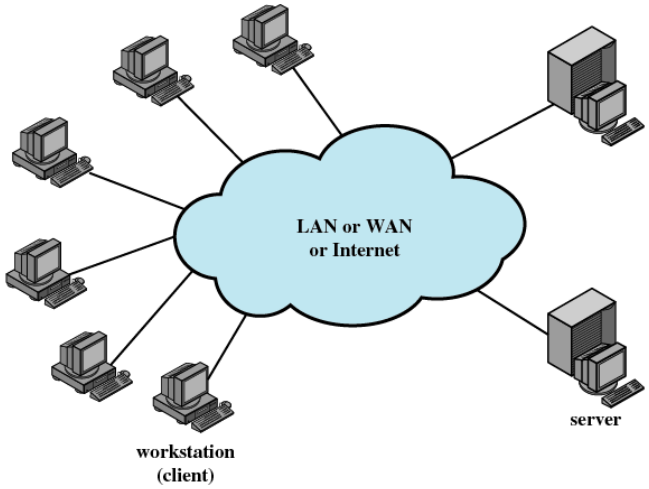
Operating Systems & Computer Networks

Networked Computer & Internet



Content (2)

Freie Universität Berlin

8. Networked Computer & Internet
- Sockets
- Internet
- Layers
- Protocols

9. Host-to-Network I
- Physical Layer
- Media
- Signals
- Modems

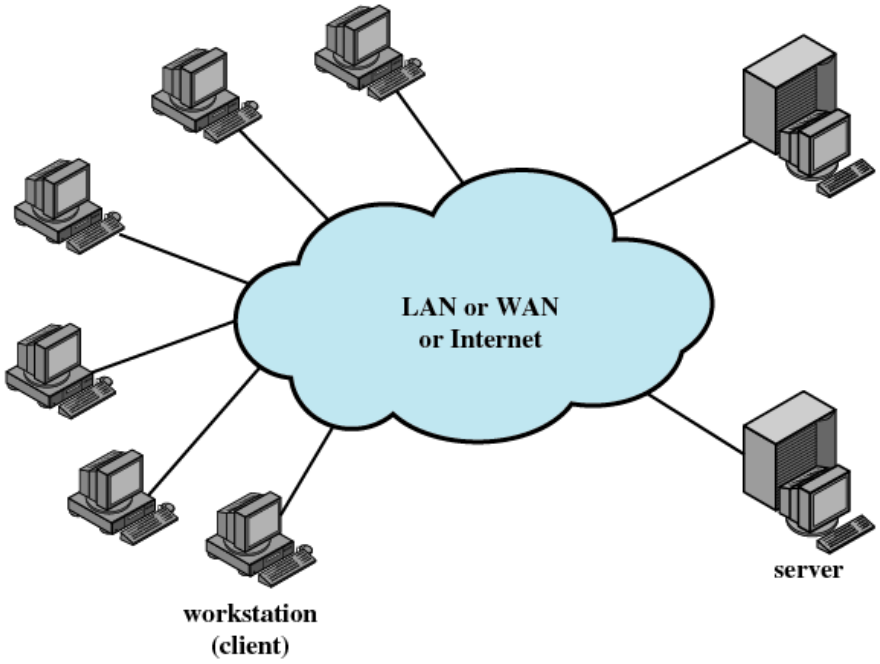
10. Host-to-Network II
- Data Link Layer
- Framing, Flow Control
- Error Detection / Correction
- Point-to-Point Protocol

11. Host-to-Network III
- Topologies
- Medium Access
- Local Area Networks
  - Ethernet, WLAN

12. Internetworking
- Switches, Routers
- Routing
- Internet Protocol
- Addressing

13. Transport Layer
- Protocol Mechanisms
- TCP, UDP
- Addressing, Ports

Motivation – Networked Computers



Questions:

- How can a user/process communicate over the network?
- How can (possibly distant) computers exchange data?
- How does a computer now which other computer it should be talking to?

Motivation – Networked Computers

www.inf.fu-berlin.de

160.45.117.200

Socket

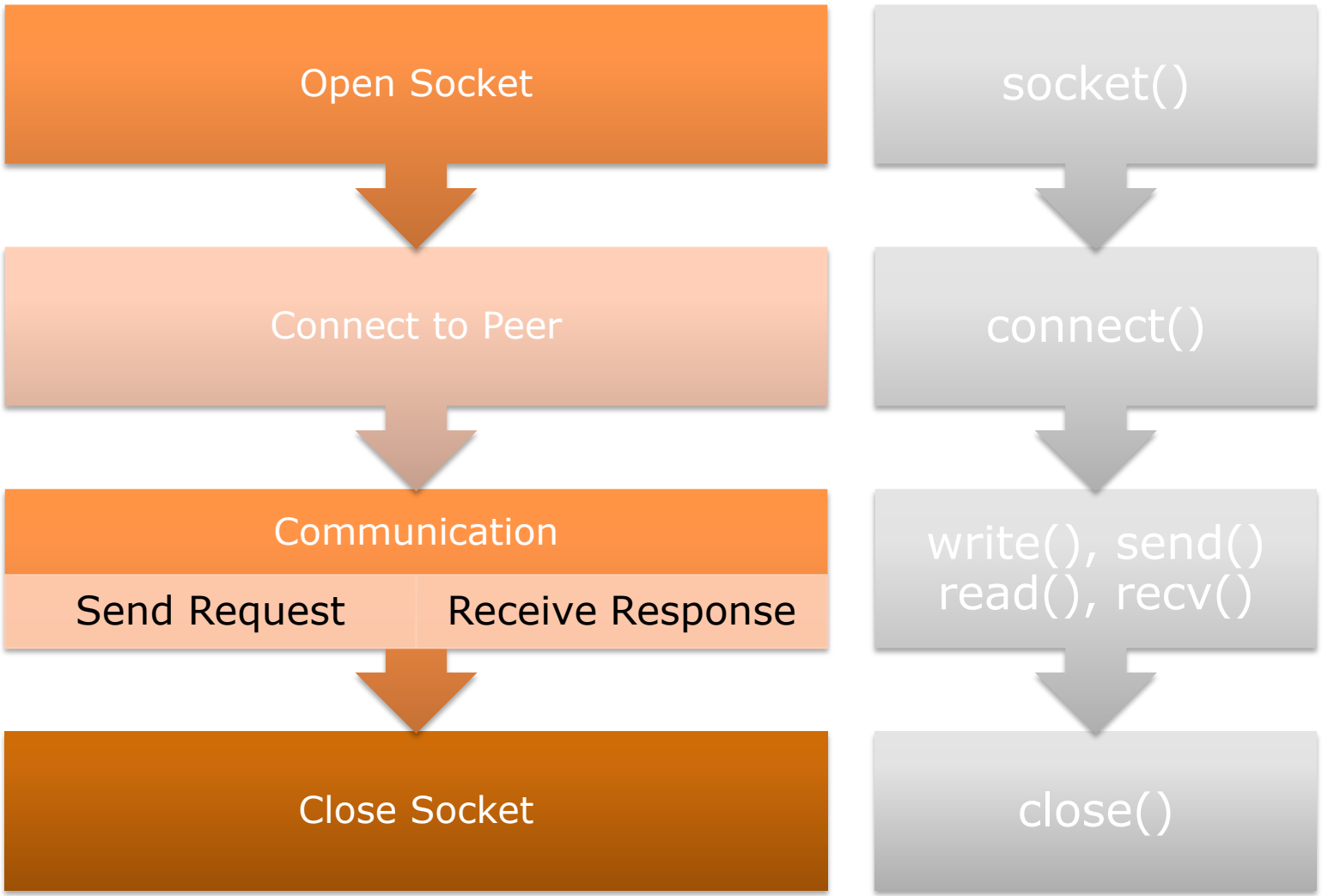
- Enable communication between a client and server
- Concatenation of a Port and an IP address form a socket, 160.45.117.200:80 (http://www.inf.fu-berlin.de)

Types of Sockets (classical Internet)

- Stream sockets
  - Use Transmission Control Protocol (TCP)
  - Reliable data transfer
- Datagram sockets
  - Use User Datagram Protocol (UDP)
  - Delivery is not guaranteed

➢ Processes may open sockets to transparently communicate with processes on remote computers

