L3 – Network Layer

Routing

- **Routing** = the process of finding a path in the network between two communicating nodes
  - the route/path has to satisfy certain constraints
  - influenced by several factors:
    - *static ones:* network topology
    - *dynamic ones:* network load

L3 – Network Layer

The Global View Problem

- the global knowledge of network topology is problematic
  - it's very difficult to acquire it
  - if yet acquired, it's not actual any more
  - it has to be locally relevant
- a local view of network topology represents a *routing table*
- the difference between local and global knowledge can lead to:
  - cycles/loops (i.e., black holes)
  - oscillation (load adaptability)

L3 – Network Layer

Routing – the goal

- the main goal of routing is:
    - to find optimal paths
        - the optimality criterion is a *metric* – a cost assigned for passing through a network
    - to deliver a data packet to its receiver
- the routing *usually* does not deal with the whole packet path
    - the router deals with just a single step – to whom should be the particular packet forwarded
        - somebody "closer" to the recipient
        - so-called *hop-by-hop* principle
    - the next router then decides, what to further do with the received packet

L3 – Network Layer

Routing – basic approaches

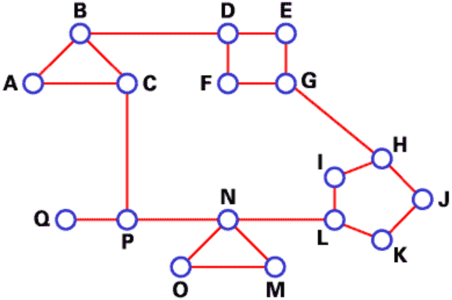
The basic approaches divide based on the routing table creation/maintenance:

- *static (non-adaptive)*
  - manually (by hand) edited records
  - suitable for a static topology and smaller networks
- *dynamic (adaptive)* – these respond to network changes
  - complex (usually distributed) algorithms
  - e.g.:
    - *centralized* – a centre controls the whole routing
    - *isolated* – every node on its own
    - *distributed* – nodes' cooperation

L3 – Network Layer

Routing – mathematical view

- the routing can be seen as a problem of graph theory
- a network can be represented by a graph, where:
  - nodes represent routers (identified by their IP addresses)
  - edges represent routers' interconnection (a data link)
  - edges' value = the communication cost
  - *the goal:* to find paths having minimal costs between any two nodes in the network



L3 – Network Layer

Routing – routing algorithms' required features

Required features of any routing algorithm:

- accuracy
- simplicity
- effectivity and scalability
    - to minimize an amount of control information ($\approx$ 5% of the whole traffic!)
    - to minimize routing tables' sizes
- robustness and stability
    - a distributed algorithm is necessary
- fairness
- optimality
    - *"What should be treated as the best path?"*

L3 – Network Layer

Routing – basic approaches to distributed routing

Basic approaches to distributed routing:

- *Distance Vector (DV)* – Bellman-Ford algorithm
  - the neighboring routers periodically (or when the topology changes) exchange complete copies of their routing tables
  - based on the content of received updates, a router updates its information and increments its *distance vector number*
    - a metric indicating the number of hops in the network
  - i.e., *"all pieces of information about the network just to my neighbors"*
- *Link State (LS)*
  - the routers periodically exchange information about states of the links, to which they are directly connected
  - they maintain complete information about the network topology – every router is aware of all the other routers in the network
  - once acquired, the Dijkstra algorithm is used for shortest paths computation
  - i.e., *"information about just my neighbors to everyone"*

L3 – Network Layer

Distance Vector – RIP protocol

- the principal actor of DV routing
  - RIPv1 (RFC 1058)
  - RIPv2 (RFC 1723) – adds several features (e.g., an authentication of routing information)
- the networks are identified using the CIDR mechanism
- the number of hops is used as a metric
  - transfer of a packet between two neighboring routers = 1 hop
  - infinity = 16
    - $\Rightarrow$ the RIP cannot be used for networks with minimal amount of hops between any two routers > 15
- the routers send the information periodically every 30 seconds
  - triggered updates when a state of a link changes
  - timeout 180s (detection of connection errors)
- usage:
  - suitable for small networks and stable links
  - not advisable for redundant networks

