Fondamenti di Internet

Network Reference Model

Foreword

- In the digital era, various information is presented as data in our life. What is data? How is data transmitted?
- In this course, we will use the network reference model to understand the "life" of data.

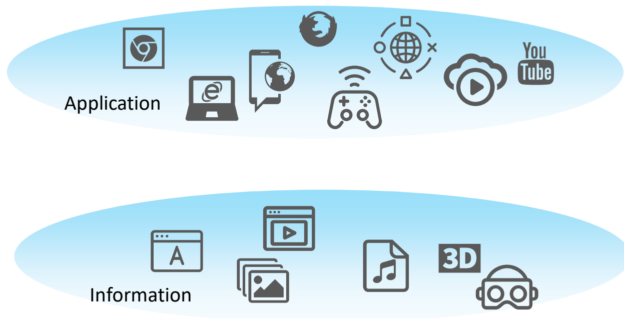
Objectives

- On completion of this course, you will be able to:
  - Understand the data definition and transmission process.
  - Understand the concepts and advantages of the network reference model.
  - Understand common standard protocols.
  - Understand the data encapsulation and decapsulation processes.

Contents

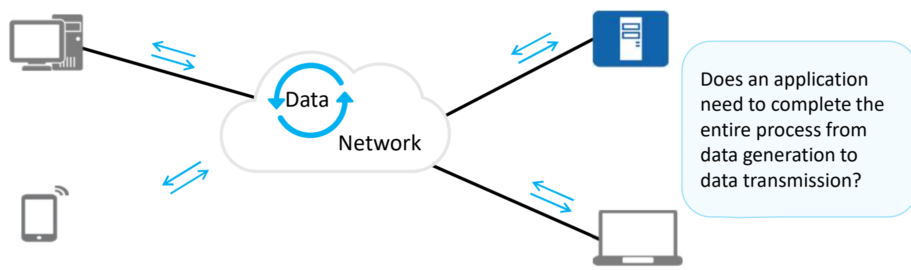
Origin of the Story - Applications

- Applications are used to meet various requirements of people, such as web page access, online gaming, and online video playback.
- Information is generated along with applications. Texts, pictures, and videos are all information presentation modes.



Application Implementation - Data

- Data generation
  - In the computer field, data is the carrier of all kinds of information.
- Data transmission
  - Data generated by most applications needs to be transmitted between devices.

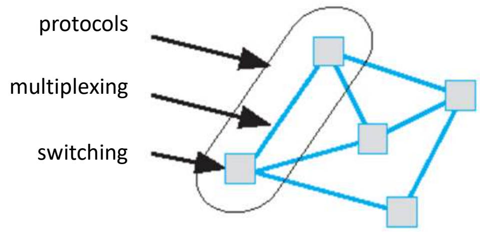


- A computer can identify only digital data consisting of 0s and 1s. It is incapable of reading other types of information, so the information needs to be translated into data by certain rules.
- However, people do not have the capability of reading electronic data. Therefore, data needs to be converted into information that can be understood by people.
- A network engineer needs to pay more attention to the end-to-end data transmission process.

Contents

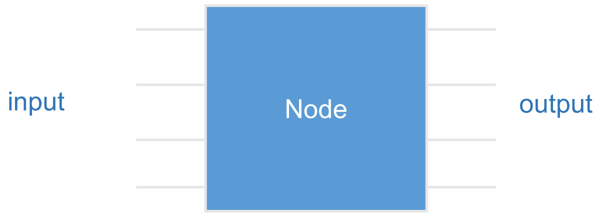
Transfer modes

- A network transfer mode specifies how data are transferred from source to destination through a network made up of links and nodes. Therefore, to describe a transfer mode, it is necessary to specify:
  - how data traverses nodes and are forwarded to destination (**switching**).
  - how and when data can be transmitted through the branches (**multiplexing techniques**).
  - how the entities in the network interact by making use of the links that interconnect them, ie the communication rules (**protocol architecture**).



Switching

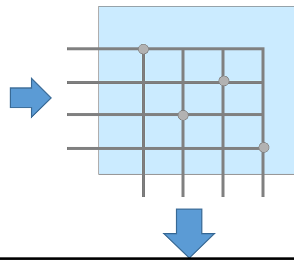
For a node of the network, the switching technique determines how a content, received through an input interface to the node, connected to an input link, is sent to one or more output interfaces, connected to the output links.

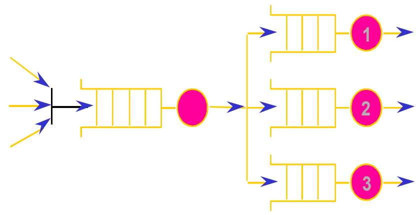


The switching technique is realized through the **routing** and **forwarding** functions

Switching

In a network node, **routing** is the decision-making function, which logically associates an input link with one or more output links to which a content received through the input branch must be **forwarded**.

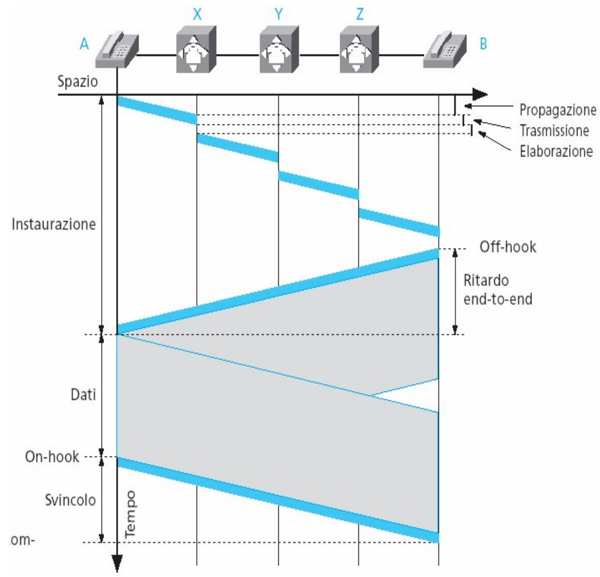




Circuit Switching

- CIRCUIT SETUP
- DATA TRANSFER
- CIRCUIT TEAR-DOWN

End-to-end pre-allocation of a **physical** communication circuit



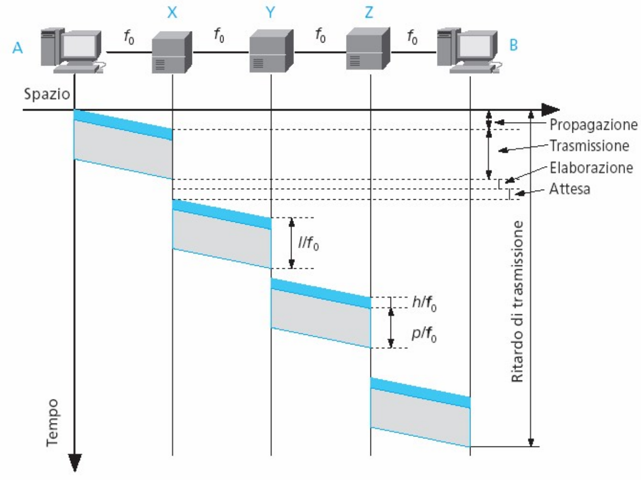
Packet Switching

Information Unit (IU)

