Layers, Models & Protocols

Roman Dunaytsev

The Bonch-Bruevich Saint-Petersburg
State University of Telecommunications

roman.dunaytsev@spbgut.ru

Lecture № 3

Outline

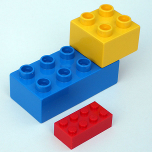
Outline

Layering

- Why layering?
- For network communications to take place, hundreds of problems must be solved (addressing, compatibility, error handling, etc.)
- Working with these hundreds of problems would be unmanageable...
  - **Decomposition method**: a complex task can be broken down into several subtasks, where each subtask is simpler than the original one and can be solved independently of the others
- Thus, we break down the many tasks involved in moving data from one machine to another into several groups
- Now instead of having hundreds of problems to solve at once, we get more manageable groups of tasks
- **2 basic design principles**:
  - Modularity
  - Hierarchy of abstraction levels (aka layering)

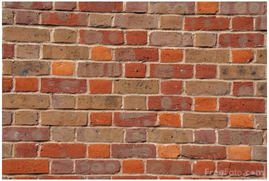
Layering (cont'd)

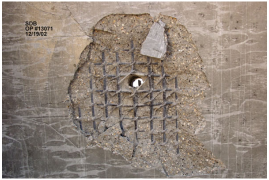
- **Modularity**
- Modules are analogous to building blocks of different shapes and sizes: when creating a building, each block has certain functions
- Designing one of these blocks is a much easier task than designing the entire building
- The blocks have standard interfaces to each other so that they fit together easily
- If the requirements for a block change, only that block needs to be changed, the other blocks are not affected



Layering (cont'd)

- Modular and unstructured design as brickwork and concrete structure:





- **Benefits** of modular design:
  - Enables a component to be placed in service or taken out of service with little or no impact on the rest of the system
  - Facilitates troubleshooting, problem isolation, and management
- **Shortcomings** of modular design:
  - Modular systems are not well optimized for performance (this is usually due to the cost of putting up interfaces between modules)

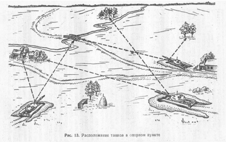
Layering (cont'd)

- **Layering**
- Layering is a common technique to break a complex system into many logical layers, which are at different levels of abstraction
  - An abstraction level is a way of hiding the implementation details of a particular set of functionality
- Layers are arranged into a vertical stack
- Each layer is responsible for performing a particular task, as well as interacting with the layers above it and below it
- The lower layers are charged with more specific tasks (e.g., hardware signaling) and provide services to the higher layers
- The higher layers in turn use these services to implement more abstract functions (e.g., establishing a connection)

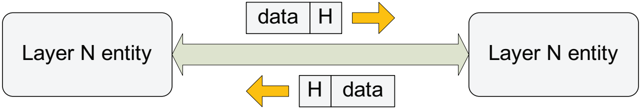
Layering (cont'd)

- Different levels of abstraction in armed forces:

- Commander (high level)
- Officer (mean level)
- Soldier (low level)

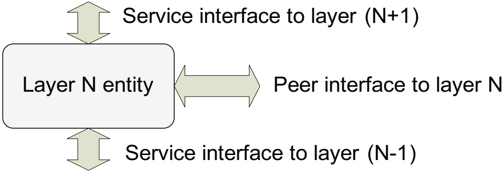


- In each layer, a process on one machine carries out a conversation with a peer process on the other machine across a peer interface
- The processes at layer N are referred to as **layer N entities**
- Layer N entities communicate by exchanging **protocol data units (PDUs)**, each of which contains control information (a header) and some data
- As a rule, the communication between peer entities is virtual (no direct communication link exists between them)



Layering (cont'd)

- The behavior of the layer N entities is specified by a set of rules called **a layer N protocol**
- Each protocol defines 2 kinds of interfaces:
  - **A service interface** to the other objects on the same machine (specifies the operations that local objects can perform)
  - **A peer interface** to its counterpart (peer) on another machine (specifies the form and meaning of messages exchanged between protocol peers to implement a communication service)



Layering (cont'd)

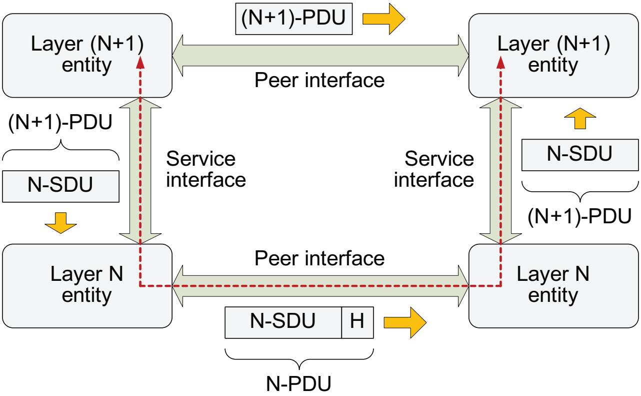
- For communication to take place, layer (N+1) entities use the services provided by layer N entities
- The transmission of a layer (N+1) PDU is performed by passing a block of information from layer (N+1) to layer N across the service interface
- The block of information passed between layer (N+1) and layer N entities is called a layer N **service data unit (SDU)**, which is the layer (N+1) PDU itself
- Usually, layer N entities do not interpret the information contained in SDUs

