Rechnernetze – Computer Networks

Lecture 6: Foundation
ISO/OSI Reference Model

Prof. Dr.-Ing. Markus Fidler

Institute of Communications Technology
Leibniz Universität Hannover

May 6, 2024

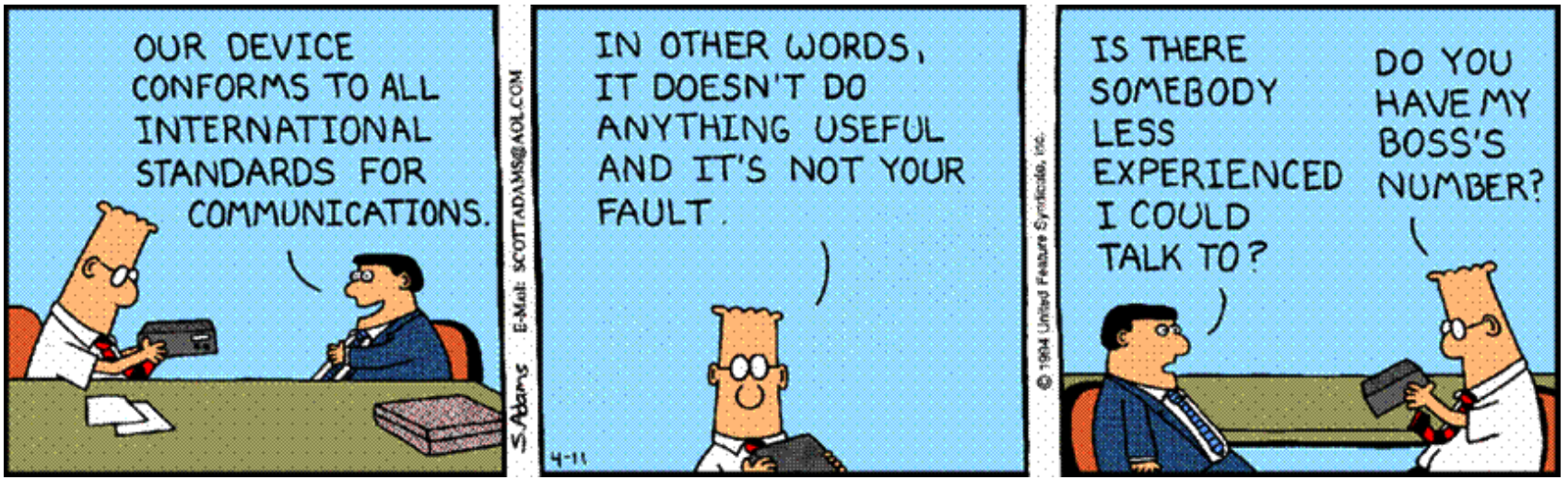
Outline

Reference model

Intermediate systems

Services

Layer functionalities



Motivation

Communication networks

- have to solve a large number of **complex** tasks
- have to achieve **inter-operability** of heterogenous systems

Design and implementation of such networks are difficult

- need a fundamental structuring of the problem
- levels of abstraction with different functionalities
- decomposition into modules, here hierarchical layers

The Open Systems Interconnection (OSI) reference model of the ISO provides an abstract architectural model for the design of networks

- it defines abstract concepts but
- it does not define their implementation

Direct link networks

So far: direct link networks

Decomposition into functional modules/layers

- layer 1: physical layer
  - line coding, modulation
  - forward error correction
  - framing
- layer 2: data link layer
  - error detection
  - automatic repeat request
  - flow control
  - medium access control

Encapsulation

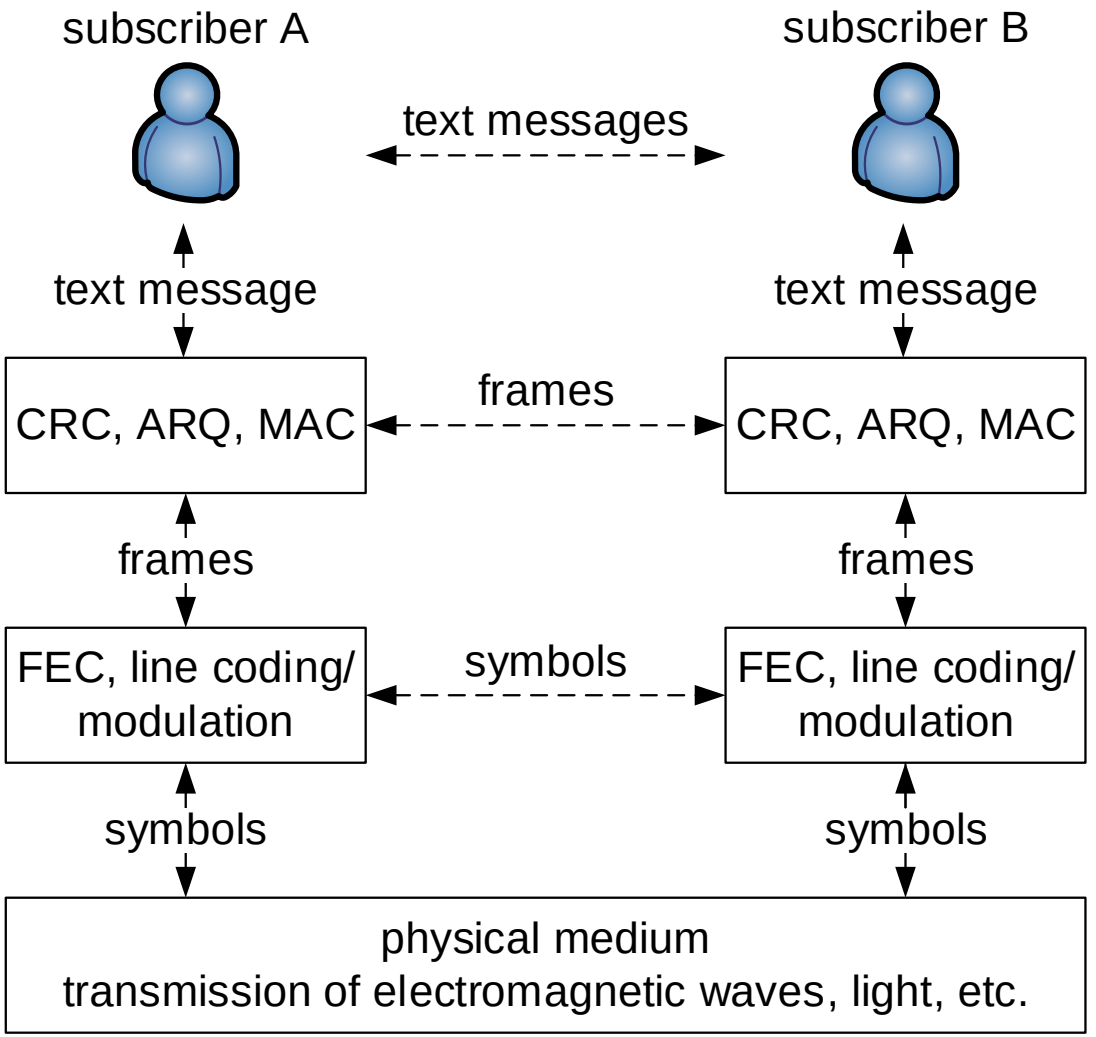
A basic principle in software development is the clear separation of interface and implementation, e.g.,

- layer 2 passes frames to layer 1 at the interface of layer 1
- how these frames are encoded is hidden in layer 1