Header	Payload

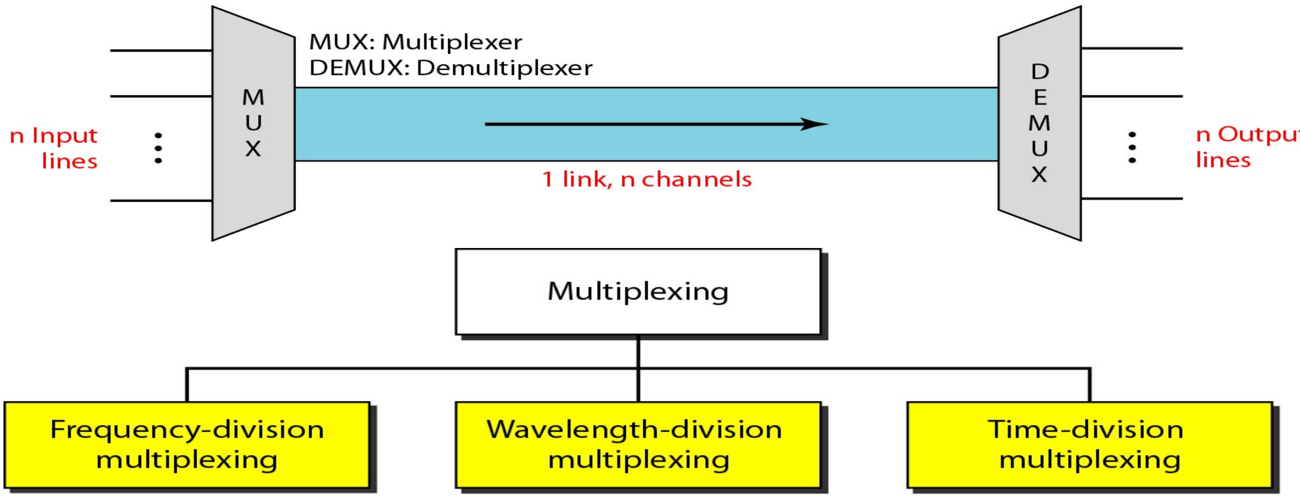
store&forward of Information Units

Possible loss of Ius and other issues



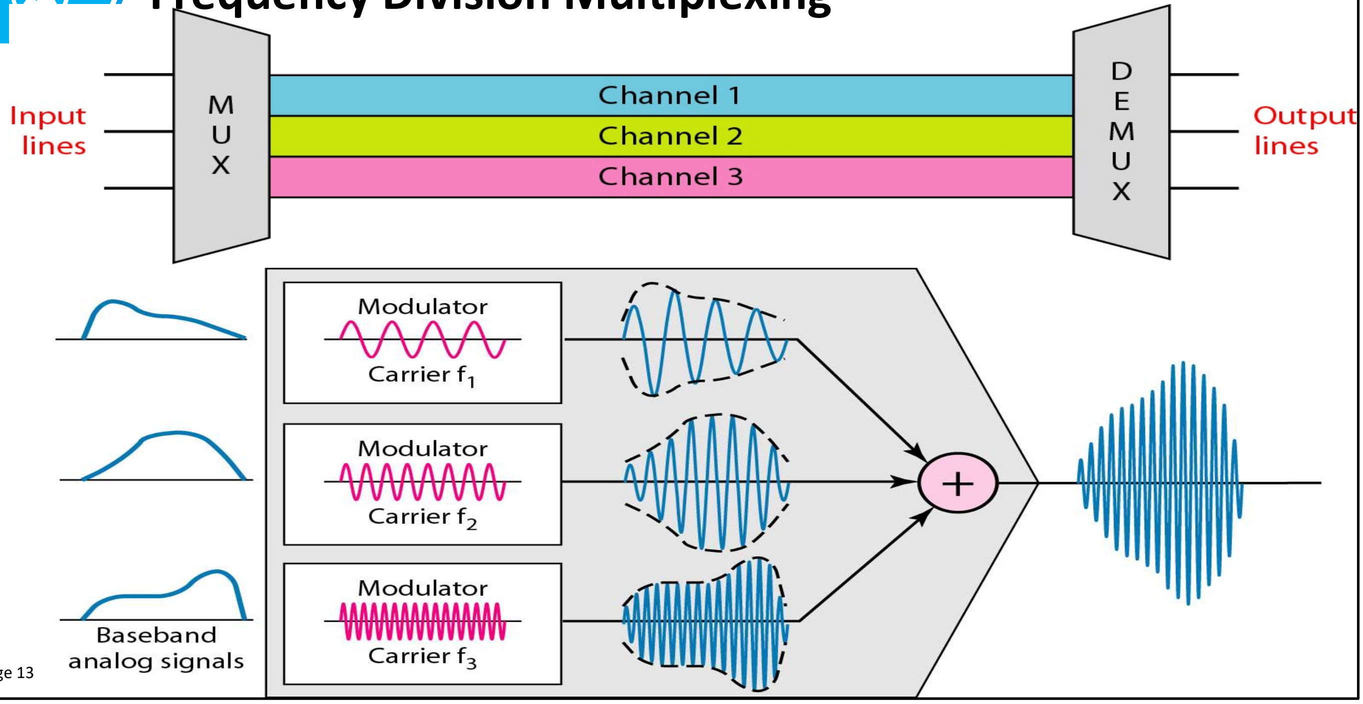
Multiplexing

- Sharing of link transfer capacity



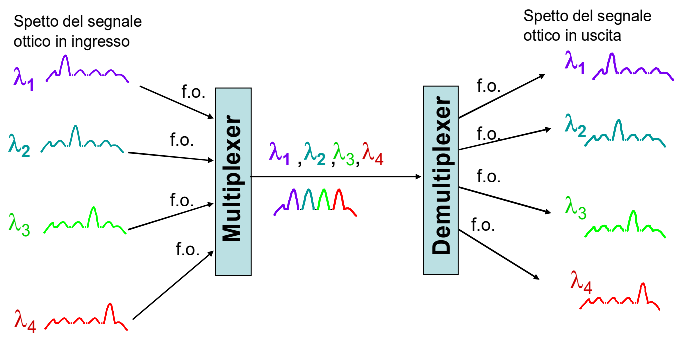
Page 12

Frequency Division Multiplexing

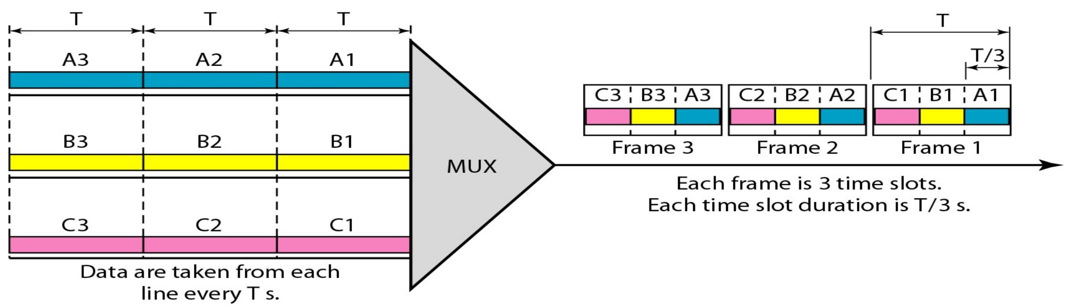


Page 13

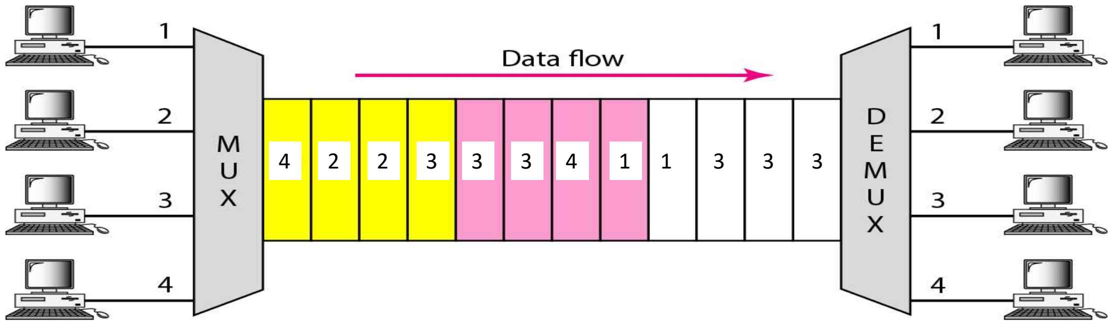
Wavelength Division Multiplexing



Time Division Multiplexing - syncronous



Time Division Multiplexing - asyncronous



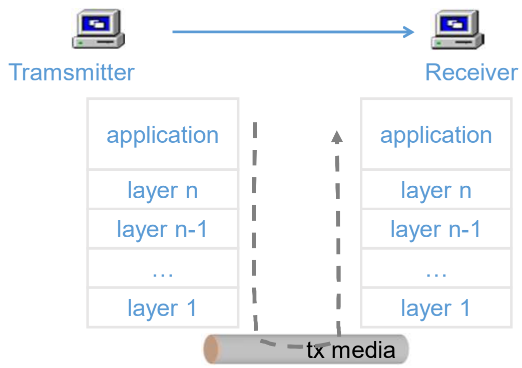
Synchronous TDM does not guarantee that the full capacity of a link is used. Because the time slots are pre-assigned, whenever a connected device is not transmitting, the corresponding slot is empty. Asynchronous time division multiplexing, or statistical time division multiplexing, is designed to avoid this kind of waste.

Protocol Architecture

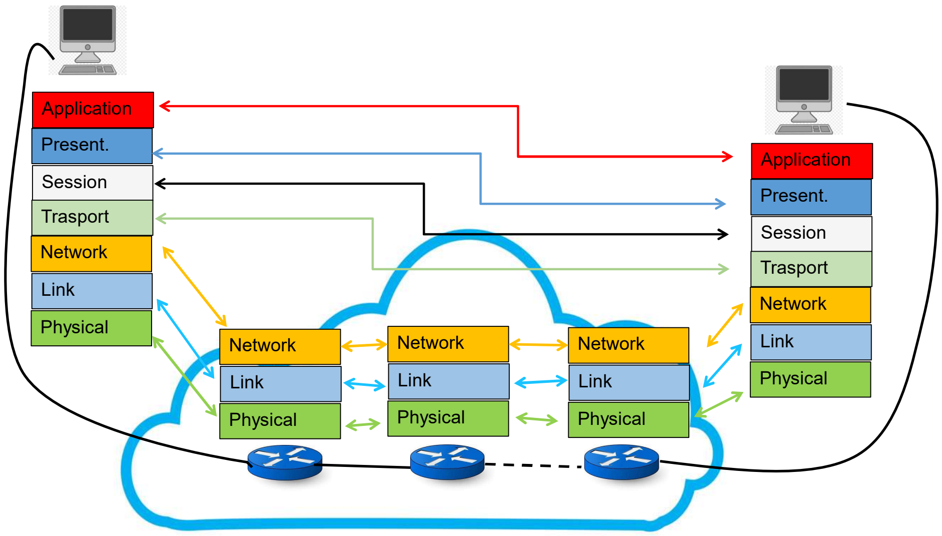
Protocol: set of rules and methods of implementation of a function or group of functions.