Layering (cont'd)

- The layer N entity uses the control information to form a header that is attached to the SDU to produce a layer N PDU
- Thus, the layer (N+1) PDU is **encapsulated** in the layer N PDU
- Upon receiving the layer N PDU, the layer N peer entity uses the header to execute the layer N protocol and, if appropriate, to deliver the SDU to the corresponding layer (N+1) entity
- The communication process is completed when the SDU (which is the layer (N+1) PDU) is passed to the layer (N+1) peer entity

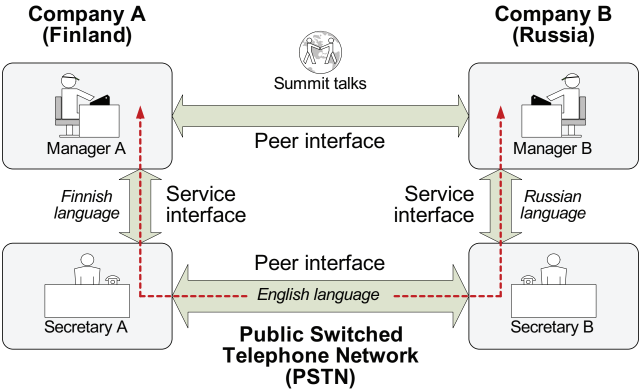
Layering (cont'd)

- Peer-to-peer communication



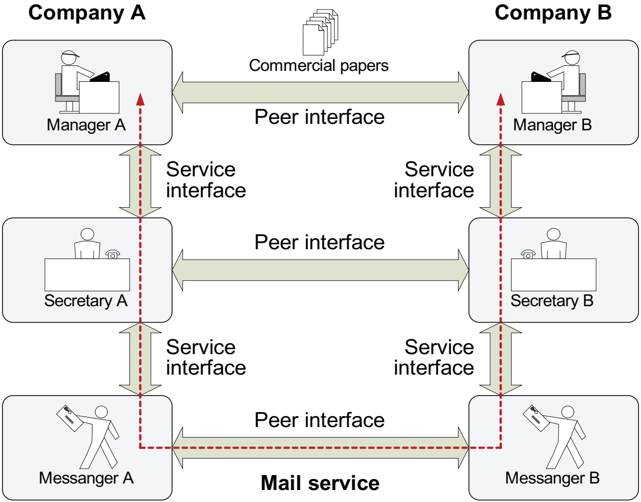
Layering (cont'd)

- Layering can be found everywhere: armed forces, educational institutions, industry, business, etc.
  - E.g., a telephone call



Layering (cont'd)

- E.g., posting



Layering (cont'd)

- Thus, hierarchical modular design provides 3 useful features:
  1. Decomposes the problem of building a network into more manageable components
  2. Provides flexibility and allows incremental evolution
  3. Simplifies **standardization** activities

Outline

Standards

- Why standards?
- Communications networks are designed to serve a wide variety of users who are using equipment from **many different vendors**
- To design and build networks effectively, standards are necessary to achieve interoperability, compatibility, and required performance in a cost-effective manner

Standards (cont'd)

- **De jure standards** – result from a consultative process that occurs on a national and possibly international basis
- **De facto standards** – arise when a certain product, or class of products, becomes dominant in a market
- **Proprietary standards** – developed and controlled by one company
- **Open standards** – available to the general public and developed (or approved) and maintained via a collaborative and consensus driven process

Standards (cont'd)

- **Benefits** of standards:
  - Make the interconnection of systems from different vendors possible
  - Make users and network operators vendor independent
  - Make international services available worldwide
  - Enable competition
  - Lead to economies of scale

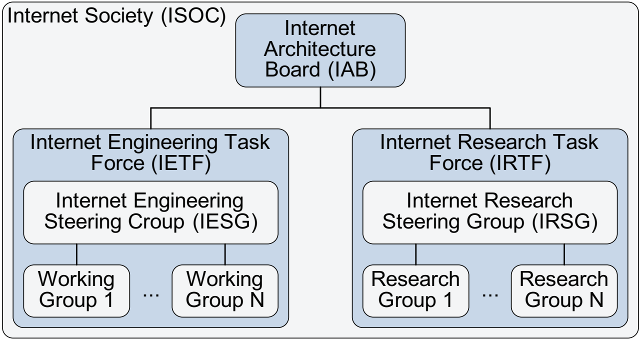
- **Shortcomings** of standards:
  - A standard tends to freeze the technology
  - Multiple standards can be released for the same thing
  - Political interests often lead to different standards in Europe, Japan, and the USA

Standards (cont'd)