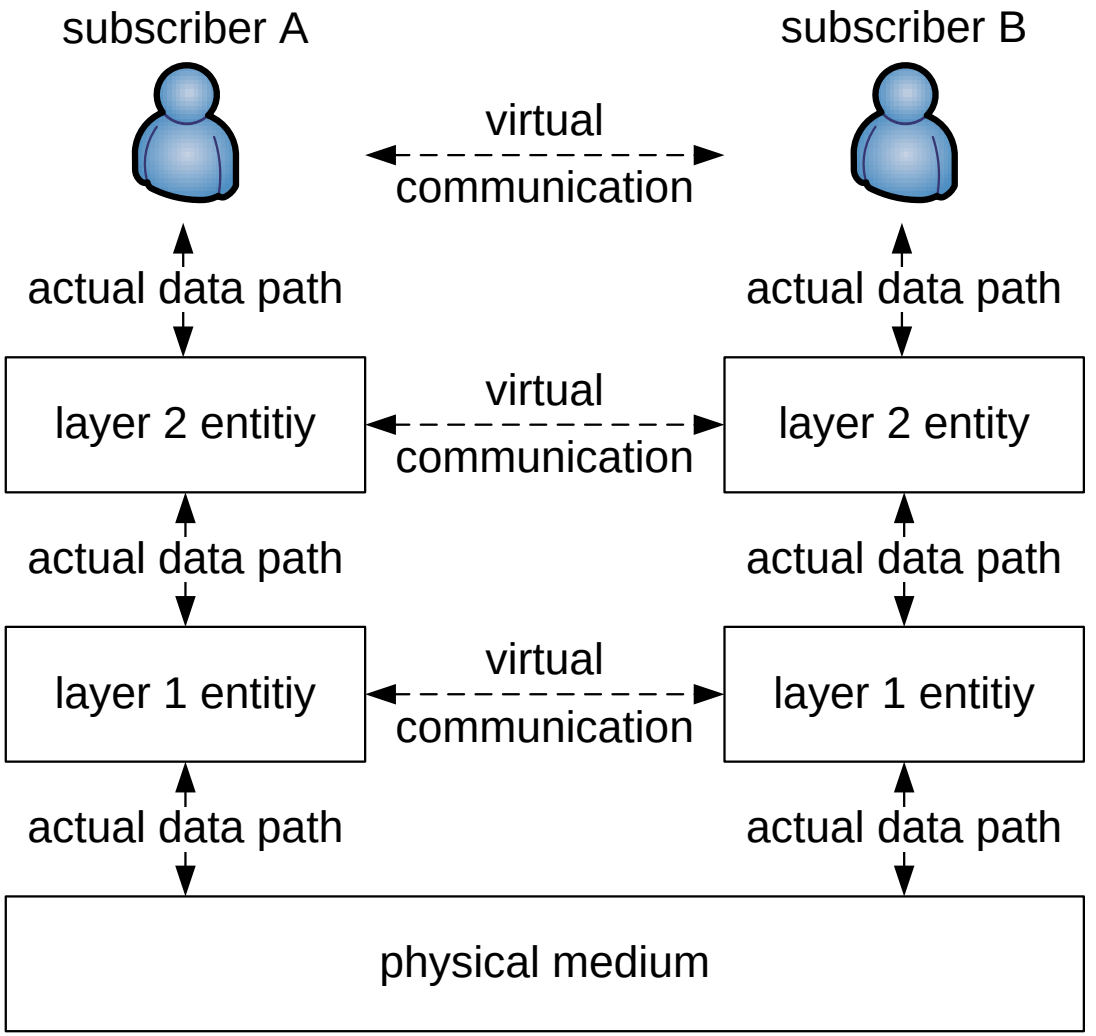
Advantage: changes inside the implementation of layer 1 do not affect layer 2 as long as the interface is left unchanged

- can use different layer 1 technologies without the need to re-implement layer 2
- can update layer 1 as technology advances to achieve, e.g., higher bit rates

Example: architecture, e.g., instant messaging



Example: actual vs. virtual communication

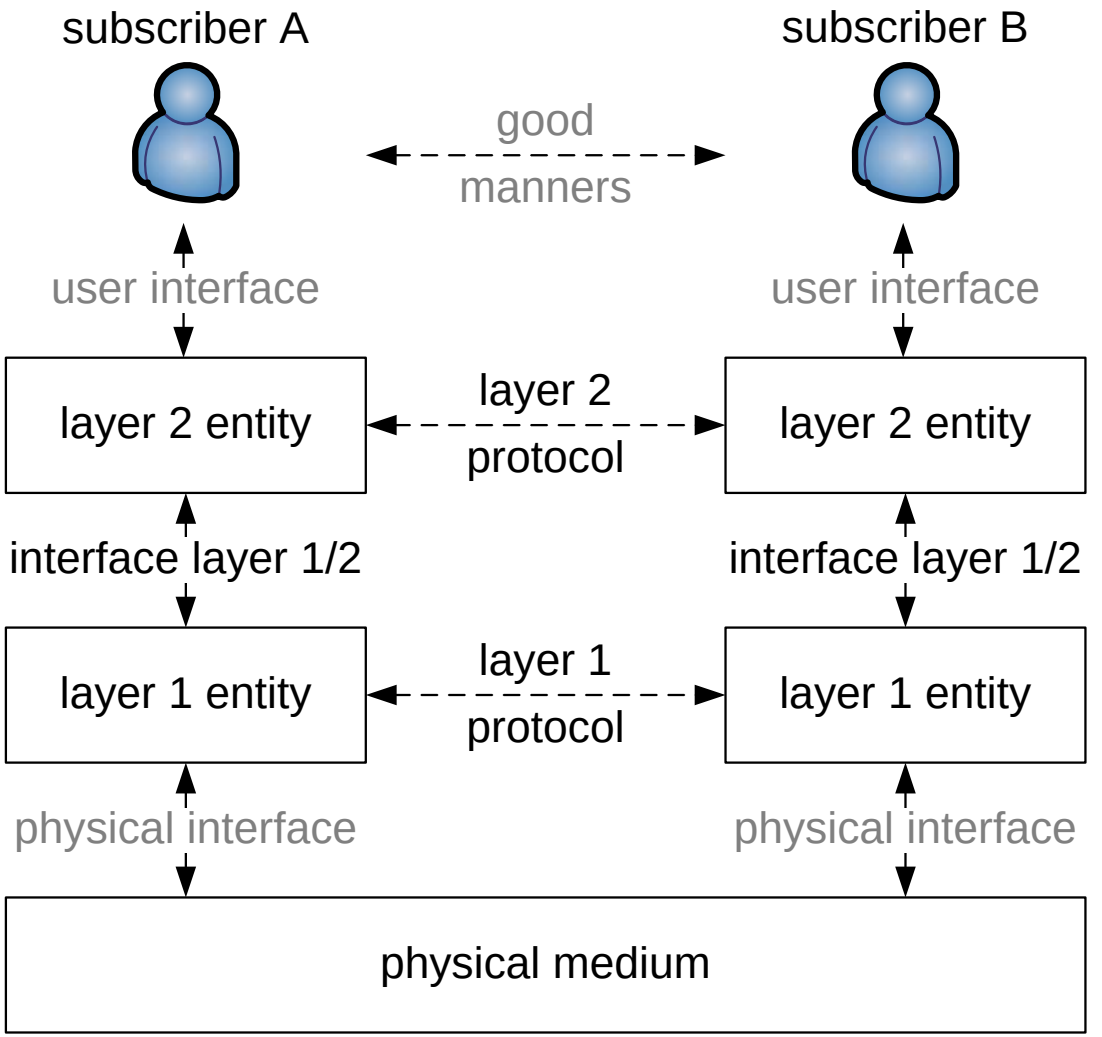


Actual vs. virtual communication

Communication applies among peer entities (within the same layer, on different machines) and between entities of adjacent layers (on the same machine)

- **virtual** communication
  - applies **horizontally** between peer entities within a single layer
  - according to syntactic (format) and semantic (meaning) rules specified by the **protocol**
- **actual** data transfer
  - is carried out **vertically** only between entities of adjacent layers at their **interfaces**
  - a layer offers **services** only to the next higher layer

Example: interface vs. protocol



Service, interface, and protocol

The OSI model strictly separates the concepts of service, interface, and protocol

