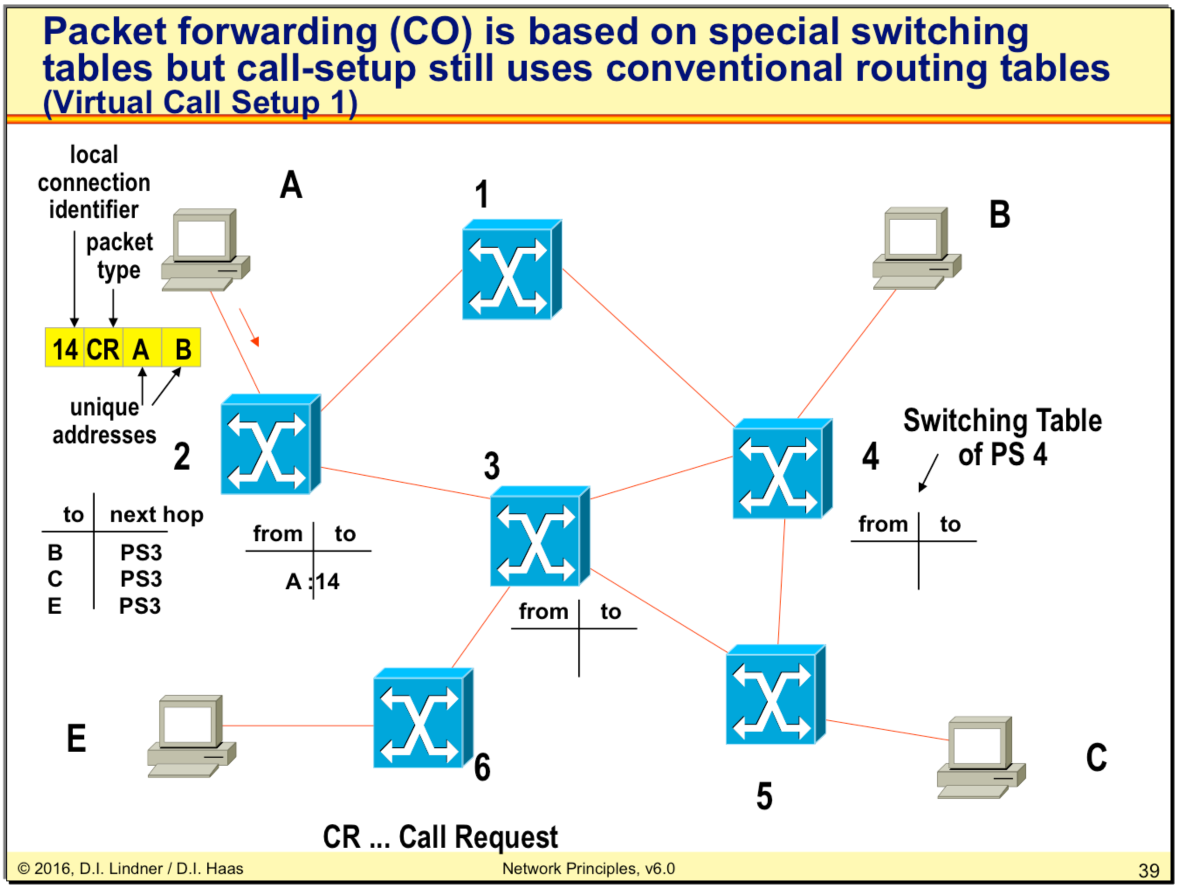
- **Connection-oriented service**
  - Special control packets (call setup packets) establish a logical (virtual) point-to-point connection between end systems first
  - We call that connection a **Virtual Circuit**
  - After connection is established
    - Data packets can be transmitted across that virtual circuit
    - Typically virtual circuit will be closed after data transfer is finished
- **Different methods are possible**
  - To establish a virtual call service

© 2016, D.I. Lindner / D.I. Haas

In Virtual Call Service technology addresses are used as well, but in a different manner than compared to datagram services. The global unique address information in Virtual Call Service systems is only used at the beginning of a conversation to setup a connection.

With an established connection data packets are forwarded according to virtual circuit identifiers which are held in switching tables.

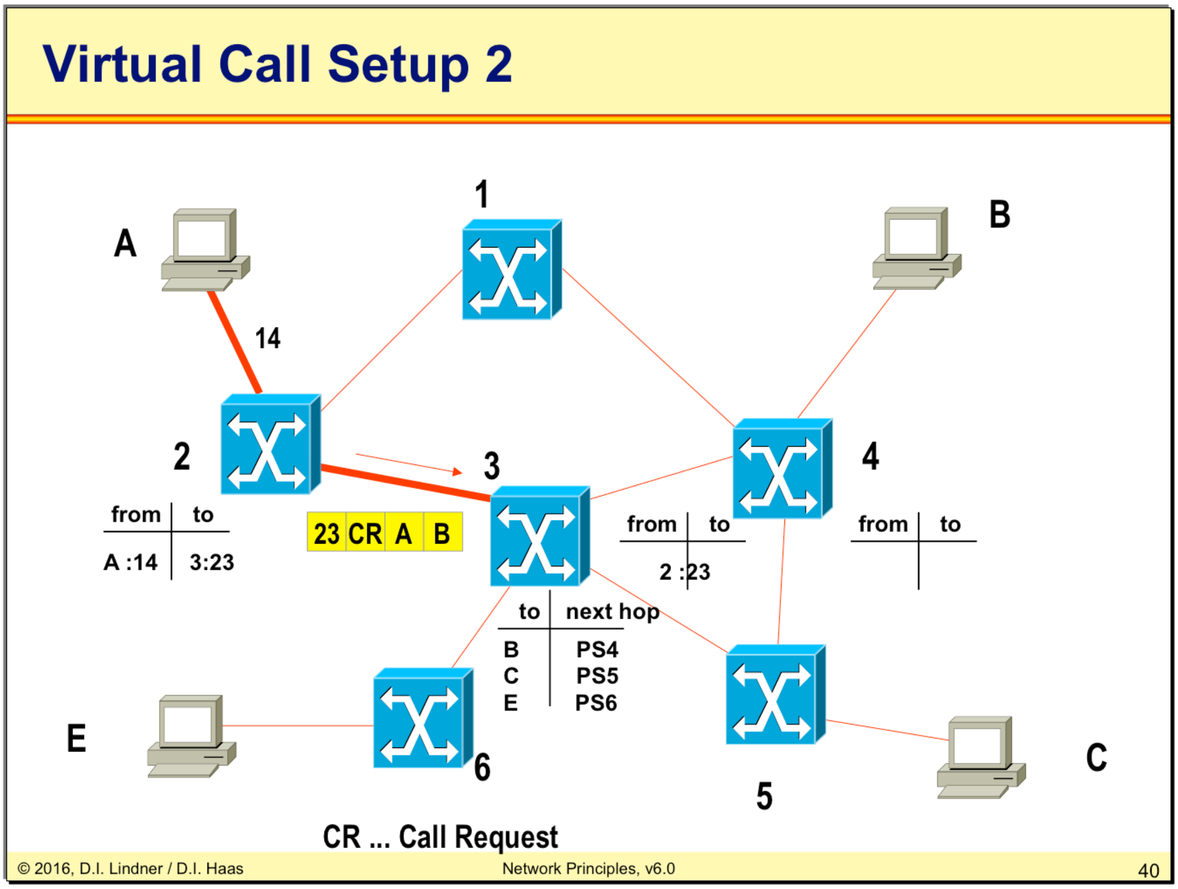
© 2016, D.I. Lindner / D.I. Haas



Call setup packets are transported across the network like datagrams hence for path decisions routing tables are used. So every packet switching network either CL or CO needs routing first.

Call setup packets contain unique address information of source and destination end systems and a local connection identifier to represent the requested connection. During proceeding of call setup packet the connection identifier on an incoming line will be mapped to a connection identifier on the outgoing line. The connection identifier has only local significance meaning that it was agreed between two directly connected devices e.g. end system and local packet switch or packet switch to next packet switch and so on.

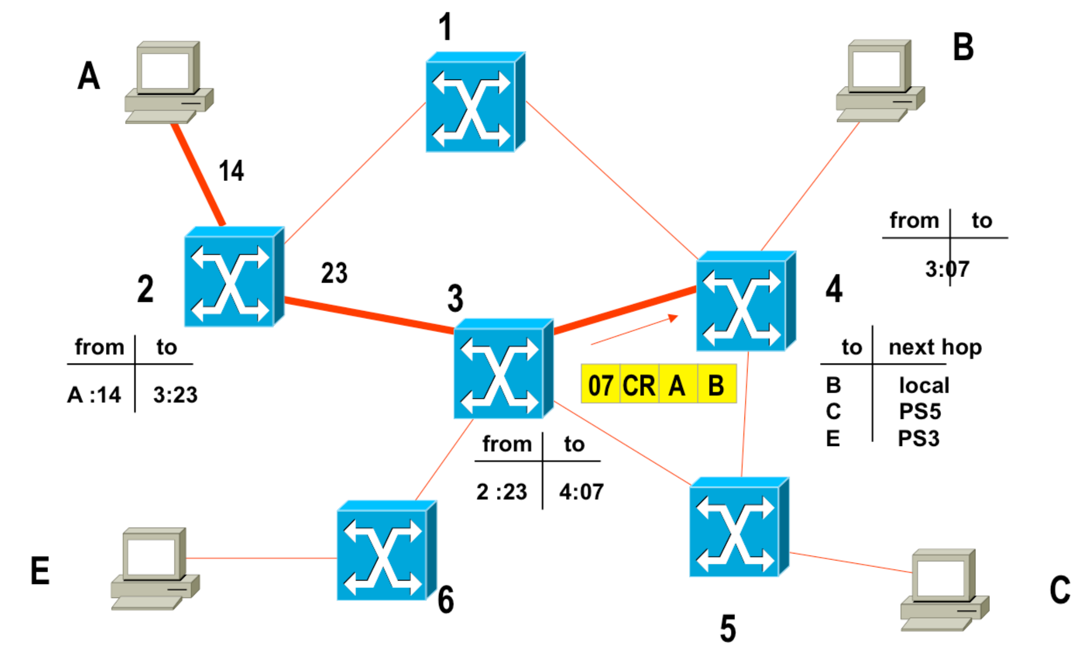
L04 - Network Principles (v6.0)

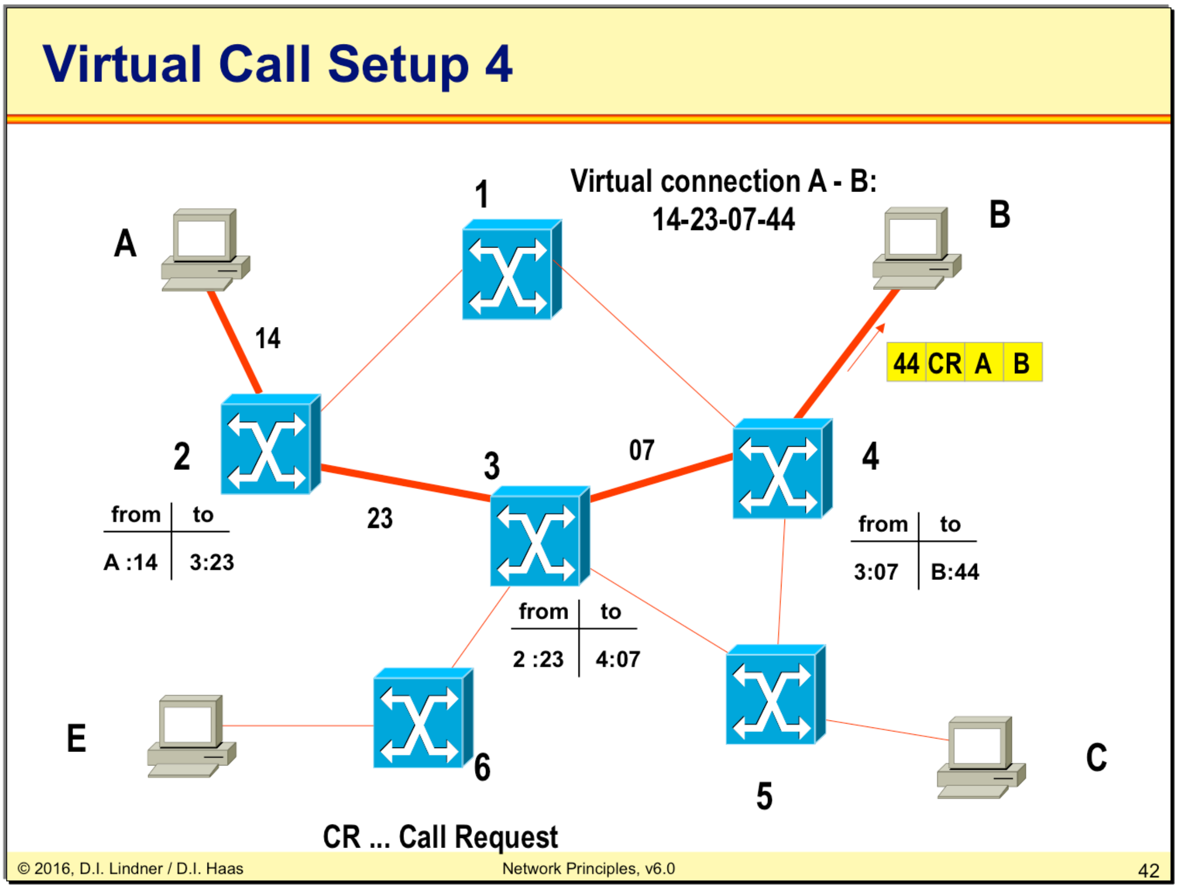


During call setup the information about incoming connection identifier/incoming port to outgoing connection identifier/outgoing port is stored in the **switching table.**

The path - the call setup packet has taken - is marked by corresponding switching table entries.

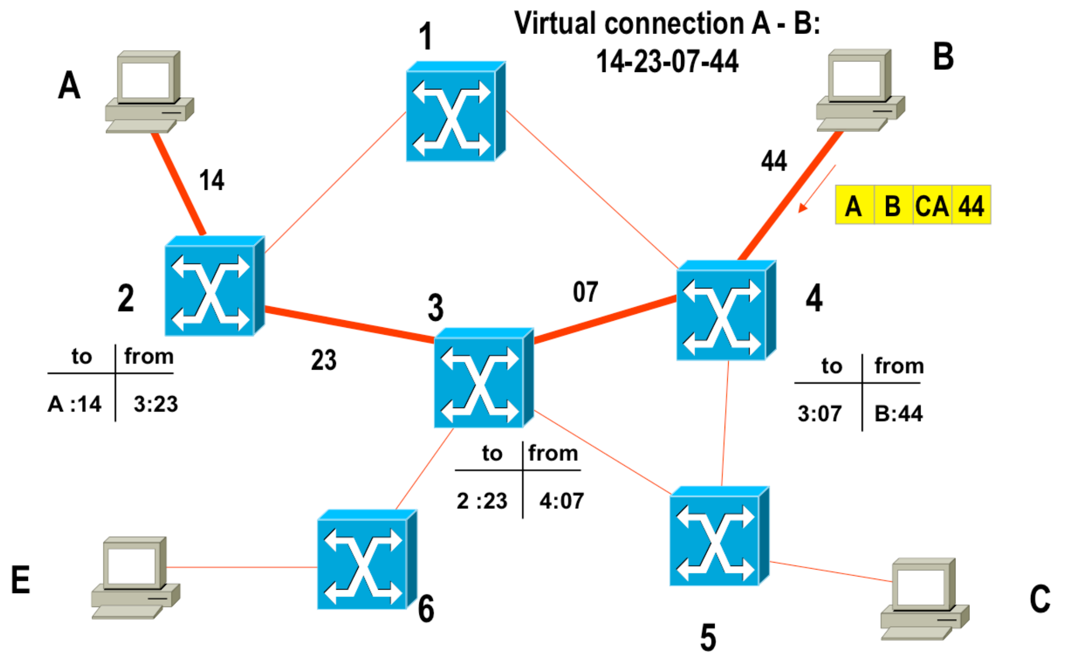
Virtual Call Setup 3



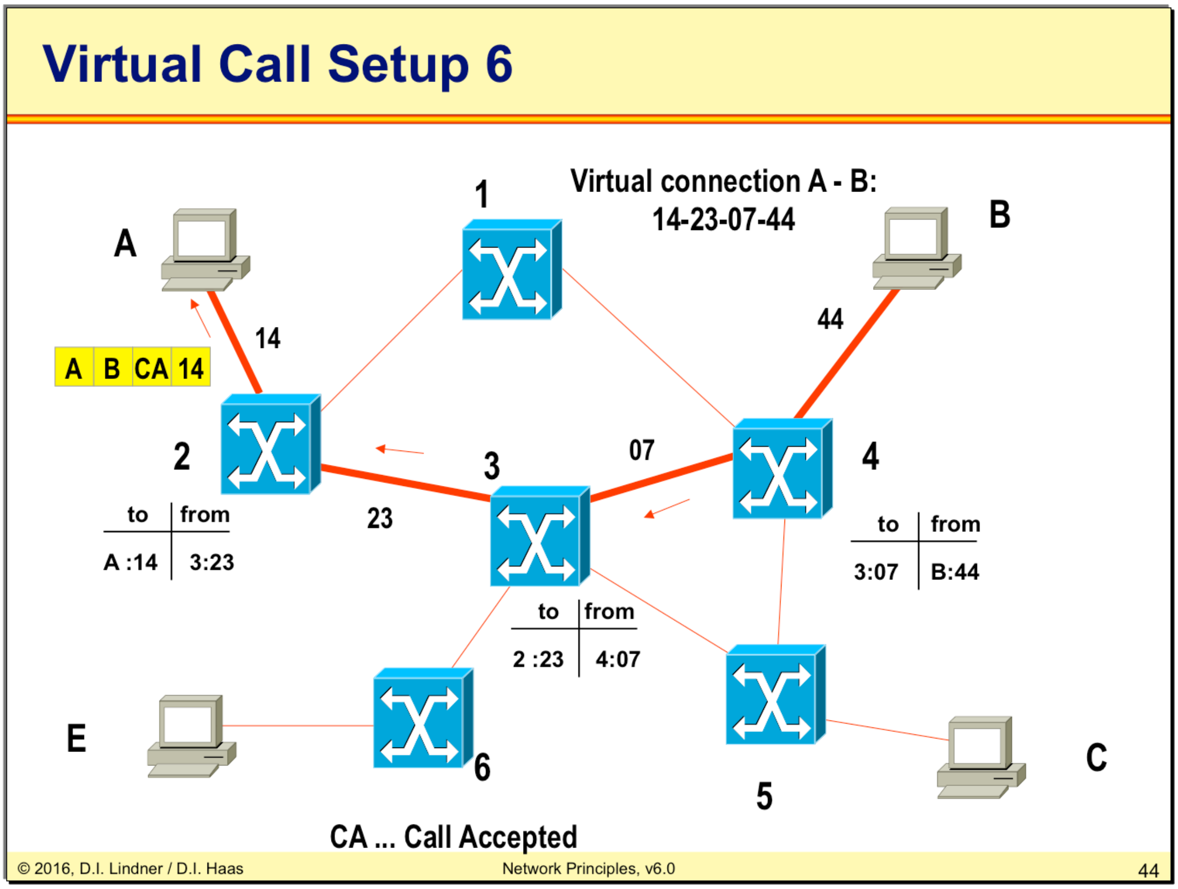


Now the call setup packet has reached the destination and will be acknowledged if the destination accepts the call.