- **International Telecommunication Union - Telecommunication Standardization Sector (ITU-T)**
  - A United Nations-sponsored agency for defining telecommunications standards
  - www.itu.int/net/home/index.aspx
- **International Organization for Standardization (ISO)**
  - An international organization that produces standards for industry and trade (e.g., film speed, screw threads, and telecommunications)
  - www.iso.org/iso/home.htm
- **Institute of Electrical and Electronics Engineers (IEEE)**
  - Sponsors the 802 committee that has developed many of the local area and metropolitan area networking standards
  - www.ieee.org/portal/site
- **European Telecommunications Standards Institute (ETSI)**
  - A standards body that represents the telecommunications companies from about 60 countries inside and outside of Europe
  - www.etsi.org/WebSite/homepage.aspx

Standards (cont'd)

- **Internet Society (ISOC)**
  - A professional society responsible for general, high-level activities related to the management and development of the Internet



Standards (cont'd)

- **Request For Comments (RFC)**
- The RFC series contains technical and organizational documents about the Internet, including the technical specifications and policy documents produced by the IETF
- The first RFC (RFC 1 - Host Software) was published on April 7, 1969 by Steve Crocker
- The RFC Editor assigns each RFC a unique serial number
  - Once assigned a number and published, an RFC is never rescinded or modified
  - If the document requires corrections, the authors publish a revised document
  - Thus, some RFCs supersede others; the superseded RFCs are said to be obsolete

Standards (cont'd)

- Not all RFCs are standards!
- RFC 2026 defines specification maturity levels:
  - Standards Track: Proposed, Draft, Standard
  - Non-Standards Track: Experimental, Informational, Historic
  - 'Best Current Practice': the BCP series usually covers technical recommendations for how to practice Internet standards
- And almost every April Fools' Day, the IETF publishes one or more humorous RFC documents

Outline

OSI Model

- The ultimate goal of open standards is to make possible both **interconnection** (i.e., physical compatibility) and **interoperability** (i.e., logical compatibility) for equipment from different vendors
- In the early days of computing, manufacturers often made their physical components and software interfaces proprietary so that their customers would be forced to buy all of their equipment from a single source
- Soon both manufacturers and their customers began to see the limitations of this proprietary approach
- The **Open Systems Interconnection (OSI)** effort began in the 1970s as a series of articles on the development of a standard reference model for networking
- It promoted the idea of a common model of protocol layers, defining interoperability between network devices and software

OSI Model (cont'd)

- As a result, the standard was approved in 1984 as ISO International Standard 7498 'Information processing systems - Open Interconnection - Basic Reference Model'
- In 1994, it was replaced by ISO International Standard 7498-1 'Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model'
- The identical text is also published as ITU-T Recommendation X.200 (previously 'CCITT Recommendation')
  - Available for free download at www.itu.int/rec/T-REC-X.200-199407-I/en
- **An open system** is one based on a common model of network architecture and built to conform open standards

OSI Model (cont'd)

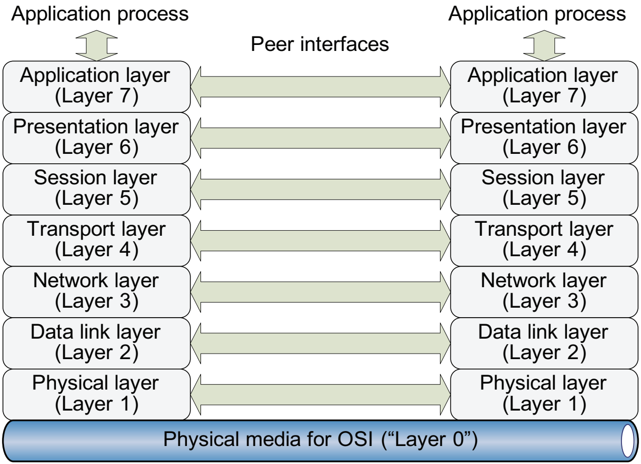
- The purpose of the OSI reference model is to provide a common basis for the coordination of standards development for the purpose of systems interconnection, while allowing existing standards to be placed into perspective within the overall reference model
- It is also the purpose of the OSI reference model to identify areas for developing or improving standards, and to provide a common reference for maintaining consistency of all related standards
- The OSI reference model **does not specify services and protocols** for OSI. It is neither an implementation specification for systems, nor a basis for evaluating the conformance of implementations
- Rather, the OSI reference model provides a **conceptual and functional framework** which allows international teams of experts to work productively and independently on the development of standards for each layer of the OSI reference model

OSI Model (cont'd)

- Principles to determine the number and nature of the layers:
  - For simplicity, keep the number of layers within a small limit
  - Create layer boundaries that minimize layer interactions (i.e., minimize information flow across layer interfaces)
  - Each layer has boundaries (interfaces) only to its higher and lower adjacent layers
  - Select layer boundaries that past experience shows have functioned successfully
  - Create boundaries that might permit the corresponding interface to be standardized
  - Separate layers so that different functions are separated from each other
  - Place similar functions in the same layer to allow a redesign with minimal effect to adjacent layers

OSI Model (cont'd)