L3 – Network Layer

Link State – OSPF protocol

- *Open Shortest Path First*
- currently the mostly used LS protocol
- metric: *cost*
  - a number (in the range between 1 and 65535) assigned to each router's network interface
  - the lower the number is, the better the link/path is (i.e., will be preferred)
  - by default, every interface is automatically assigned a cost derived from the link's throughput
    - $cost = 100000000/bandwidth$ (bw in bps)
    - might be manually edited
- extensions:
  - message authentication
  - routing areas – next layer of hierarchy
  - load-balancing – more links/paths with the same cost

L3 – Network Layer

Routing – Link State vs. Distance Vector

Link State

- *Complexity:*
  - every node has to know the cost of every link in the network ⇒ $O(nE)$ messages
  - once a link state changes, the change has to be propagated to *every* node
- *Speed of convergence:*
  - $O(n^2)$ alg., sends $O(nE)$ messages
  - sustains from oscillations
- *Robustness:*
  - wrongly functional/compromised router spreads wrong information just about the links it is directly connected to
  - every router computes routing tables on its own ⇒ separated from routing information propagation ⇒ a form of robustness
- *Usage:*
  - suitable for large networks

Distance Vector

- *Complexity:*
  - once a link state changes, the change has to be propagated just to the *closest neighbors*; it is further propagated just in cases, when the changed state leads to a change in the current shortest paths tree
- *Speed of convergence:*
  - may converge more slowly than LS
  - problems with routing loops/cycles, *count-to-infinity* problem
- *Robustness:*
  - bad computation is spread through the network ⇒ may lead to a "confusion" of other routers (bad routing tables)
- *Usage:*
  - suitable just for smaller networks

L3 – Network Layer

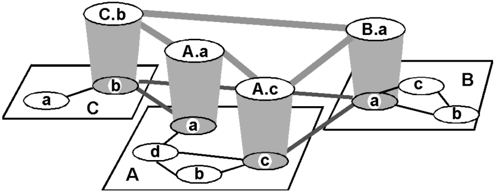
Autonomous Systems

- the goal of Internet's division into *Autonomous Systems* is
  - a reduction of routing overhead
    - simpler routing tables, a reduction of exchanged information, etc.
  - a simplification of the whole network management
    - particular internets are managed by various institutions/organizations
- autonomous systems = domains
  - a 16bit identifier is assigned to every AS/domain
    - *Autonomous System Number (ASN)* – RFC 1930
    - assigned by *ICANN (Internet Corporation For Assigned Names and Numbers)*
  - correspond to administrative domains
    - networks and routers inside a single AS are managed by a single organization/institution
    - e.g., CESNET, PASNET, ...
  - a distinction according to the way an AS is connected to the Internet:
    - *Stub AS*
    - *Multihomed AS*
    - *Transit AS*

L3 – Network Layer

Autonomous Systems – routing

- separated routing because of scalability reasons:
  - *interior routing*
    - routing inside an AS
    - under the full control of AS's administrator(s)
    - the primary goal is the performance
    - so-called *Interior Gateway Protocols (IGP)* (e.g., RIP, OSPF)
  - *exterior routing*
    - routing among ASs
    - the primary goal is the support of defined policies and scalability
    - so-called *Exterior Gateway Protocols (EGP)* (e.g., EGP, BGP-4)
  - a cooperation of interior and exterior routing protocols is necessary



L3 – Network Layer

Autonomous Systems – exterior routing (BGP)

- *Border Gateway Protocol*
  - currently version 4 (BGP-4)
- proposed due to Internet's grow and demands on complex topologies support
  - supports redundant topologies, deals with loops/cycles
- employs so-called *Path Vector* routing
  - not only paths' costs, but the full descriptions of the whole paths are exchanged
- allows a definition of routing rules (policies)
- makes use of the fully reliable TCP protocol
- uses CIDR for paths' aggregation

L3 – Network Layer

IP Multicast

A classical solution of group communication in the network:

- Just a single data copy goes every network link
- A feature of the network (hop-by-hop service, no end-to-end service)
- Non-reliable delivery (best effort, UDP, group address)
- Spread wideness restricted by TTL (Time To Live) field of packets

How to identify a group?