Communication protocol: set of rules that define the methods of interaction between systems.

Protocol Architecture: describes the stratification of the communication functions and establishes their allocation to the network equipment.



OSI Reference Model - Example

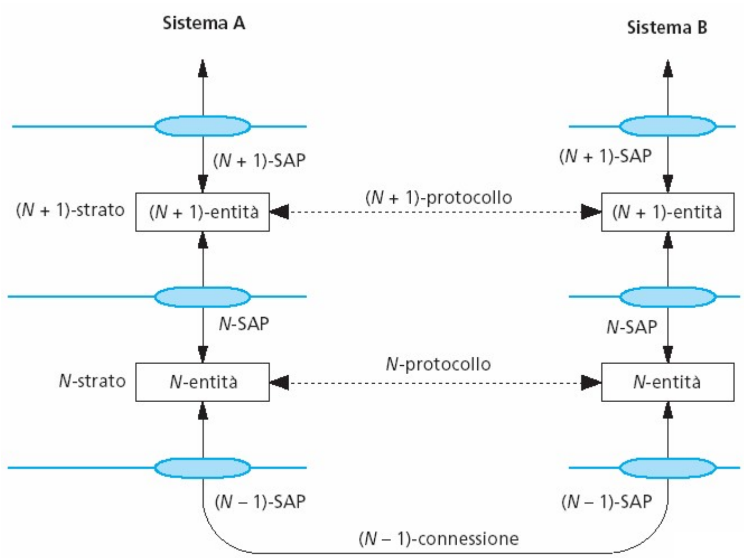


OSI Reference Model

7. Application Layer	Provides interfaces for applications.
6. Presentation Layer	Translates data formats to ensure that the application-layer data of one system can be identified by the application layer of another system.
5. Session Layer	Establishes, manages, and terminates sessions between communicating parties.
4. Transport Layer	Establishes, maintains, and cancels an end-to-end data transmission process; controls transmission speeds and adjusts data sequences.
3. Network Layer	Defines logical addresses and transfers data from sources to destinations.
2. Data Link Layer	Encapsulates packets into frames, transmits frames in P2P or P2MP mode, and implements error checking.
1. Physical Layer	Transmits bitstreams over transmission media and defines electrical and physical specifications.