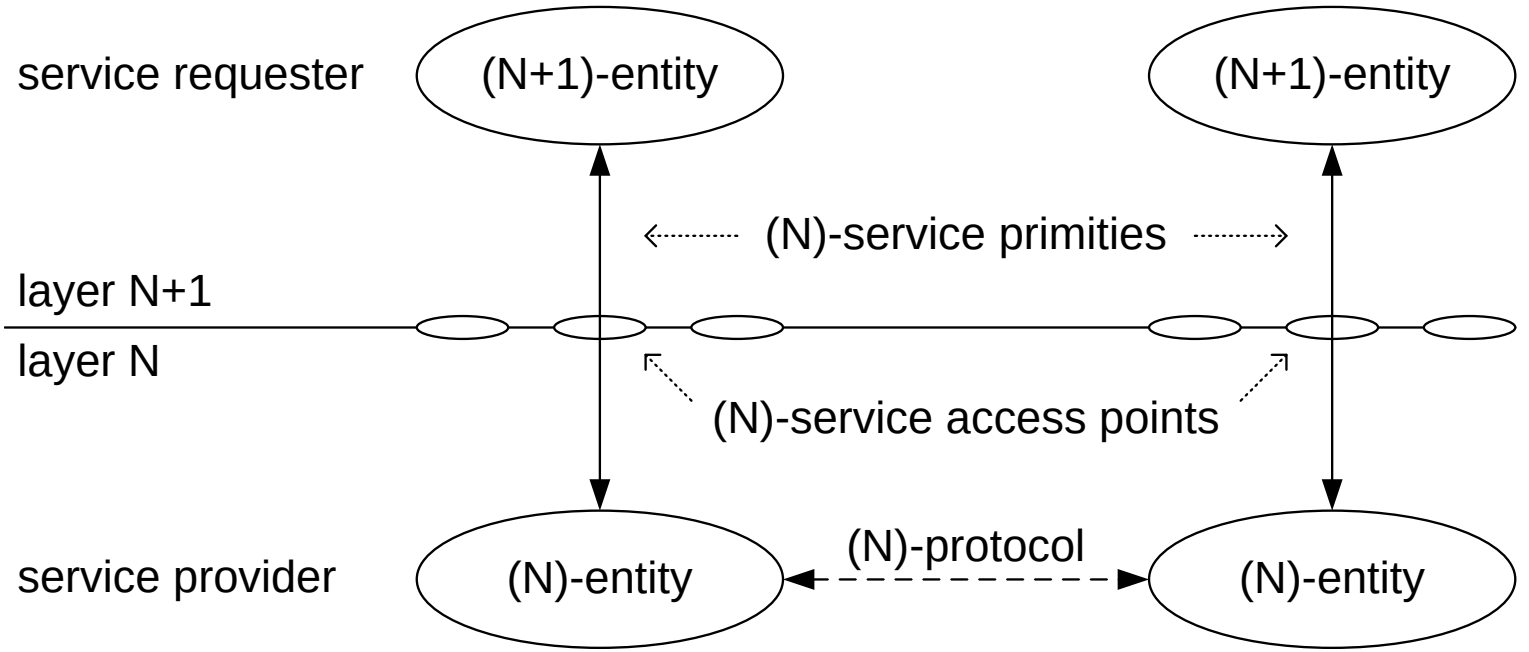
- **service:** the service defines the functions that are provided by a layer for the next higher layer
- **interface:** the interface defines how a service is accessed
- **protocol:** the protocol defines how the layer implements the service

Encapsulation

The strict separation of service, interface, and protocol has significant advantages (known also in software development)

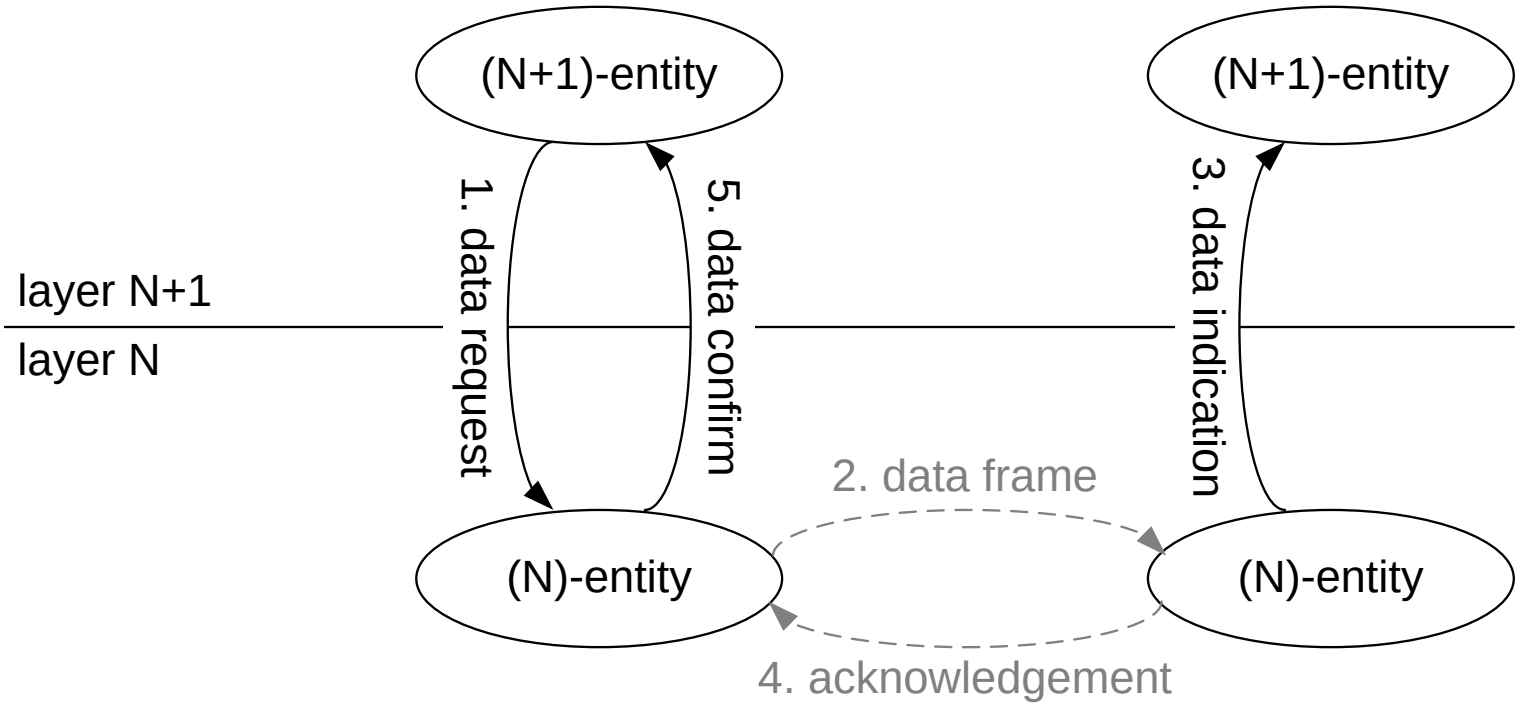
- the protocol is hidden from the next higher layer at the interface; the service does not specify how the protocol implements the service
  Advantage: protocols are modifiable and exchangeable
  - service: transmission of symbols
  - protocol: Manchester coding or 64-QAM
- the implementation of a system including its interface is hidden from other peer systems by the protocol
  Advantage: communication between heterogeneous systems
  - interface on system A: big-endian byte order, ASCII characters
  - interface on system B: little-endian, Unicode characters

Service provider and requester



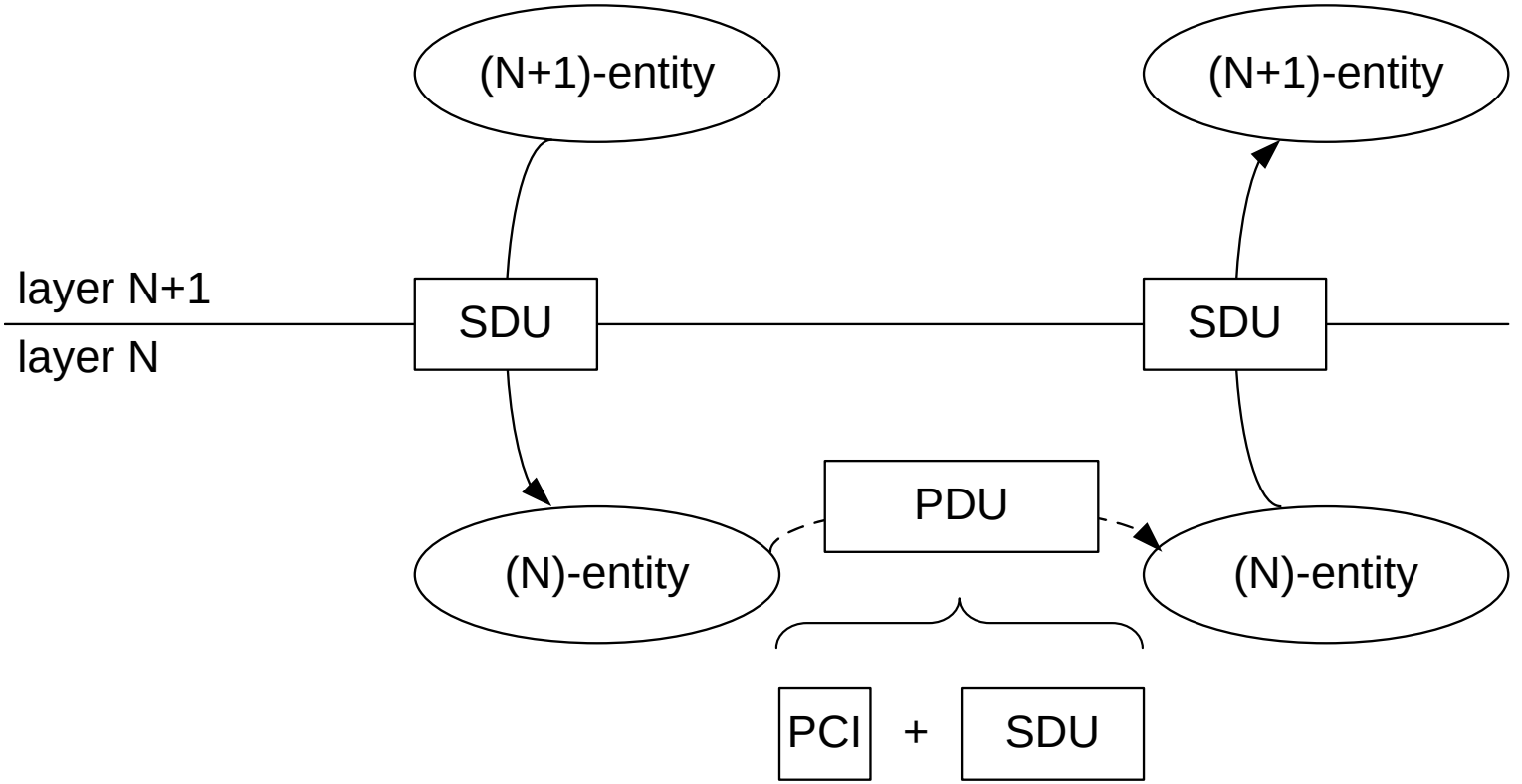
- layer N is service provider for layer N+1
- service primitives are used to access services
- service access points are used for unambiguous addressing