OS Support for Networking

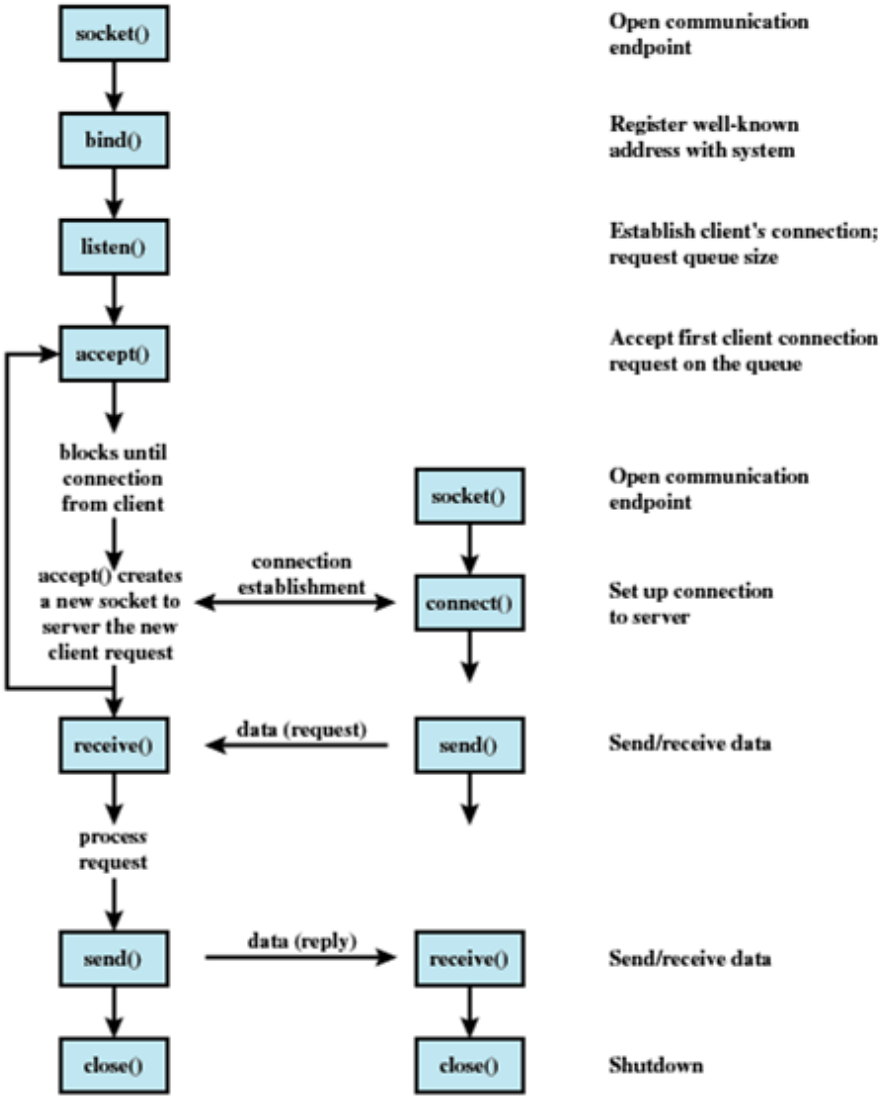


Socket Creation and Operation

- System call

```
int socket(int domain, int type, int protocol)
```

- Parameters
  - **`domain`** Protocol family
    - e.g. **`PF_INET`** for TCP/IP
  - **`type`**
    - Stream or datagram
  - **`protocol`** (optional)
    - e.g. TCP or UDP (for TCP/IP networking)



Datagram Communication

- Simplest possible service: unreliable datagrams

Sender

```
1. int s = socket(…);
2. sendto(s,
      buffer,
      datasize,
      0,
      to_addr,
      addr_length);
```

- `to_addr` and `addr_length` specify destination

Receiver

```
1. int s = socket(…);
2. bind(s, local_addr, …);
3. recv(s,
      buffer,
      max_buff_length,
      0);
```

- Will wait until data is available on socket `s` and put the data into `buffer`

Byte Streams over Connection-Oriented Socket

- For reliable byte streams, sockets have to be connected first
- Receiver has to accept connection

Client

1. `int s = socket(…);`
2. `connect(s, destination_addr, addr_length);`
3. `send(s,buffer, datasize, 0);`
4. Arbitrary `recv()/send()`
5. `close (s);`

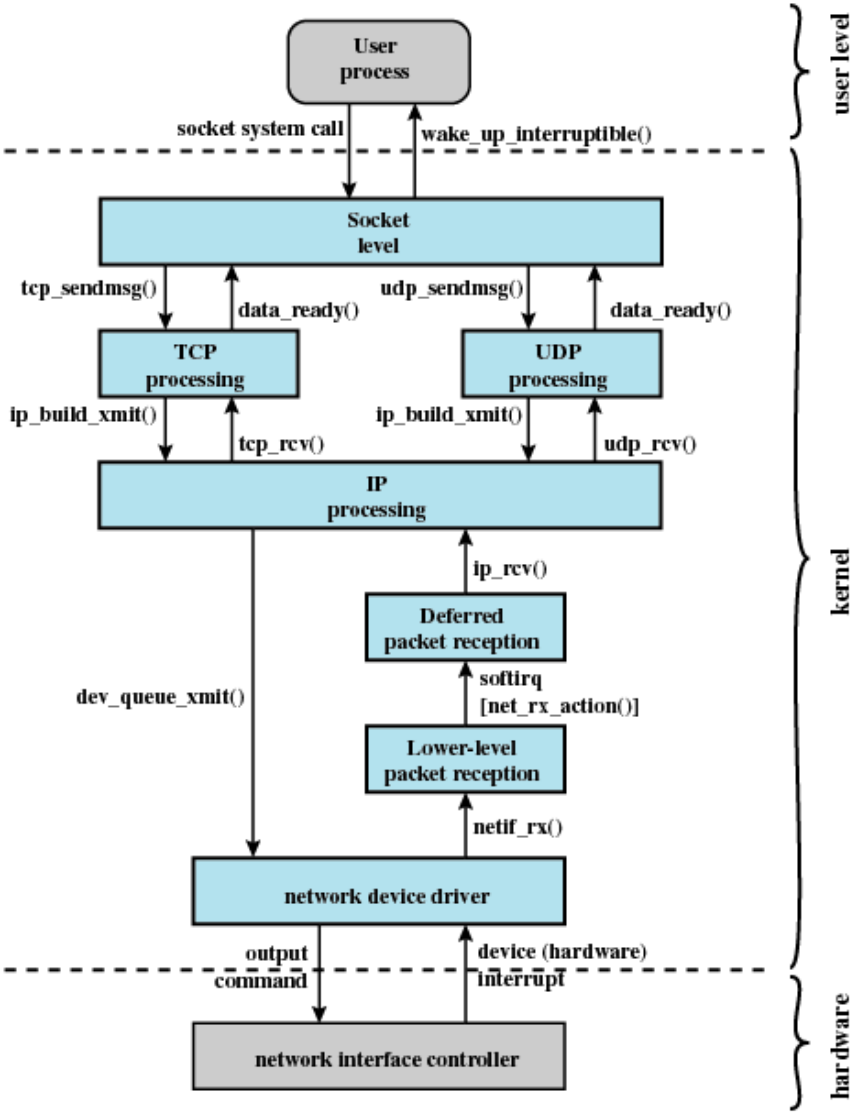
- Connected sockets use a `send` without address information

Server

1. `int s = socket(…);`
2. `bind(s, local_addr, …);`
3. `listen(s, …);`
4. `int newsock = accept(s, *remote_addr, …);`
5. `recv(newsock, buffer, max_buff_length, 0);`
6. Arbitrary `recv()/send()`
7. `close (newsock);`

   `...`
8. `close(s);`

Kernel-level Socket Support



API to user space

TCP / UDP implementation

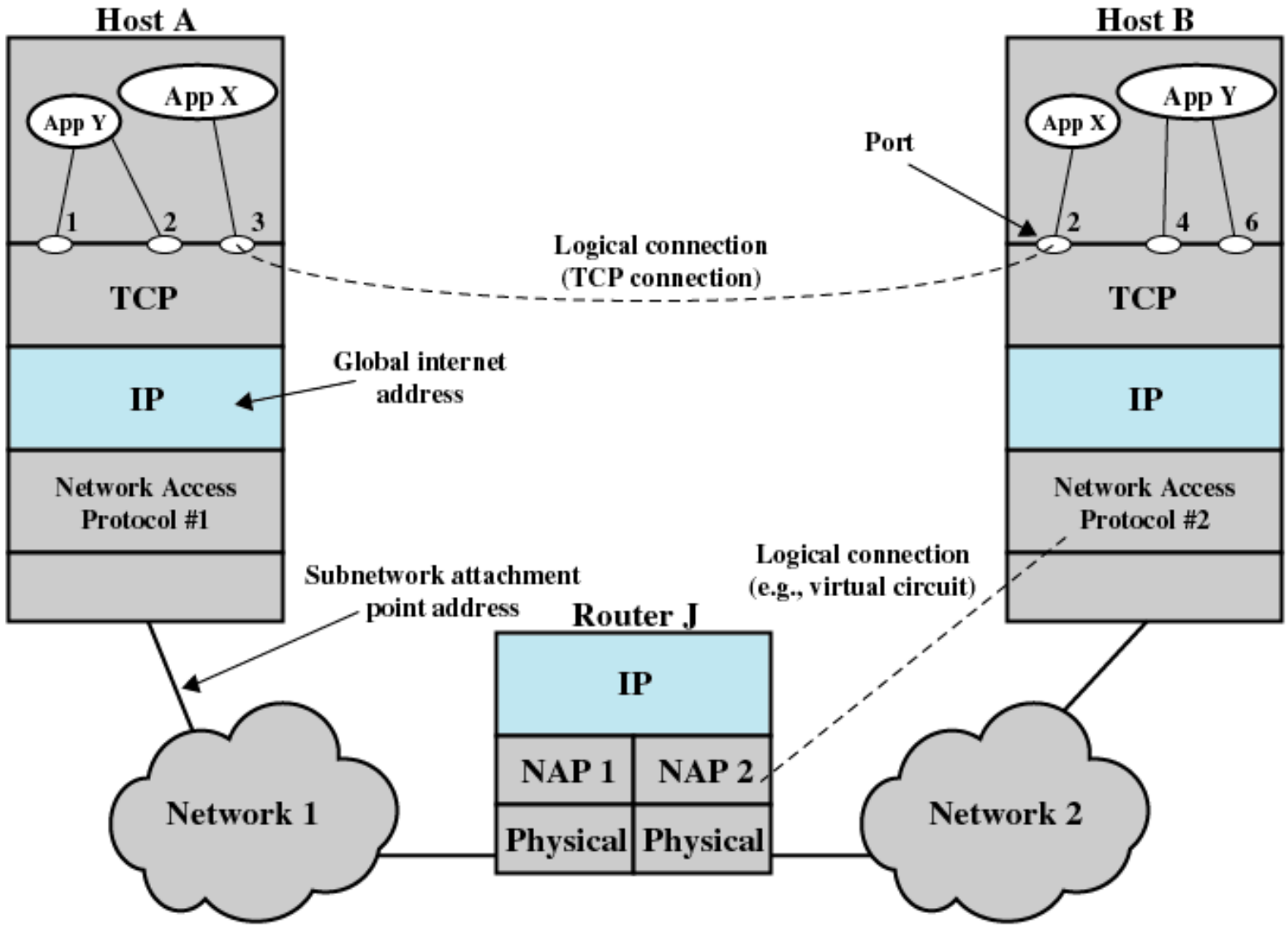
Internet Protocol (IP) implementation

Device driver (includes method to access communication medium)