Virtual Call Setup 5

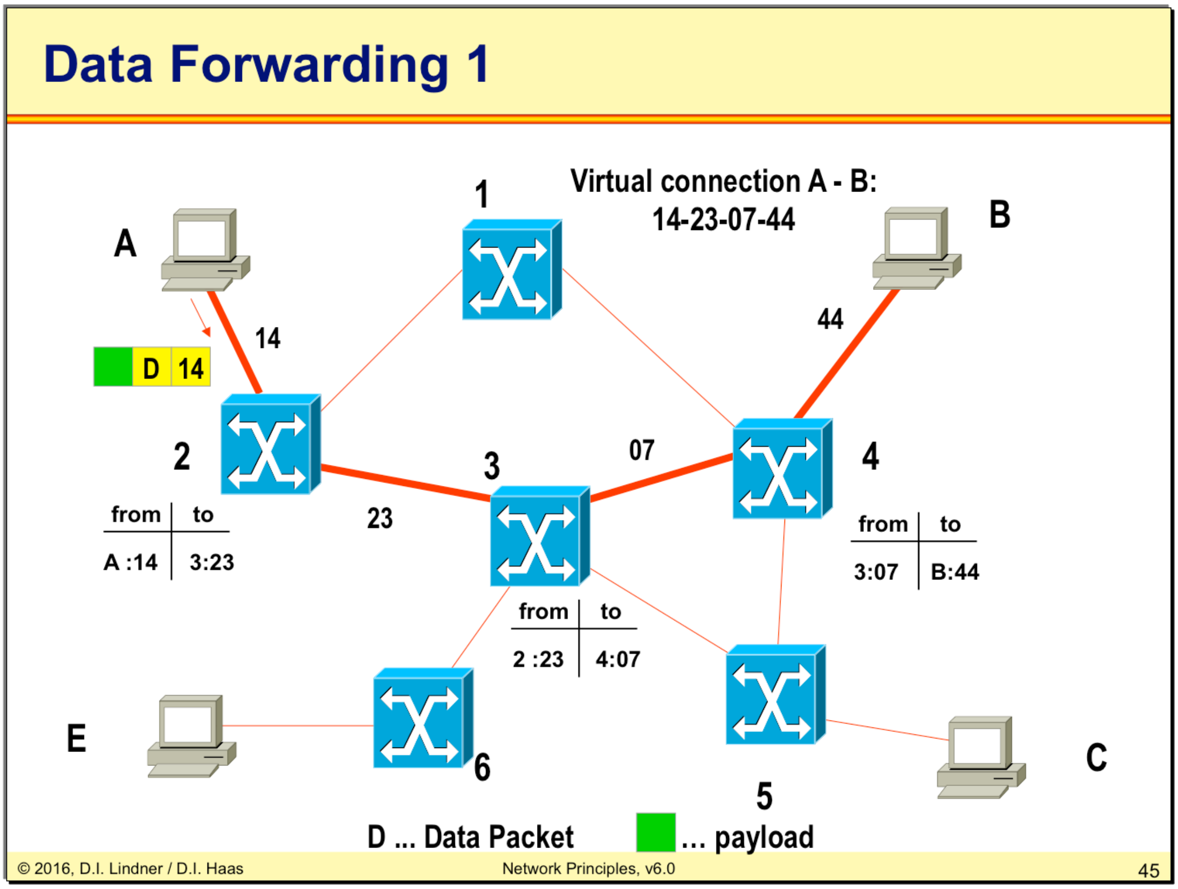


CA ... Call Accepted



Now the call accepted packet has reached the caller and you can see that the logical point-to-point connection between end system A and B can be identified in the network by the connection identifier sequence 14-23-07-44.

© 2016, D.I. Lindner / D.I. Haas



Now all data packets and even control packets use local identifiers as only address seen in these packets

1) to indicate to which connection they belong

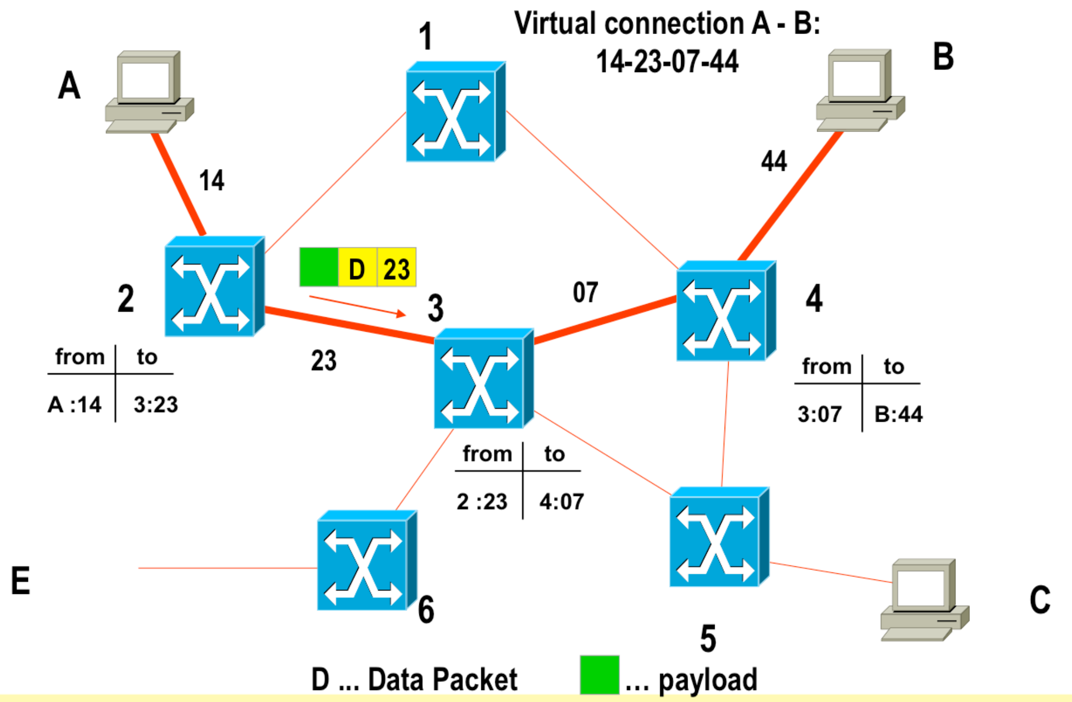
2) to which destination they should be delivered

Hence unique source and destination addresses are not required during data transfer phase anymore.

Swapping of incoming identifiers to outgoing identifiers is done by packet switches hop by hop by consulting the switching table only.

Forwarding decision based on switching table only, routing table not necessary in that phase anymore

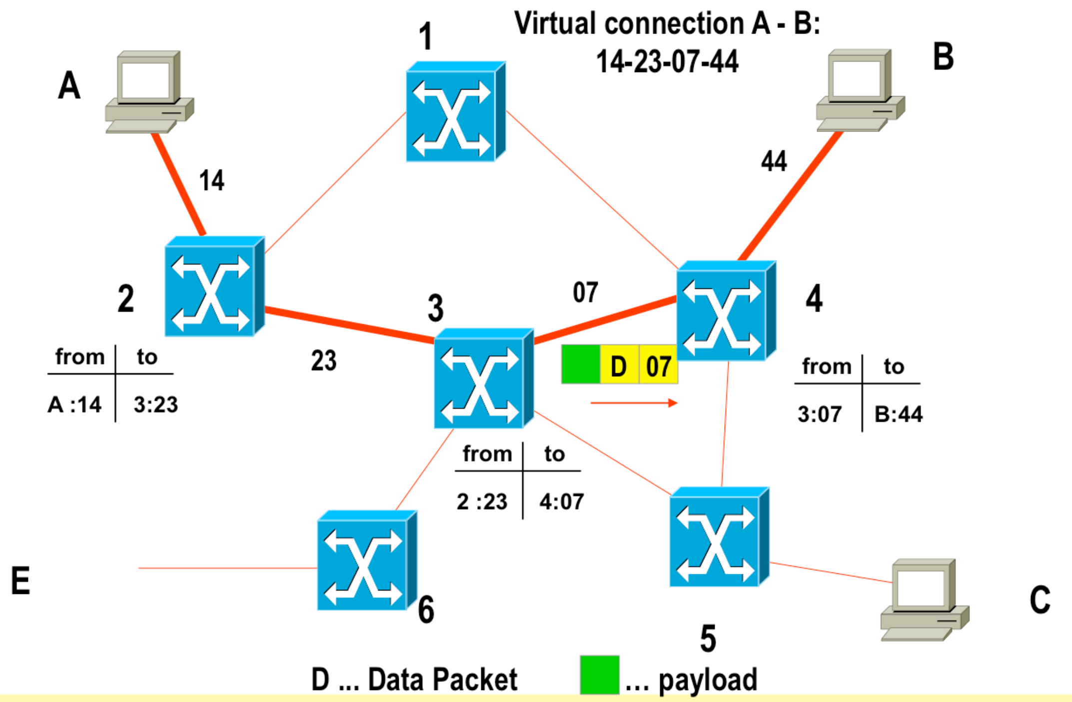
Data Forwarding 2



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 46

© 2016, D.I. Lindner / D.I. Haas

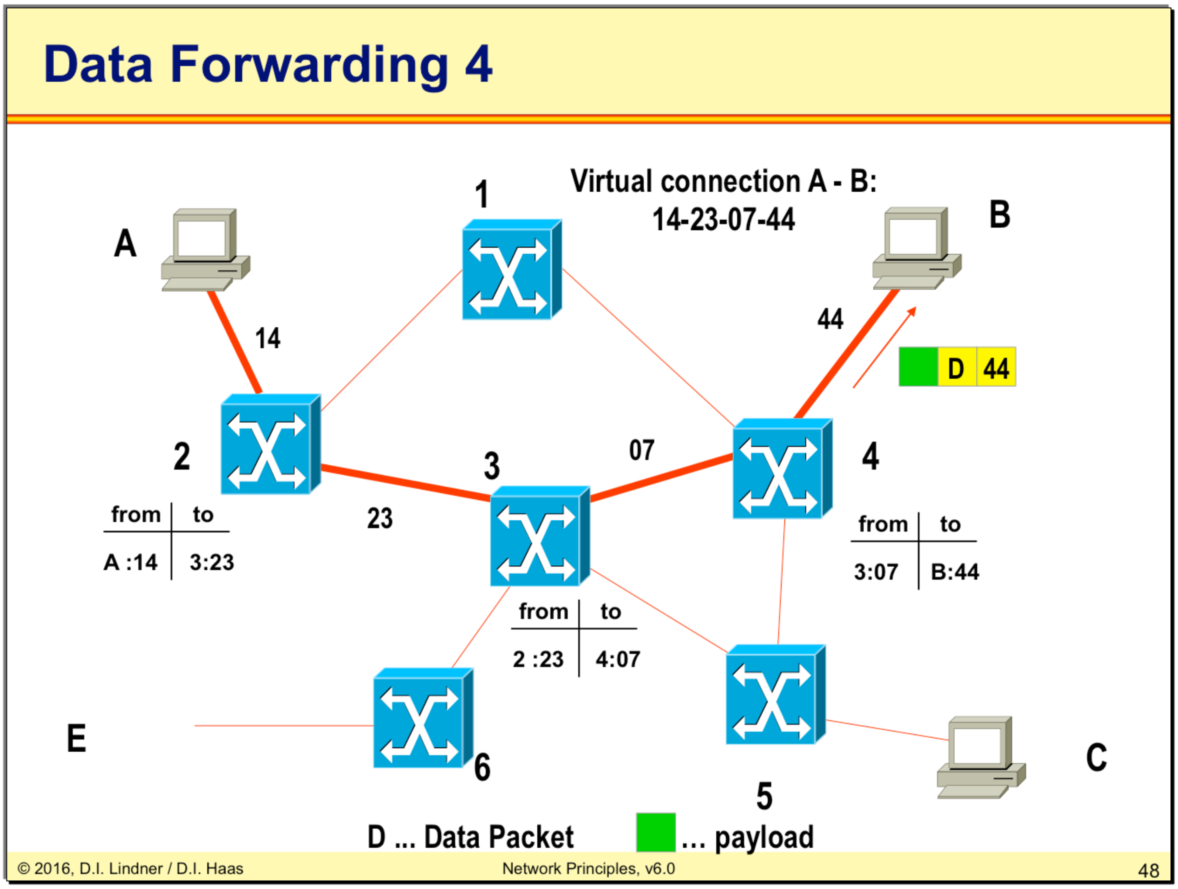
Data Forwarding 3



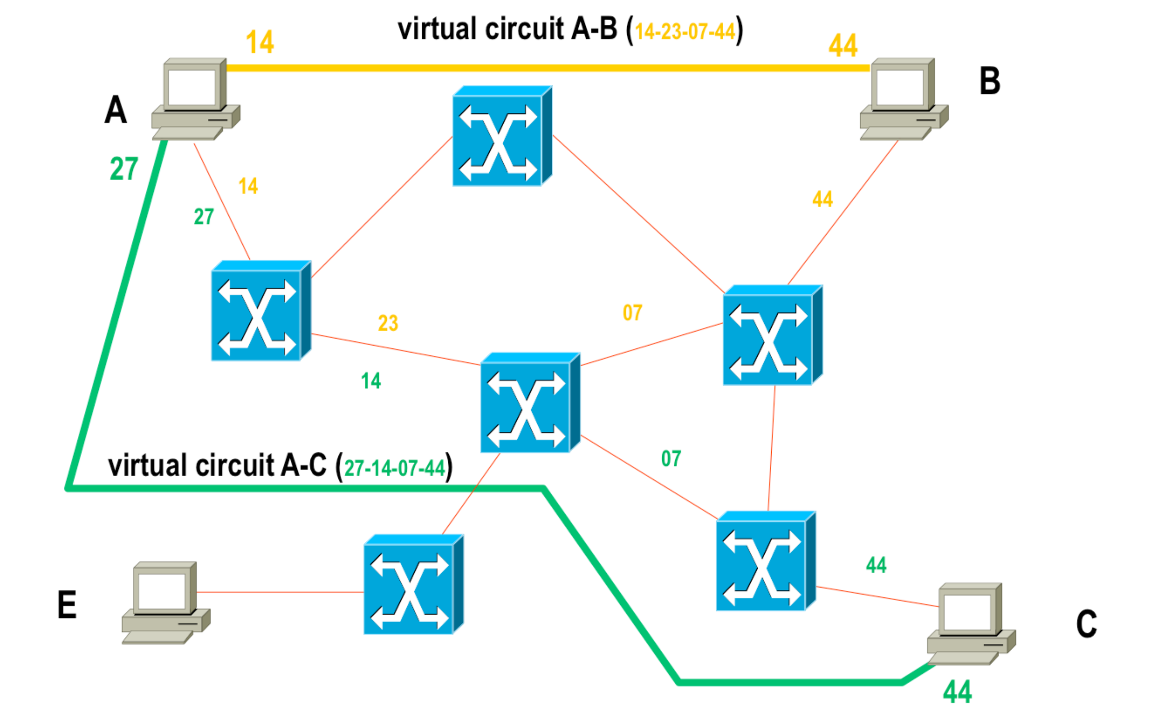
© 2016, D.I. Lindner / D.I. Haas

© 2016, D.I. Lindner / D.I. Haas

L04 - Network Principles (v6.0)



Multiplexing of Virtual Circuits



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 49

Connection identifiers and their corresponding switching tables are the base for maintaining/ multiplexing several virtual circuits (logical channels) over one physical link.

Therefore multiplexing several logical channels (virtual circuits) over such a packet switching infrastructure at the same time is not a problem.

The picture shows two virtual circuits which were already established by the set up procedure. Please recognize the local meaning of the connection identifier. Virtual circuit from device A to B is identified by the sequence 14-23-07-44 but virtual Circuit from device A to D has a sequence of 27-14-07-44. Only on a single link the numbers for a virtual circuit must be different in order to distinguish the circuits.

In principle connection identifiers have the same meaning - as port identifiers used for asynchronous TDM on a point-to-point line (as we have already seen in the TDM Techniques chapter).

Some examples for the name of local connection identifiers in famous network technologies:

- X.25 -> LCN (logical channel number)
- Frame Relay -> DLCI (data link connection identifier)
- ATM -> VPI/VCI (virtual path/channel identifier)

© 2016, D.I. Lindner / D.I. Haas

Two Service Types

- **Switched Virtual Circuit (SVC)**
  - Dynamic establishment as shown
  - At the end a proper disconnection procedure is necessary
  - Virtual circuits require establishing and clearing
- **Permanent Virtual Circuit (PVC)**
  - No establishment and disconnection procedures are necessary
  - Switching tables preconfigured by administrator
  - Circuits are permanently available for data transfer

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 50

In Virtual Call Service technique we find two basic types of connections Switched Virtual Circuits (SVC) and Permanent Virtual Circuits (PVC).

SVC´s dynamically establish a connection when needed and tear down the connection when the data transfer is finished. SVC technique is mainly used in combination with X25 and ATM services.

PVC´s are permanently up and can be seen like leased line services. Please recognize that this kind of leased line is not comparable with the leased line service in circuit switching networks. The later has constant delay guaranteed by synchronous TDM – the former has variable delay caused by the nature of asynchronous TDM. PVC´s are mainly used in Frame-relay and ATM services.