Service primitives



- example service primitives (1,3,5) for reliable data transmission
- layer N requires an acknowledgement from the receiving entity
- if a data frame is not acknowledged the sender retransmits it

Data units



- SDU: service data unit
- PCI: protocol control information
- PDU: protocol data unit; (N)-PDUs become (N-1)-SDUs

OSI reference model

The OSI reference model is an abstract model for network protocol architectures; it defines 7 layers and their functionalities:

7. **application:** application related services, e.g. email transfer
6. **presentation:** machine independent representation, e.g. UTF
5. **session:** synchronization, e.g. after lost connection
4. **transport:** end-to-end connection between hosts
3. **network:** addressing and routing
2. **data link:** per-hop data transfer, medium access control
1. **physical:** coding, modulation, and physical transmission

The OSI model seeks to balance the functionality per layer, achieve small interfaces, and a manageable number of layers.

Network architecture

A **network architecture** is a concrete specification of

- layers
- protocols

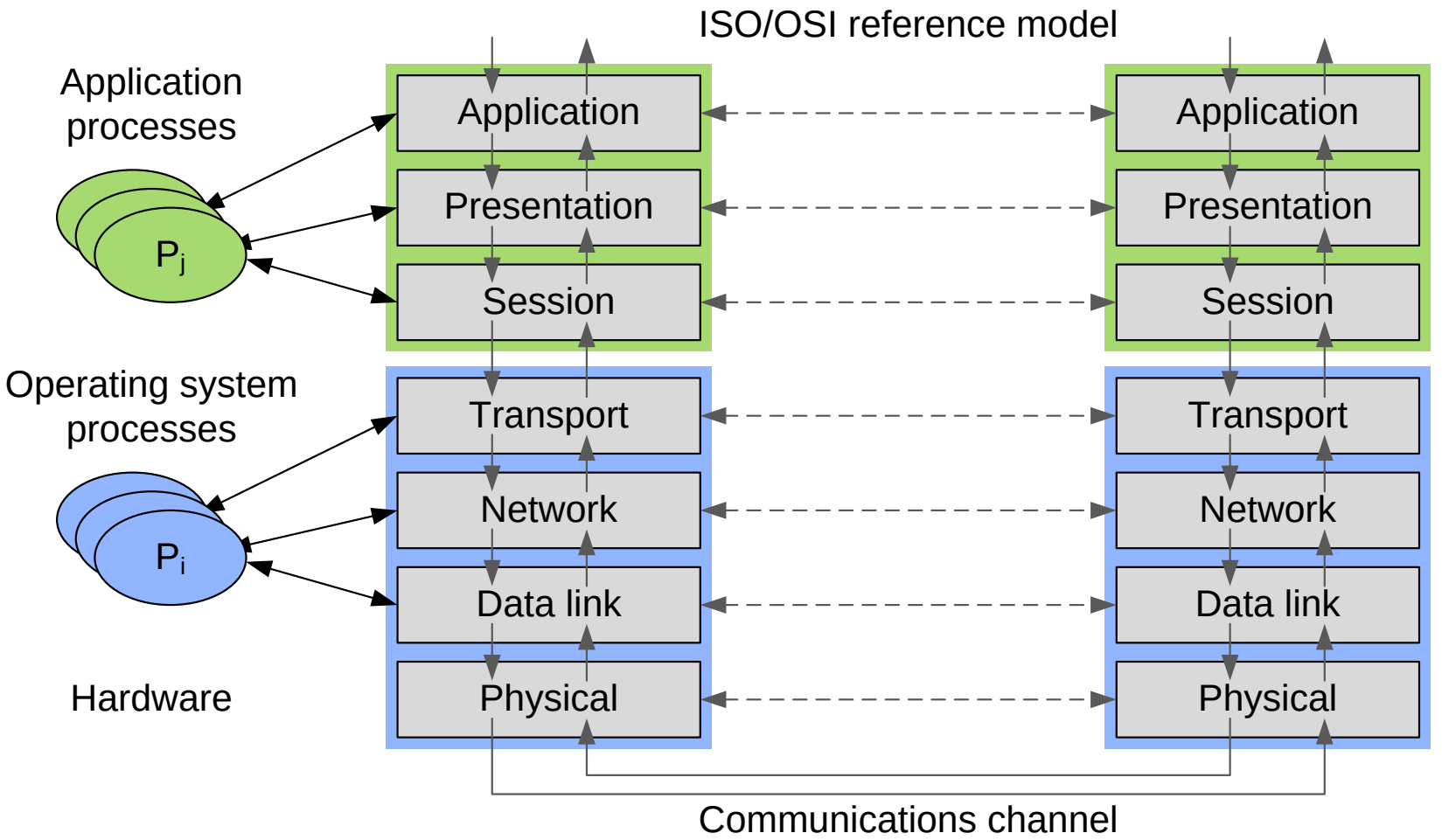
that contains enough information to build a system that obeys the defined protocols. The hierarchy of protocols implemented is called the **protocol stack**.

The specification of a network architecture does **not** contain

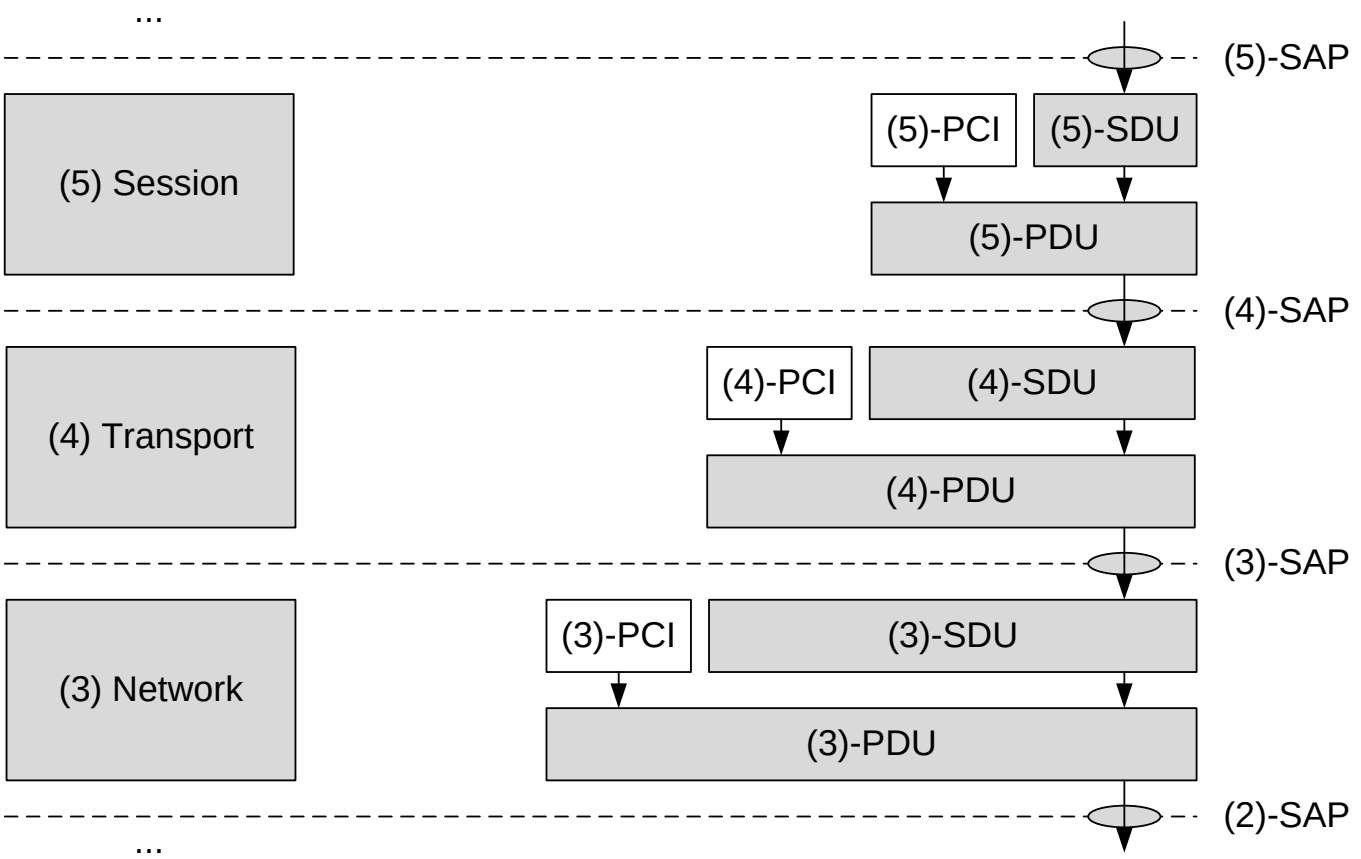
- the specification of the interfaces
- the details of the implementation