Layered communication architecture

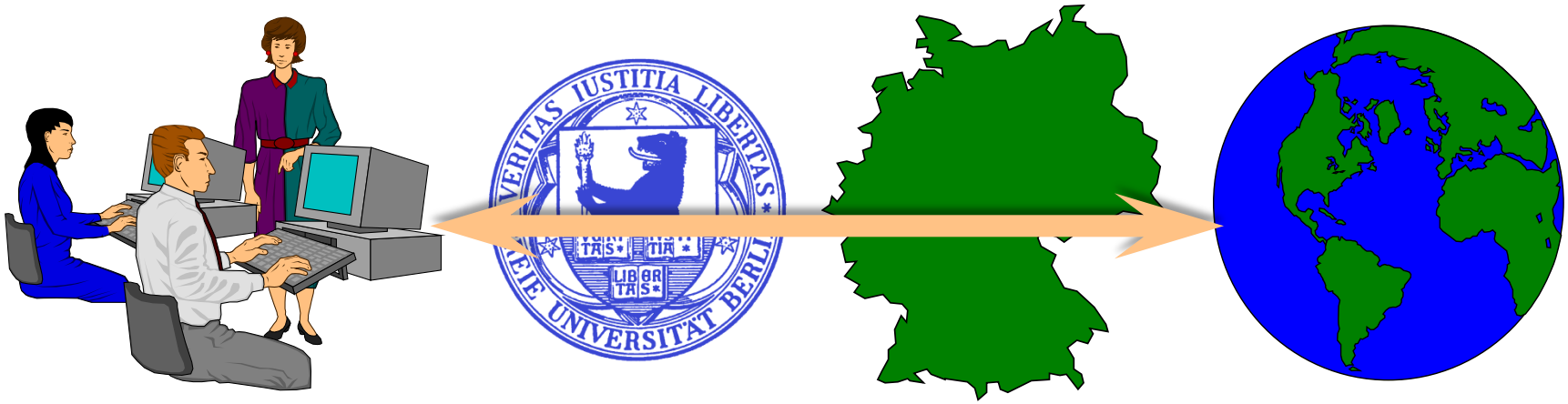
The Internet

Internet / TCP/IP Network Stack



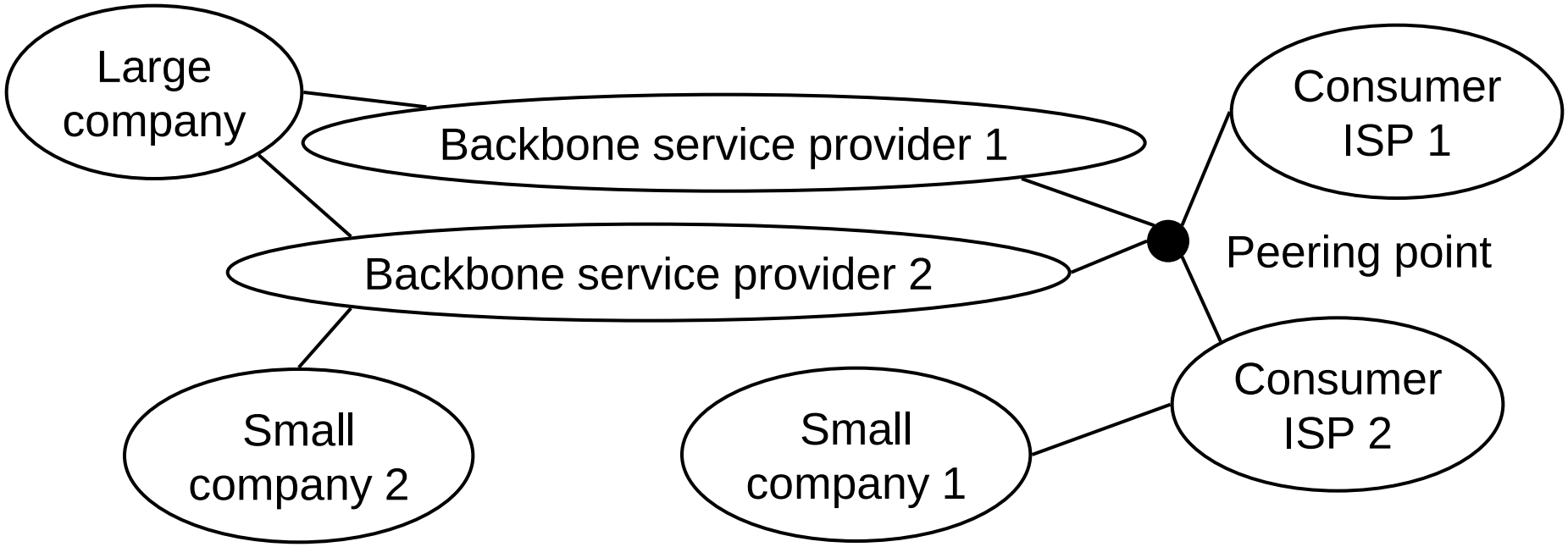
The Internet

- The Internet consists of
  - many computers
    - using same network protocol family TCP/IP
      - IP on top of lower-level protocol (Ethernet, WLAN, Bluetooth, ...)
    - that are (directly or indirectly) connected to each other
    - that offer or use certain services
  - many users that have direct access to the services
  - many networks interconnected via gateways