- The Open Systems Interconnection Model (OSI) was included in the ISO 7489 standard and released in 1984. ISO stands for International Organization for Standardization.
- The OSI reference model is also called the seven-layer model. The seven layers from bottom to top are as follows:
  - Physical layer: transmits bit flows between devices and defines physical specifications such as electrical levels, speeds, and cable pins.
  - Data link layer: encapsulates bits into octets and octets into frames, uses MAC addresses to access media, and implements error checking.
  - Network layer: defines logical addresses for routers to determine paths and transmits data from source networks to destination networks.
  - Transport layer: implements connection-oriented and non-connection-oriented data transmission, as well as error checking before retransmission.
  - Session layer: establishes, manages, and terminates sessions between entities at the presentation layer. Communication at this layer is implemented through service requests and responses transmitted between applications on different devices.
  - Presentation layer: provides data encoding and conversion so that data sent by the application layer of one system can be identified by the application layer of another system.
  - Application layer: provides network services for applications and the OSI layer closest to end users.

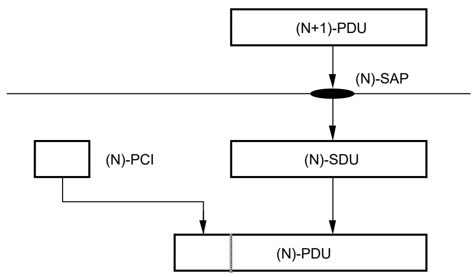
Formal definition of the protocol architecture



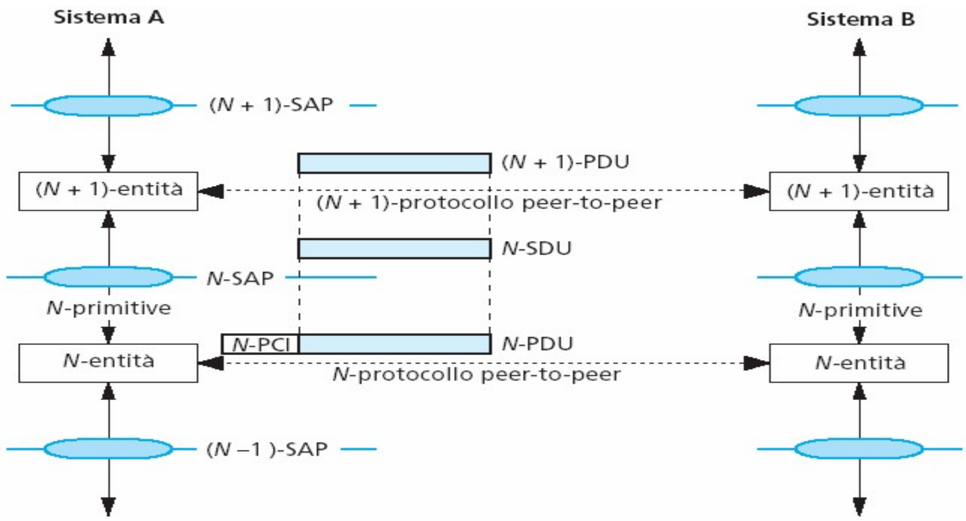
- Examples of SAPs are the type field in the Medium Access Control (MAC) protocol, the address field in HDLC, the protocol field in the IP network header, and the port identifier in UDP and TCP.

Protocol Architecture – Information Units

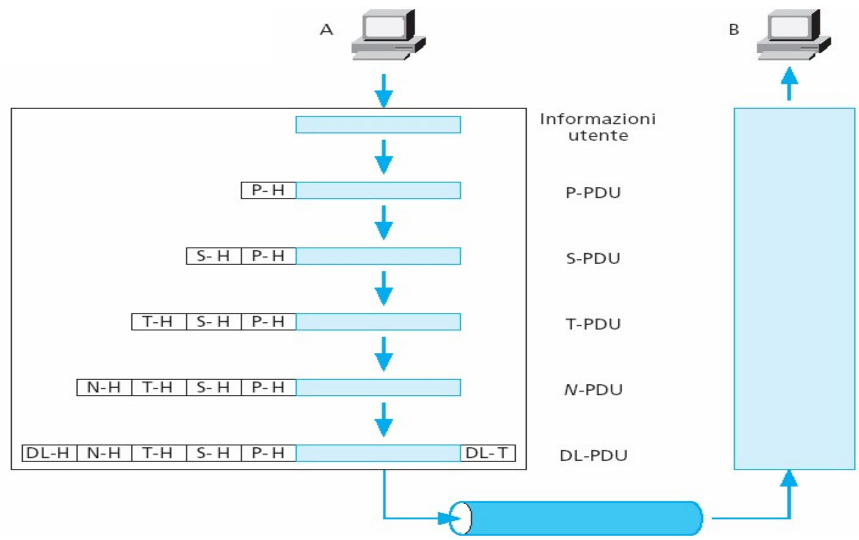
- Each layer manages Information Units, also referred to as PDU (Protocol Data Unit). Thet include a **payload** to betransferred (SDU, Service Data Unit) and a **header** (PCI, Protocol Control Information)
- A PDU of a layer is encapsulated as a PDU of the lower layer



Protocol Architecture – global view



Protocol Architecture – IU size inflation



TCP/IP Reference Model

- The OSI protocol stack is complex, and the TCP and IP protocols are widely used in the industry. Therefore, the TCP/IP reference model becomes the mainstream reference model of the Internet.

Standard TCP/IP model	OSI model	Equivalent TCP/IP model
Application Layer	Application Layer	Application Layer
	Presentation Layer	
	Session Layer	
Host-to-Host Layer	Transport Layer	Transport Layer
Internet Layer	Network Layer	Network Layer
Network Access Layer	Data Link Layer	Data Link Layer
	Physical Layer	Physical Layer

- The TCP/IP model is similar to the OSI model in structure and adopts a hierarchical architecture. Adjacent TCP/IP layers are closely related.
- The standard TCP/IP model combines the data link layer and physical layer in the OSI model into the network access layer. This division mode is contrary to the actual protocol formulation. Therefore, the equivalent TCP/IP model that integrates the TCP/IP standard model and the OSI model is proposed. Contents in the following slides are based on the equivalent TCP/IP model.