- ⇒ multicast IP address
  - *IPv4:* class D (`224.0.0.0` – `239.255.255.255`)
  - *IPv6:* prefix `ff00::/8`

Two basic approaches to multicast routing:

- *Source Based Tree*
- *Shared Tree (Core Based Tree)*

L3 – Network Layer

IP Multicast – Source Based Tree vs. Core Based Tree

Source Based Tree

- Top-down activity (from the constituent)
- Periodic broadcast
- Cutting the subtrees with no clients
- Wideness restriction – TTL
- Suitable for closely located groups
- Drawbacks: overhead, flooding by broadcasts
- Protocols: DVMRP (RIP), MOSPF (OSPF), PIM–DM

Core Based Tree

- A core is established – ensured by meeting points (MPs)
- A client contacts a MP
- Down-top activity (from the receiver)
- Reduces broadcast $\longrightarrow$ better scalability
- Drawback: a dependence on the core availability
- Protocols: CBT, PIM–SM

L4 – Transport Layer

Introduction

Transport Layer:

- provides its services to the *Application Layer*:
  - obtains data coming from sending application and transforms them into *segments*
  - received segments delivers to the destination application
- in cooperation with the network layer ensures data (segments) delivery between communicating *applications/processes*
  - providing transmission reliability, if required
  - provides them with a logical communication channel
    - an illusion of direct physical interconnection
  - so-called *process-to-process delivery*
- the lowest layer providing so-called *end-to-end* services
  - the headers generated on the sender's side are interpreted "only" on the receiver's side
  - the transport layer data are seen by routers as a payload of transmitted packets

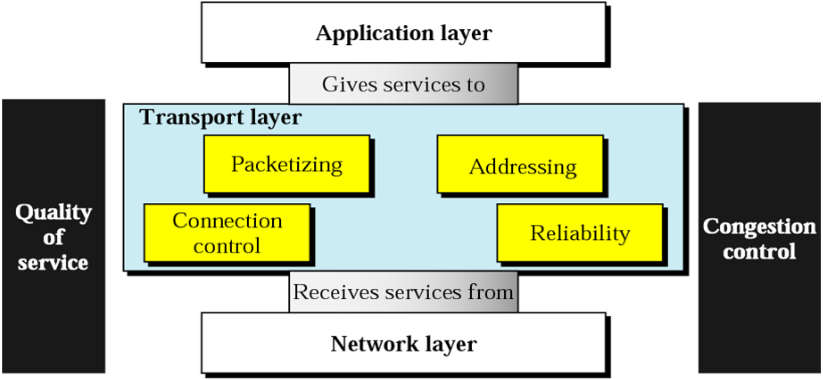


Figure: Position of the Transport Layer.

L4 – Transport Layer

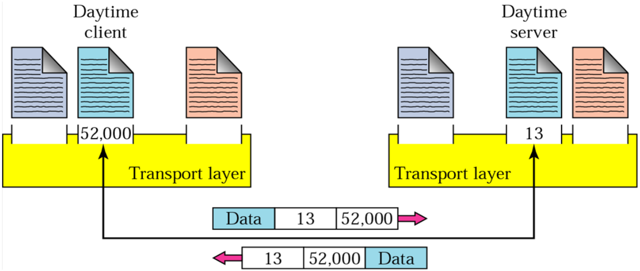
Services

- *Packetizing*
  - the data provided by an application are transformed into packets (having a transport header added)
- *Connection Control*
  - *connection-oriented* and *connectionless* services
- *Addressing*
  - the addresses of transport layer entities (= network applications/services) – so-called *ports*
  - the packets contain source and destination ports (an identification of source and destination application)
    - an application is uniquely identified in the network by the pair *IP_address:port*
- *Connection Reliability*
  - *Flow Control* and *Error Control*
    - provided on the node-to-node principle by lower layers, L4 provides it on the *end-to-end* principle
  - ensures a reliability over *best-effort* service (IP)
- *Congestion Control and Quality of Service (QoS) ensurance*