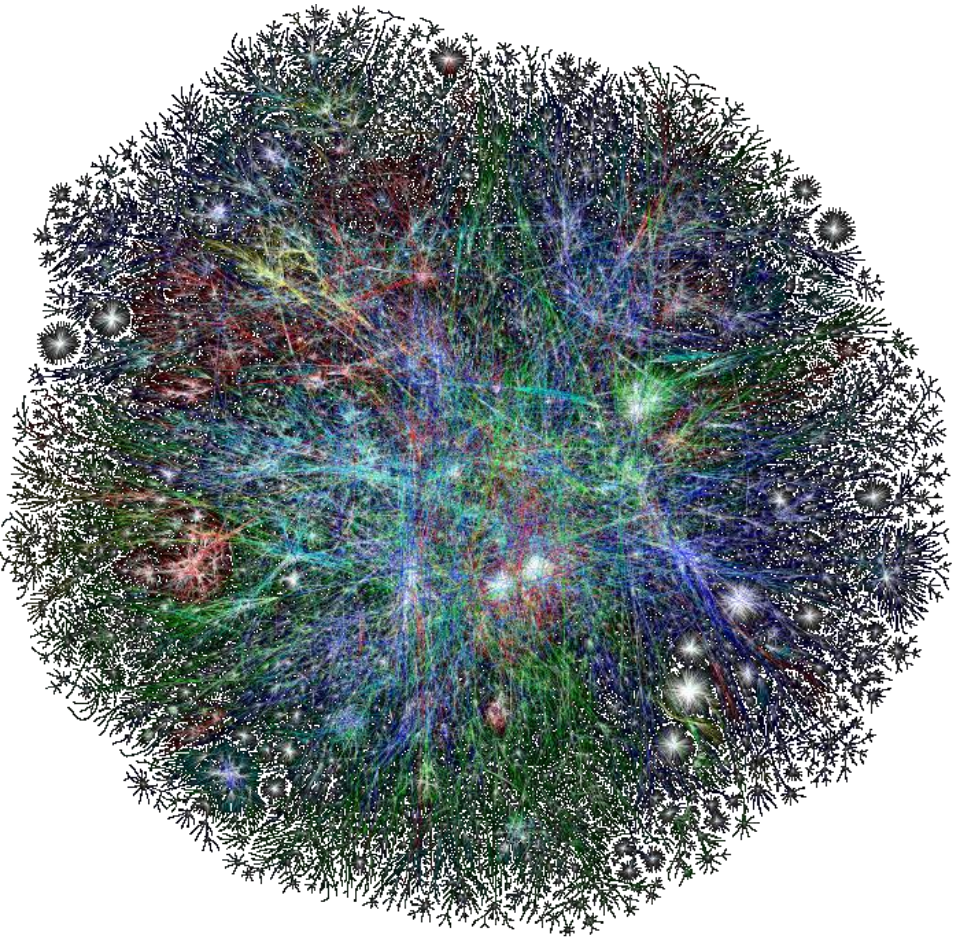


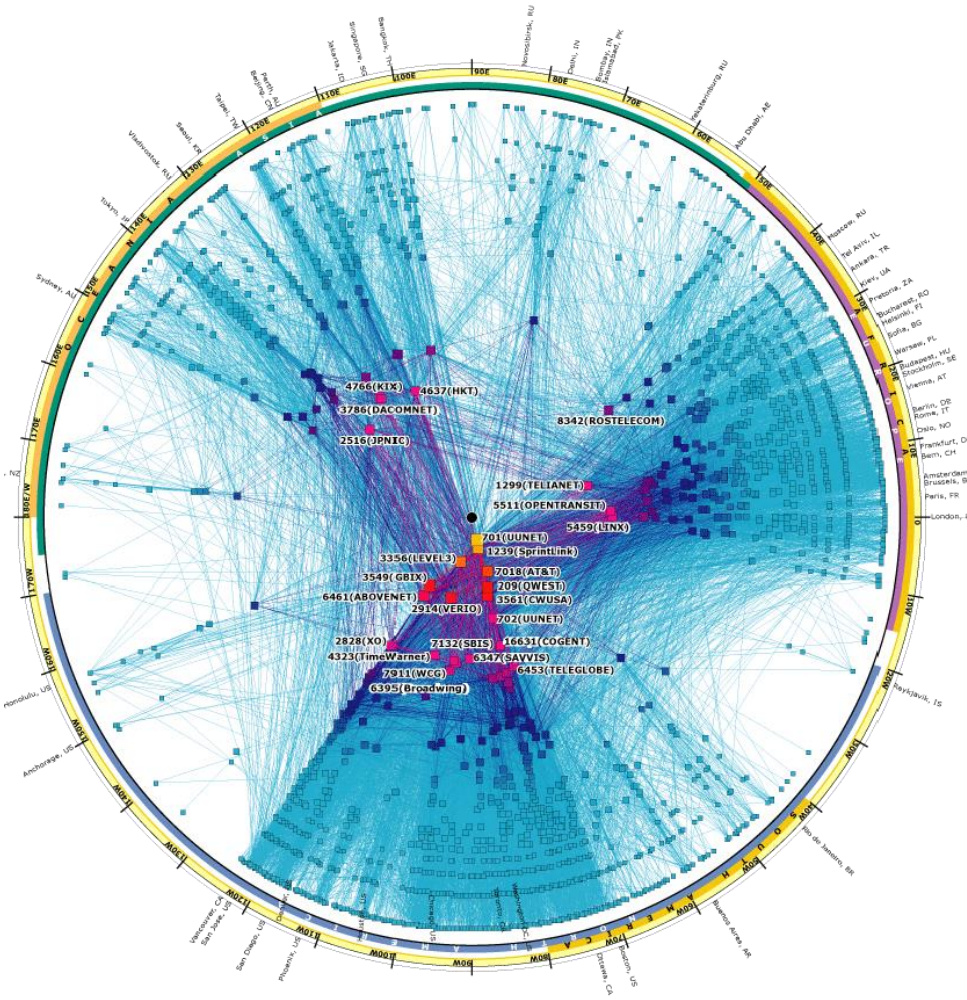
Structure of the Internet (Concept)

- Backbone service providers
- Consumer Internet Service Provider (ISP)
- Peering Points – shortcuts between operators
- Consumers
  - Direct backbone connectivity (companies) or ISP (private)



Structure of the Internet (Reality)





Exemplary Services in the Internet

- World Wide Web (WWW)
  - World-wide interlinked resources
  - Based on “Hypertext Transfer Protocol” (HTTP)
- Electronic mail (email)
  - Exchange of digital multimedia messages
  - Based on “Simple Mail Transfer Protocol” (SMTP)
- File transfer
  - Exchange of files
  - Based on “File Transfer Protocol” (FTP)
- Network management
  - Monitoring and control of networked systems
  - Based on “Simple Network Management Protocol” (SNMP)
- P2P, VoIP, IPTV, ...

- Many company-specific services: Skype, Gaming, ...

Internet Design Principles

- Minimalism and autonomy
  - Independent operation of the network, no internal changes necessary if connected to other networks
- "Best-Effort" services
  - Network tries as best as possible to transmit data end-to-end
  - Reliable communication is feasible through retransmission
    - Today several extensions towards quality-of-service (QoS) support exist
- Stateless intermediate systems
  - No intermediate system (routers) should keep state related to any end-to-end communication
    - Big difference to classical telephone networks (circuit vs. packet switched)
    - Alternatives necessary for quality-of-service support
- Decentralized control
  - No global, centralized control of all interconnected networks

Some (Historical) IP Design Principles

RFC 1958, based on papers from mid-80s

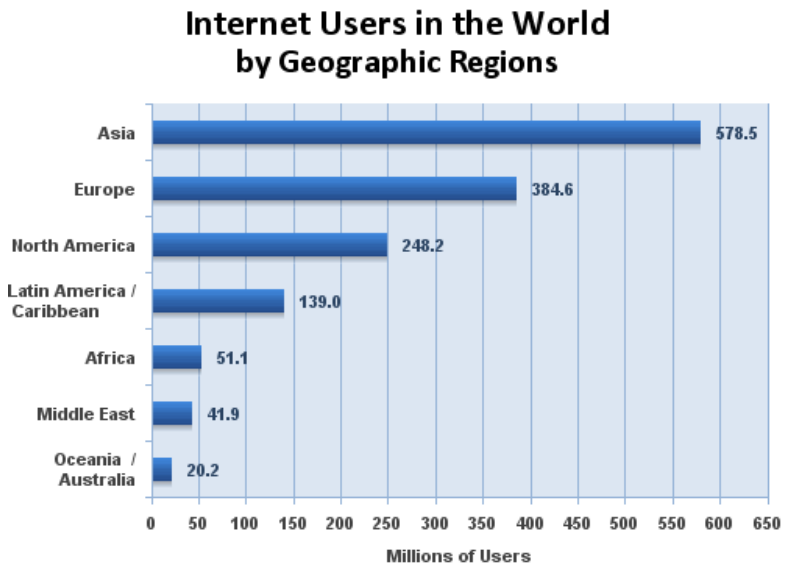
- Make sure it works – before writing the standard
- Keep it simple
- Make clear choices
- Exploit modularity
- Expect heterogeneity
- Avoid *static* options and parameters
- Look for a good design; it need not be perfect
  - 80-20 rule: 80% of effects comes from 20% of causes
- Be strict when sending and tolerant when receiving
- Think about scalability (with regard to nodes and traffic)
- Consider performance and cost

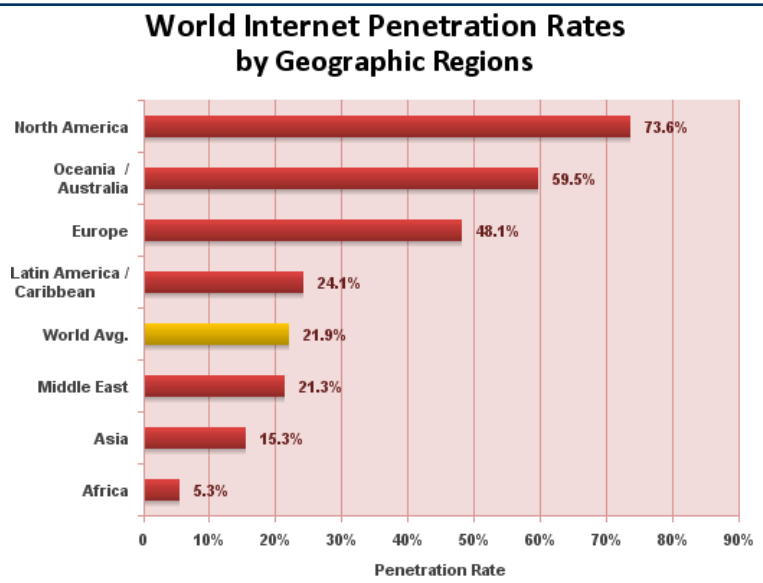
➢ Looking back, some choices are not optimal anymore.

Development of the Internet

- 1962 DoD (Department of Defense): "Defense depends on communication."
- 1967 ARPA (Advanced Research Project Agency) of the DoD: Project reliable packet network at Stanford Research Inst.
- 1969 First "Internet" (4 hosts)
- 1971 Start of ARPAnet, the first Internet backbone
- 1974 New protocol suite: TCP/IP (Transmission Control Protocol/Internet Protocol)
- 1980 Integration of TCP/IP protocols into UNIX (BSD)
- 1988 IP connection to the Internet from Germany via EUnet-IRB Dortmund and XLink (eXtended Lokales Informatik Netz Karlsruhe)
- 1991 EBONE: European backbone
- 1995 Internet becomes visible due to WWW
- 1996 University Corporation for Advanced Internet Development - Internet2
- 1999 Second Internet2-Backbone: Abilene
- 1998-2002 Rise and fall of dotcoms
- 2006 VoIP, Web 2.0 hype (and history repeats…)
- 2009 Clouds, more clouds
- 2011 Everything is mobile (> 4.5bn subscribers), apps rule…
- 20xy Internet of Things, IPv6 finally everywhere