Common TCP/IP Protocols

- The TCP/IP protocol stack defines a series of standard protocols.

Application Layer	Telnet	FTP	TFTP	SNMP
	HTTP	SMTP	DNS	DHCP
Transport Layer	TCP		UDP	
Network Layer	ICMP		IGMP	
	IP			
Data Link Layer	PPPoE			
	Ethernet		PPP	
Physical Layer	...			

- Application Layer
  - Hypertext Transfer Protocol (HTTP): is used to access various pages on web servers.
  - File Transfer Protocol (FTP): provides a method for transferring files. It allows data to be transferred from one host to another.
  - Domain name service (DNS): translates from host domain names to IP addresses.
- Transport layer
  - Transmission Control Protocol (TCP): provides reliable connection-oriented communication services for applications. Currently, TCP is used by many popular applications.
  - User Datagram Protocol (UDP): provides connectionless communication and does not guarantee the reliability of packet transmission. The reliability can be ensured by the application layer.
- Network layer
  - Internet Protocol (IP): encapsulates transport-layer data into data packets and forwards packets from source sites to destination sites. IP provides a connectionless and unreliable service.
  - Internet Group Management Protocol (IGMP): manages multicast group memberships. Specifically, IGMP sets up and maintains memberships between IP hosts and their directly connected multicast routers.
  - Internet Control Message Protocol (ICMP): sends control messages based on the IP protocol and provides information about various problems that may exist in the communication environment. Such information helps administrators diagnose problems and take proper measures to resolve the problems.
- Data link layer

- Point-to-Point Protocol (PPP): is a data link layer protocol that works in point-to-point mode. PPP is mainly used on wide area networks (WANs).
- Ethernet: is a multi-access and broadcast protocol at the data link layer, which is the most widely used local area network (LAN) technology.
- Point-to-Point Protocol over Ethernet (PPPoE): connects multiple hosts on a network to a remote access concentrator through a simple bridge device (access device). Common applications include home broadband dialup access.

Common Protocol Standardization Organizations

- Internet Engineering Task Force (IETF)
  - IETF is a voluntary organization responsible for developing and promoting Internet protocols (especially protocols that constitute the TCP/IP protocol suite), and releasing new or replacing old protocol standards through RFCs.
- Institute of Electrical and Electronics Engineers (IEEE)
  - IEEE has formulated about 30% of standards in the electronics, electrical, and computer science fields worldwide. Those standards include well-known IEEE802.3 (Ethernet) and IEEE802.11 (Wi-Fi).
- International Organization for Standardization (ISO)
  - ISO is an international organization that plays an important role in the formulation of computer network standards, such as the OSI model defined in ISO/IEC 7498-1.

Application Layer

- The application layer provides interfaces for application software so that applications can use network services. The application layer protocol designates transport layer protocols and ports.
- PDUs transmitted at the network layer are called data.

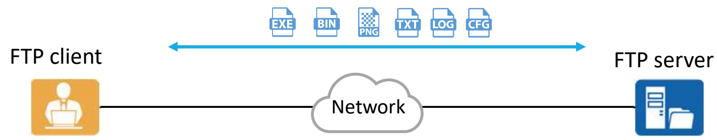
Application Layer (Data)
Transport Layer
Network Layer
Data Link Layer
Physical Layer

- HTTP 80 (TCP) (HTTPS 443)
  Hypertext transfer protocol, which provides web browsing services.
- Telnet 23 (TCP)
  Remote login protocol, which provides remote management services.
- FTP 20 and 21 (TCP)
  File transfer protocol, which provides Internet file resource sharing services.
- SMTP 25 (TCP)
  Simple mail transfer protocol, which provides Internet email services.
- TFTP 69 (UDP)
  Simple file transfer protocol, which provides simple file transfer services.

- The TCP/IP suite enables data to be transmitted over a network. The layers use packet data units (PDUs) to exchange data, implementing communication between network devices.
- PDUs transmitted at different layers contain different information. Therefore, PDUs have different names at different layers.

Common Application Layer Protocols - FTP

- The File Transfer Protocol (FTP) transfers files from one host to another to implement file download and upload. This protocol adopts the client/server (C/S) structure.

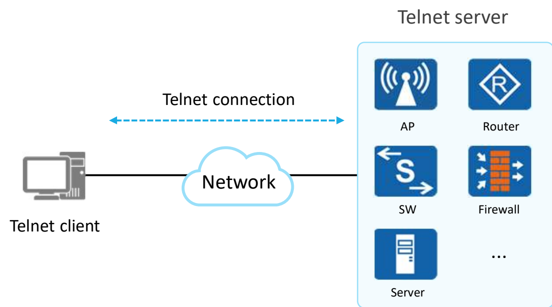


FTP client: provides commands for local users to operate files on a remote server. A user can install an FTP client program on a PC and set up a connection with an FTP server to operate files on the server.

FTP server: a device that runs the FTP service. It provides the access and operation functions for remote clients, allowing users to access the FTP server through the FTP client program and access files on the server.

Common Application Layer Protocols - Telnet

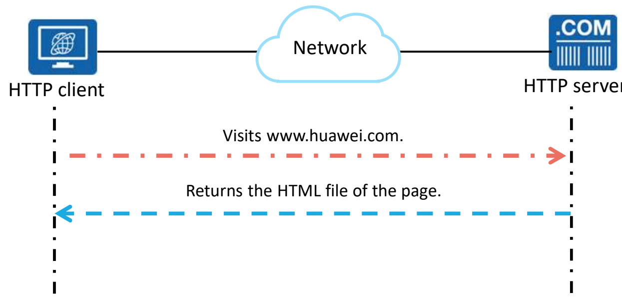
- Telnet is a standard protocol that provides remote login services on a network. It provides users with the ability to operate remote devices through local PCs.



A user connects to a Telnet server through the Telnet client program. The commands entered on the Telnet client are executed on the server, as if the commands were entered on the console of the server.

Common Application Layer Protocols - HTTP

- Hypertext Transfer Protocol (HTTP): is one of the most widely used network protocols on the Internet. HTTP was originally designed to provide a method for publishing and receiving HTML pages.



Transport Layer

- A transport layer protocol receives data from an application layer protocol, encapsulates the data with the corresponding transport layer protocol header, and helps establish an end-to-end (port-to-port) connection.
- PDUs transmitted at the transport layer are called segments.