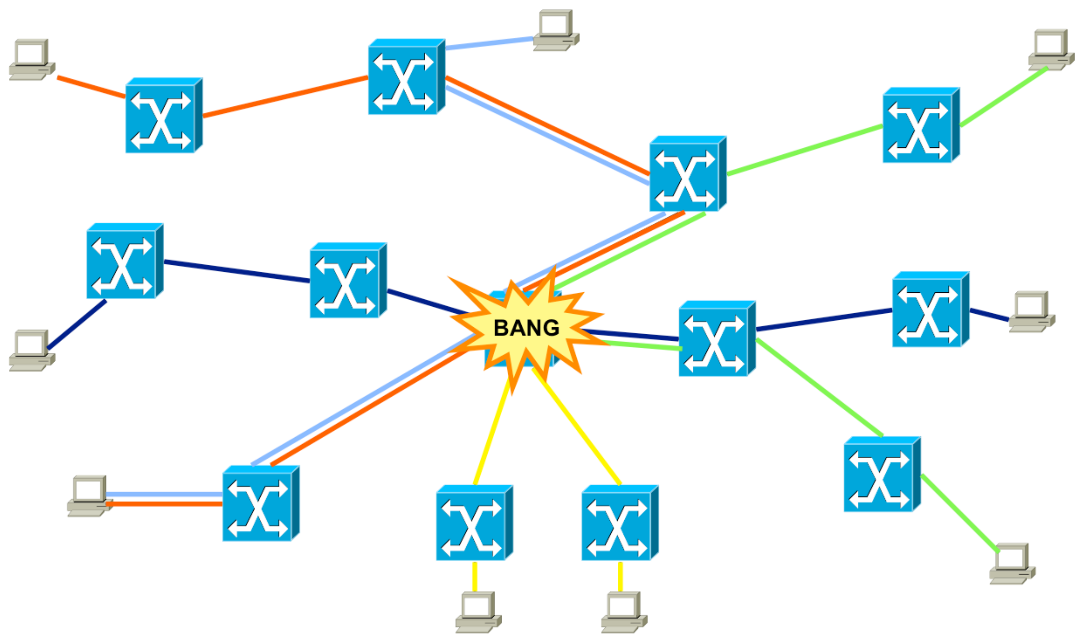
© 2016, D.I. Lindner / D.I. Haas

Virtual Call Service Facts (1)

- **The sequence of data packets is guaranteed by the network**
  - Packets can use the established path only
- **Path selection is done during connection setup**
  - Afterwards, entries of routing table are not used
- **In case of trunk line or packet switch failure**
  - Virtual circuits will be closed and must be reestablished again by end systems using call setup packets
  - If there is at least one redundant path, packet switches can establish a new virtual circuit taking the redundant path

L04 - Network Principles (v6.0)

Example



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 52

This example shows us what will happen if a node in the center of a network collapses. All connection through the collapsed node are torn down and new connections using signaling needs to be established. This causes a lot of overhead through to new connection setup requests. In Virtual Call Service technology its up to the end devices to set up a new connection through the network.

In Datagram technology this problem would be fixed by the network itself by rerouting.

Virtual Call Service Facts (2)

- **Connection-oriented packet switching**
  - Allows flow control procedures between end system and packet switch because of connection-oriented approach
    - In connectionless packet switching networks flow control is not possible or only poorly implemented
  - Flow control procedures can avoid buffer overflow and hence network congestion
  - Allows reservation of resources
    - Capacity, buffers, cpu time, etc.
  - Can offer Quality of Service (QoS)
  - Call setup can be denied by network if QoS can not be guaranteed

© 2016, D.I. Lindner / D.I. Haas

Virtual Call Service Facts (3)

- **Advantages**
  - Required resources of packet switches can be reserved during call setup and hence QoS could be provided
  - End system view of the network
    - Reliable point-to-point transport pipe based on network internal error recovery, flow control and sequencing procedures (X.25)
    - Higher protocol layers can rely on network services (X.25)
  - Readiness for receipt is tested in advance
    - Call setup of SVC service
- **Disadvantages**
  - Call setup takes time
  - More complex protocols for end systems and packet switches than datagram service

Virtual Call Service Facts (4)

- **Packet switching like that was much faster than packet forwarding of IP routers in the past**
  - Routing process is complex, typically implemented in software
  - Switching is simple, typically implemented in hardware
- **Nowadays high speed packet forwarding of IP routers**
  - Is done in a more optimized way which allows support in hardware
  - e.g. Cisco's cef-switching

© 2016, D.I. Lindner / D.I. Haas

Why is routing slower? We give just a short explanation here: First, a router must determine which part of the address is topology relevant – with IP addresses this so-called network-identifier has variable length. Second, the router must find the best ("longest") match of the destination net-ID with the routing table entries. Third, the next-hop might not be the physical next hop. In this case a recursive routing table lookup is necessary. Fourth, because of the topology-related addresses (and the associated complex forwarding processes) the routing table cannot easily be stored in a high-performance data structure. All this is typically implemented in software.

Switching is completely different. The addresses are unstructured and not topology related. The switching process is simply to look up the correct entry in the switching table and determine the outgoing interface, hereby modifying the logical channel number (the local connection identifier). The whole process can be implemented in hardware. Additionally, switching is greatly accelerated using hashing-functions (CAM-tables).

© 2016, D.I. Lindner / D.I. Haas

Virtual Call – Summary (1)

- **Connection establishment**
  - Through routing process (!)
  - Globally unique topology-related addresses necessary
  - Creates entries in switching tables
  - Can reserve switching resources (QoS)
- **Packet forwarding relies on local identifiers**
  - Not topology related
  - Only unique per port
  - Swapping of local identifiers (labels) according to the switching table

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 56

Remember routing processes are needed even in Virtual Call Service technologies to allow the setup of a connection. The addresses used for connection setup need to be structured and globally unique.

The connection setup procedure creates entries in switching tables to support the data forwarding phase.

Its quite easy to reserve transport resources (QoS) during connection establishment, because the path through the network remains the same for one conversation.

Data packet forwarding is performed according to local and only per port unique virtual circuit identifiers.

© 2016, D.I. Lindner / D.I. Haas

Virtual Call – Summary (2)

- **Connection can be regarded as virtual pipe**
  - Sequence is guaranteed
  - Resources can be guaranteed
- **Virtual call multiplex**
  - Multiple virtual pipes per switch and interface possible
  - Pipes are locally distinguished through connection identifier
- **Network failures disrupt pipe**
  - Connection re-establishment necessary
  - Datagram networks are more robust

© 2016, D.I. Lindner / D.I. Haas

Remember a connection used by Virtual Call Service technologies can be seen like a virtual pipe or tunnel. Therefore the correct sequence of data packets is guaranteed and resources can be reserved quite easily.

Network failures will lead to an tear down of the connection and a new connection setup procedure.

Datagram networks are more robust because to setup a proper connection is more difficult than data packet forwarding on a hop by hop basis. The connection setup procedure needs more sophisticated protocols especially when QoS parameters should be taken into account.

© 2016, D.I. Lindner / D.I. Haas

Network Technologies based on Virtual Call Method

- **X.25**
  - Reliable transport pipe because of protocol inherent error recovery and flow control
  - Local identifier = LCN
  - In-band signaling
- **Frame Relay**
  - Virtual circuit technique but no error recovery
  - Congestion indication instead of flow control
  - Local identifier = DLCI
  - Out-band signaling
- **ATM (Asynchronous Transfer Mode)**
  - Same as Frame Relay but packets with fixed length
  - Hence called cell switching
  - Local identifier = VPI/VCI
  - Out-band signaling

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 58

All WAN-switching technologies utilize the same principle that has been described above. But the connection identifer has different names. In X.25 we call it the Logical Channel Number (LCN). With Frame Relay we talk about the Data Link Connection Identifier (DLCI). And ATM packets are switched using the Virtual Path Identifier/Virtual Circuit Identifier (VPI/VCI). No matter what complicated names are used, it is simply a dumb identifier without any special meaning.

© 2016, D.I. Lindner / D.I. Haas

Agenda

- **Introduction**
- **Circuit Switching**
- **Packet Switching**
  - Principles
  - Datagram Service
  - Virtual Call Service
- **OSI Reference Model**
- **Summary of Network Methods**

L04 - Network Principles (v6.0)

"The good thing about standards is that there are so many to choose from"

Andrew S. Tanenbaum

Standards

- **We need networking standards**
  - Ensure interoperability
  - Large market, lower cost (mass production)
- **Vendors need standards**
  - Good for marketing
- **Vendors create standards**
  - Bad for competitors, hard to catch up
- **But: Slow standardization processes freeze technology...**

We need standards. Unfortunately. Otherwise, each vendor would create what he wants and we would not be able to communicate across networks. This situation occurred very often in history. For example the United Nations initiated a world-wide Telephony standardization board, known as CCITT (today ITU-T). Or in the pre-Ethernet age, many vendors built completely incompatible LAN protocols.

Especially to force interoperability, many vendors for Internet-equipment initiated the TCP/IP Interoperability Conference in 1987, today known as "INTEROP".

© 2016, D.I. Lindner / D.I. Haas

Who Defines Standards?

- **ISO - Anything**
  - International Standards Organization (ISO)
  - International agency for the development of standards in many areas, founded 1946, currently 89 member countries
- **IETF - Internet**
- **ITU-T (former CCITT) – Telco Technologies**
- **CEPT - PTT Technologies**
- **ETSI - European Standards**
- **ANSI - North American Standards**
- **ATM-Forum, Frame Relay Forum, MPLS Forum**
- **IEEE - LAN Protocols**
- **DIN, ÖNORM - National Standards**

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 62

The above slide mentions the most important standardization organizations.

The Internet Engineering Task Force (IETF) is "actually" the most important technical organization for the Internet working groups and is organized in several areas. Area manager and IETF chairman form the IESG (Internet Engineering Steering Group). The IETF is also responsible to maintain the RFCs.

© 2016, D.I. Lindner / D.I. Haas

Standards Types

- **De facto standards**
  - Anyone can create them
  - E.g. Internet RFCs
- **De jure standards**
  - Created by a standardization organization
  - E.g. ISO/OSI, ITU-T

© 2016, D.I. Lindner / D.I. Haas

Not all standards are like the others. De facto standards are more flexible and speed-up the implementation. Usually everybody is allowed to extend them. The whole Internet is built on such loosely standards. Unfortunately misinterpretations can occur (RFCs).

De jure standards are like acts of law. For example ITU-T standards explain nearly every detail implementers may ask.

© 2016, D.I. Lindner / D.I. Haas

Note

Standardization is applied to *network layers* and *interfaces* between them

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 64

The above sentence leads us to network layers. Break big problems into smaller ones and write standards for them ("divide and conquer"). Of course the interfaces between the layers must be standardized too. Eventually, multiple developers can work on different parts of the whole story.

© 2016, D.I. Lindner / D.I. Haas

Idea of Layering and Services

- **Because communication between systems can be a very complex task**
  - Divide task of communication in multiple sub-tasks
    - So called layers
  - Hence every layer implements only a part of the overall communication systems
- **Hierarchically organized**
  - Each layer *receives* services from the layer below
  - Each layer *serves* for the layer above
- **Good for interoperability**
  - Capsulated entities and interfaces
- **But increases complexity / overhead**

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 65

Network layers are an abstraction to hide complexity. Layers are organized hierarchically, that is there is a predefined command direction. Imagine what would happen if we have a democratic model?

Note that network layers force a more complex development. Many high-performance communication technologies have been developed in an ad-hoc act, or alternatively consists of only a few layers.

© 2016, D.I. Lindner / D.I. Haas

Where to Define Layers? Why to define Layers?

- **Where:**
  - Group functions (services) together
  - When changes in technology occur
  - To expose services
- **Why:**
  - To allow changes in protocol and HW
  - To utilize existing protocols and HW

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 66

A good layering structure requires a intelligent grouping of functions. Ideally, technology improvements can be implemented immediately.

For example the X.25 packetizing algorithm, which is written in software and part of a network driver of the operating system can remain untouched, while the serial line hardware can be updated, and vice versa.

© 2016, D.I. Lindner / D.I. Haas

The ISO/OSI Model

- **International Standards Organization (ISO)**
  - International agency for the development of standards in many areas
  - Founded 1946
  - Currently 89 member countries
  - More than 5000 standards until today
  - Network standardization is just one part of it
- **OSI (Open Systems Interconnection) Reference Model**
  - Defines tasks and interactions of seven layers
  - Framework for development of communication standards
  - System-internal implementation is out of the scope
  - Only external behavior of a system is defined by open standards
- **1988 US Government OSI Profile (GOSIP)**
  - Requires Government products to support OSI layering

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 67

The ISO standardized anything–character sets, paper sizes, screws, ..., and network layers.

When viewing communication between or among computer systems, it is helpful to implement a common set of standards or conventions. The International Standardization Organization (ISO) has developed an architecture or model, called the Open Systems Interconnection (OSI) model, that is a framework for defining standards for linking heterogeneous computers.

In 1988 the US Government required any communication device to comply with the ISO/OSI model (GOSIP). Note that the non-OSI Internet was built much earlier, so many people expected the end of the Internet. But the Internet (which was created as nuclear-bomb resistant) not only survived the ISO/OSI model but also displaced many OSI-compliant protocols, such as CLNP.

Similarly, in Europe the "European Procedure Handbook for Open Systems" (EPHOS) had been released.

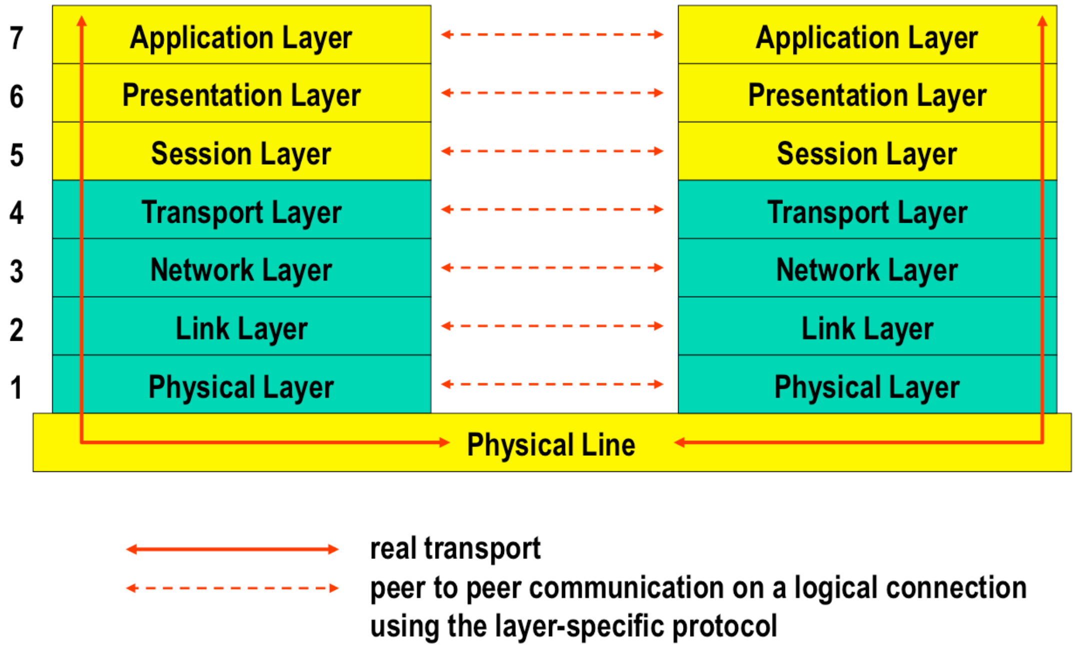
© 2016, D.I. Lindner / D.I. Haas

Basic Idea of the OSI Model

- **Each layer depends on services of the layer below in order to provide its own services to the upper layer**
  - **Service specification standards**
- **Representation of a layer within a system**
  - Is called entity (e.g. a particular task or subroutine)
- **In order to fulfill the task of a layer**
  - Entities use their own system internal resources as well as peer-to-peer communication based on layer specific protocol
  - **Protocol specification standards**

© 2016, D.I. Lindner / D.I. Haas

The 7 OSI Layers



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 69

Because the communication between different systems can be a very complex task, OSI splits the communication aspects into smaller tasks. All layering is based on the OSI reference Model, which defines tasks and interactions of seven layers.

The user's data moves from the first layer (Application Layer) through all other layers. When two systems communicate with each other, then only the different layers talk. The application layer only talk with the application layer or the network layer only communicate with the network layer of system B. We can talk about a parallel communication between the layers. Every layer works for its own, it is not interested what the other layer does.

© 2016, D.I. Lindner / D.I. Haas

Purpose

- **OSI model *describes* communication *services* and *protocols***
- **No assumption about**
  - Operating system
  - Programming Language
- **Practically, the OSI model**
  - Organizes knowledge
  - Provides a common discussion base

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 70

Although every book of data communication mentions the ISO/OSI 7-layer model it is not that important in the real world: most technologies do not comply to this model. It is merely a reference model so that we can refer to it when we want to explain certain functions in our protocols. From this point of view the OSI model is indeed important today.

© 2016, D.I. Lindner / D.I. Haas

OSI Basics

- **Point-to-Point, no shared media**
- **Nodes are called**
  - End Systems (ES)
  - Intermediate Systems (IS)
- **Each layer of the OSI model detects and handles errors**
- **Hence connection-oriented per se**
- **Dumb hosts and intelligent network**
  - Compared with Internet: dumb network, intelligent hosts

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 71

The original OSI modes was created for point-to-point connections only (for example there was no specification for LAN-like shared media originally).

The OSI Truth

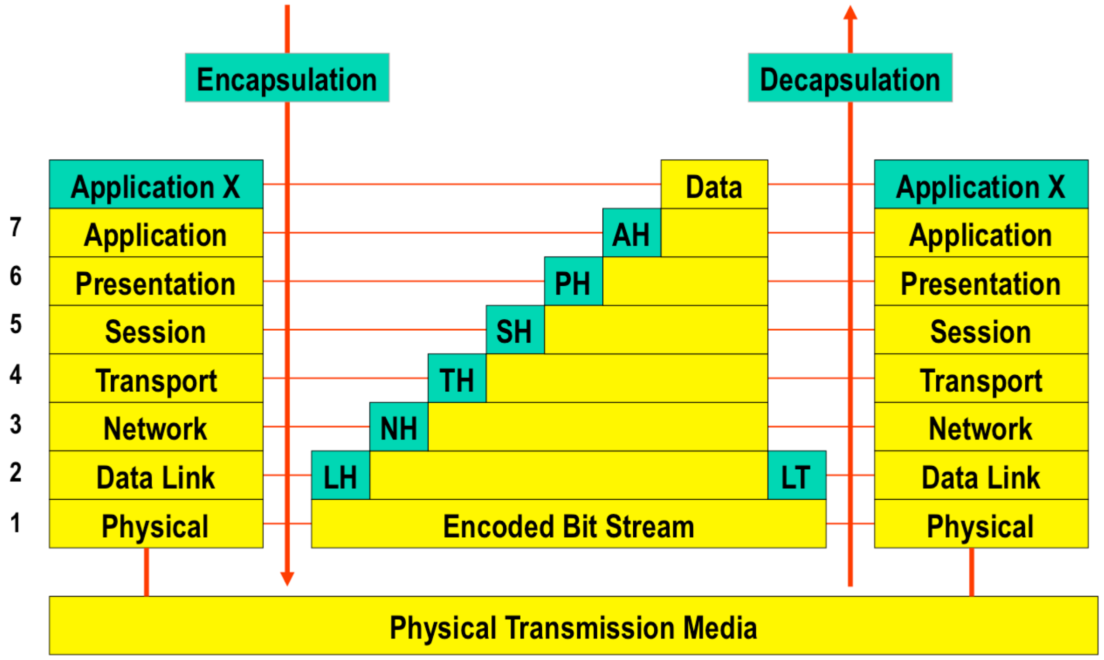
- **OSI model was created before protocols**
  - Good: Not biased, general approach
  - Bad: Designers had little experience, no ideas in which layers to put which functionality...
- **Not widespread (complex, expensive)**
- **But serves as good teaching aid !!!**

© 2016, D.I. Lindner / D.I. Haas

Although the OSI Model was created before any OSI based protocols were created - and so the complete model is very complex and not practically elaborated - its widely used today to define and category most of the important protocols. OSI is not biased because this reference framework is not associated with any particular vendor philosophy. OSI represents a general approach for describing data communication procedures but this property is often considered as a big disadvantage, because practical implementations typically can be described with a much simpler model and on the other hand the OSI architects had only little experience with real life implementations. Therefore, genuine OSI protocols are not really widespread today, because of its complexity. Nevertheless, the OSI model serves as reference frame when discussing or learning about protocols.