Internet Users World-Wide

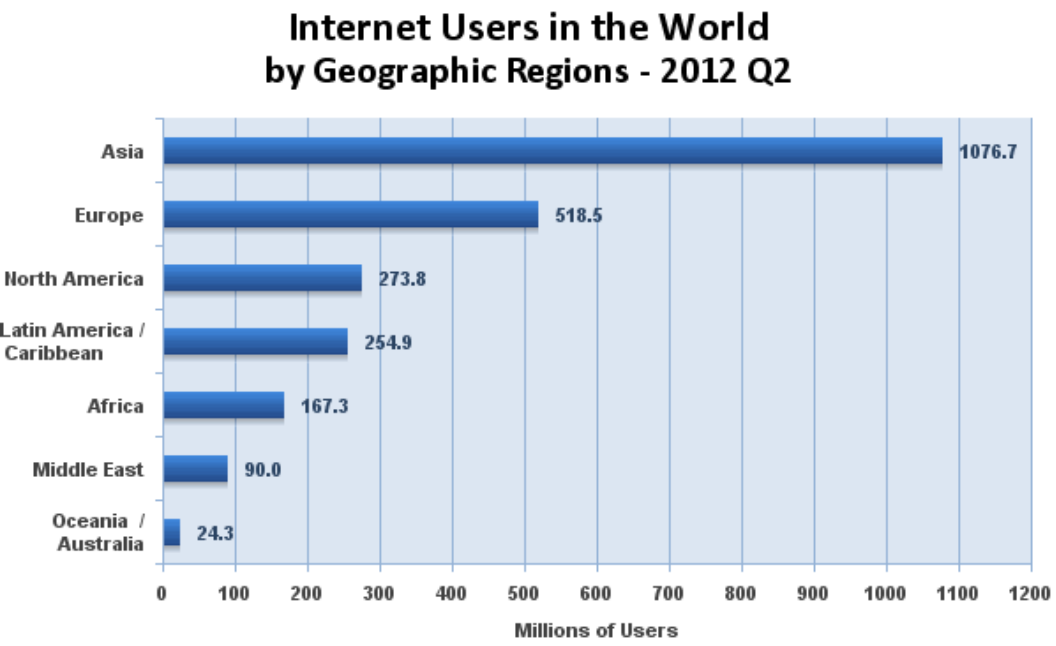




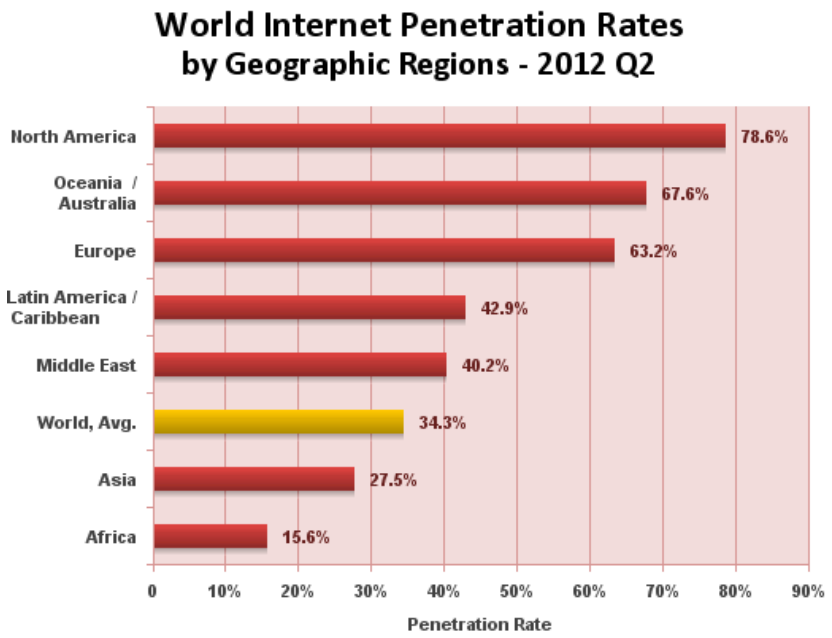
World Regions	Population (2008 Est.)	Internet Users Dec/31, 2000	Internet Usage, Latest Data	% Population (Penetration)	Usage % of World	Usage Growth 2000-2008
Africa	955,206,348	4,514,400	**51,065,630**	5.3 %	3.5 %	1,031.2 %
Asia	3,776,181,949	114,304,000	**578,538,257**	15.3 %	39.5 %	406.1 %
Europe	800,401,065	105,096,093	**384,633,765**	48.1 %	26.3 %	266.0 %
Middle East	197,090,443	3,284,800	**41,939,200**	21.3 %	2.9 %	1,176.8 %
North America	337,167,248	108,096,800	**248,241,969**	73.6 %	17.0 %	129.6 %
Latin America/Caribbean	576,091,673	18,068,919	**139,009,209**	24.1 %	9.5 %	669.3 %
Oceania / Australia	33,981,562	7,620,480	**20,204,331**	59.5 %	1.4 %	165.1 %
WORLD TOTAL	6,676,120,288	360,985,492	**1,463,632,361**	21.9 %	100.0 %	305.5 %

Source: http://www.internetworldstats.com/stats.htm (09/11/06)

Internet Users World-Wide



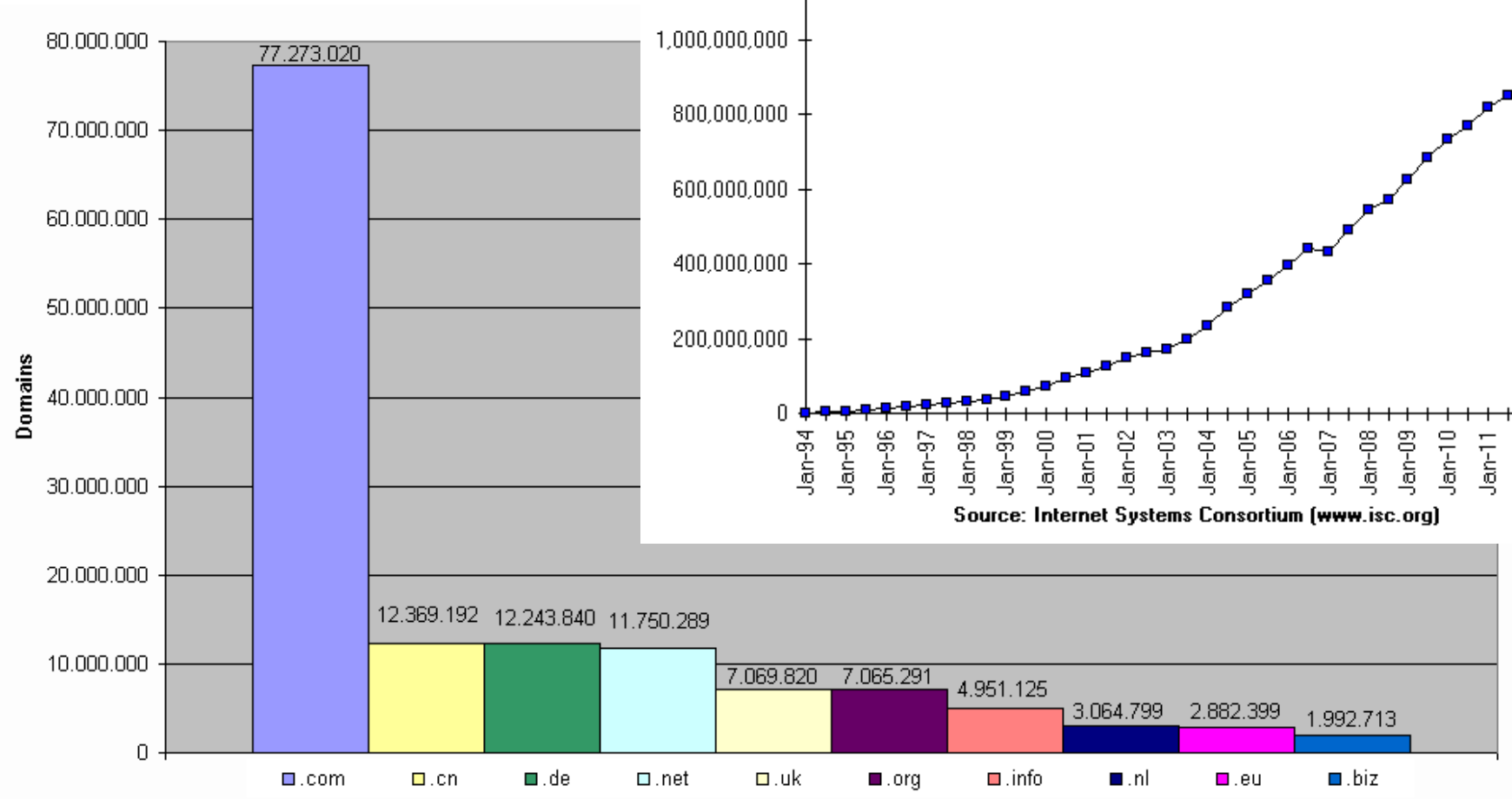
Source: Internet World Stats - www.internetworldstats.com/stats.htm
2,405,518,376 Internet users estimated for June 30, 2012
Copyright © 2012, Miniwatts Marketing Group



Source: Internet World Stats - www.internetworldststs.com/stats.htm
Penetration Rates are based on a world population of 7,017,846,922
and 2,405,518,376 estimated Internet users on June 30, 2012.
Copyright © 2012, Miniwatts Marketing Group

Hosts and Internet Domains

14.5 million .de – domains
818 million hosts world-wide

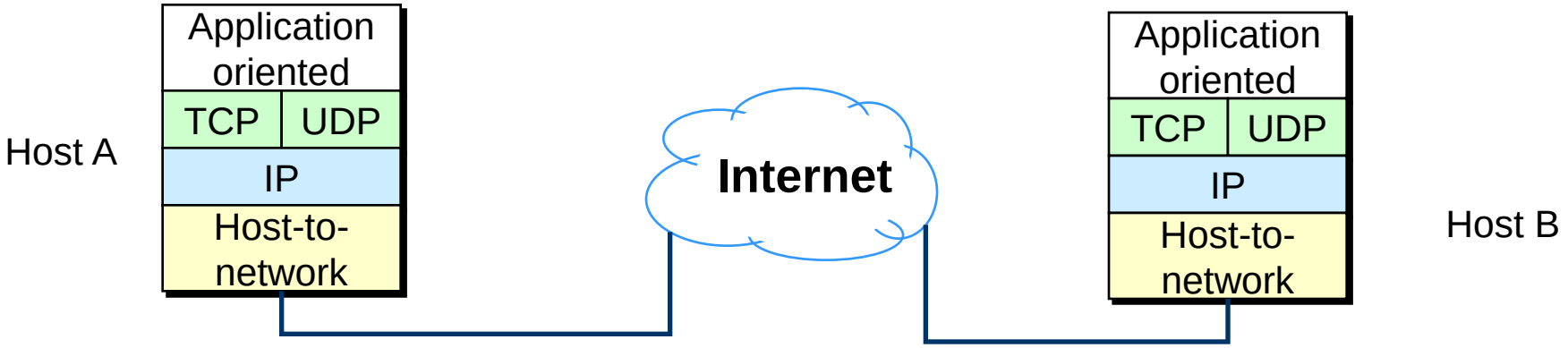


Sources: DENIC (www.denic.de), Internet Systems Consortium, Inc. (http://www.isc.org/)