L4 – Transport Layer

Addressing – ports

- addresses on L4 – *port numbers (ports)*
  - ≈ addresses of services
  - identify a sending application on the sender node (identified by its IP address)
  - identify a receiving application on the receiver node (identified by its IP address)
- ports are identified by *16-bit number*
  - range 0 – 65535



L4 – Transport Layer

Connection-oriented vs. Connection-less Services

Connection-oriented services

- prior to the transmission, a connection is established (and maintained during the whole transmission)
- packets are numbered
  - their delivery/undelivery is explicitly acknowledged

Connection-less services

- packets are sent to the destination application without any connection being established
- packets are not numbered ($\Rightarrow$ they aren't acknowledged)
  - might be lost, delayed, delivered out-of-order, etc.

L4 – Transport Layer

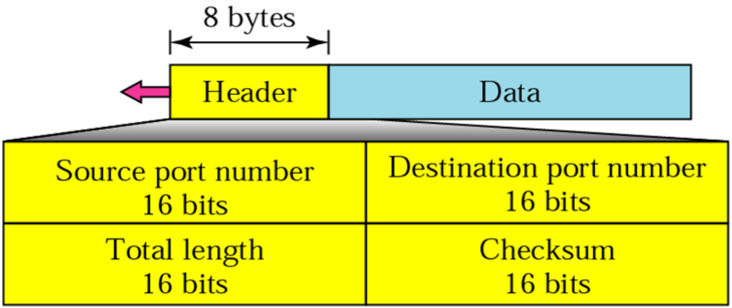
User Datagram Protocol (UDP)

User Datagram Protocol (UDP)

- the simplest transport protocol providing a **connection-less** and **unreliable** service
  - provides *best-effort* service
  - enriches the IP layer services just by process-to-process communication and simple error control
  - if a reliability has to be ensured, it must be provided by the application
- *main features:* simplicity, minimal overhead
  - no connection establishment/maintenance necessity (brings a delay in the beginning of the transmission)
  - no necessity to maintain state information by the communicating nodes
  - small/simple header
- selected applications:
  - processes requiring just a simple "request – reply" communication (e.g., the DNS (Domain Name Service))
  - processes/protocols with internal flow and error control (e.g., TFTP (Trivial File Transport Protocol))
  - real-time transfers
  - multicast transfers

L4 – Transport Layer

UDP header



- **source port** – the identification of sending service/application
- **destination port** – the identification of receiving service/application
- **length** – the total length of the UDP packet (header + data)
- **checksum** – the UDP packet checksum (header + data)

L4 – Transport Layer

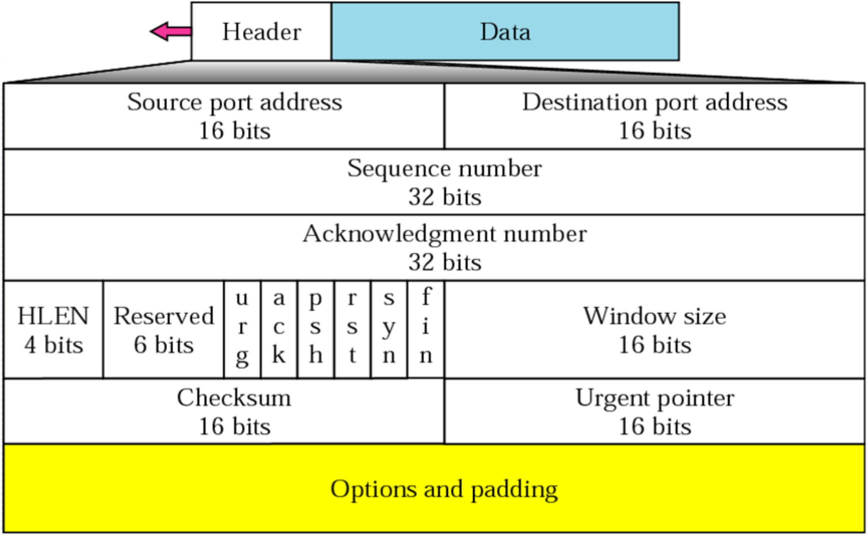
Transmission Control Protocol (TCP)

Transmission Control Protocol (TCP)