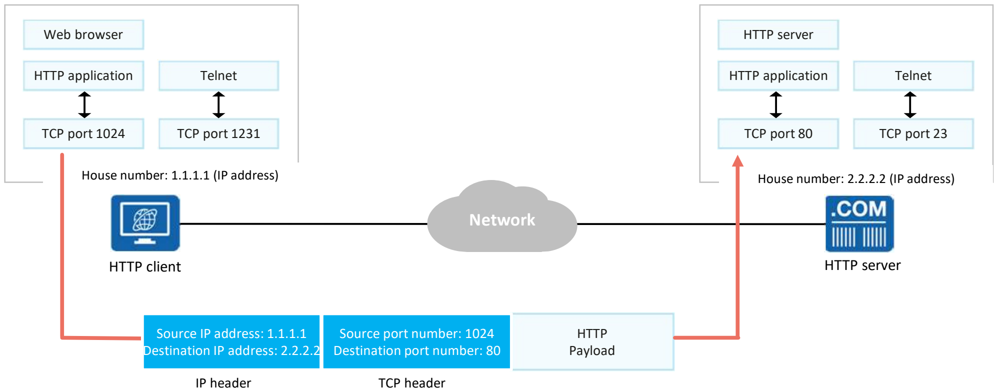
Application Layer
Transport Layer (Segment)
Network Layer
Data Link Layer
Physical Layer

Transport layer protocols:

TCP: a connection-oriented reliable protocol defined by IETF in RFC 793.

UDP: a simple connectionless protocol defined by IETF in RFC 768.

TCP and UDP - Port Numbers



- Generally, the source port used by a client is randomly allocated, and the destination port is specified by the application of a server.
- The system generally selects a source port number that is greater than 1023 and is not being used.
- The destination port number is the listening port of the application (service) enabled on the server. For example, the default port number for HTTP is 80.

Application Layer | Transport Layer | Network Layer | Data Link Layer | Physical Layer

TCP and UDP - Header Formats

C	E	U	A	P	R	S	F
W	C	R	C	S	S	Y	I
R	E	G	K	H	T	N	N

TCP header 20 bytes

<table>
<tr><td colspan="3">Source Port (16)</td><td>Destination Port (16)</td></tr>
<tr><td colspan="4">Sequence Number (32)</td></tr>
<tr><td colspan="4">Acknowledgement Number (32)</td></tr>
<tr><td>Header Length (4)</td><td>Reserved (6)</td><td>Control Bits (6)</td><td>Window (16)</td></tr>
<tr><td colspan="3">Checksum (16)</td><td>Urgent (16)</td></tr>
<tr><td colspan="4">Options</td></tr>
<tr><td colspan="4">Data (varies)</td></tr>
</table>

UDP header 8 bytes

<table>
<tr><td>Source Port (16)</td><td>Destination Port (16)</td></tr>
<tr><td>Length (16)</td><td>Checksum (16)</td></tr>
<tr><td colspan="2">Data (if any)</td></tr>
</table>

- TCP header:
  - Source Port: identifies the application that sends the segment. This field is 16 bits long.
  - Destination Port: identifies the application that receives the segment. This field is 16 bits long.
  - Sequence Number: Every byte of data sent over a TCP connection has a sequence number. The value of the Sequence Number field equals the sequence number of the first byte in a sent segment. This field is 32 bits long.
  - Acknowledgment Number: indicates the sequence number of the next segment's first byte that the receiver is expecting to receive. The value of this field is 1 plus the sequence number of the last byte in the previous segment that is successfully received. This field is valid only when the ACK flag is set. This field is 32 bits long.
  - Header Length: indicates the length of the TCP header. The unit is 32 bits (4 bytes). If there is no option content, the value of this field is 5, indicating that the header contains 20 bytes.
  - Reserved: This field is reserved and must be set to 0. This field is 6 bits long.
  - Control Bits: control bits, includes FIN, ACK, and SYN flags, indicating TCP data segments in different states.
  - Window: used for TCP flow control. The value is the maximum number of bytes that are allowed by the receiver. The maximum window size is 65535 bytes. This field is 16 bits long.
  - Checksum: a mandatory field. It is calculated and stored by the sender and verified by the receiver. During checksum computation, the TCP header and TCP data are included, and a 12-byte pseudo header is added before the TCP segment. This field is 16 bits long.
  - Urgent: indicates the urgent pointer. The urgent pointer is valid only when the URG flag is set. The Urgent field indicates that the sender transmits data in emergency mode. The

urgent pointer indicates the number of urgent data bytes in a segment (urgent data is placed at the beginning of the segment). This field is 16 bits long.

  - Options: This field is optional. This field is 0 to 40 bytes long.

- UDP header:
  - Source Port: identifies the application that sends the segment. This field is 16 bits long.
  - Destination Port: identifies the application that receives the segment. This field is 16 bits long.
  - Length: specifies the total length of the UDP header and data. The possible minimum length is 8 bytes because the UDP header already occupies 8 bytes. Due to the existence of this field, the total length of a UDP segment does not exceed 65535 bytes (including an 8-byte header and 65527-byte data).
  - Checksum: checksum of the UDP header and UDP data. This field is 16 bits long.

- **CWR** (Congestion Window Reduced) - se impostato a 1 indica che l'host sorgente ha ricevuto un segmento TCP con il flag ECE impostato a 1 (aggiunto all'header in RFC 3168).
- **ECE** [ECN (Explicit Congestion Notification) -Echo] - se impostato a 1 indica che l'host supporta ECN durante il 3-way handshake (aggiunto all'header in RFC 3168).
- **URG** - se impostato a 1 indica che nel flusso sono presenti *dati urgenti* alla posizione (offset) indicata dal campo *Urgent pointer*. *Urgent Pointer* punta alla fine dei dati urgenti;
- **ACK** - se impostato a 1 indica che il campo *Acknowledgment number* è valido;
- **PSH** - se impostato a 1 indica che i dati in arrivo non devono essere bufferizzati ma passati subito ai livelli superiori dell'applicazione;
- **RST** - se impostato a 1 indica che la connessione non è valida; viene utilizzato in caso di grave errore; a volte utilizzato insieme al flag ACK per la chiusura di una connessione.
- **SYN** - se impostato a 1 indica che l'host mittente del segmento vuole *aprire una connessione TCP* con l'host destinatario e specifica nel campo *Sequence number* il valore dell'Initial Sequence Number (*ISN*); ha lo scopo di sincronizzare i numeri di sequenza dei due host. L'host che ha inviato il SYN deve attendere dall'host remoto un pacchetto SYN/ACK.
- **FIN** - se impostato a 1 indica che l'host mittente del segmento vuole *chiudere la connessione TCP* aperta con l'host destinatario. Il mittente attende la conferma dal