Protocols

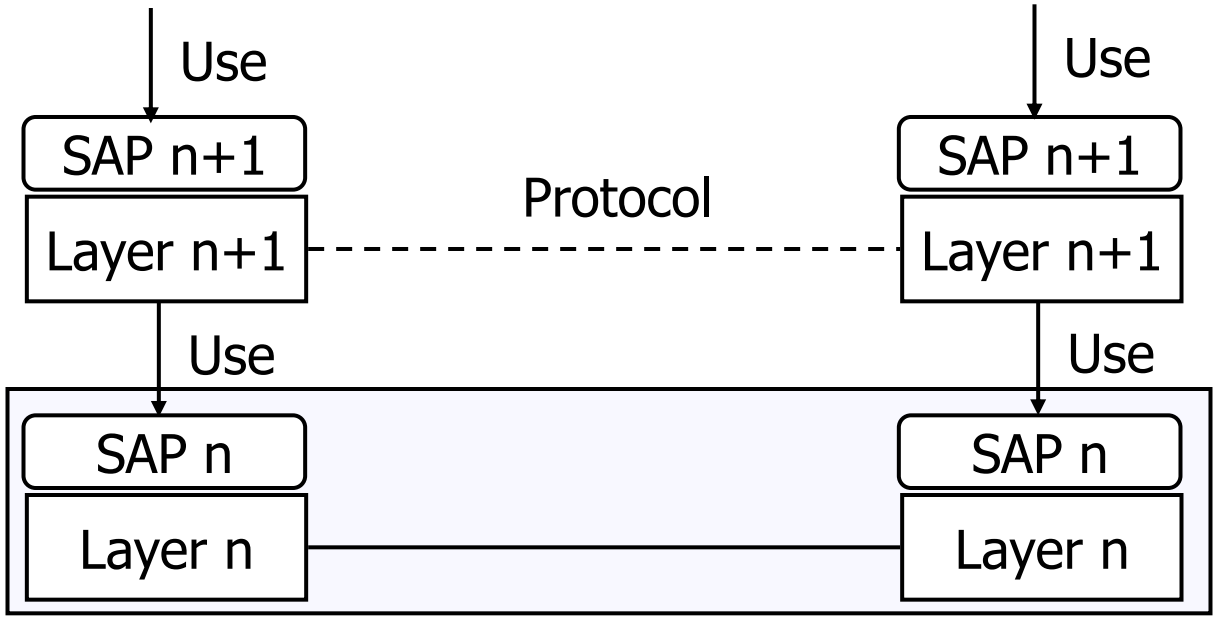
The Internet Protocol Suite

- TCP (Transmission Control Protocol)
  - Reliable, connection oriented transport protocol over unreliable IP (Internet Protocol)
- UDP (User Datagram Protocol)
  - Connectionless transport protocol, offers application interface to IP plus multiplexing
- Examples for application oriented protocols
  - HTTP: HyperText Transfer Protocol
  - FTP: File Transfer Protocol
  - Telnet: Simple terminal protocol



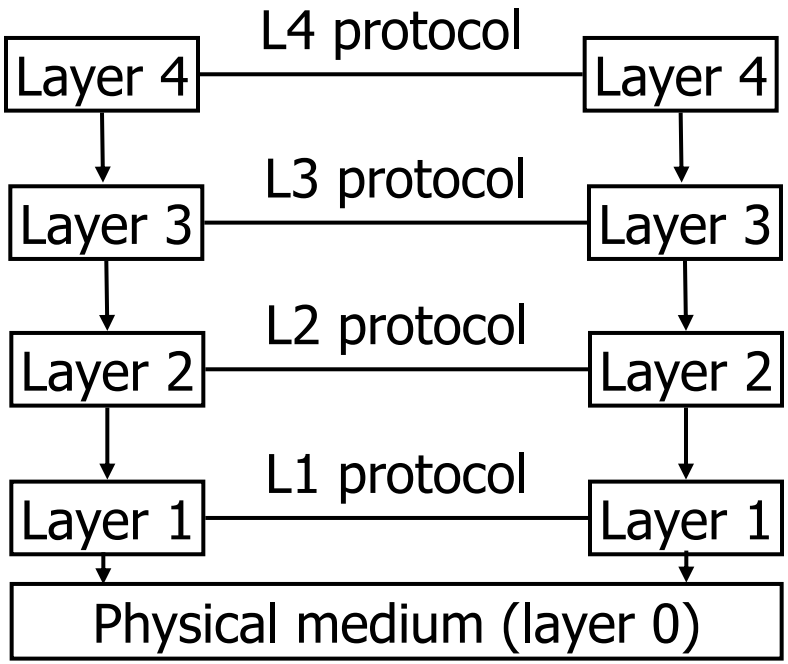
Protocols

- Protocols are a set of rules
  - Describe how two (or more) remote parts of a layer cooperate to *implement the service* of the given layer
    - Behavior, packet formats
  - These remote parts are called *peer protocol entities* or simply *peers*
  - Use the service of underlying layer to exchange data with peer



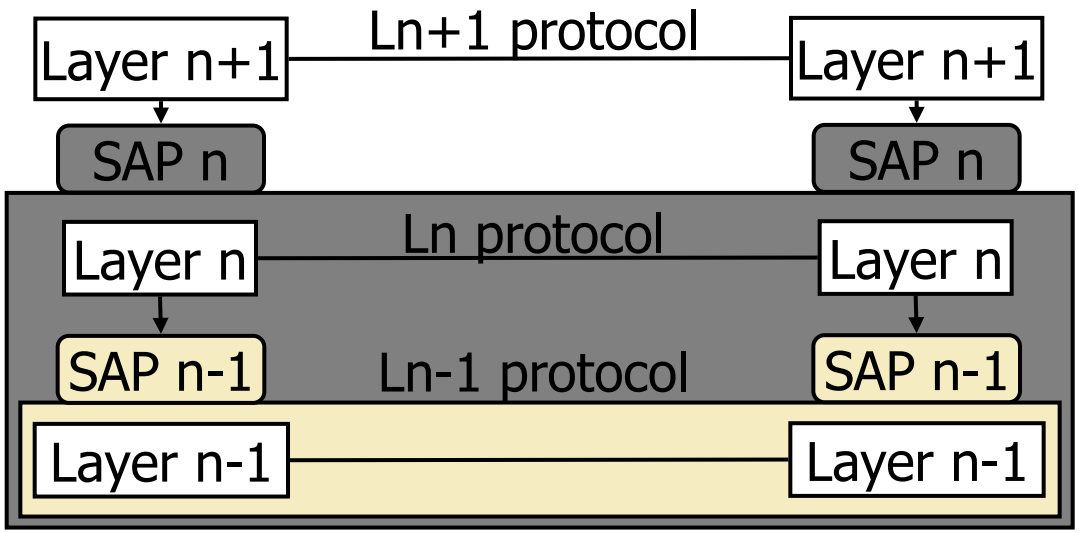
Protocol Stacks

- Typically, several layers and thus several protocols in real system
- Layers/protocols are arranged as *(protocol) stack*
  - One atop the other, *only* using services from directly beneath (so-called *strict layering*)