© 2016, D.I. Lindner / D.I. Haas

Encapsulation and Decapsulation



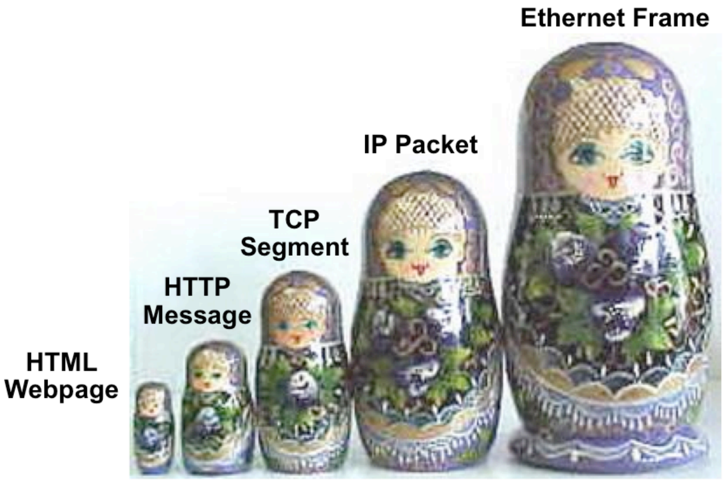
© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 73

One of the most important principles:

Every layer adds its own protocol header by going downstairs in the stack -> Encapsulation at the source.

Every layer removes its own protocol header by going upstairs in the stack -> Decapsulation at the destination.

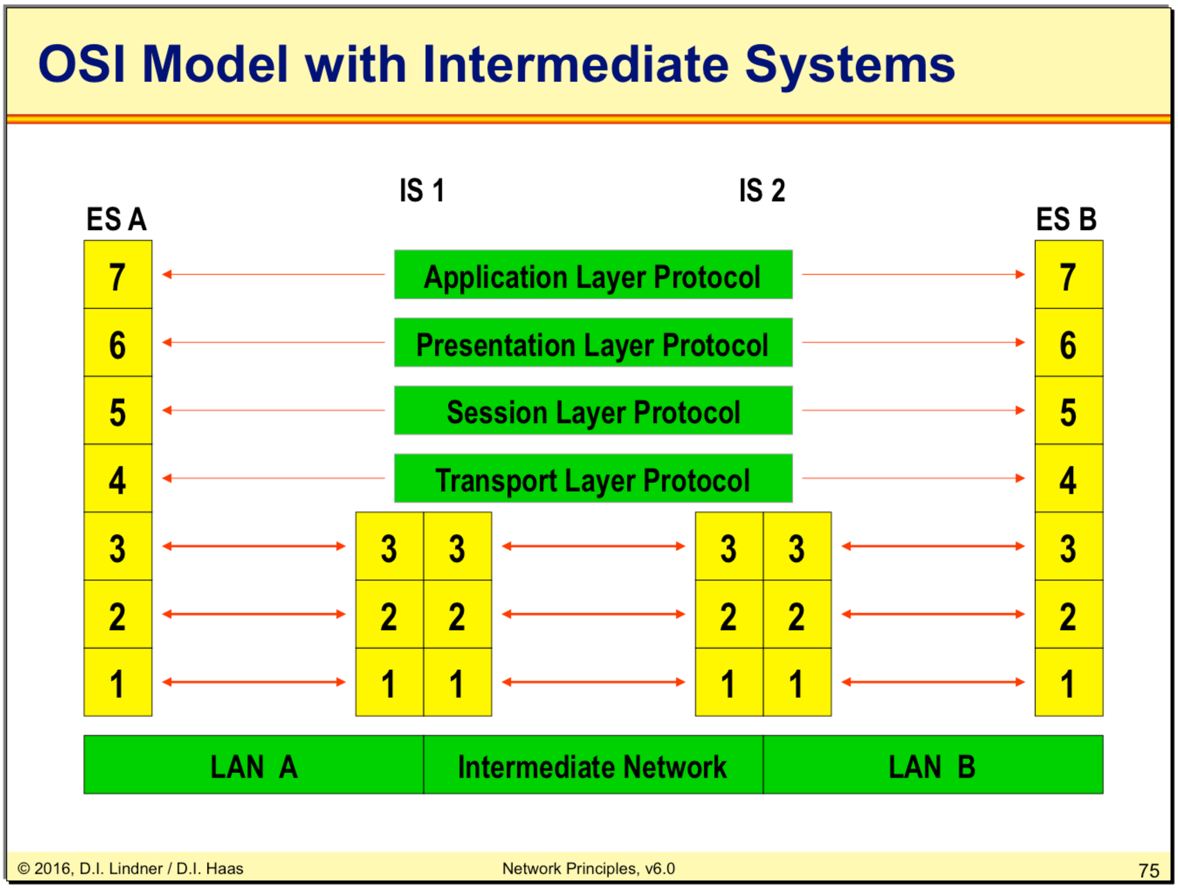
Practical Encapsulation



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 74

The idea of encapsulation is fundamental in the data communication world. Adjacent layers encapsulate or decapsulate information by adding/removing additional "overheads" or "headers" in order to implement layer-specific functionalities. The whole process can be regarded as Matroschka-puppet principle.

In our example let's suppose a web-server sends a webpage (HTML code) to a client. The webpage is carried via the Hyper Text Transfer Protocol (HTTP) which provides for error and status messages, encoding styles and other things. The HTTP header and body is carried via TCP segments, which are sent via IP packets. On some links in-between, the IP packets might be carried inside Ethernet frames.



In end systems (ES) all seven layers must be implemented for communication between network applications of different computers.

If two end systems are not directly connected via one physical link so called OSI relay systems / intermediate systems are necessary

Intermediate systems (IS):

- Store and forward devices
- Packet switches
- Require routing / switching functionality
- Only lower layers (1-3) are necessary

OSI Speak (1)

- **Entities**
  - Anything capable of sending or receiving information
- **System**
  - Physically distinct object which contains one or more entities
- **Protocol**
  - Set of rules governing the exchange of data between two entities

© 2016, D.I. Lindner / D.I. Haas

Entities:

Any hardware or software module that acts upon a single layer is called an "entity". Several entities exist peer to peer within a given layer and are capable to communicate which each other. This type of communication is referred as "horizontal" communication -- this is actually what we mean when we talk about a "protocol".

System:

Several entities make up a "system". For example a PC is a "system" because it consists of the entities Ethernet PHY entity, MAC entity, LLC entity, IP entity, TCP entity, and several L7 entities. A system is merely a term that reflects the physically separation of groups of entities.

Protocol:

We already described the meaning of protocol above together with the definition of a entity, but a protocol can be explained more simple: A protocol is a set of rules that are necessary to exchange data in an ordered and unmistakable way.

© 2016, D.I. Lindner / D.I. Haas

OSI Speak (2)

- **Layer**
  - A set of entities
- **Interface**
  - Boundary between two layers
- **Service Access Point (SAP)**
  - Virtual port where services are passed through

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 77

Layer:

A "layer" in the OSI jargon is a set of entities--but do not confuse layers with systems! The entities of a layer reside on the same hierarchy level and a single layer comprises several systems. On the other hand a system comprises several layers but typically only one (or a limited number) of entities are available on each layer of a system. For example: In order to communicate in the Internet, all devices must support layer 3 (the IP layer). That is, each system must provide at least one IP-entity.

Interface:

An "interface" is simply the logical boundary between two layers. Note that interfaces are typically not physically visible because they represent the boundary between two layers at a whole. The local representation of an interface is called a "Service Access Point" or SAP. The Service Access Point is one of the most frequently used terms in data communication and simply reflects the piece of hardware or software that acts as an interface between two layers. The previously OSI-interface is meant globally, while the SAP has local meaning, i. e. at one system. A SAP is a practical term, in some technologies such as IEEE 802.2 it is just a field in the header indicating the destination and source layer. If you use an Ethernet NIC with an AUI interface, than this electrical interface can be also considered a SAP because "service primitives" are passed through this interface. Service Primitives are explained below...

Service Access Point:

An "Interface Data Unit" (IDU) is practically spoken the piece of data that is passed through a SAP to the next layer's entity. It contains ICI and SDU which is described below. When an IDU is passed through a SAP to the next layer, this layer extracts and processes the Interface Control Information (ICI).

© 2016, D.I. Lindner / D.I. Haas

OSI Speak (3)

- **Interface Data Unit (IDU)**
  - Data unit for vertical communication (between adjacent layers of same system)
- **Protocol Data Unit (PDU)**
  - Data unit for horizontal communication (between same layers of peering systems)
- **Interface Control Information (ICI)**
  - Part of IDU
  - Destined for entity in target-layer
- **Service Data Unit (SDU)**
  - Part of IDU
  - Destined for further communication
  - Contains actual data ;-)

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 78

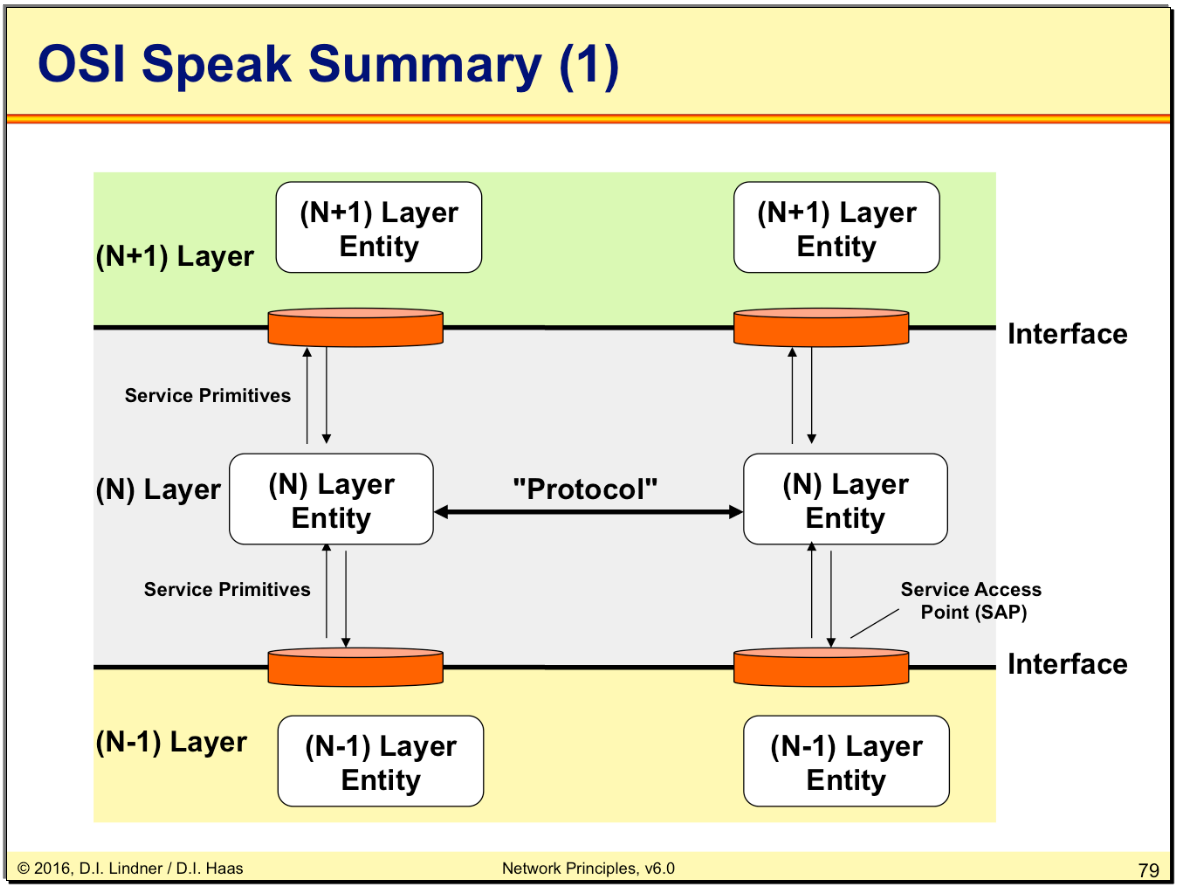
Interface Data Unit:

An "Interface Data Unit" (IDU) is practically spoken the piece of data that is passed through a SAP to the next layer's entity. It contains ICI and SDU which is described below. When an IDU is passed through a SAP to the next layer, this layer extracts and processes the Interface Control Information (ICI).
Note that data is passed through a SAP using "service primitives". Service primitives are functions that are implementation specific (for example an API) and are used to pass data from one layer to another on the same system. These service primitives actually pass on these IDUs.

Protocol Data Unit:

The SDU actually represents the payload plus headers for upper layers. The SDU is transported horizontally with an header used at this layer. Both SDU and Header is called a "Protocol Data Unit" (PDU). The PDU is the most often used term of all these terms mentioned here. At least you should remember the PDU.

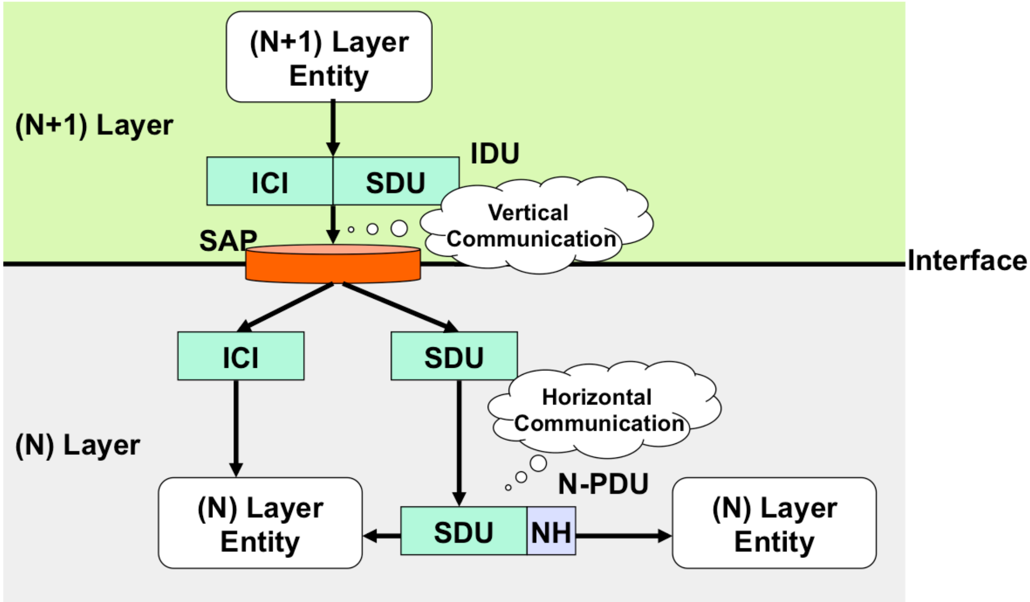
© 2016, D.I. Lindner / D.I. Haas



The ISO/OSI model defines four service primitives: request, indication, response and confirm. Note that the service primitives are only used for vertical communication.

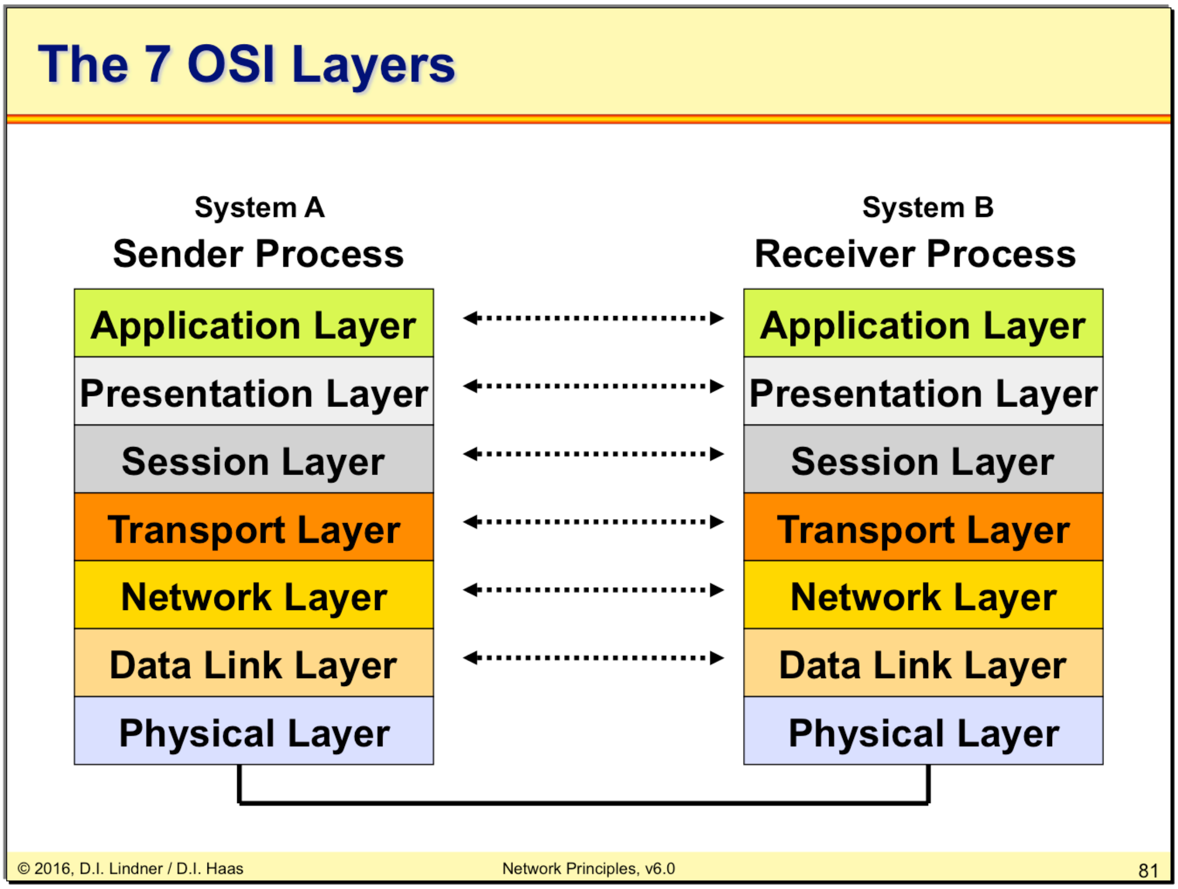
© 2016, D.I. Lindner / D.I. Haas

OSI Speak Summary (2)