since these are hidden and not visible from the outside, as long as the protocols are obeyed.

Protocol hierarchy



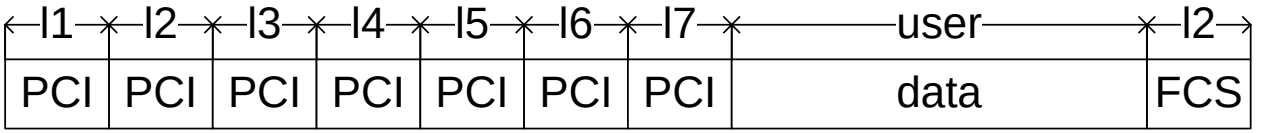
Nesting of data units



SDU: service data unit
PDU: protocol data unit

PCI: protocol control information
SAP: service access point

Nesting of data units continued



PCI: protocol control information
FCS: frame check sequence

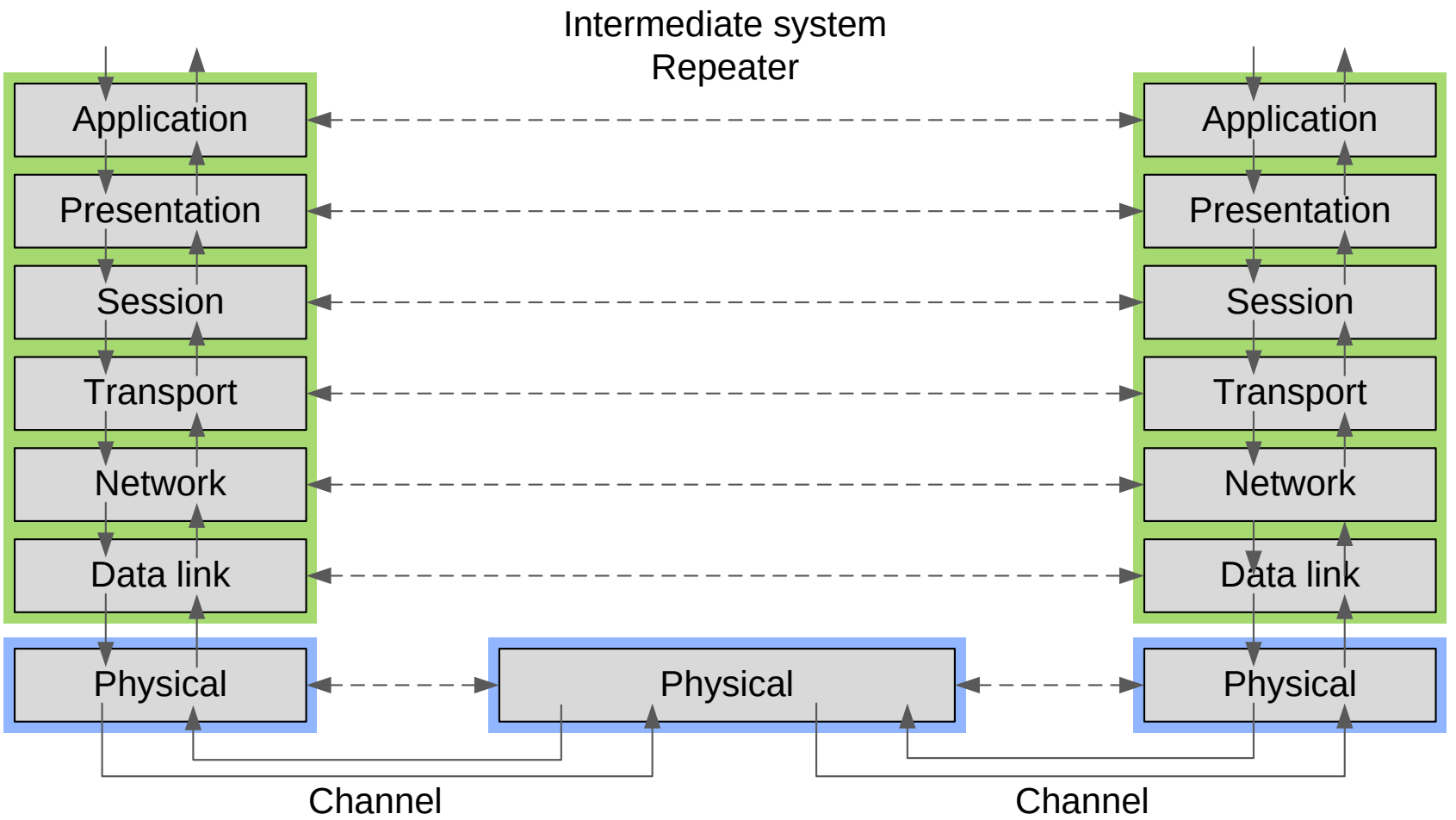
- at the sender each layer adds its PCI
- at the receiver each layer removes its PCI
- peering entities within a layer may exchange PCI without user data, e.g. layer 4 PCI is nested into layer 3 and layer 2 PCI
- typically layer 2 (but possibly also other layers) may add a frame check sequence to detect bit errors
- layer 1 is rather concerned with transmitting bits and does not necessarily add PCI