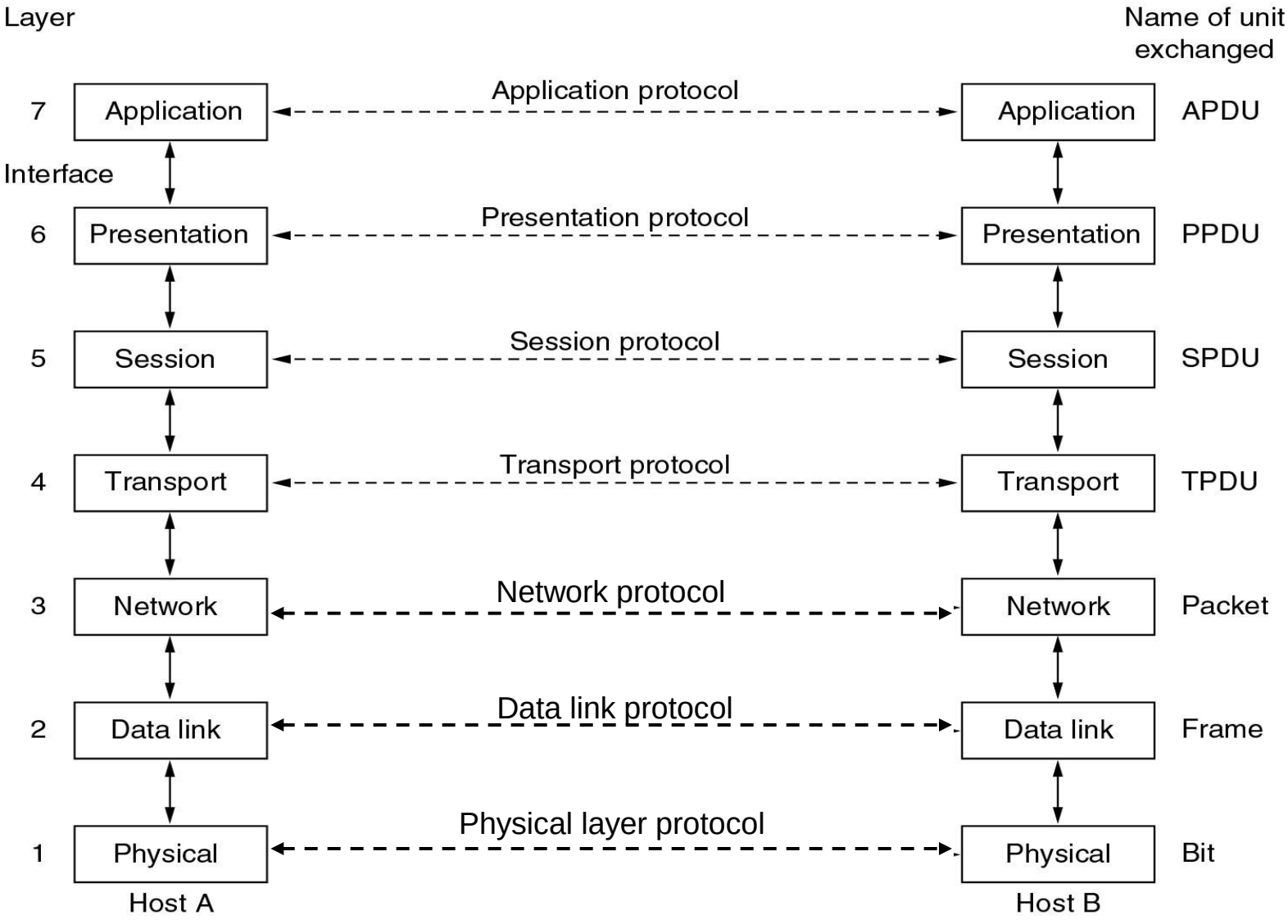


Layers Do Not Care About Distributed Lower Layers

- A given layer n+1 does not care about the fact that its lower layer is actually distributed …
  - Layer n+1 imagines layer n as something that "just works", has service access points where they are necessary
  - In reality, layer n of course is distributed in turn, relying on yet lower layers
  - At the end, the physical medium (layer 0) is transporting signals (as physical representation of data)



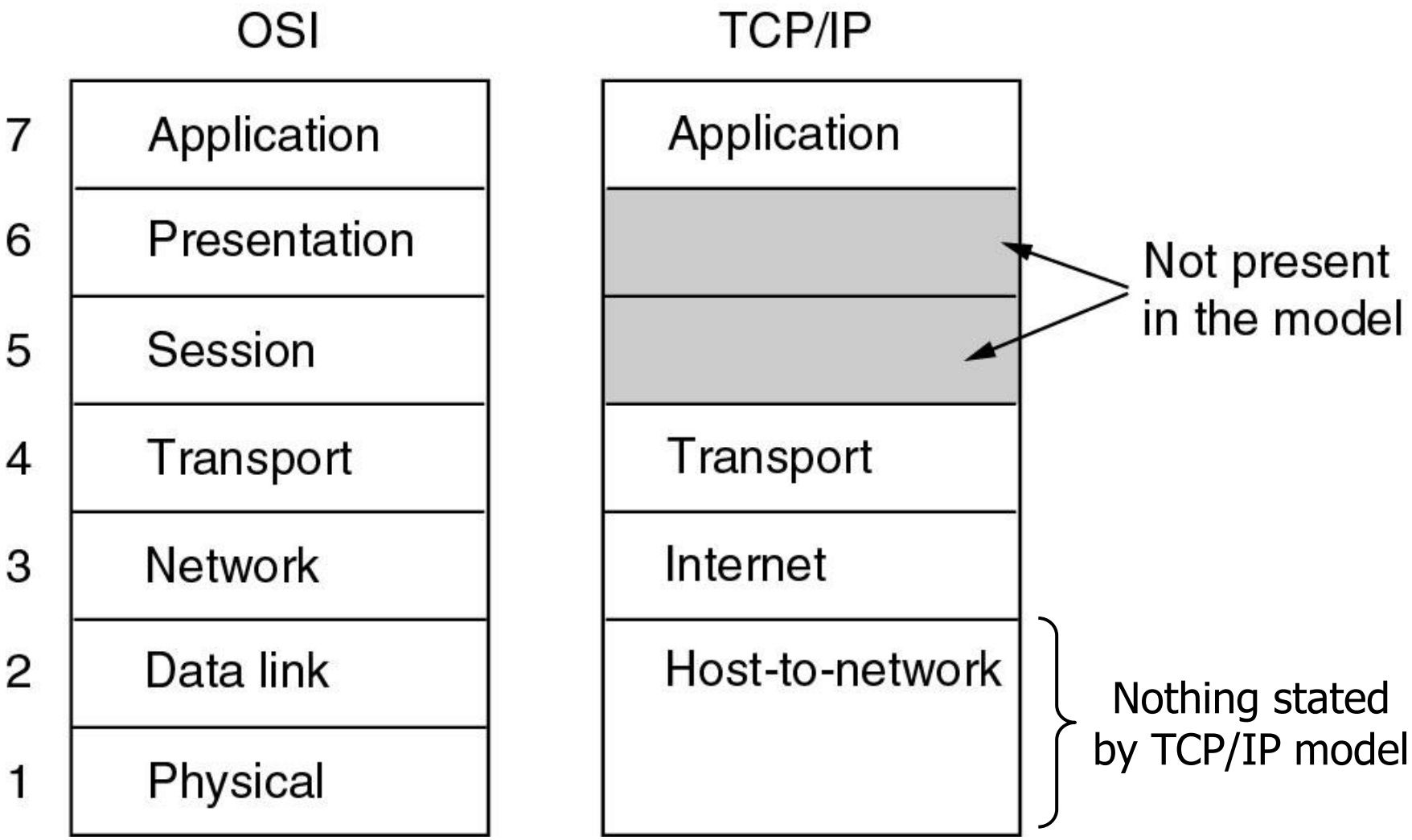
ISO/OSI 7-layer Reference Model



Seven Layers (in brief)

1. **Physical layer**: Transmit raw bits over a physical medium
2. **Data Link layer**: Provide a (more or less) error-free transmission service for data frames over a shared medium
3. **Network layer**: Solve the forwarding and routing problem for a network
4. **Transport layer**: Provide (possibly reliable, in order) end-to-end communication, overload protection, fragmentation
5. **Session layer**: Group communication into *sessions* which can be synchronized, checkpointed, …
6. **Presentation layer**: Ensure that syntax and semantic of data is uniform between all types of terminals
7. **Application layer**: Actual application, e.g., protocols to transport web pages

TCP/IP Protocol Stack



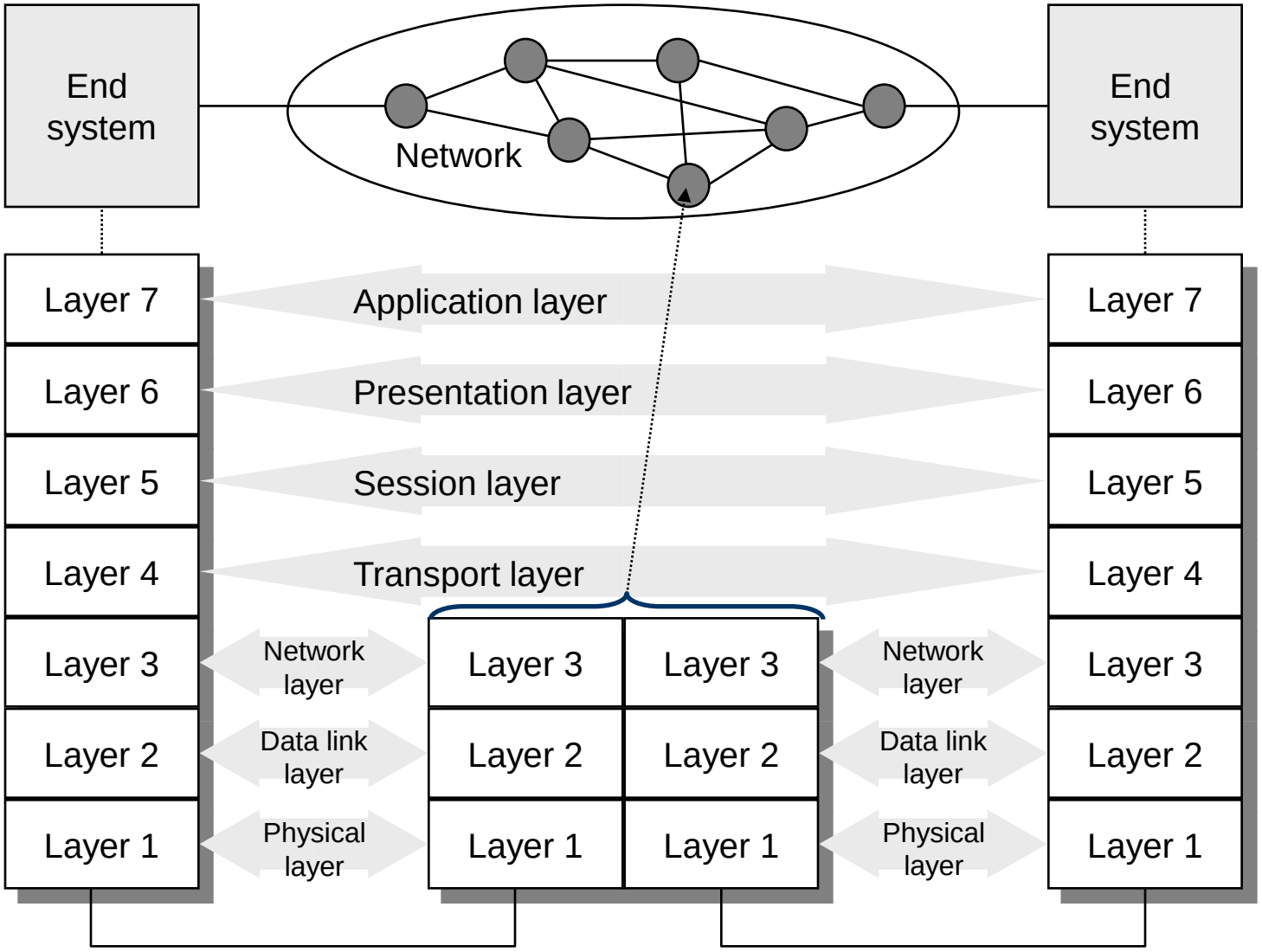
ISO/OSI versus TCP/IP

- ISO/OSI: Very useful model, non-existing protocols
- TCP/IP: Non-existing model, very useful protocols