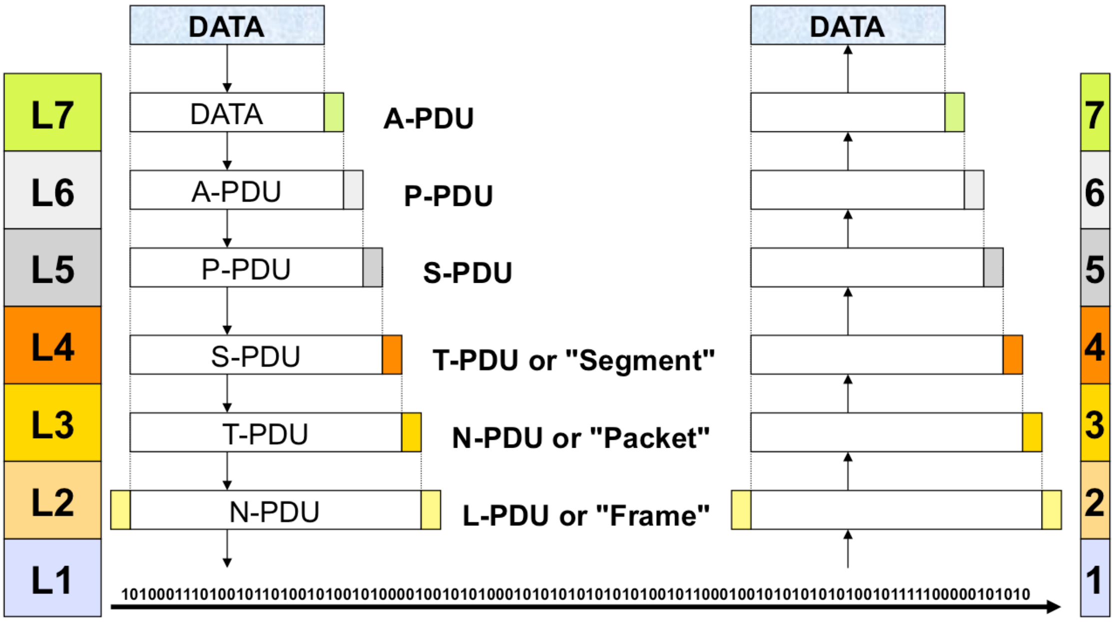
© 2016, D.I. Lindner / D.I. Haas

© 2016, D.I. Lindner / D.I. Haas


The 7 OSI Layers
System A
Sender Process
System B
Receiver Process
Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer
Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer
© 2016, D.I. Lindner / D.I. Haas
Network Principles, v6.0
81

L04 - Network Principles (v6.0)

Encapsulation and Decapsulation Principle

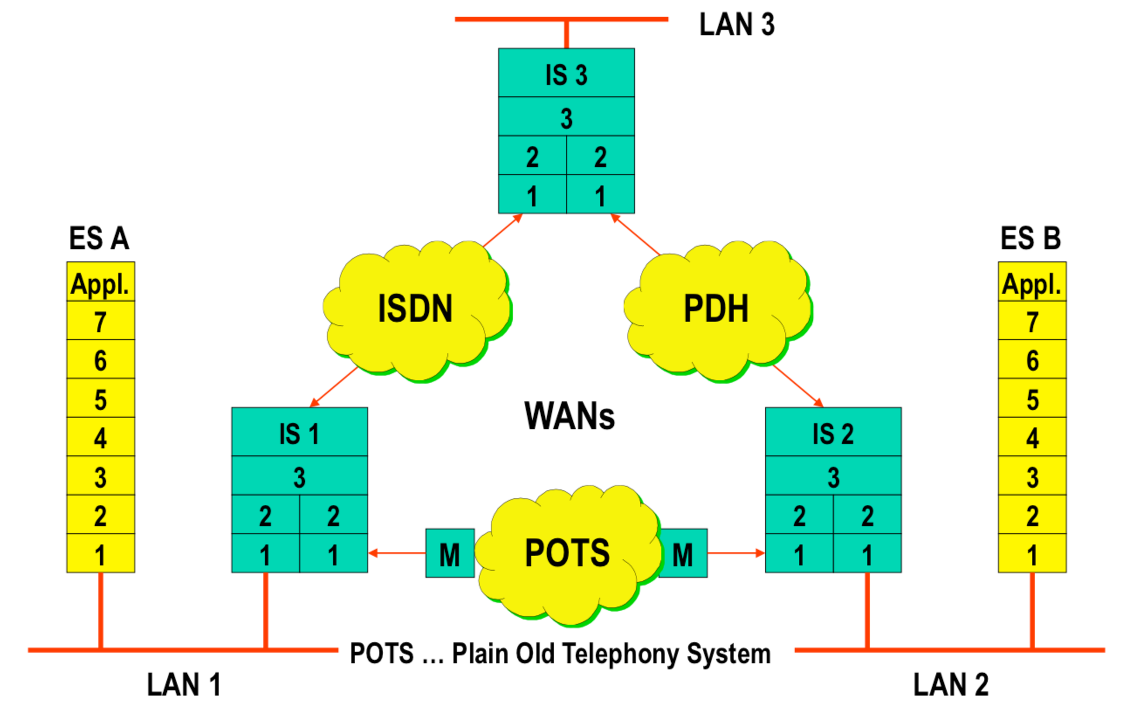


© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 82

The data moves through all 7 layers. Every layer add his own header. The data with layer 4,5,6 and 7 header is called “segment” in the IP world. A segment plus layer 3 header is called “packet” in case of CO packet switching and “datagram” in case of CL packet switching (IP). The so called “frame” (data plus six headers) will be transport over layer 1 to the destination system (frame means a block of bits at layer 2). In the destination system the frame will move through all 7 layers again. At each layer the corresponding protocol header will be removed, processed according to the layer-specific protocol and the data part – if present – will be given to the layer above. Of course the data part at every layer except the application layer will contain further protocol headers of higher layers.

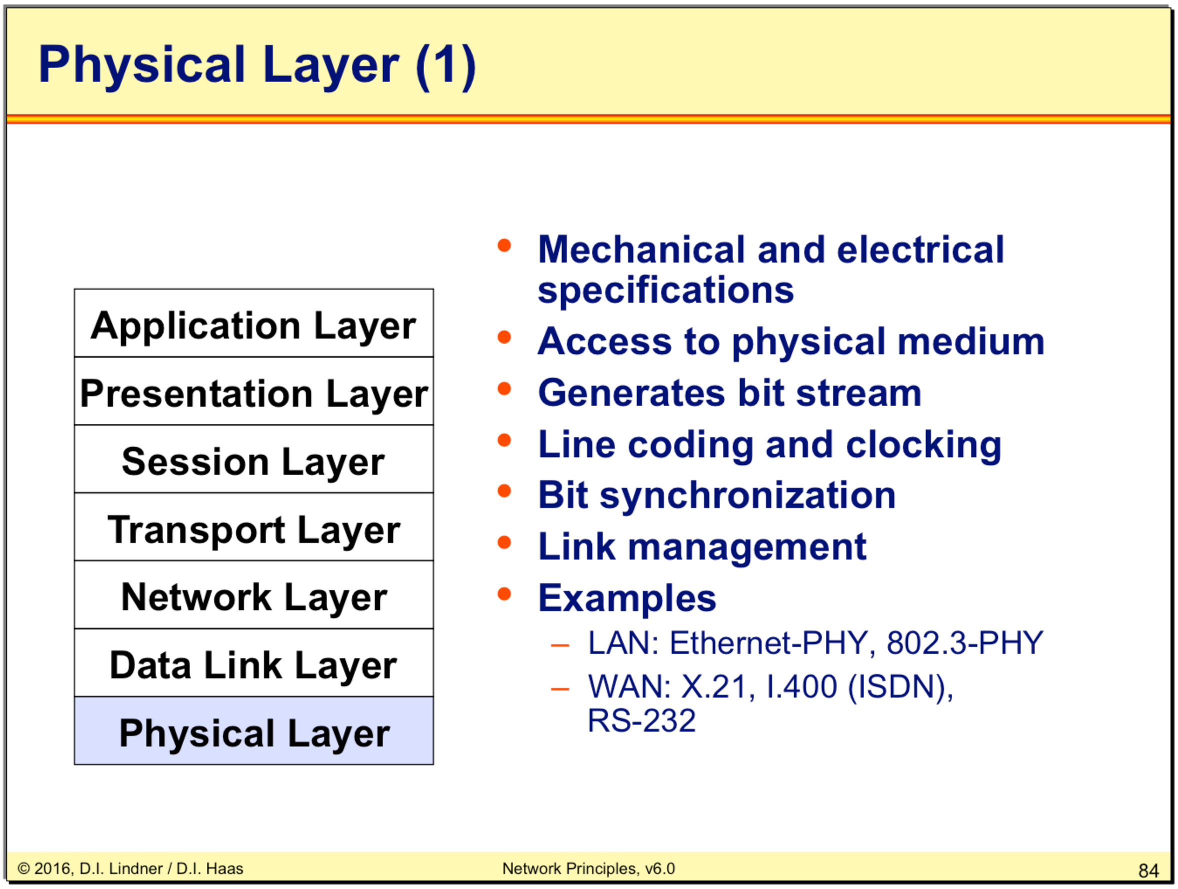
© 2016, D.I. Lindner / D.I. Haas

Example Topology with ES and IS



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 83

POTS ...	Plain Old Telephony System
M ...	Modem
ISDN ...	Integrated Services Digital Network
PDH ...	Plesiosynchronous Digital Network
LAN ...	Local Area Network
ES ...	Endsystem

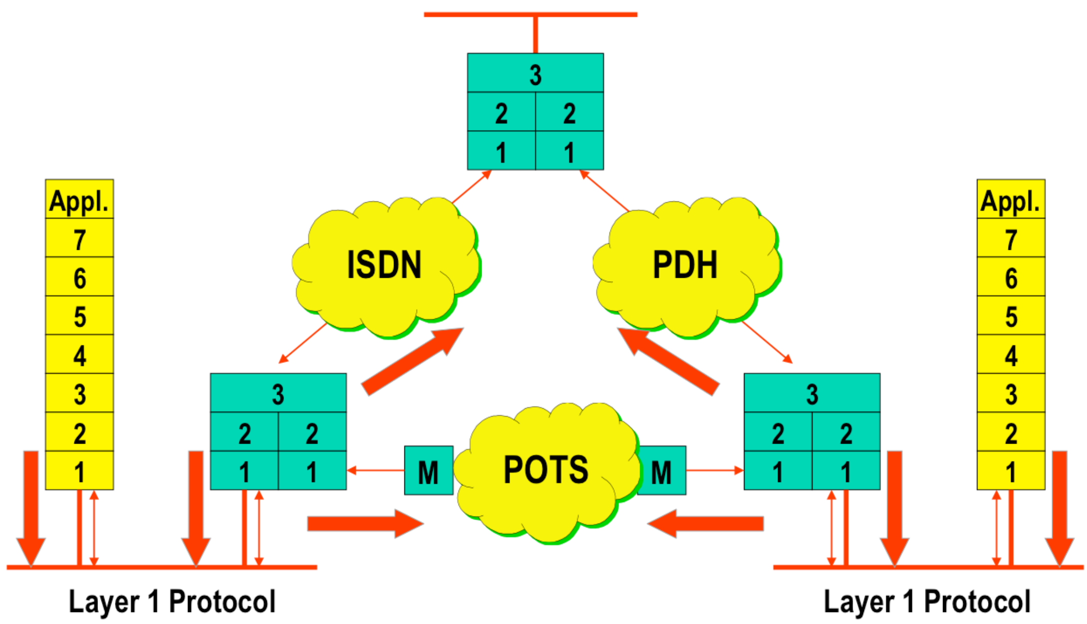


The Physical Layer generates the bit stream. This layer provides access to the physical medium by applying line coding (NRZ, Manchester, etc), bit synchronization (clocking, PLL), but also includes mechanical, electrical and optical specifications. Layer 1 also can activate or deactivate the links between end systems (link management).

The physical part of the Ethernet NIC is called "PHY" and is perhaps the most complex entity of Ethernet because the PHY consists of a number of sub-layers that care for interoperability with different Ethernet speeds (10, 100, 1000, 10000 Mbit/s) and coding (Manchester, 4B5B, 8B10B, ...). Note that there is a fundamental difference between "Ethernet" and IEEE "802.3": these are two separate LAN specifications but typically implemented on the same NIC—they just share the same topology and use the same media access strategy—most people are not aware of that.

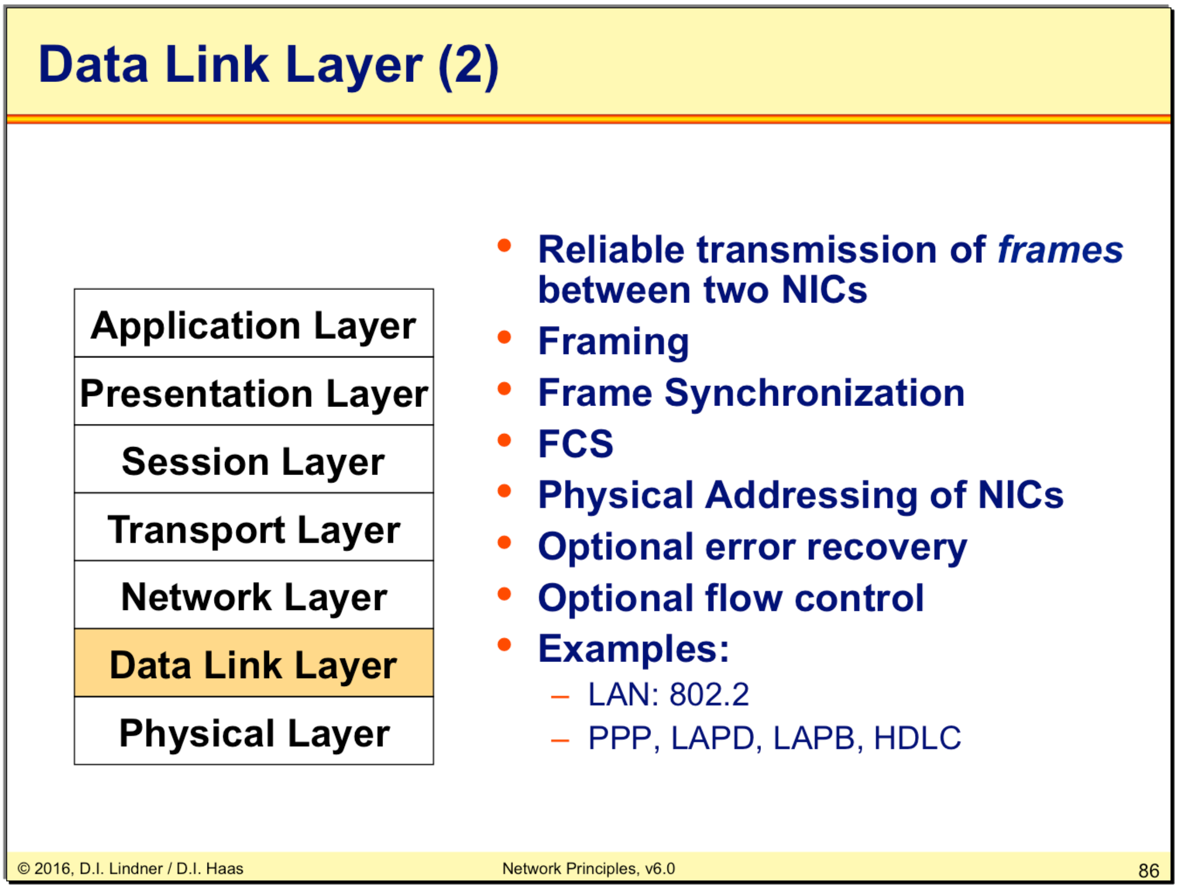
The X.21 is a typical and widely available interface on a Cisco router. The ISDN-layer 1 is specified in the ITU-T standard I.400 and describes both a 192 kbit/s synchronous multiplexing interface capable to transport 2 B channels and one D channel and secondly a high speed 2.048 Mbit/s interface capable to carry 30 B channels and one D channel. These ISDN specifics are presented in the N-ISDN chapter in more details. The old well-known Recommended Standard (RS) 232 specifies the classical serial interface found on many PCs and other peripheral devices.

Physical Layer (1)



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 85

© 2016, D.I. Lindner / D.I. Haas

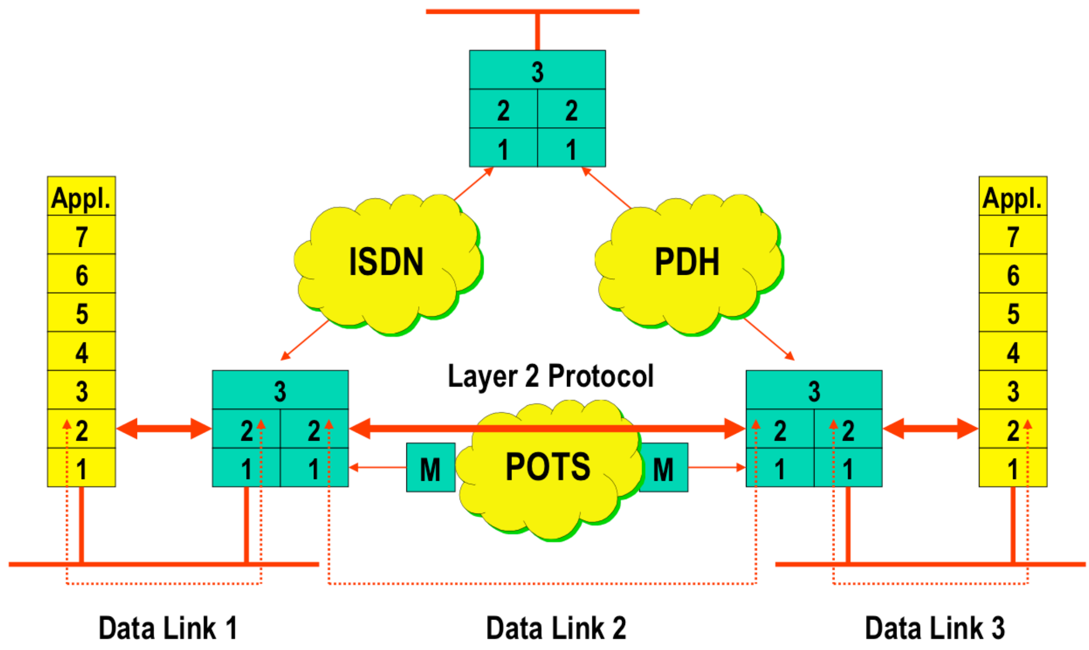


The data link layer builds the frame. In that way, framing or frame synchronization is the most important thing on layer 2. Where is the beginning of the frame ? Where is the end ? With a special Bit-Code the layer 2 protocols, such as HDLC or PPP, guarantee the framing of the data. That's important for the MTU (maximum transfer unit).