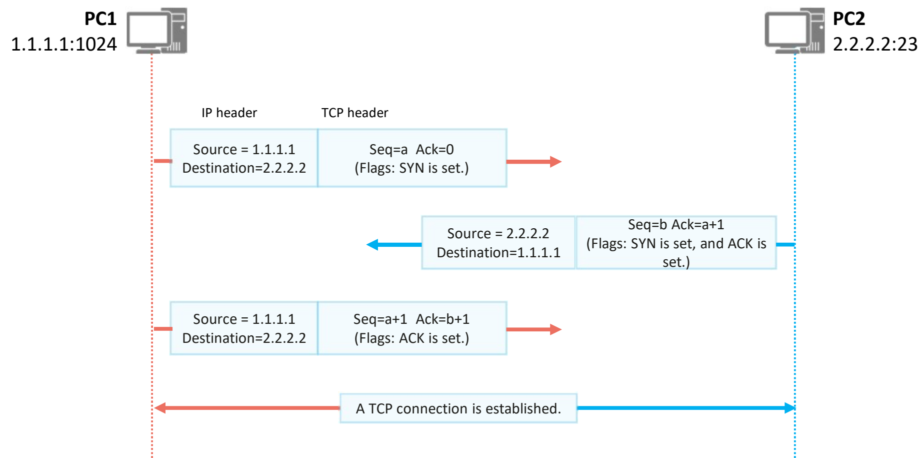
ricevente (con un FIN-ACK). A questo punto la connessione è ritenuta chiusa per metà: l'host che ha inviato FIN non potrà più inviare dati, mentre l'altro host ha il canale di comunicazione ancora disponibile. Quando anche l'altro host invierà il pacchetto con FIN impostato, la connessione, dopo il relativo FIN-ACK, sarà considerata completamente chiusa.

- **Managing Congestion**
- Enter RFC2481 “A Proposal to Add Explicit Congestion Notification (ECN) to IP” which was superseded by RFC3168 with the same name. To assist in notifying the end-points, changes were made to the TCP and IP headers. First, two one-bit flags were added to the reserved field of the TCP header: bit 8 (CWR – Congestion Window Reduced) and bit 9 (ECE – ECN-Echo). Lastly, two flags were changed in the IP header in the differentiated services field: bit 14 (ECT – ECN Capable Transport) and 15 (CE – Congestion Experienced).
- So how does it work? To accomplish this, the sender sends a SYN packet with the ECE and CWR flags set, and the receiver sends back the SYN-ACK with only the ECE flag set. Any other configuration indicates a non-ECN setup. Assuming an ECN-aware network, an oversimplification of the control process looks like this: when a router detects congestion, rather than dropping packets destined to a receiver, it marks them with the CE flag in the IP header and delivers the packet to the receiver. Prior to acknowledging the receipt of the packet, the receiver sets the ECE flag in the TCP header of the ACK and sends it back to the sender. The sender having received the ECE marked ACK responds by halving the send window and reducing the slow start threshold.

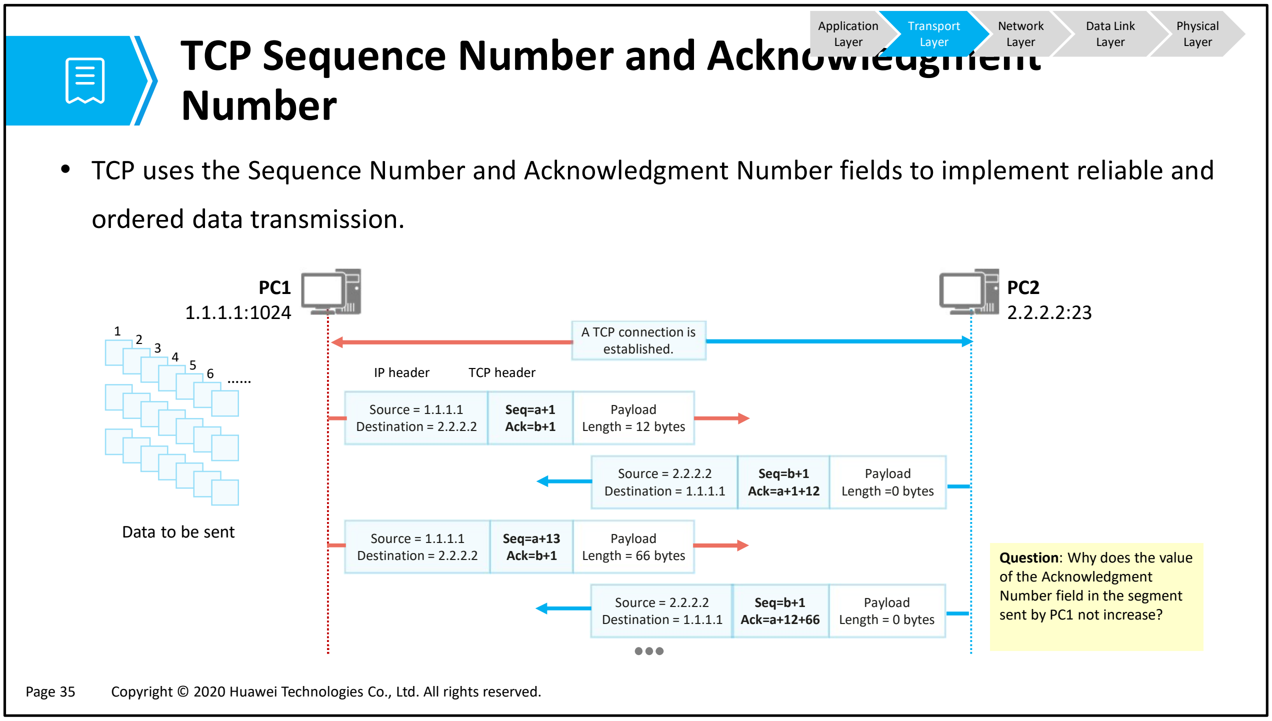
Application Layer | Transport Layer | Network Layer | Data Link Layer | Physical Layer