- It was decided to use **7 layers**, numbered 1 through 7 from bottom to top

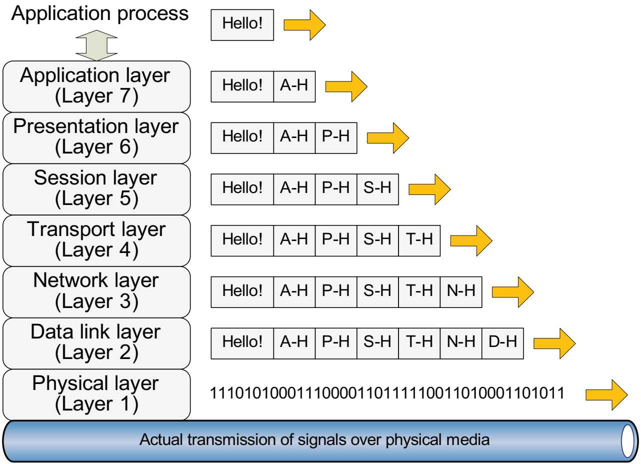


OSI Model (cont'd)

- The OSI layers use various forms of control information to communicate with their peer layers in other systems
- This control information consists of specific requests and instructions that are exchanged between peer OSI entities
- Control information typically takes one of two forms: **headers** and **trailers**
- **Headers are prepended** to data that has been passed down from upper layers
- **Trailers are appended** to that data

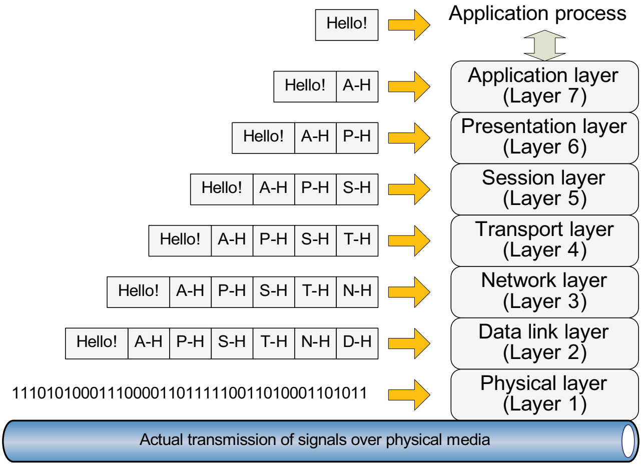
OSI Model (cont'd)

- Each layer in the source system adds control information to the transmitted data



OSI Model (cont'd)

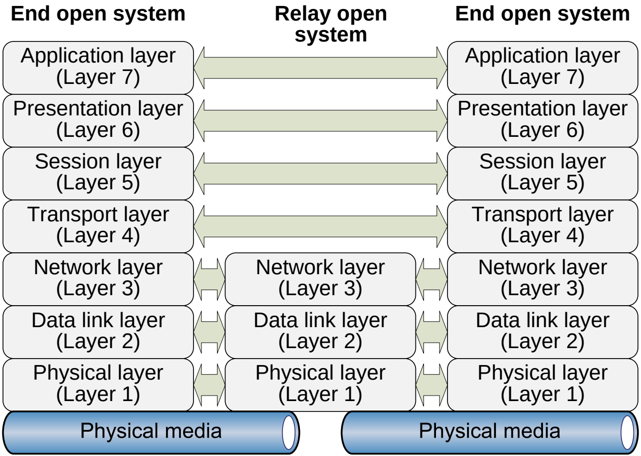
- And each layer in the destination system analyzes and removes this control information



OSI Model (cont'd)

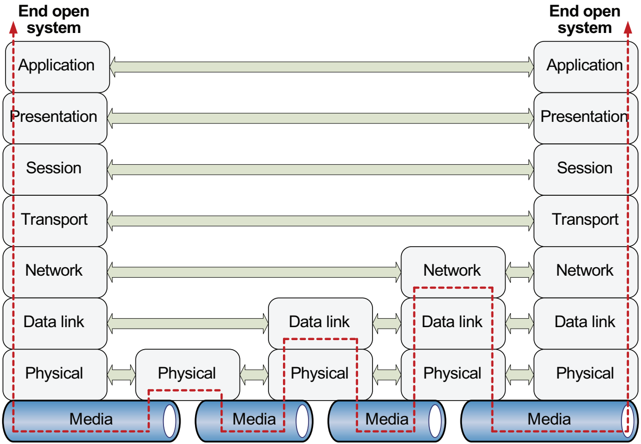
- The highest is the application layer and it interfaces directly to and performs common application services for **application processes**
- The lower layers provide the services through which application processes on different open systems communicate
- Layers 1 to 6, together with the physical media for OSI provide a step-by-step enhancement of communication services
- Layers 2 to 7 perform encapsulation/decapsulation
- When the physical media for OSI do not link all open systems directly, some intermediate open systems act only as **relay open systems**, passing data to other open systems
  - The functions and protocols which support the forwarding of data are then provided in the lower layers

- Communication involving a relay open system



OSI Model (cont'd)

- Communication involving different relay open systems



OSI Layers

- **The physical layer** – provides the mechanical, electrical, functional, and procedural means to activate, maintain, and deactivate physical connections for bit transmission between data link entities
- A physical connection may involve intermediate open systems, each relaying bit transmission within the physical layer
- Physical layer entities are interconnected by means of physical media
- Physical connections may be made using a variety of materials such as the following:
  - Twisted-pair cable
  - Coaxial cable
  - Fiber-optic cable
  - Wireless communications