➢ Use simplified ISO/OSI model, but treat TCP/IP protocol stack in context of this model
  - With suitable add-ons especially for the lower layers

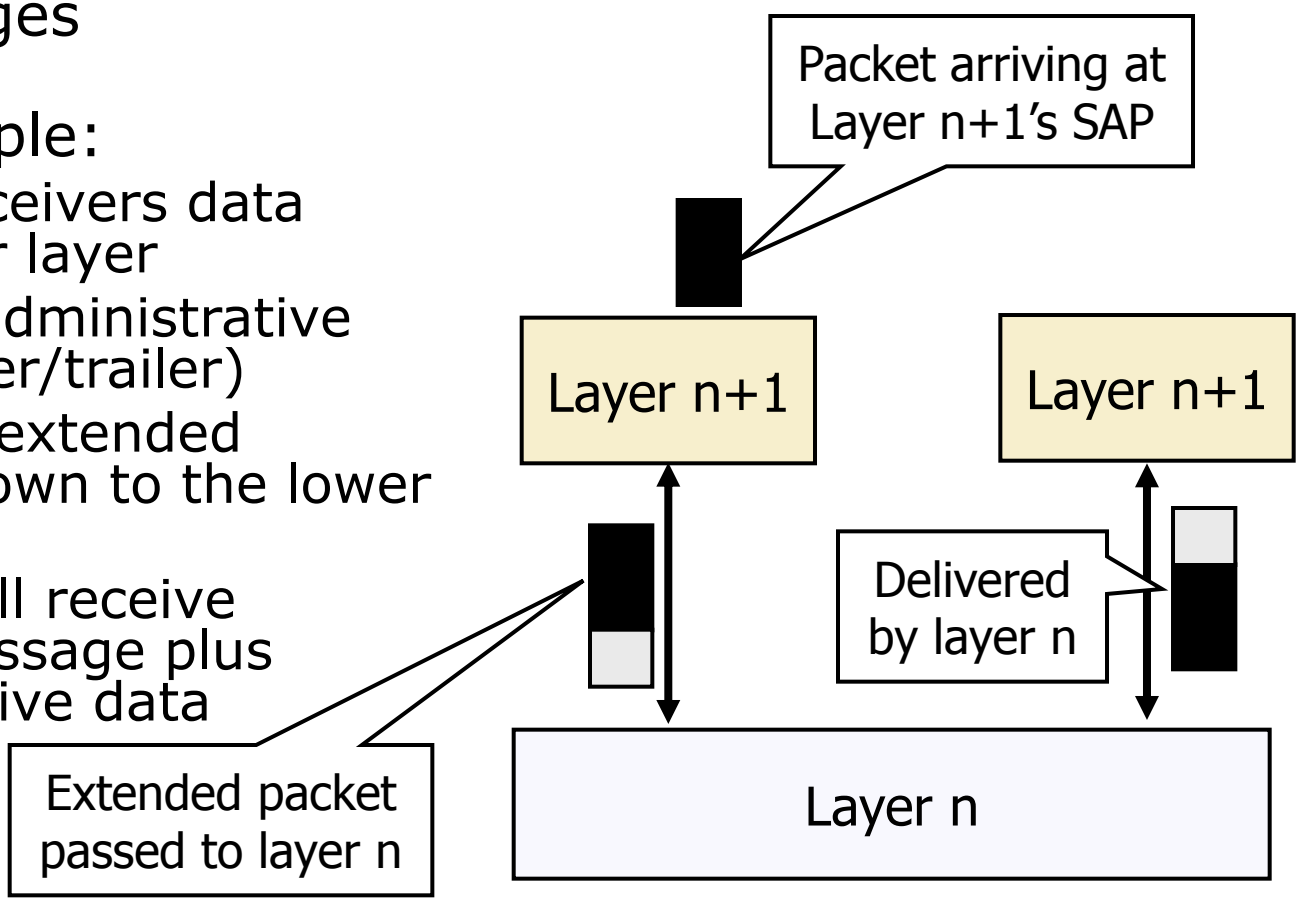
5	Application layer
4	Transport layer
3	Network layer
2	Data link layer
1	Physical layer

7 Layers with Intermediate System

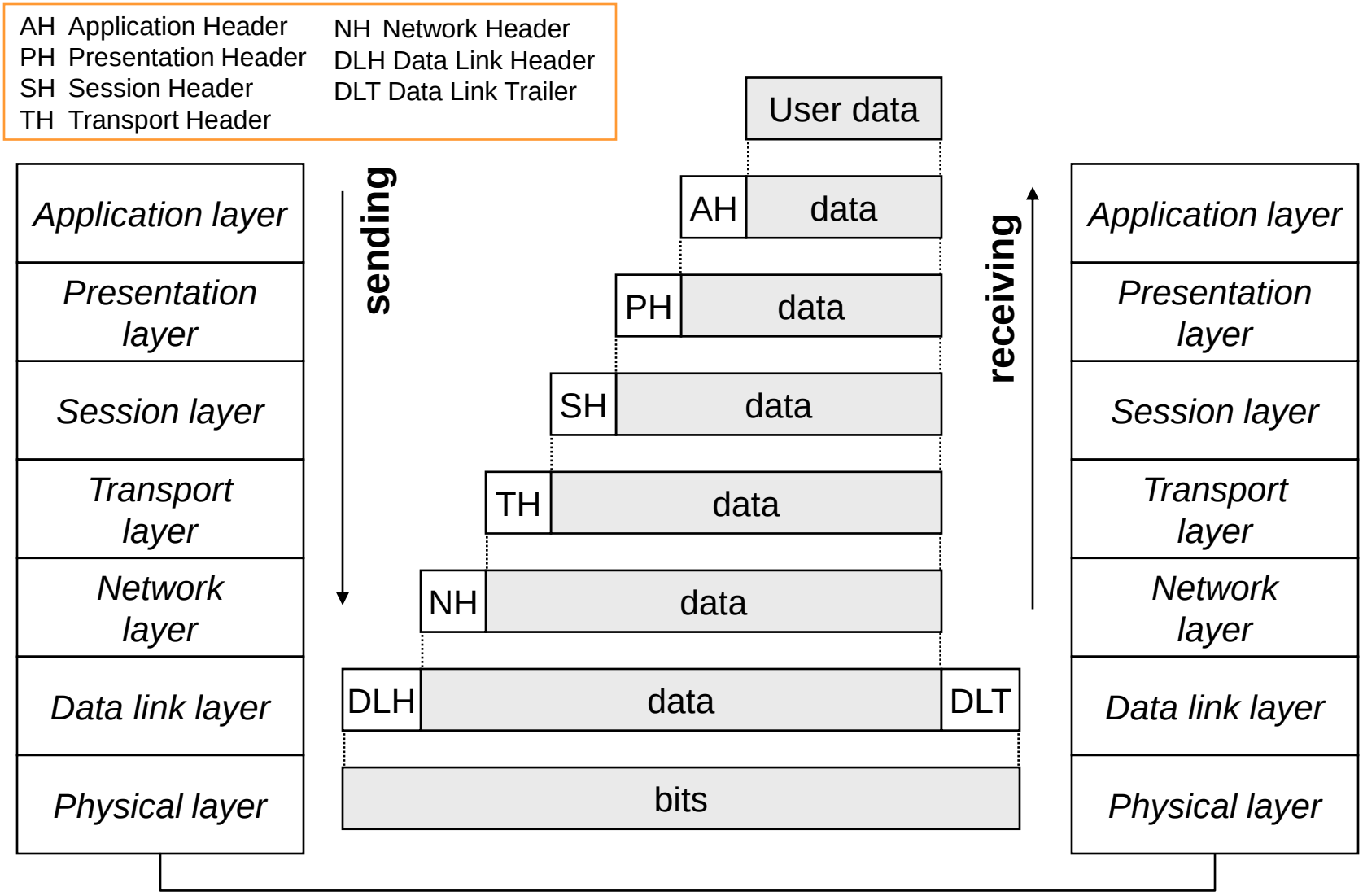


Protocols and Messages

- When using lower-layer services to communicate with remote peer, administrative data is usually included in those messages
- Typical example:
  - Protocol receivers data from higher layer
  - Adds own administrative data (header/trailer)
  - Passes the extended message down to the lower layer
  - Receiver will receive original message plus administrative data



Encapsulation of Data



Content (2)

8. Networked Computer & Internet
- Sockets
- Internet
- Layers
- Protocols

9. Host-to-Network I
- Physical Layer
- Media
- Signals
- Modems

10. Host-to-Network II
- Data Link Layer
- Framing, Flow Control
- Error Detection / Correction
- Point-to-Point Protocol

11. Host-to-Network III
- Topologies
- Medium Access
- Local Area Networks
  - Ethernet, WLAN

12. Internetworking
- Switches, Routers
- Routing
- Internet Protocol
- Addressing

13. Transport Layer
- Protocol Mechanisms
- TCP, UDP
- Addressing, Ports