Intermediate systems

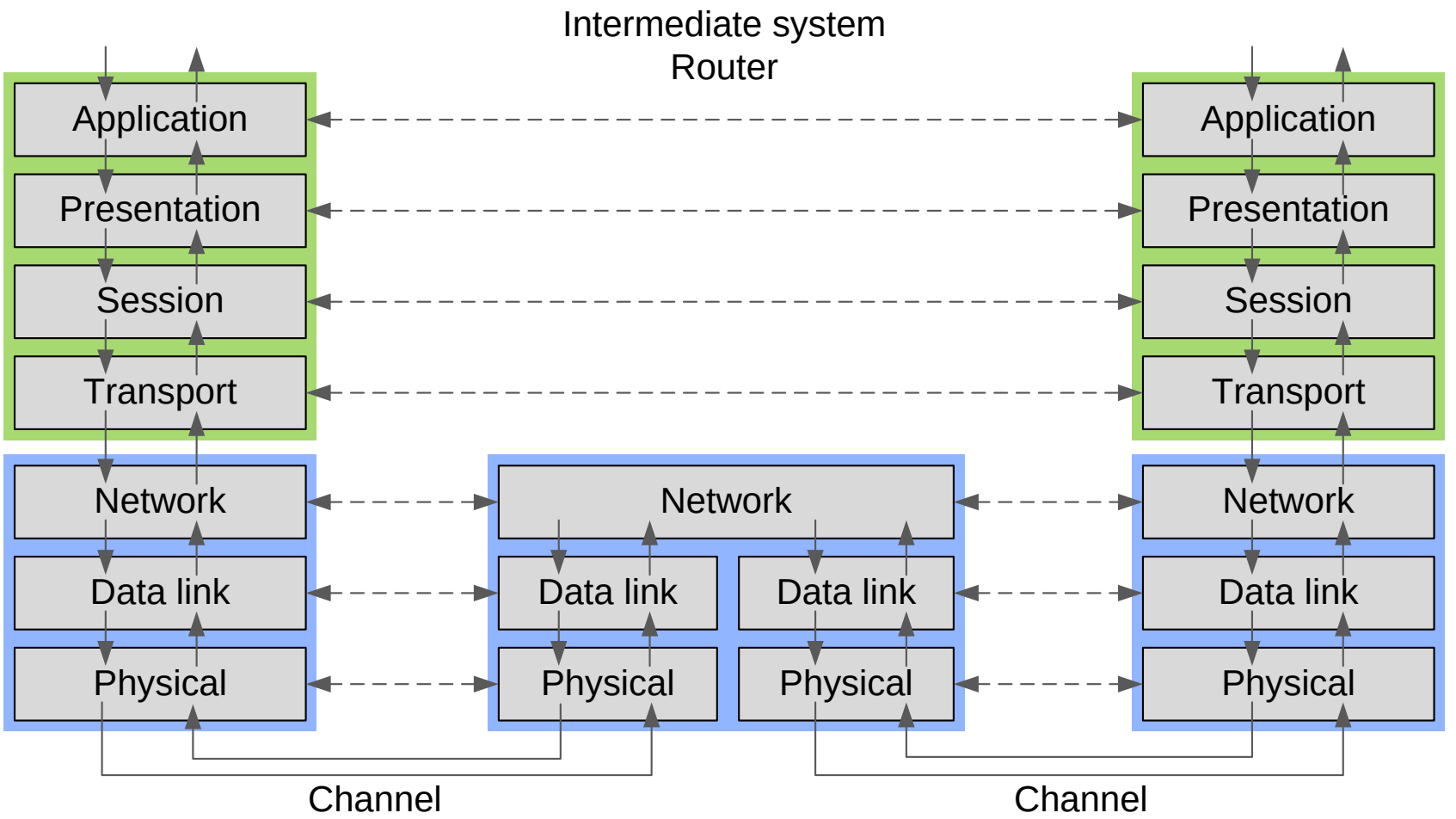
Intermediate systems, i.e. the network infrastructure, forward data from source to destination. The task can involve different layers

- layer 1: repeater
  - reconstruction and amplification of the physical signal
- layer 2: switch/bridge
  - forwarding of data link protocol data units
  - can involve different layer 1 protocols, e.g. bridge from Ethernet to Wifi within the IEEE 802 family of protocols
- layer 3: router
  - routing and forwarding of network protocol data units
  - can involve different layer 1 and layer 2 protocols, e.g. Ethernet, ATM, SONET
- layers 4...7: gateway
  - anything above the network layer, e.g. transition between VoIP and ISDN telephony or deep packet inspection by a firewall

Repeater

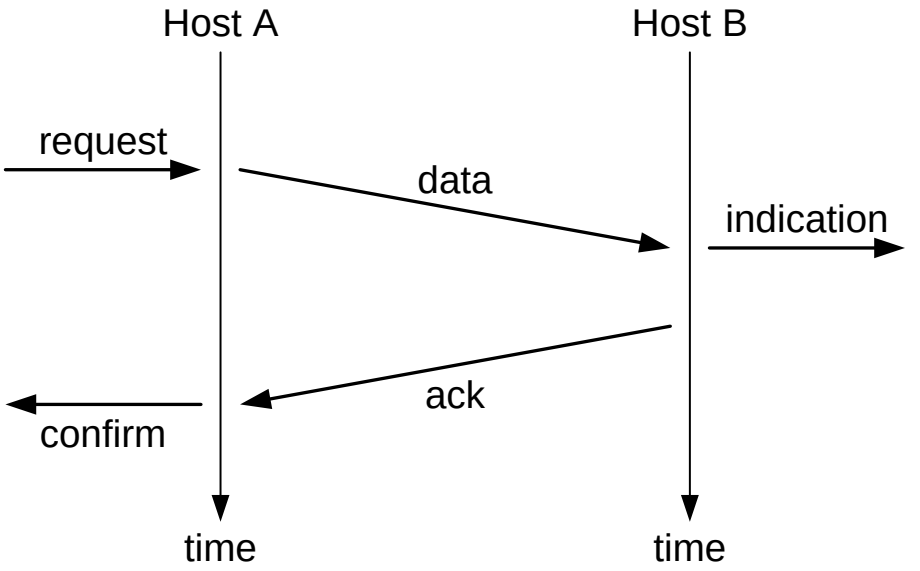

Intermediate system
Repeater
Application
Presentation
Session
Transport
Network
Data link
Physical
Physical
Application
Presentation
Session
Transport
Network
Data link
Physical
Channel
Channel

Router

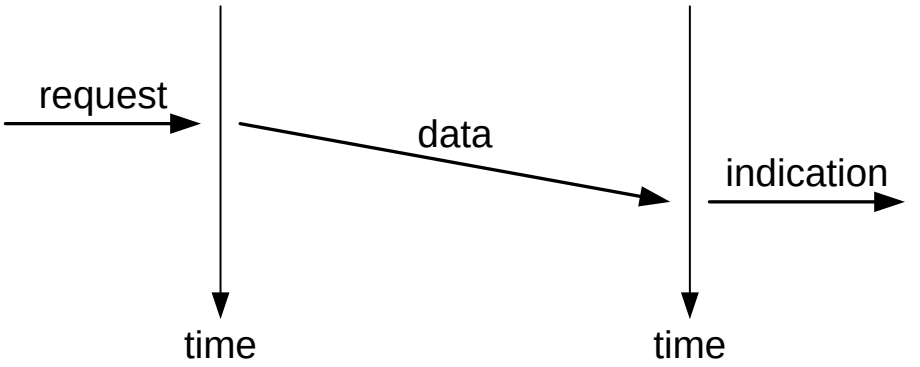

Intermediate system
Router
Application
Presentation
Session
Transport
Network
Data link
Physical
Network
Data link
Data link
Physical
Physical
Application
Presentation
Session
Transport
Network
Data link
Physical
Channel
Channel

Reliable vs. unreliable service

- reliable service
- acknowledged
- retransmission of lost data



- unreliable service
- unacknowledged



Connection-oriented vs. connectionless service

Connection-oriented service

- three phases
  - connection establishment (optional parameter negotiation)
  - data transfer (discrete messages or continuous byte stream)
  - connection release
- example: telephone system

Connectionless service

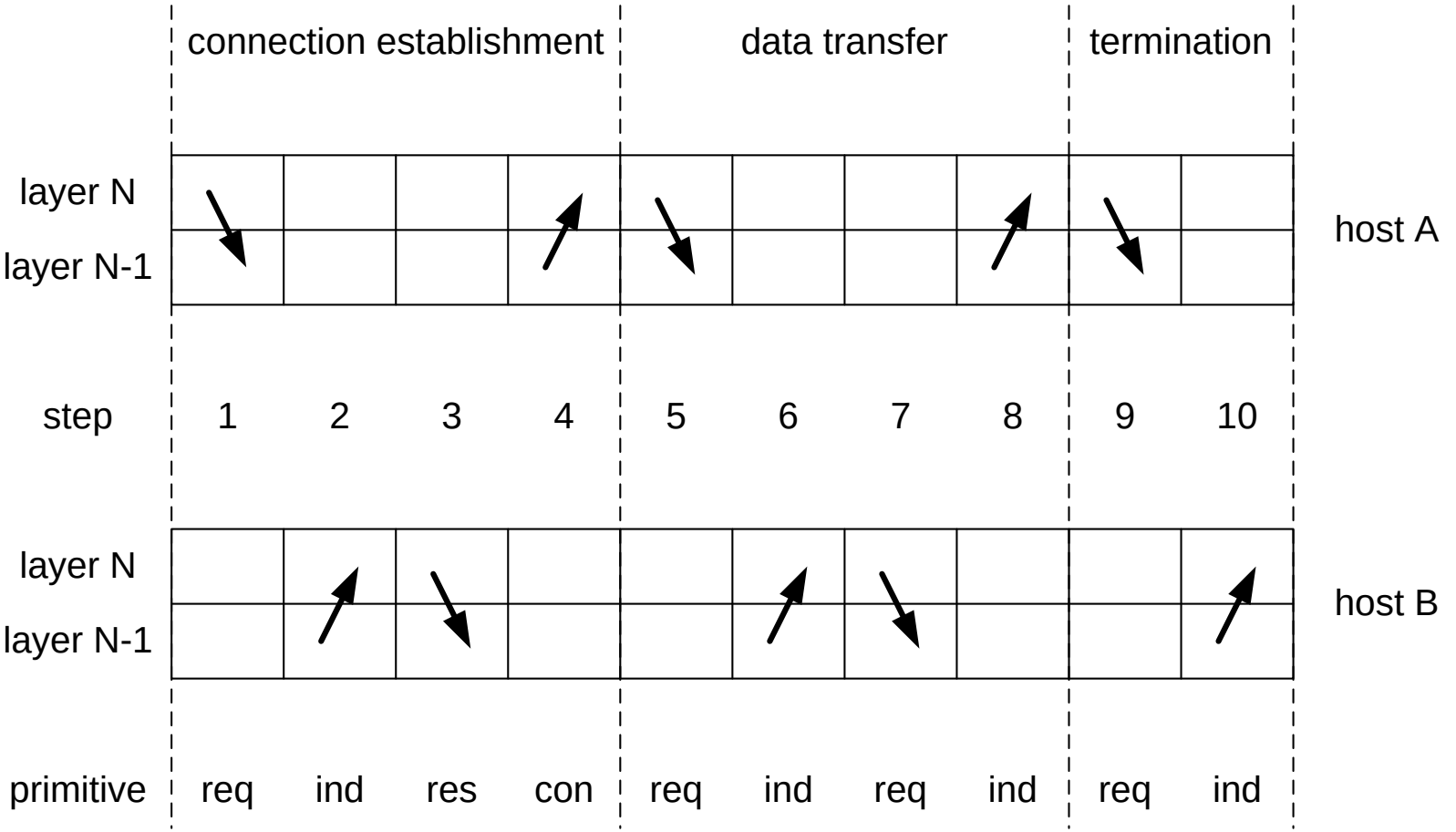
- data (messages) can be sent immediately
- no connection establishment and termination
- example: post system