OSI Layers (cont'd)

- The physical layer gets PDUs from the data link layer above it and converts them into a series of signals
- These signals are sent across a transmission medium to the physical layer at the receiving end
- At the destination, the physical layer converts those signals into a series of bit values
- These values are grouped into PDUs and passed up to the data link layer

OSI Layers (cont'd)

- Physical layer standards specify **4 types of characteristics**:
- Mechanical
  - Physical dimensions of plugs or connectors, assignment of circuits to pins, connector latching, mounting arrangements, etc.
- Electrical (or optical if an optical medium is used)
  - Voltages or current levels, timings of signals (pulse rise times and durations), etc.
- Functional
  - Meanings to circuits or pins like Data, Control, Timing, and Ground (alternative representations are needed with media such as fiber optics and air)
- Procedural
  - Sequences of control and data messages to set up, use, and deactivate physical connections

OSI Layers (cont'd)

- **The data link layer** – provides functional and procedural means for connectionless mode of operation among network entities, and for connection-oriented mode of operation, including the establishment, maintenance, and release data link connections among network entities
- A data link connection is built upon one or several physical connections
- The data link layer detects and possibly corrects errors which may occur in the physical layer

OSI Layers (cont'd)

- Data link layer protocols specify different characteristics and mechanisms including:
  - Framing
  - Error control
  - Flow control
  - Addressing

OSI Layers (cont'd)

- **The network layer** – provides the functional and procedural means for connectionless mode of operation or connection-oriented mode of operation, supports transmission among transport entities and, therefore, provides to the transport entities independence of routing and relay considerations
- Any relay functions and hop-by-hop service enhancement protocols used to support the network service between end open systems are operating below the transport layer, i.e. within the network layer or below
- Thus, the network layer also provides for the end-to-end routing and delivery of packets through multiple networks

OSI Layers (cont'd)

- Network layer protocols specify the following:
- Determining addresses or translating from physical to logical (aka network) addresses
  - One of the functions of the network layer is, in fact, to provide capabilities needed to **communicate on an internetwork**
- Finding a route between the source and the destination or between two intermediate devices
- Establishing and maintaining a logical connection between these two nodes, to establish either a connectionless or a connection-oriented communication
- Fragmentation of large packets of data into fragments which are small enough to be transmitted by the underlying data link layer

OSI Layers (cont'd)

- **The transport layer** – provides transparent transfer of data between session entities and releases them from any concern with the detailed way in which reliable and cost effective transfer of data is achieved
- It also optimizes the use of the available network service to provide the performance required by each session entity at minimum cost
  - In the OSI reference model, the transport layer is responsible for providing data transfer at an agreed-upon level of quality, such as at specified transmission speeds and error rates
- It is released from any concern with routing and relaying since the network service provides data transfer from any transport entity to any other
  - The transport layer protocols **operate only between end systems**
- The transport layer detects and possibly corrects errors which may occur at the lower layers

OSI Layers (cont'd)

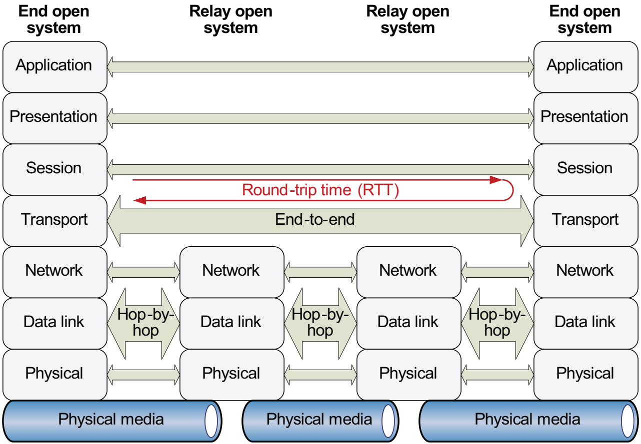
- Transport layer protocols provide **5 main functions**:
  - Set up, maintain, and tear down a connection between two session entities
  - Provide the reliable or unreliable delivery of data across this connection
  - Segment data into smaller, more manageable sizes (segmentation)
  - Multiplex connections, allowing multiple application processes to send and receive data simultaneously on the same networking device
  - Implement flow control and congestion control to ensure one component does not overflow another with too much data

OSI Layers (cont'd)

- The transport layer error control is **end-to-end**
  - End systems are responsible for loss detection, notification, and recovery
  - Lost or erroneous PDUs are recovered by retransmissions from the original sender (i.e., end system/end system)
  - It takes a lot of time but provides end-to-end reliable data delivery, even in the presence of relay system failures
- The data link layer error control is **hop-by-hop**
  - A pair of neighboring sender and receiver is responsible for loss detection, notification, and recovery
  - Lost or erroneous PDUs are recovered by local retransmissions (i.e., end system/relay system, relay system/relay system, relay system/end system)
  - It provides quick detection, notification, and recovery but cannot assure end-to-end reliable data delivery