Also frame checking, correction of transmission errors on a physical link, is implemented on layer 2. There are also a physical address of the network interface cards. This address is transported with the data link layer too (e.g. MAC-Address with Ethernet).

Error recovery and flow control may be realized in connection-oriented mode.

Data Link Layer (2)



© 2016, D.I. Lindner / D.I. Haas

© 2016, D.I. Lindner / D.I. Haas

Network Layer (3)

Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

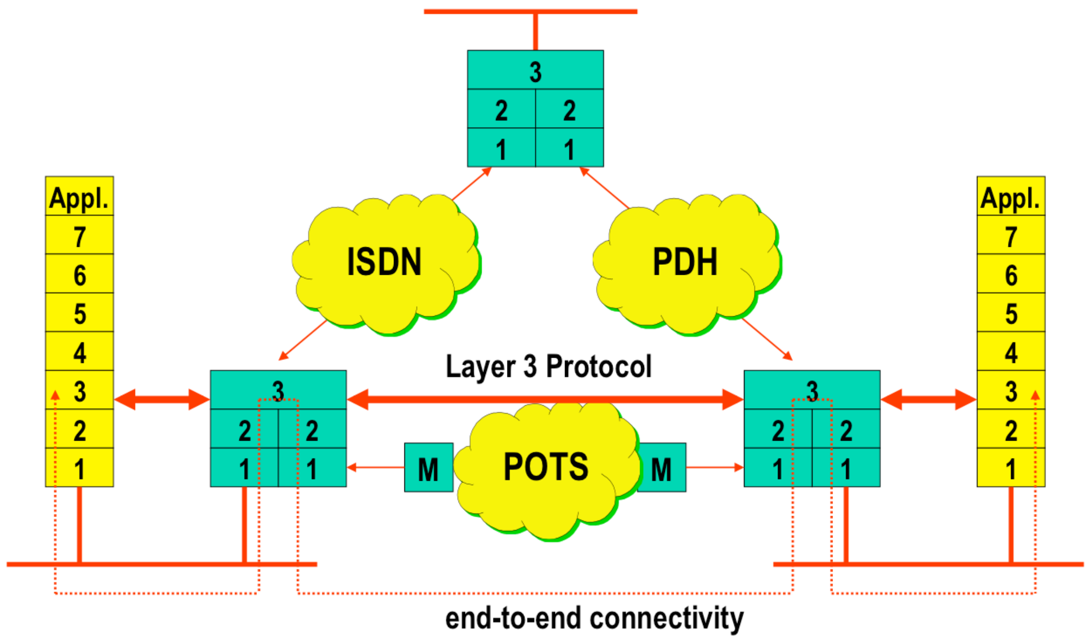
- **Transports *packets* over packet switching networks**
  - Routing / Switching
- **Provides structured addresses to name networks**
  - N-SAP address
- **Fragmentation and reassembling**
- **Examples:**
  - CLNP
  - IP, IPX
  - Q.931, X.25

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 88

The network layer builds the so-called "packet". Layer 3 transports the packets between the different networks. Therefore layer 3 needs structured and routable addresses to find the right networks. IP is the most important Layer 3 protocol today (IPv4 has a structured 4 byte address). The OSI Connectionless Network Protocol (CLNP) is another example for a layer-3 protocol but it is not so widely used today, except some Telcos and Carriers use it for internal purposes. IPX has been developed by Novell in order to extend Novell networks over different data-link layer worlds. Q.931 is the ISDN layer 3 carried over the D-channel and is used for signaling purposes. Basically Q.931 conveys the telephone numbers and other service parameters. The classical packet-switched WAN standard X.25 actually specifies only the layer 3 of this technology and is used to set up a number of virtual calls over an asynchronous link layer (LAPB).

© 2016, D.I. Lindner / D.I. Haas

Network Layer (3)



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 89

© 2016, D.I. Lindner / D.I. Haas

Transport Layer (4)

- Application Layer
- Presentation Layer
- Session Layer
- **Transport Layer**
- Network Layer
- Data Link Layer
- Physical Layer

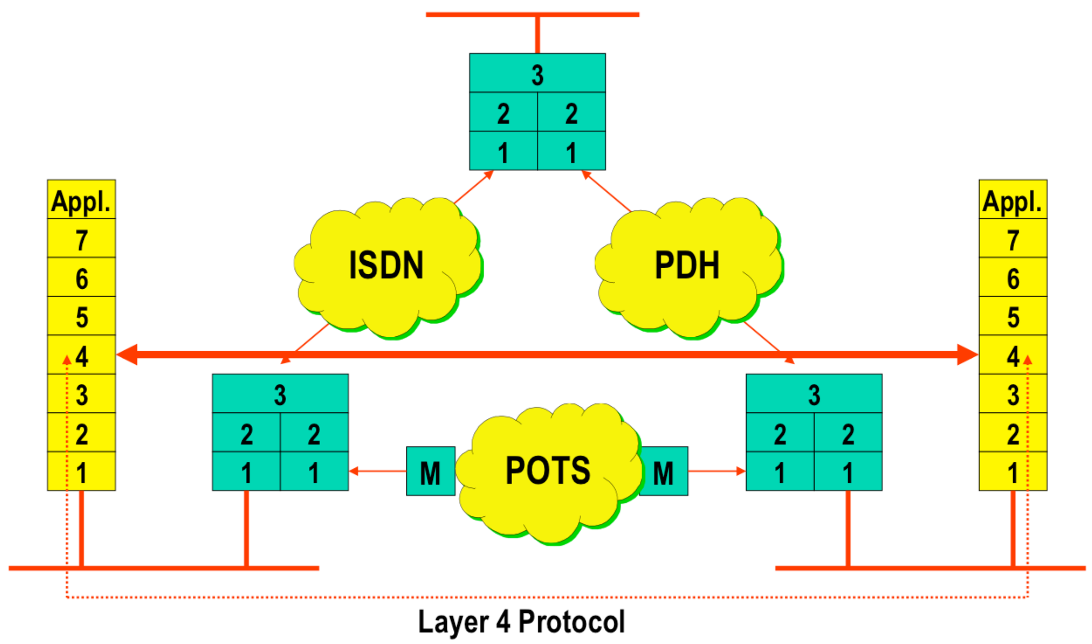
- **Transport of *segments* between applications (end systems)**
  - Reliable if error recovery by ARQ
- **Application multiplexing through T-SAPs**
  - Transport Addresses
- **Sequence numbers and Flow control**
- **Optional QoS Capabilities**
- **Examples:**
  - TCP (UDP)
  - ISO 8073 Transport Protocol

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 90

The transport layer is necessary to build a logical connection to the application in order to send data in so-called "segments". With the help of port numbers (by TCP and UDP), a layer 4 protocol guarantees the transport of the segments to the right application. These port numbers are called T-SAPs in the OSI world. The transport layer optionally takes care about flow control, reliable transmission between end systems, and is most important for QoS capabilities. Flow control requires connection oriented mode. Depending on the capabilities of the underlying layers regarding error recovery connection oriented mode is necessary for reliable transport

© 2016, D.I. Lindner / D.I. Haas

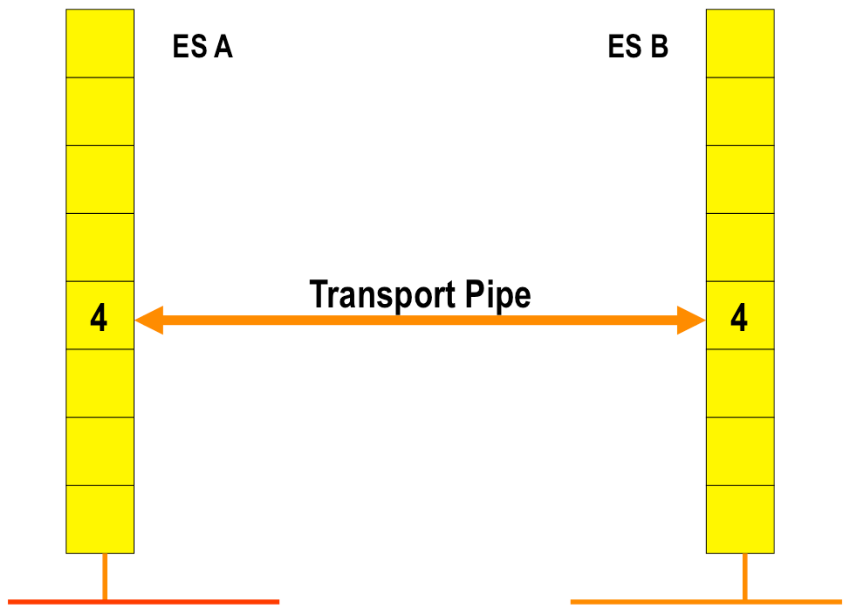
Transport Layer (4)



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 91

L04 - Network Principles (v6.0)

How Layer 5 sees the Network



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 92

Functions of Higher Layers

- **Layer 5 (session layer)**
  – Coordinates and controls dialogue between different end systems
- **Layer 6 (presentation layer)**
  – Responsible for common language between end systems (conversion, adaptation of data)
- **Layer 7 (application layer)**
  – Supports user with common network applications (e.g. file transfer, virtual terminal) or basic network procedures in order to implement distributed applications (e.g. transaction systems)

© 2016, D.I. Lindner / D.I. Haas

Establishing Sessions

The Session Layer, one layer above the transport layer, is responsible for establishing sessions between applications. Please note the difference: the transport layer establishes connections between end nodes, the session layer establishes sessions between applications (or processes) residing in those nodes. Because one transport layer connection may be used by many session, every session layer uses a kind of session identifier to distinguish between different session. Functions of the session layer include:

Establishing and maintaining sessions between host processes.

Flow control, session recovery and synchronization.

Translation between names and addresses.

Presenting the Information

The Presentation Layer is concerned with syntax (language used in application messages to transfer commands and responses) and context (protocols to achieve a certain purpose) of the application protocols.

Application Support

The Application Layer deals with the actual communications support of applications. Please note, that no actual application like word processing or database access is part of the application layer. The OSI model is a model for communications, therefore the application layer contains protocols used to support the operating system of, or the application within the end node concerning to their communications needs. For this reason, the application layer supports basic communications functions like remote terminal access, file transfer, email connectivity, transaction processing, etc.

© 2016, D.I. Lindner / D.I. Haas

Session Layer (5)

Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

- **Provides a user-oriented connection service**
  - *Synchronization Points*
- **Little capabilities, usually not implemented or part of application layer**
  - Telnet: GA and SYNCH
  - FTP: re-get allows to continue an interrupted download
  - ISO 8327 Session Protocol

© 2016, D.I. Lindner / D.I. Haas

The Session Layer coordinates and controls dialogue between different end systems. This layer is only seldom or sparsely implemented. For example a Telnet server gives the sending permission to the Telnet client via a Go Ahead (GA) sequence. Using the BRK-Key, a SYNCH sequence is triggered and the server must synchronize with the client by flushing the buffered stream. FTP keeps track of the data blocks transmitted and is able to continue an interrupted session from this checkpoint on.

Session protocols are important with telephony applications such as H.323 which employs H. 225 to establish sessions. Another example is the IETF Session Initiation Protocol (SIP). The ISO 8327 is an OSI basic connection oriented session protocol specification.

© 2016, D.I. Lindner / D.I. Haas

Presentation Layer (6)

Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

- **Specifies the data representation format for the application**
- **Examples:**
  - MIME (part of L7) and UUENCODING (part of L7)
  - ISO: ASN.1 and BER

© 2016, D.I. Lindner / D.I. Haas

The layer 6 is responsible for common language between end systems. The presentation layer specifies the "meaning" of the data and how each byte should be interpreted.

In the Internet the presentation layer uses ASCII coding and the meaning of the data is specified by a so-called "Multipurpose Internet Mail Extension" (MIME) header. MIME is used by SMTP (Email) and HTTP (Web browsing) for example. UUENCODING is one example of how to transform 8-bit-bytes into 7-bit-bytes and it is typically used with Internet Mail attachments. The ISO/OSI world generally uses the "Abstract Syntax Notation Language Number One" (ASN.1) as common presentation layer. This language is used to specify data structures and contents. On the wire the data is transmitted using the "Basic Encoding Rules" (BER).

© 2016, D.I. Lindner / D.I. Haas

Application Layer (7)

Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

- **Provides network-access for applications**
- **Examples:**
  - ISO 8571 FTAM File Transfer Access + Management,
    X.400 Electronic Mail, CMIP
  - SMTP, FTP, SNMP, HTTP, Telnet, DNS, …

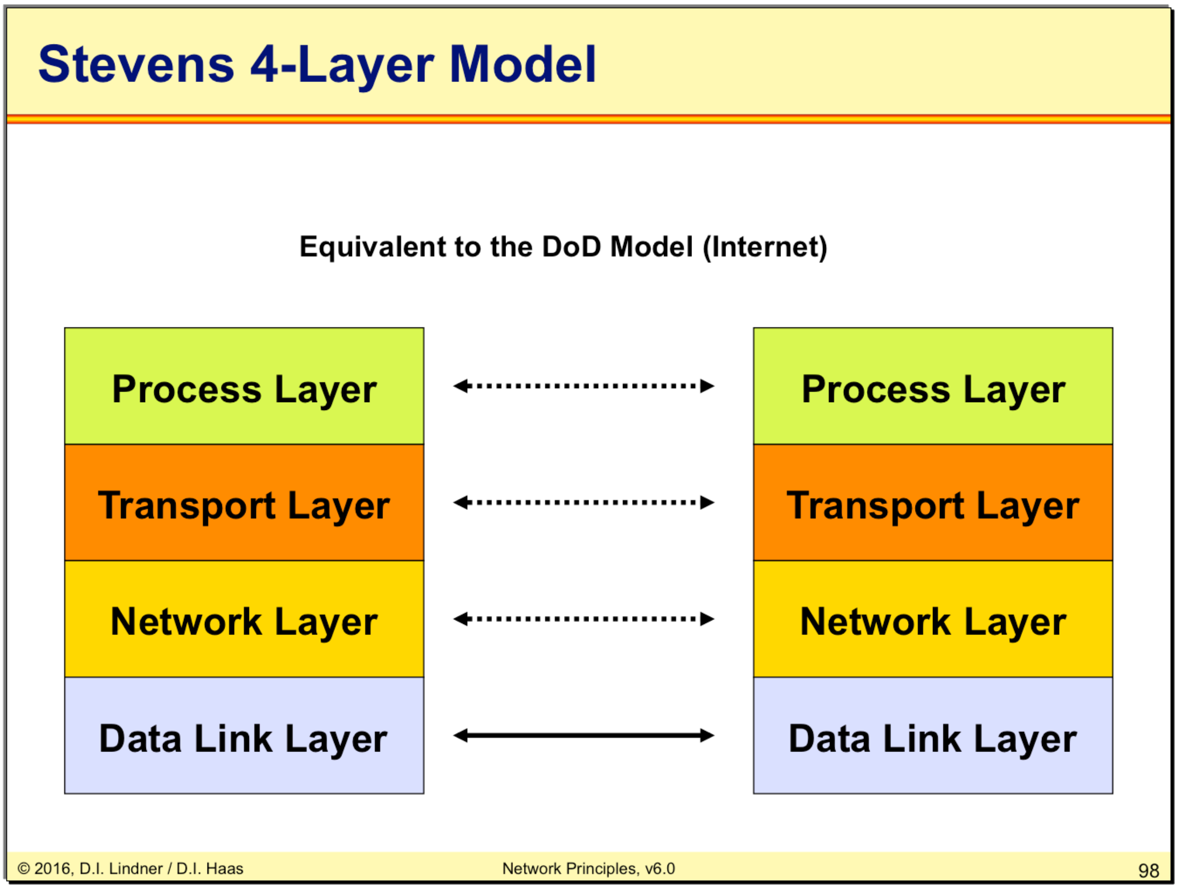
The Application layer supports user with common network applications. For example: file transfer or virtual terminals. Layer 7 also supports basic network procedures in order to implement distributed applications (e.g. transaction systems). Note that the application layer is not identical with the application! The application itself "sits" upon the application layer and uses the service primitives provided by the application layer to access the network.

Application layer protocols either use "inline" or "inband" control sequences (as it is used with Telnet), where control bytes are mixed with the data stream, or it might use a predefined frame structure, consisting of header and body. Another method is to open a dedicated logical control connection only to exchange control information (as it is used with FTP).

Padlipsky's Rule

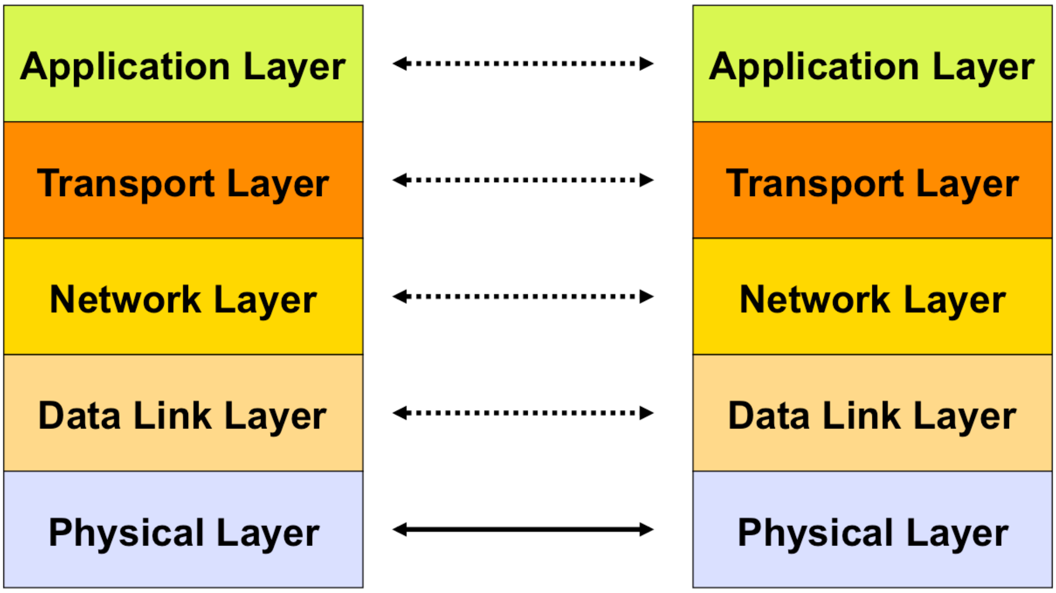
If you know what you're doing, three layers is enough. If you don't, even seventeen won't help.