examples	connection-oriented	connectionless
reliable	remote login	email with receipt request
unreliable	video streaming	junk email

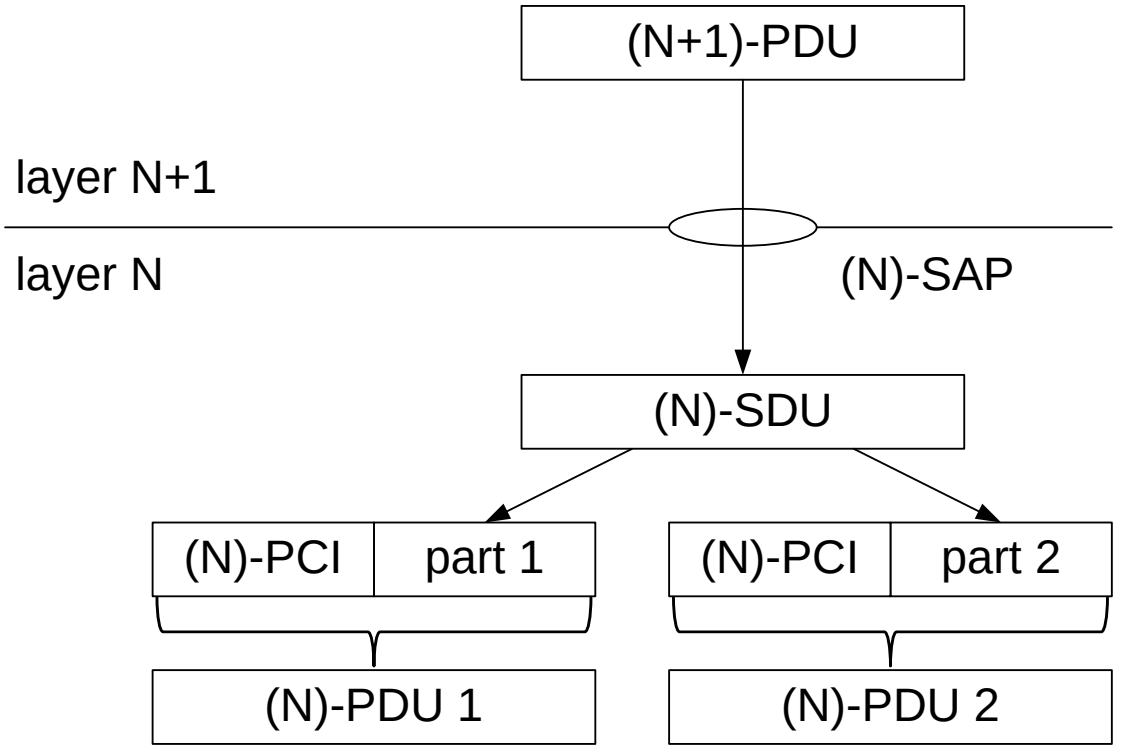
Example: Connection-oriented service

	connection establishment	
1	connect.request	A dials B's phone number
2	connect.indication	B's phone rings
3	connect.response	B answers the phone
4	connect.confirm	A hears that ringing stops
	data transfer	
5	data.request	A spreads the latest news
6	data.indication	B listens
7	data.request	B says that he understands
8	data.indication	A hears that B understands
	connection termination	
9	disconnect.request	A hangs up
10	disconnect.indication	B hears signal and hangs up