OSI Layers (cont'd)

- End-to-end vs. hop-by-hop



OSI Layers (cont'd)

- **The session layer** – provides the means necessary for cooperating presentation entities to organize and to synchronize their dialogue and to manage their data exchange
  - To do this, the session layer provides services to establish a session connection between two presentation entities, to support orderly data exchange interactions, and to release the connection in an orderly manner
- The only function of the session layer for connectionless mode of operation is to provide a mapping of transport addresses to session addresses

OSI Layers (cont'd)

- Session layer protocols are responsible for the following:
- Establishing, maintaining, and ending a session
  - A session might be used to log into another machine or to transfer a file
- **Dialogue control**
  - When a device is contacted, the session layer is responsible for determining which device participating in the communication will transmit at a given time, as well as controlling the amount of data that can be sent in a transmission
- **Token management**
  - It is useful when both sides are not allowed to perform the same operation at the same time
  - To schedule these operations, a token is issued only to one process at each given time, allowing only the process that holds the token to perform the critical task

OSI Layers (cont'd)

- **The presentation layer** – provides for common representation of the data transferred between application entities
- This releases application entities from any concern with the problem of common representation of information, since it provides them with syntax independence
- The presentation layer ensures that the information content of the application layer data is preserved during transfer
- It also provides identification of a set of transfer syntaxes, selection of transfer syntax, and access to session layer services
- The presentation layer formats data for screen display or for printing and also takes care of the job of sending the bits in the correct order
  - As with bit order, different computers read the order of bytes in different ways (some computers read the least significant byte first; others read the most significant byte first)

OSI Layers (cont'd)

- Presentation layer protocols have **3 main jobs**:
- **Data presentation**
  - The main task of the presentation layer is the representation of data (e.g., integers, floating point numbers, or character strings)
  - Data presentation ensures that the data being sent to the recipient is in a format that the recipient can process
  - As different systems may use varying internal data representations (abstract syntax), the data sent are converted to an appropriate transfer syntax and are transformed back to the receiver's internal data format upon receipt
  - **Abstract syntax** – the format of data in the application layer before it is converted to any other format by the presentation layer (e.g., Extended Binary Coded Decimal Interchange Code, EBCDIC)
  - **Transfer syntax** – the format of data after it has been converted by the presentation layer into a 'common language' format (e.g., American Standard Code for Information Interchange, ASCII)

OSI Layers (cont'd)

- **Data compression**
  - **Data compression** – the reformatting of data to make it smaller
  - This allows data to be transferred more quickly across a network
- **Data encryption**
  - **Encryption** – the process of converting data into a random set of characters that is unrecognizable to everyone except the intended recipient

OSI Layers (cont'd)

- **The application layer** – provides the sole means for application processes to access the OSI environment
- Application processes use application layer protocols
- Application layer protocols contain the functionality to perform the following tasks (although other tasks do exist):
  - File transfer services (the most common of the services provided by the application layer)
  - E-mail services
  - Database access services

OSI Protocols

- **A protocol suite** – a set of protocols
- **A protocol stack** – a practical implementation of a protocol suite
  - The terms are often used interchangeably
- **The OSI protocol suite** – a complete, 7-layer protocol suite conforming to the OSI reference model
- The OSI specifications were developed and implemented by 2 international standards organizations: the ISO and the ITU-T
- The OSI protocol suite supports numerous non-ISO/ITU-T protocols at the physical and data link layers

OSI Protocols (cont'd)

- For more information on the OSI protocols, see:
  - www.protocols.com/pbook/iso.htm

OSI model	OSI protocol suite (example)
Application layer (Layer 7)	Message Handling System / ITU-T X.400-series
Presentation layer (Layer 6)	Presentation Protocol (PP) / ISO 8823, ITU-T X.226
Session layer (Layer 5)	Session Protocol (SP) / ISO 8327, ITU-T X.225
Transport layer (Layer 4)	Transport Protocol Class 4 (TP4) / ISO 8073
Network layer (Layer 3)	Packet-Layer Protocol (PLP) / X.25
Data link layer (Layer 2)	Link Access Procedure, Balanced (LAPB) / X.25
Physical layer (Layer 1)	Physical layer synchronous serial protocol (X.21bis) / X.25

OSI Protocols (cont'd)

- The OSI reference model is a widely known and accepted reference model in the data communications field
  - E.g., it is widely used as a framework for describing all functions required of an open interconnected network
- However, the OSI protocol suite has not received widespread use
  - *Andrew L. Russell, "Rough consensus and running code' and the Internet-OSI standards war,' IEEE Annals of the History of Computing, vol. 28(3), pp. 48-61, July 2006*

Outline

TCP/IP Model

- During the 1960s the US **Department of Defense (DoD)** sponsored research to interconnect a number of data centers located in laboratories and educational institutions
  - The initial goal of this networking research was to obtain a communications network that would provide the ability for data centers to exchange information even if certain network locations lost their operational capability
- The predecessor of today's Internet was the **ARPANET**, a network that was created by the Advanced Research Projects Agency (ARPA) and launched in 1969
  - The ARPANET was one of the first layered communications networks, preceding the development of the OSI reference model by approximately a decade