Until now we discussed the famous OSI seven layer reference model, but real implementations typically consist of a subset of this 7-layer model. On the one hand, not all OSI layers are necessary in real-world applications, on the other hand, many important technologies had been created before the OSI standard.



The picture above shows the W. Stevens 4 layer model which is used also in the Internet. The Internet layer model is also called "Department of Defense" (DoD) model.

Tanenbaum 5-Layer Model



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 99

The famous computer scientist Andrew S. Tanenbaum defined a more practical approach utilizing five layers. Other than the DoD or Stevens 4-layer model the physical specifications are defined in a separate layer.

© 2016, D.I. Lindner / D.I. Haas

The Internet perspective is implement it, make it work well, then write it down.

The OSI perspective is to agree on it, write it down, circulate it a lot and now we'll see if anyone can implement it after it's an international standard and every vendor in the world is committed to it.

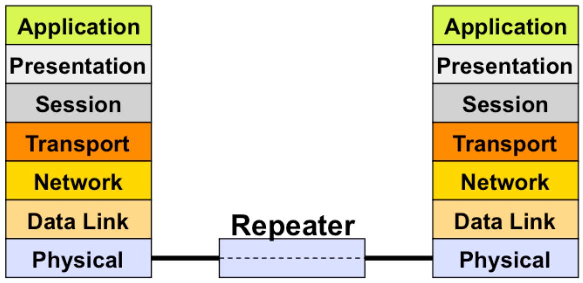
One of those processes is backwards, and I don't think it takes a Lucasian professor of physics at Oxford to figure out which.

Marshall Rose, "The Pied Piper of OSI"

© 2016, D.I. Lindner / D.I. Haas

Layer 1 Devices

- Adapts to different physical interfaces
- Amplifies and/or refreshes the physical signal
- No intelligence
- Examples: Repeater, Hub, ISDN NT1



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 101

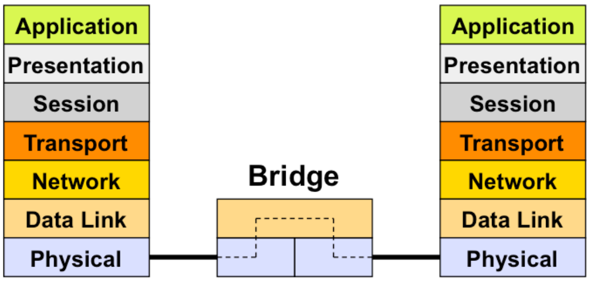
To connect different system with each other we need special devices. If you want to connect two systems only per physical layer you need a so called “hub” or “repeater”.

This kind of devices are not intelligence and only used to amplifies or refresh the physical signal, or to connect systems with different physical interfaces.

© 2016, D.I. Lindner / D.I. Haas

Layer 2 Devices

- Filter/Forwards frames according Link Layer Address
- Incorporates Layer 1-2
- LAN-Bridge (“Ethernet Switch")



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 102

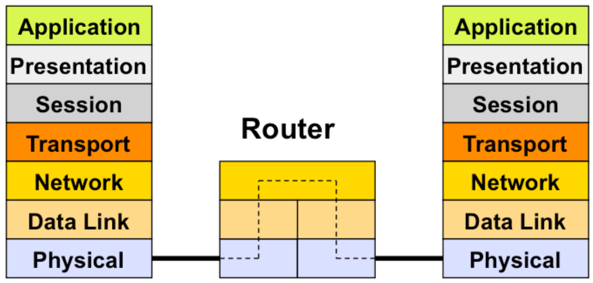
A so called “bridge” or “switch” is a device to connect systems per data link layer. This kind of devices determine the physical layer and can forward datagram's according the link layer address. For example: MAC address with Ethernet. Note that a bridge utilizes two or more physical layer entities (NICs) that is a bridge is able to convert encodings and signal-rates.

Note the difference between “bridge” and “switch”: A bridge is implemented in software, whereas a switch is built in hardware. Today only switches are used, because they are much faster.

© 2016, D.I. Lindner / D.I. Haas

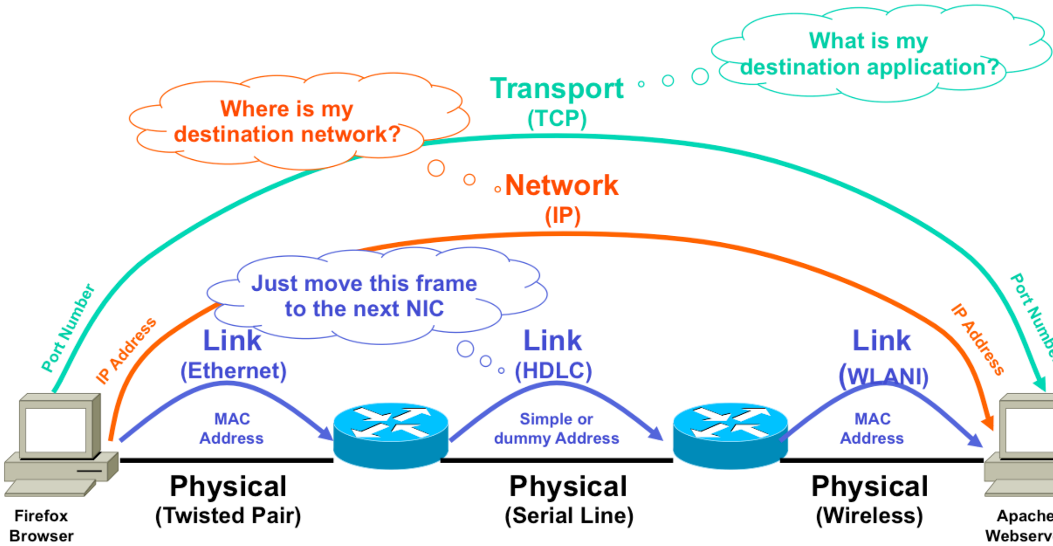
Layer 3 Devices

- "Packet Switch" or "Intermediate System"
- Forwards packets to other *networks* according *structured* address
- Terminates Links
- IP Router, WAN-Switch
  - X.25
  - Frame-Relay
  - ATM



The most important device in the Internet is the so called “router”. A router consists of several layer 1 and layer 2 entities and a single layer 3 entity, thus it can forward packets to other networks according structured addresses (remember IP-Addresses). By terminating layer 1 and 2 a router is able to connect total different network technologies with each other. For example: on one side there is Ethernet on the other side ATM.

A Practical Example



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 104

In the picture above you see a good example in which "symbolic" way the different layers talk with each other. The link layer only searches for the right NIC address. IP only wants to the destination network, and TCP is the protocol to communicate between applications. Most importantly, notice that packets are sent over different link layer technologies such as Ethernet, HDLC, or WLAN. Exactly this is the reason why a common network layer is needed to allow communication over different "networks" (=links).

Don't be confused about the different usages of the term "network". People say "network" and mean "bunch of devices interconnected with each other". To be more exact, a network is identified by a unique network identifier, such as the network-ID of the IP-address. Since a contiguous link layer implementation (such as an Ethernet LAN) can have assigned a single IP net-ID, each link can be regarded as network.

© 2016, D.I. Lindner / D.I. Haas

OSI Summary

- Network layers ensures interoperability and eases standardization
- ISO/OSI 7 layer model is an important reference model
- Practical technologies employ a different layer set, but it's always possible to refer to OSI

© 2016, D.I. Lindner / D.I. Haas

© 2016, D.I. Lindner / D.I. Haas

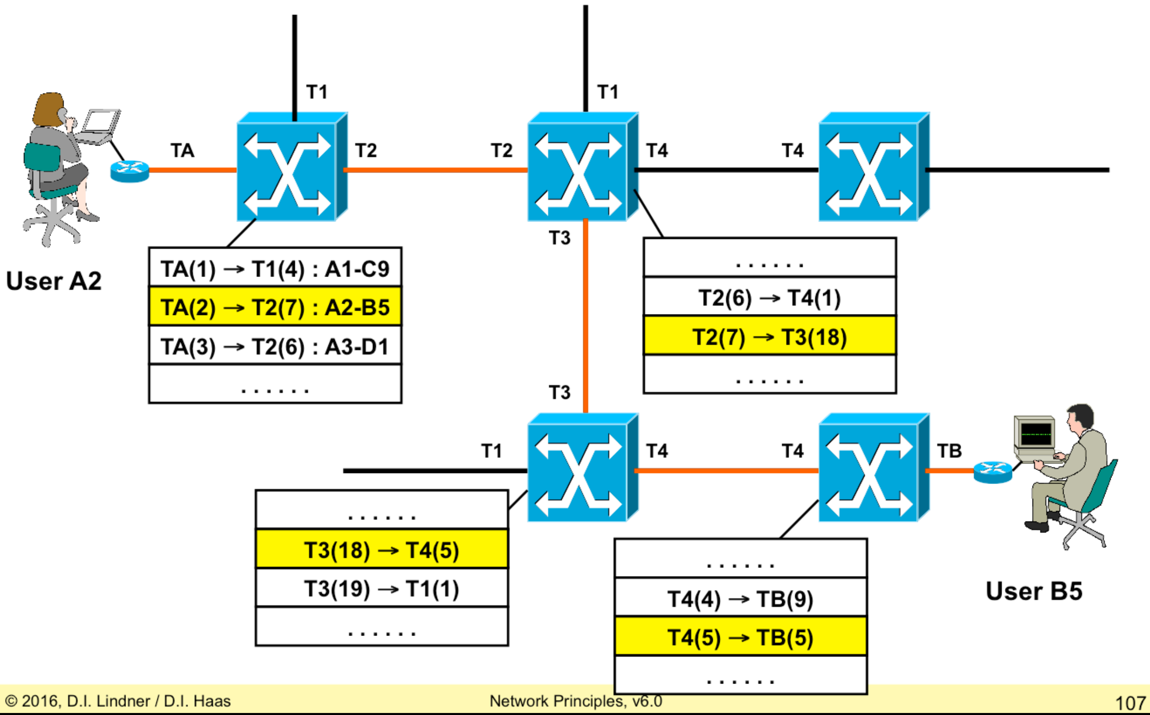
Agenda

- Introduction
- Circuit Switching
- Packet Switching
  - Principles
  - Datagram Service
  - Virtual Call Service
- OSI Reference Model
- Summary of Network Methods

© 2016, D.I. Lindner / D.I. Haas

© 2016, D.I. Lindner / D.I. Haas

Circuit Switching



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 107

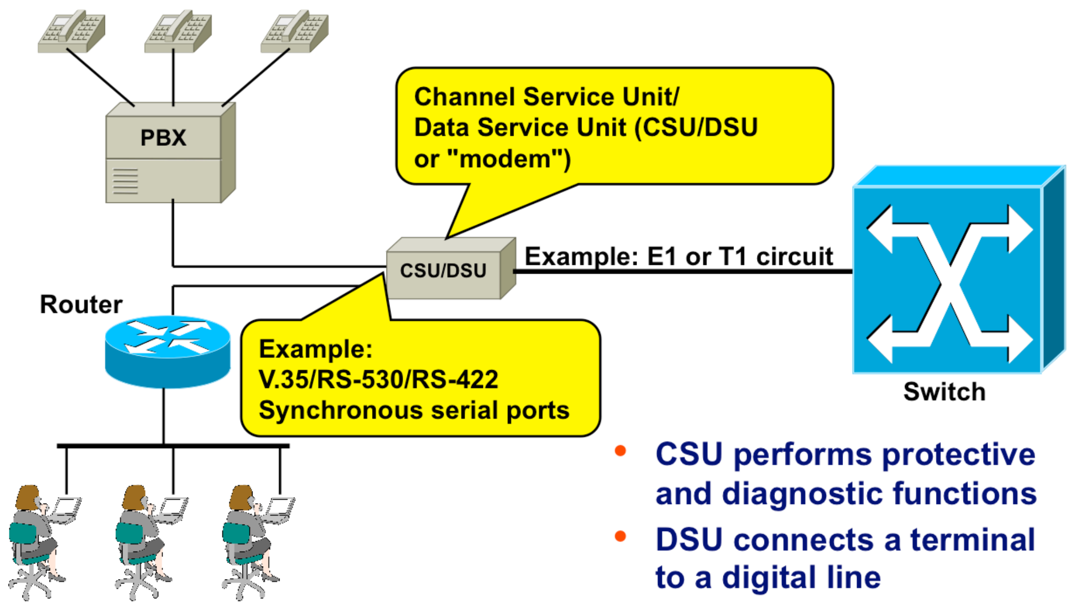
Circuit switching technology is based on deterministic TDM.

All network switches in circuit switching technology hold a switching table which determines the correlation between incoming trunk/timeslot and outgoing trunk/timeslot.

In our example the connection between user A2 and B5 is established by four network switches and their according switching tables. For both users this connection looks like a dedicated point to point link, they are not aware what's going on inside the network cloud.

© 2016, D.I. Lindner / D.I. Haas

Legacy Company Access (in the past)



- CSU performs protective and diagnostic functions
- DSU connects a terminal to a digital line

© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 108

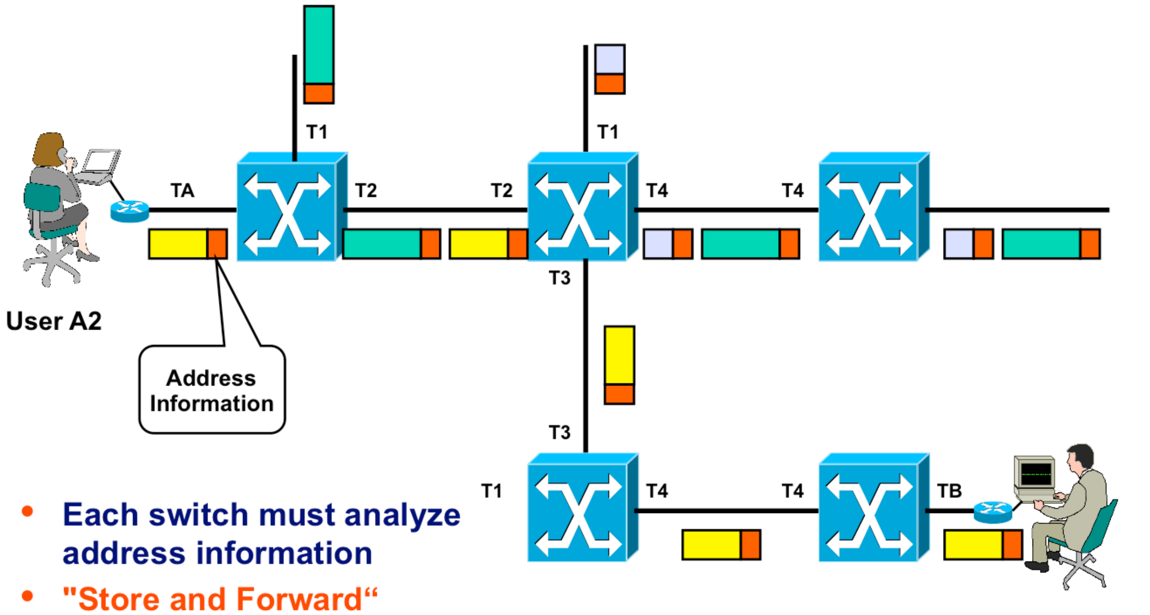
In real life a typical user configuration looks like the one shown in our example.

We find some users that are connected via a shared media (Ethernet LAN) to an IP router. The router itself is connected to a Channel Service Unit (CSU) or Data Service Unit (DSU) using an synchronous serial interface with a data rate of up to 2Mbit/s.

This CSU/DSU is responsible for terminating the TDM circuit which is supplied by the service provider as well as for the conversion between the synchronous serial interface and the TDM interface. In our scenario an PDH E1 (2048 Mbit/s) or T1 (1544 Mbit/s) circuit is used.

The connection supplied by the service provider might be shared between the router and the Private Branch Exchange. So the router uses reserved timeslots of the E1/T1 trunk for data traffic while the PBX is using some other timeslots to establish phone calls.

Packet Switching



- Each switch must analyze address information
- "Store and Forward"
- Decision about where to forward is based on tables

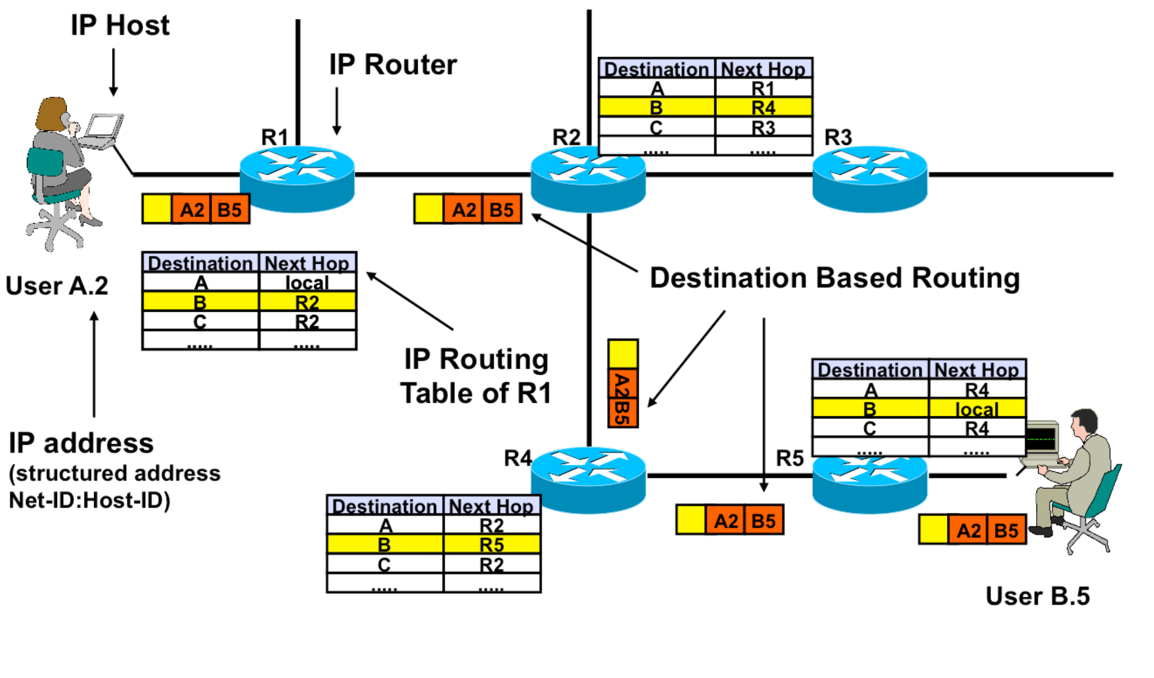
© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 109

In packet switching technology which is based on statistical time division multiplexing addresses are needed, remember there is no correlation between timeslot and destination.

Each switch must analyze the destination address of every data packet to be able to forward it according to some forwarding table.

In our example user A2 communicates with user B2 by the help of addresses.