- transport protocol providing **connection-oriented** and fully **reliable** service
  - if possible, the data sent by the sender will be received by the receiver – complete and in the right order
  - in comparison with the UDP protocol, the TCP is byte-stream oriented (UDP works with blocks of data)
- prior to a communication, a *connection* has to be established between sender and receiver
  - so-called *three-way handshake* taking place prior to the communication ensures the exchange of all necessary information
  - the connection is distinguishable just on the end nodes (end-to-end service)
    - the routers are not aware about the connections
  - an established connection might be used for fully duplex communication
    - the control data are enclosed in the backward data (so-called *piggybacking*)
  - just **point-to-point** connections are supported
    - the communication among more peers (a-la multicast) is not supported
- multiplexing/demultiplexing and error control same as in the UDP

L4 – Transport Layer

TCP header I.



L4 – Transport Layer

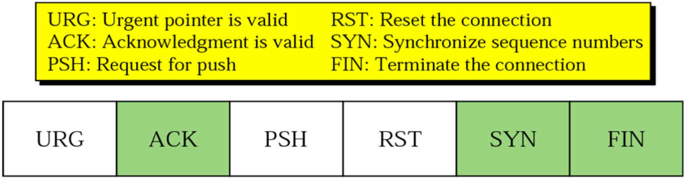
TCP header II.

- **source port** – the identification of sending service/application
- **destination port** – the identification of receiving service/application
- **sequence number** – the number assigned to the first byte of data contained in the segment
- **acknowledgement number**
  - the byte number that the receiver is expecting to receive in the next segment
  - *piggybacking*
- **header length** – the total length of the TCP header (in 4B words)
- **reserved**

L4 – Transport Layer

TCP header III.

- **control** – 6 control bits identifying various control information



- **window size** – the size of the window that the other party must maintain
  - used for the *Flow Control* service (see the next slide)
- **checksum** – the checksum of the TCP segment (header + data)
- **urgent pointer** – used when the segment contains urgent data (out-of-order delivery)
- **options**

L4 – Transport Layer

Flow Control vs. Congestion Control I.

TCP controls the amount of sent data in such a way, that:

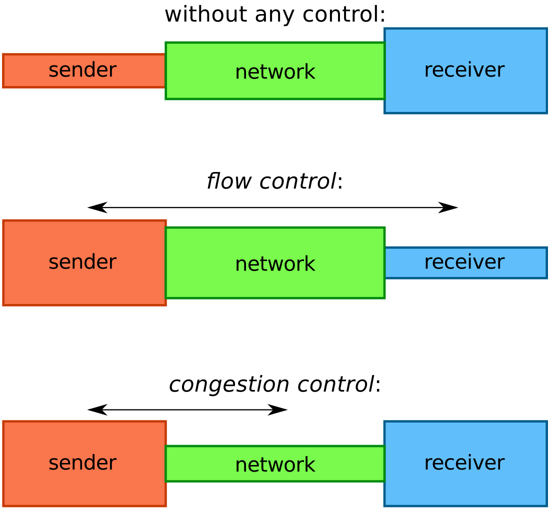
- *protects the receiver from being congested* = **Flow Control**
- *protects the network from being congested* = **Congestion Control**

The amount of data allowed to be sent to the network is defined by:

- the receiver's window size (flow control)
- by the size of so-called *congestion window* (congestion control)
  - maintained on the sender side
- the amount of data allowed to be sent to the network – limited by the **lower value** of both parameters

L4 – Transport Layer

Flow Control vs. Congestion Control II.



L4 – Transport Layer

Résumé

- ensures the communication of particular applications
- providing an optional reliability ensurance
  - UDP protocol for fast, but non-reliable packet transmission
    - just the error control (using checksums) is provided
  - TCP protocol for fully-reliable byte-stream transmission
    - the transmission reliability ensured by repeated sending (ARQ mechanisms)
    - provides a mechanism for flow control (receiver protection from a congestion) – explicit information provided by the receiver
    - provides a mechanism for congestion control (network protection from a congestion) – an estimation of available throughput (AIMD mechanism)
- *further information:*
  - PB156: Computer Networks (doc. Hladká)
  - PV183: Computer Networks Technology (dr. Pelikán)