TCP/IP Model (cont'd)

- **The TCP/IP protocol suite** – was initially developed in the 1970s as a part of an effort to define a set of technologies to operate the ARPANET
  - It is so named for two of its most important protocols: **the Transmission Control Protocol (TCP)** and **the Internet Protocol (IP)**
- The first modern versions of these two key protocols were specified in 1981 as TCP (RFC 793) and IPv4 (RFC 791)
- In 1983, the TCP/IP protocol suite was deployed as the principal protocol suite of the ARPANET

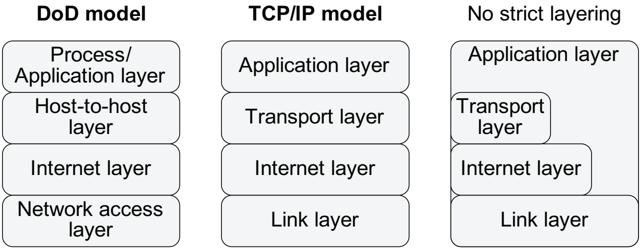
TCP/IP Model (cont'd)

- **The TCP/IP model** – a layered abstract description for communications and computer network protocol design
  - Sometimes called **the Internet reference model**, **the DoD model**, or **the ARPANET reference model**
- In contrast to the OSI reference model, the TCP/IP model has never been fully specified
  - The OSI reference model is a conceptual and functional framework for standards development and, therefore, is a **prescriptive model**
  - The TCP/IP model is a **descriptive model** of the existing TCP/IP protocol suite
- There is no universal agreement regarding how to describe TCP/IP by means of the OSI reference model

TCP/IP Model (cont'd)

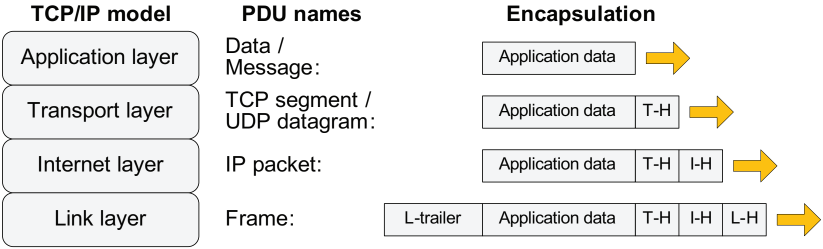
- The TCP/IP model consists of **4 layers**
  - Some descriptions present from 3 to 5 layers

- The TCP/IP model does not require strict layering

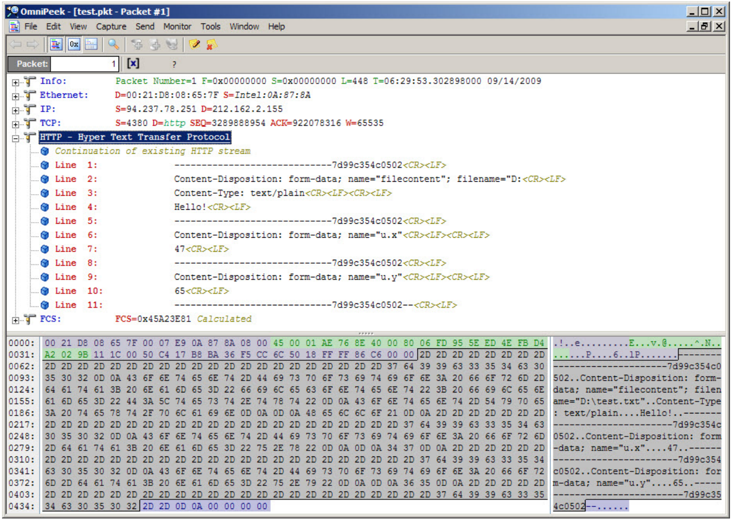


TCP/IP Model (cont'd)

- The layers communicate by performing **encapsulation/decapsulation** of PDUs
- The different layers follow different conventions for naming the PDUs



- An example of TCP/IP encapsulation



TCP/IP Layers

- **The link layer** (aka **the network access layer**, **the network interface layer**, or **the data link layer**)

- The link layer of the TCP/IP model corresponds to the data link layer of the OSI reference model

- The TCP/IP protocol suite defines only 2 protocols to be used at the link layer:
  - Serial Line Internet Protocol (SLIP)
  - Point-to-Point Protocol (PPP)

TCP/IP Layers (cont'd)

- Usually, the TCP/IP protocol suite uses non-TCP/IP protocols at the link layer

- There is a wide variety of standard protocols corresponding to different types of networks
  - Supported protocols include Ethernet/IEEE 802.3, Token Ring/IEEE 802.5, Fiber Distributed Data Interface (FDDI), WiFi/IEEE 802.11, WiMAX/IEEE 802.16, etc.
  - As a rule, the data link layer protocols are closely coupled with the physical layer protocols

TCP/IP Layers (cont'd)

- **The internet layer** (aka **the IP layer**)
- The internet layer of the TCP/IP model corresponds to the network layer of the OSI reference model
- The TCP/IP protocol suite defines several protocols to be used at the internet layer:
  - Internet Protocol (IP)
  - Internet Control Message Protocol (ICMP)
  - Address Resolution Protocol (ARP)
  - Reverse Address Resolution Protocol (RARP)
  - etc.