Connection-oriented service continued

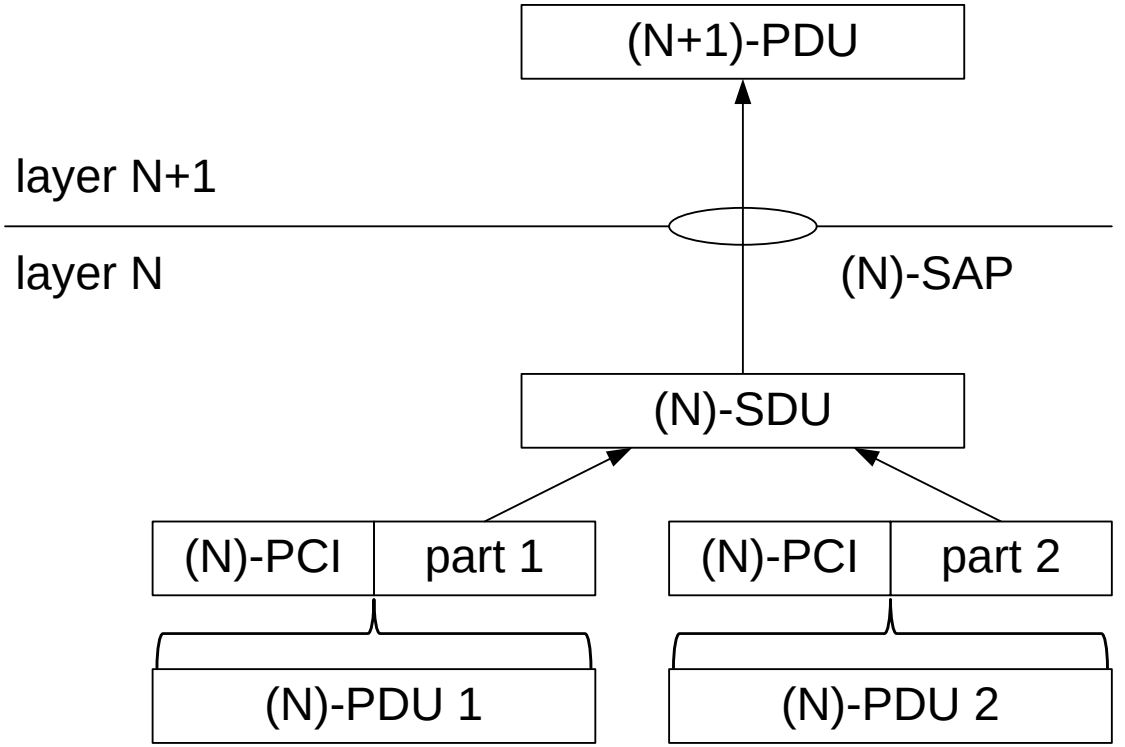


Segmentation



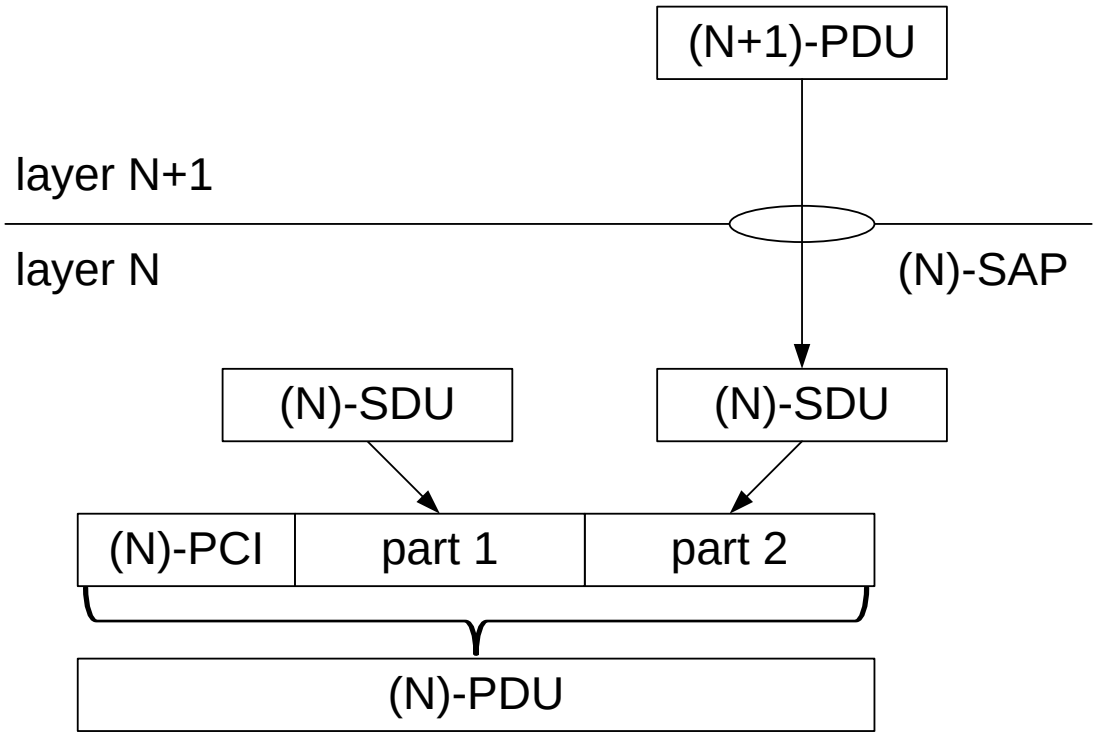
SDU: Service Data Unit
PDU: Protocol Data Unit
PCI: Protocol Control Information

Reassembly



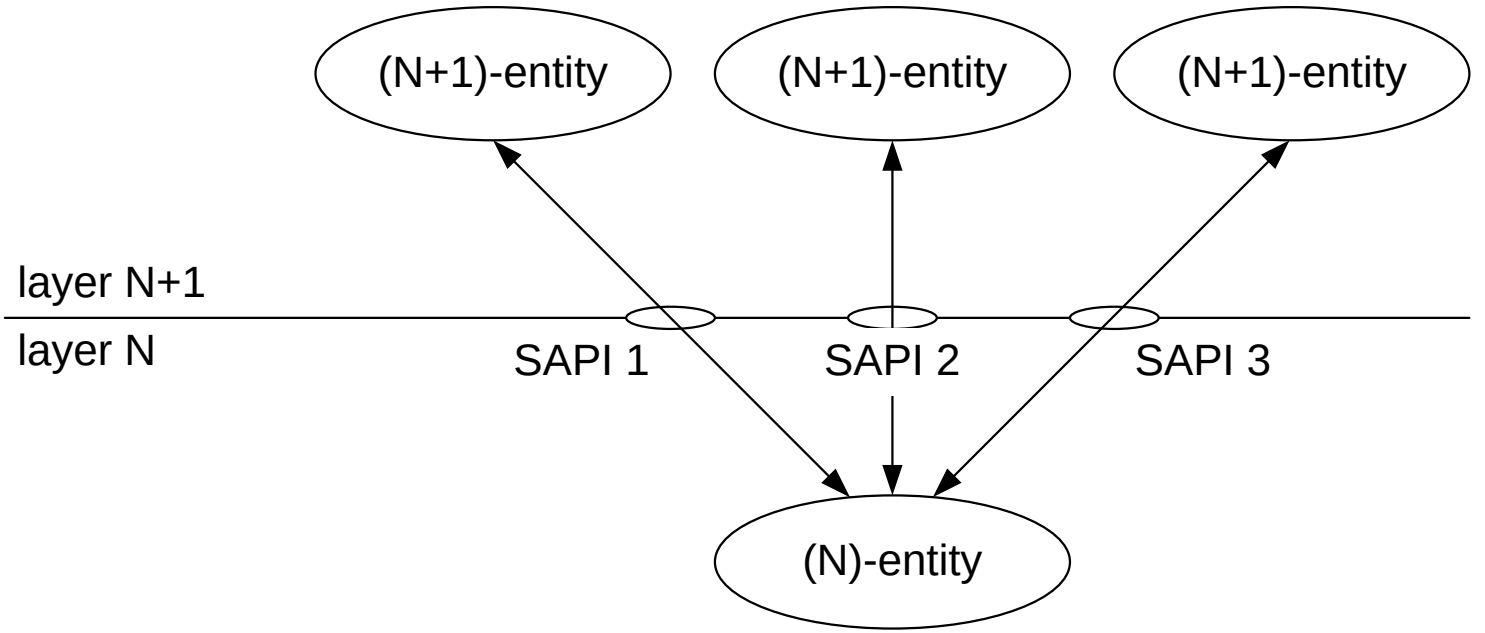
SDU: Service Data Unit
PDU: Protocol Data Unit
PCI: Protocol Control Information

Grouping and ungrouping



SDU: Service Data Unit
PDU: Protocol Data Unit
PCI: Protocol Control Information

Multiplexing and demultiplexing



SAPI: service access point identifier

- several (N+1)-entities use one N-entity: multiplexing
- (N)-PDUs specify the (N)-SAPI in the (N)-PCI for demu'xing

Basic protocol functions

Traditionally seven basic protocol functions

- segmentation/reassembly
- encapsulation
- connection control
- ordered delivery
- flow control
- error control
- multiplexing/demultiplexing

Further functions that are frequently itemized comprise

- addressing/routing
- congestion control
- quality of service

Physical layer

- task: transmit bits and ensure that bits are received correctly
- transmission medium: cable, fibre, air
- mechanics: connector, pins
- electronics: symbols, voltage, bit duration
- procedures
  - initialization and termination
  - synchronization
  - uni- or bidirectional communications
- framing, frame delimiters

Data link layer

- reliable data transfer
  - detection of remaining bit errors
  - retransmission of erroneous or lost data
  - detection of duplicated data
- frame format, header and trailer
  - data frames of tens up to thousands of bytes
  - segmentation and reassembly
  - data and acknowledgement frames
  - piggybacking of acknowledgements
- flow control
  - fast sender, slow receiver
  - control of transmission speed
- medium access control (sublayer)
  - multiple access to broadcast channels

Network layer

- end-to-end communication from a source to a destination over intermediate systems (network) as opposed to communication over a single channel (layers 1 and 2)
- routing, i.e. determining a path through the network
  - static routing, i.e. hard-wired routes
  - dynamic routing
- addressing of hosts
- interconnection of sub-networks
- congestion control
  - control of transmission speed to avoid network overload (as opposed to flow control between sender and receiver)
- quality of service (delay, delay jitter, packet loss)
- fundamental design choices
  - connection-oriented or connectionless
  - reliable or unreliable

Transport layer

- end-to-end communication between source and destination, typically not implemented by intermediate systems
- segmentation and reassembly
- end-to-end flow control
- multiplexing and demultiplexing of higher layer processes
- enhancement of layer 3 services
  - congestion control if not implemented by layer 3
  - reliable and/or unreliable services
  - connection-oriented and/or connectionless services

Session layer

- establishment of sessions
- session support over longer periods of time
- synchronization of sessions, e.g. if connectivity is interrupted
- dialogue control, e.g. manages the direction of a half-duplex channel
- token management, i.e. communications partners may not be allowed to perform certain operations simultaneously
- often the session layer implements few functions and may even be omitted or integrated into the application layer

Presentation layer

- problem: heterogenous end systems use different (incompatible) data representations
- this concerns syntax and semantics of the information (as opposed to raw bits transmitted by lower layers)
- negotiation of data structures
- conversion of information into a common data structure
- e.g. data types such as integer etc. may differ on end systems, characters may be encoded in ASCII, Unicode etc.
- often the presentation layer implements few functions and may even be omitted or integrated into the application layer

Layer 7: Application layer

Application layer

- application tailored services
- not the application itself
- examples
  - application layer protocol HTTP, application web
  - application layer protocol SMTP, application email

User, control, and management plane

Intermediate systems

- user/data plane
  - data transport
  - flow control
  - error control
- control plane
  - connection control
  - determines routes
- management plane
  - resource management
  - network management