L7 – Application Layer

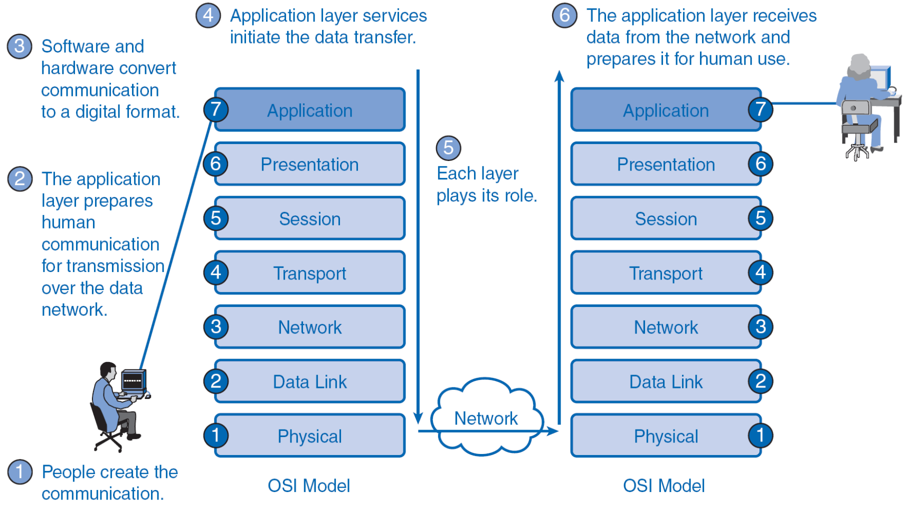
Introduction I.

Application Layer:

- provides services to *users*:
  - application programs specific for a particular purpose
    - e.g., electronic mail, WWW, DNS, etc. etc.
  - applications = the main reason for computer networks existence
- comprises *network applications/programs* and *application protocols*
  - application protocols (HTTP, SMTP, etc.) are **parts of** network applications (web, email)
    - they are not applications on their own
    - the protocols define a form of communication between communicating applications
  - application protocols define:
    - types of messages, which the applications exchange (*request/response*)
    - messages' syntax
    - messages' semantics (a semantics of particular fields)
    - rules, when and how the messages are exchanged

L7 – Application Layer

Introduction II.



L7 – Application Layer

Basic Application Classification/Distinction

According to employed communication model:

- Client-Server model
  - Thin vs. Fat clients
- Peer-to-peer model

According to the way of accessing the information:

- pull model – the data transfer is initiated by a client
- push model – the data transfer is initiated by a server

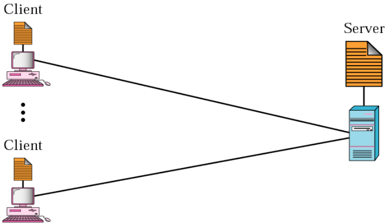
According to the demands on the computer network:

- applications with low demands on the computer network
- applications with high demands on the computer network

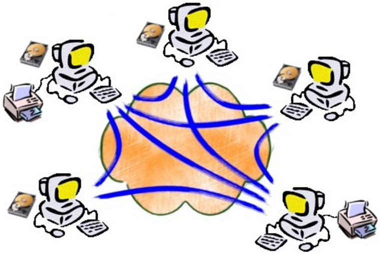
L7 – Application Layer

Client-Server vs. Peer-to-peer

Client-Server



Peer-to-peer



L7 – Application Layer

Résumé

- provides services to users
  - acts as an interface between users and computer network
- the applications can be distincted according to various criteria
  - client/server vs. peer-to-peer, pull vs. push model, demands on the computer network, etc.
- examples of Internet's fundamental applications and application protocols:
  - name service (DNS)
  - World-Wide-Web (HTTP)
  - electronic email (SMTP)
  - file transfer (FTP)
  - multimedia transmissions (RTP/RTCP)
- *further information:*
  - PB156: Computer Networks (doc. Hladká)
  - PV160: Net-centric computing II. (prof. Matyska)
  - PV188: Principles of Multimedia Processing and Transport (doc. Hladká, dr. Liška, Ing. Šiler)