TCP/IP Layers (cont'd)

- **The transport layer** (previously known as **the host-to-host layer**)
- The transport layer of the TCP/IP model corresponds to the transport layer of the OSI reference model
- The TCP/IP protocol suite defines 2 protocols to be used at the transport layer:
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)

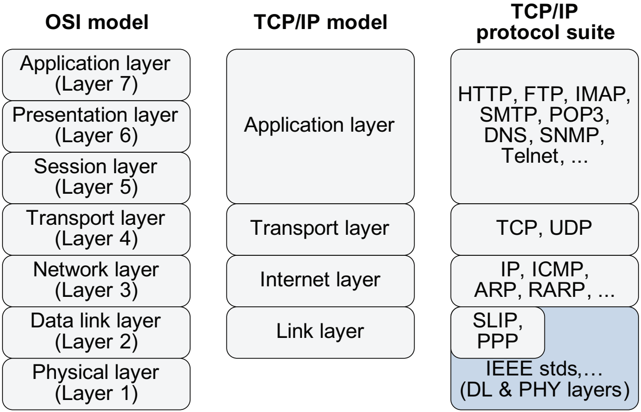
TCP/IP Layers (cont'd)

- **The application layer** of the TCP/IP model corresponds to the application layer, the presentation layer, and the session layer of the OSI reference model

- The TCP/IP protocol suite defines numerous protocols to be used at the application layer (both for end-user applications and network services):
  - HyperText Transfer Protocol (HTTP)
  - File Transfer Protocol (FTP)
  - Internet Message Access Protocol (IMAP)
  - Post Office Protocol Version 3 (POP3)
  - Domain Name Service (DNS)
  - Simple Network Management Protocol (SNMP)
  - etc.

TCP/IP Layers (cont'd)

- The OSI reference model vs. the TCP/IP model



Outline

Hardware/Software Implementation

- Layers 1 and 2 (i.e., the physical layer and the data link layer) are generally hardware layers
  - Typically, Layer 1 (the physical layer) is wholly concerned with hardware
  - Some procedures (such as the bit-stuffing/stripping and cyclic redundancy check computation) at Layer 2 (the data link layer) are usually performed in hardware for speed, although they can be done in software
- The higher layers are generally carried out in software

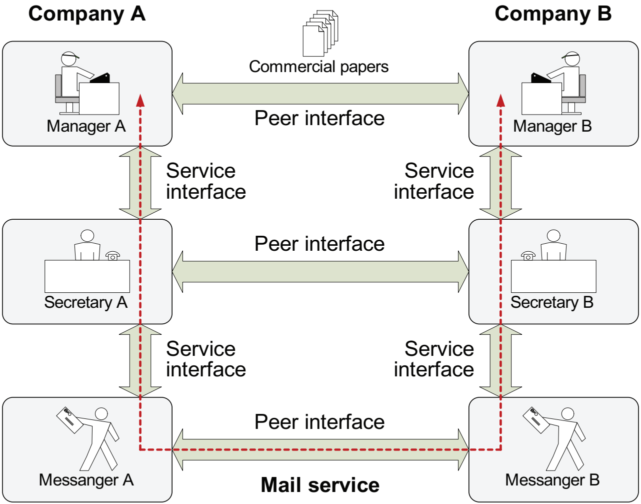
Hardware/Software Implementation (cont'd)

- However, hardware or software implementation is not stated in the protocols or the layered reference models
  - High-performance routers are to a large extent based on fast non-programmable digital electronics, carrying out Layer 3 (the network/Internet layer) switching
  - In modern modems and wireless equipment, the physical layer may partly be implemented using programmable digital signal processors or software radio (soft radio) programmable chipsets, allowing the chip to be reused in several alternative standards and radio interfaces instead of separate circuits for each standard

Outline

Cross-Layering

- Do we always need strict layering?



Cross-Layering (cont'd)

- The TCP/IP protocol suite is based on a layered model
- The layered model defines a restrictive interaction between the layers
  - Every layer is allowed to interact only with its adjacent layers
  - Communication resulting from **overjumping is prohibited**
- The TCP/IP protocol suite was developed primary for wired networks and performs suboptimal or poorly over wireless networks
- Wireless networks have pushed the activities around cross-layering

Cross-Layering (cont'd)

- Cross-layering approach neglects or weakens this protocol design rule for the purpose of performance optimization, resource preservation, or error/delay tolerance
- The basic idea behind the cross-layering is to optimize TCP/IP performance over wireless networks (in terms of data rate, losses, delay, jitter, battery power, etc.)
- The background for considering cross-layered approaches is derived from interlayer dependencies
- Currently, it is mostly a research topic...

Cross-Layering (cont'd)

- Different kinds of cross-layer design proposals