IP Datagram Service



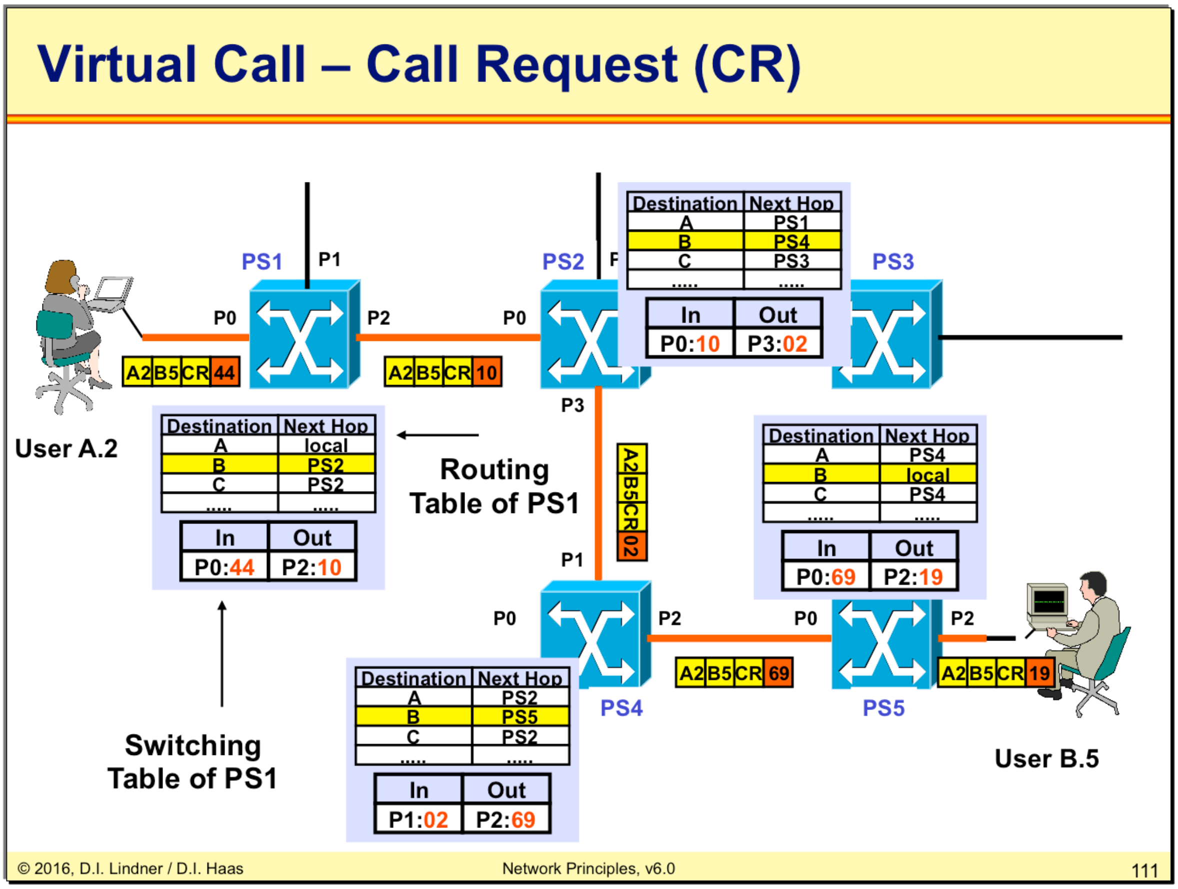
© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 110

In the Datagram technology user A.2 sends out data packets destined for the user B.5. Each single datagram holds the information about sender and receiver address.

The datagram forwarding devices in our example routers hold a routing table in memory. In the routing table we find a correlation between the destination address of a data packet and the corresponding outgoing interface as well as the next hop router. So data packets are forwarded through the network on a hop by hop basis.

The routing tables can be set up either by manual configuration of the administrator or by the help of dynamic routing protocols like RIP, OSPF, IS-IS, etc. The use of dynamic routing protocols may lead to rerouting decisions in case of network failure and so packet overtaking may happen in these systems.

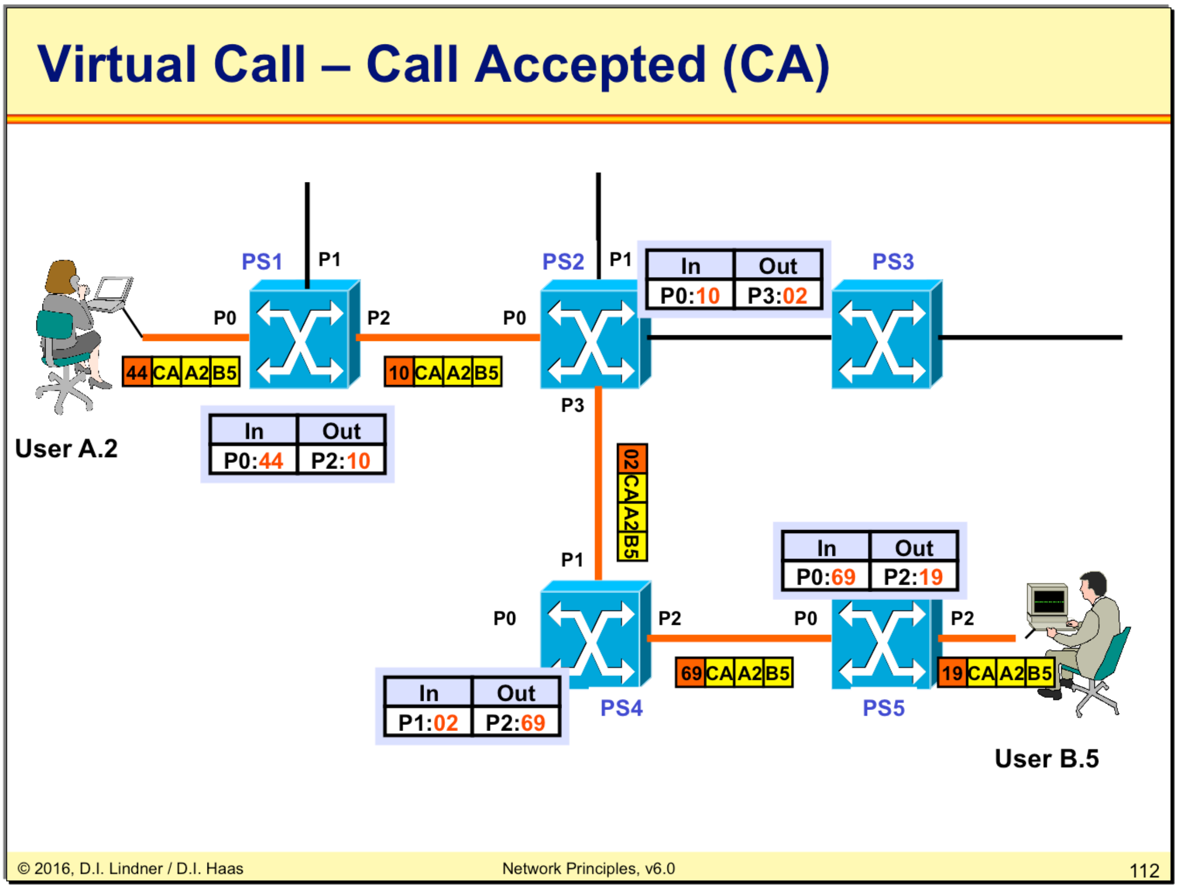
© 2016, D.I. Lindner / D.I. Haas



In Virtual Call Service technology addresses are used as well, but in a different manner than compared to datagram services. The address information in Virtual Call Service systems is only used at the beginning of a conversation to setup a connection.

With an established connection data packets are forwarded according to virtual circuit identifiers which are held in switching tables.

In our example user A.2 sends a connection setup request to user B.5. This connection setup request is forwarded by the network under the use of routing tables. This routing tables can be configured manually by an administrator or dynamically by the help of routing protocols e.g. PNNI.



The connection setup request builds up a tunnel-like connection of virtual circuit identifiers held in switching tables.

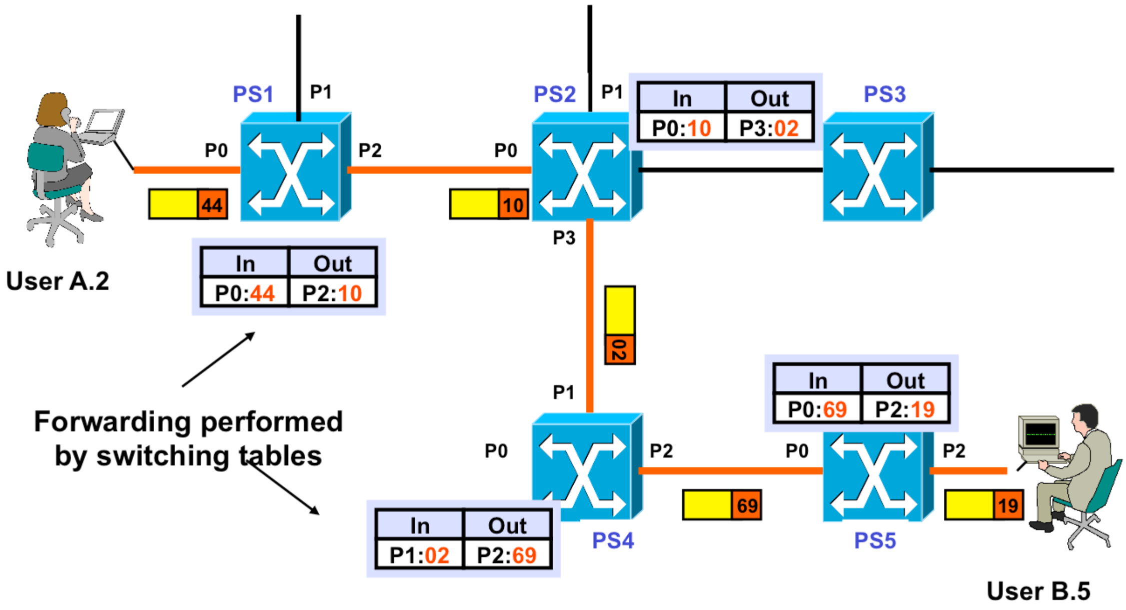
User B.5 hopefully answers with a connection accept message back through the already established tunnel. From now on only switching tables with their circuit identifiers are used to forward the data packets.

The entries in the switching tables are created dynamically during the connection setup procedure by each network node.

© 2016, D.I. Lindner / D.I. Haas

L04 - Network Principles (v6.0)

Virtual Call – Data Transfer



© 2016, D.I. Lindner / D.I. Haas Network Principles, v6.0 113

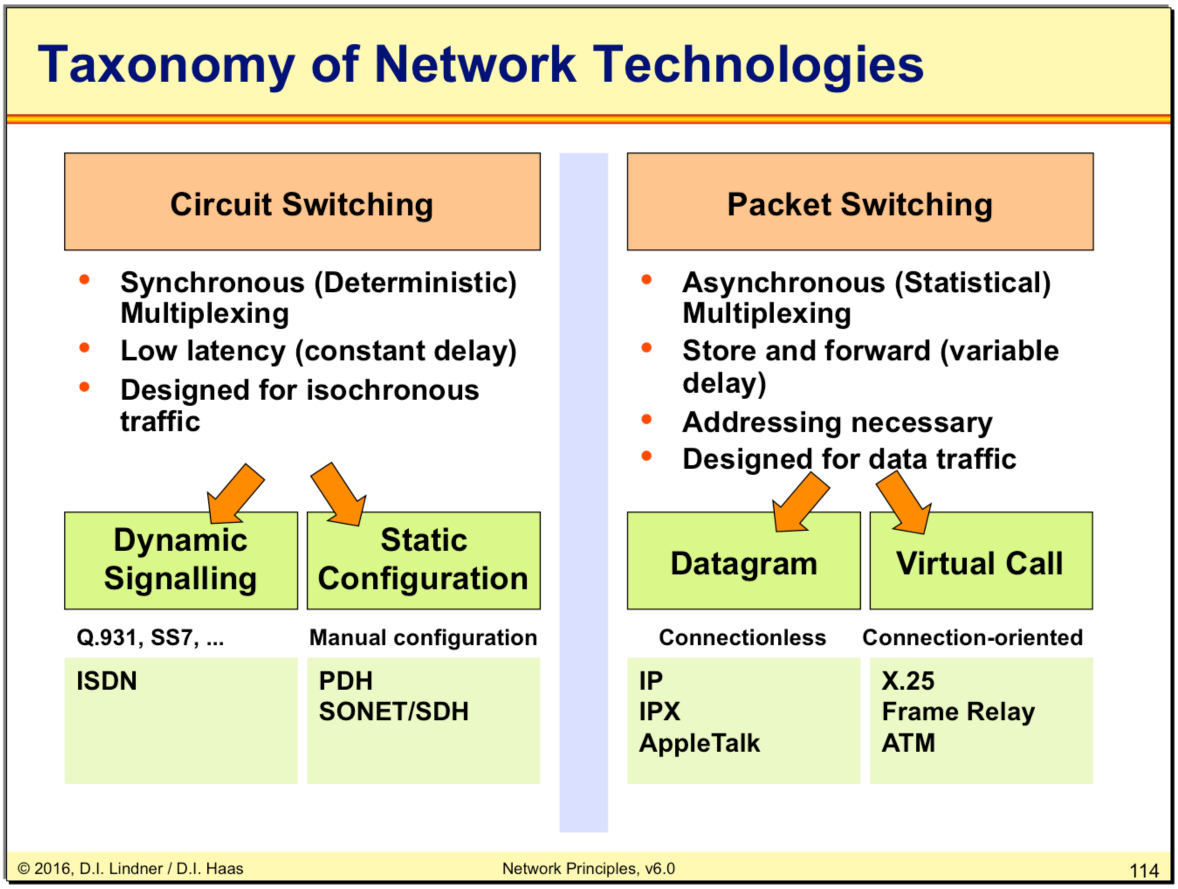
During the data transport phase there is no more need for addresses.

Data packets are forwarded using virtual circuit identifiers, which change on a hop per hop basis. Circuit identifiers have only local meaning in combination with their according trunk connection.

This behavior also prevents things like packet overtaking and makes it easier to implement QoS technologies in the network.

If a connection between two nodes is lost due to network failure, a new connection is established, starting with the connection setup procedure right from the beginning.

© 2016, D.I. Lindner / D.I. Haas

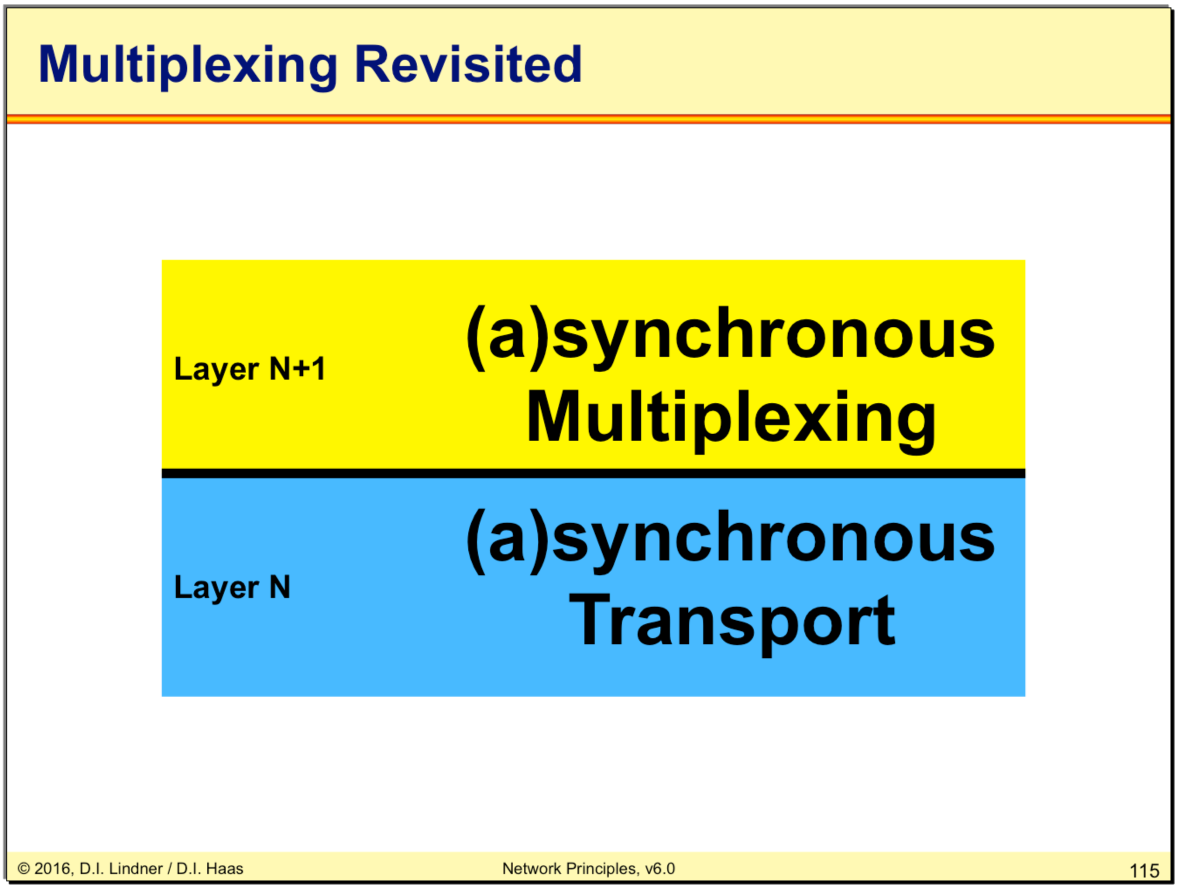


This slide gives us an good overview about the TDM technologies discussed so far.

On the top of this graphic we find the two basic flavors of TDM systems circuit switching based on deterministic TDM and packet switching based on statistical TDM.

Current circuit switching technologies are ISDN and PDH systems which can be used for SVC services using the signaling protocols Q931 and Signaling System Seven (SS7) or based on PVC technique using manually configured SONET/SDH channels.

Current packet switching technologies can be split up in Datagram Services like IP, IPX etc. or Virtual Call Services like X25, ATM, Frame-relay etc.



This slide wants to tell you that the world is not black and white only, but is always made up of some kind of colored grey.

The same is true for networking. Networks are made up of layers and each layer has its own identity and properties with interfaces to the next higher or lower layer.

So its quite easy to take a synchronous layer and put something asynchronous on top of it. Like ATM or IP on top of SONET/SDH.

Summary

- **Only two fundamental network principles**
  - Circuit switching
  - Packet switching
  - The first is good for real-time voice (traffic) the latter is good for data traffic
  - But everybody wants to have the best of both worlds
- **Packet switching allows two basic types:**
  - Datagram (CL) versus Virtual Call (CO)
  - Different address types (!) for forwarding decision of data packet
    - Unique routable address (CL)
    - Local connection identifier (CO)

© 2016, D.I. Lindner / D.I. Haas