TCP Connection Setup - Three-Way Handshake

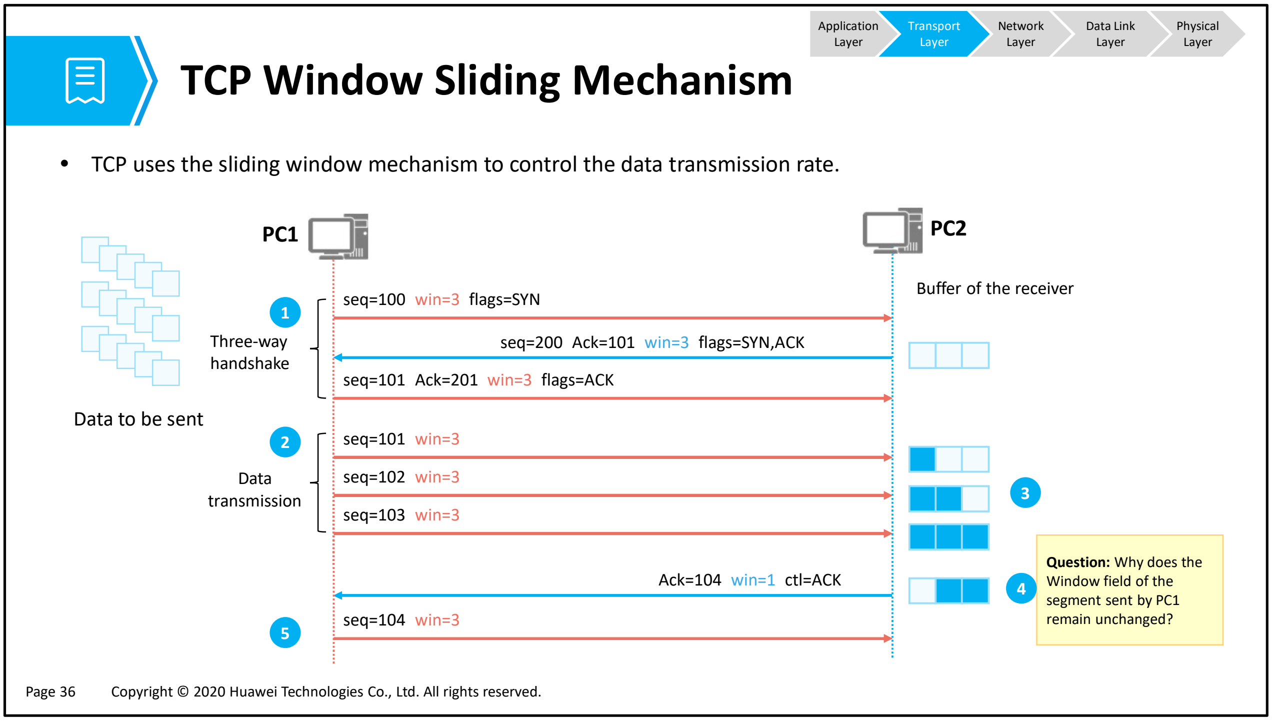
- Before sending data, a TCP-based application needs to establish a connection through three-way handshake.



- The TCP connection setup process is as follows:
  - The TCP connection initiator (PC1 in the figure) sends the first TCP segment with SYN being set. The initial sequence number a is a randomly generated number. The acknowledgment number is 0 because no segment has ever been received from PC2.
  - After receiving a valid TCP segment with the SYN flag being set, the receiver (PC2) replies with a TCP segment with SYN and ACK being set. The initial sequence number b is a randomly generated number. Because the segment is a response one to PC1, the acknowledgment number is a+1.
  - After receiving the TCP segment in which SYN and ACK are set, PC1 replies with a segment in which ACK is set, the sequence number is a+1, and the acknowledgment number is b+1. After PC2 receives the segment, a TCP connection is established.

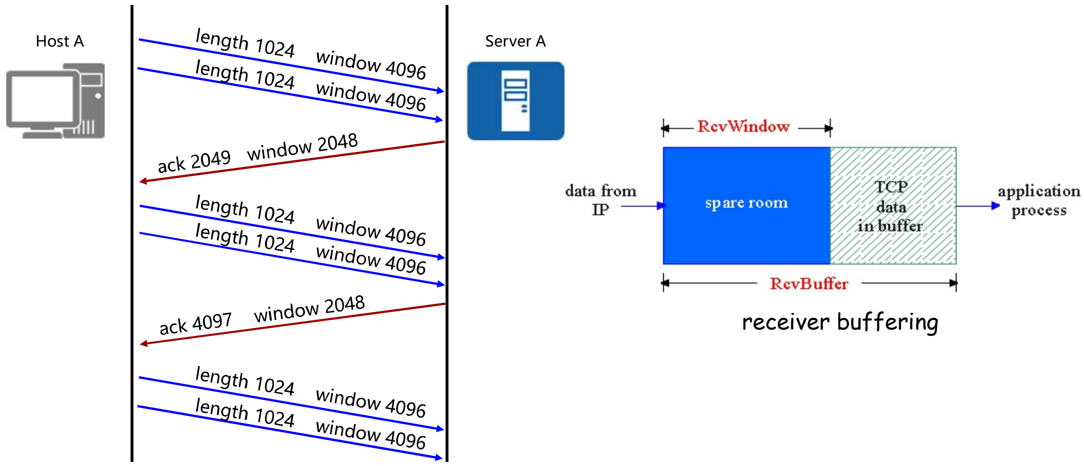


- Assume that PC1 needs to send segments of data to PC2. The transmission process is as follows:
  1. PC1 numbers each byte to be sent by TCP. Assume that the number of the first byte is a+1. Then, the number of the second byte is a+2, the number of the third byte is a+3, and so on.
  2. PC1 uses the number of the first byte of each segment of data as the sequence number and sends out the TCP segment.
  3. After receiving the TCP segment from PC1, PC2 needs to acknowledge the segment and request the next segment of data. How is the next segment of data determined? Sequence number (a+1) + Payload length = Sequence number of the first byte of the next segment (a+1+12)
  4. After receiving the TCP segment sent by PC2, PC1 finds that the acknowledgment number is a+1+12, indicating that the segments from a+1 to a+12 have been received and the sequence number of the upcoming segment to be sent should be a+1+12.
- To improve the sending efficiency, multiple segments of data can be sent at a time by the sender and then acknowledged at a time by the receiver.



1. During the TCP three-way handshake, both ends notify each other of the maximum number of bytes (buffer size) that can be received by the local end through the Window field.
2. After the TCP connection is set up, the sender sends data of the specified number of bytes based on the window size declared by the receiver.
3. After receiving the data, the receiver stores the data in the buffer and waits for the upper-layer application to obtain the buffered data. After the data is obtained by the upper-layer application, the corresponding buffer space is released.
4. The receiver notifies the current acceptable data size (window) according to its buffer size.
5. The sender sends a certain amount of data based on the current window size of the receiver.

TCP Flow Control



- The TCP window field provides a means of flow control that governs the amount of data sent by the sender. This is achieved by returning a "window" with every TCP segment for which the ACK field is set, indicating a range of acceptable sequence numbers beyond the last segment successfully received. The window indicates the permitted number of octets that the sender may transmit before receiving further permission.
- In the example, TCP transmission from host A to server A contains the current window size for host A. The window size for server A is determined as part of the handshake, which based on the transmission can be assumed as 2048. Once data equivalent to the window size has been received, an acknowledgement will be returned, relative to the number of bytes received, plus one. Following this, host A will proceed to transmit the next batch of data.
- A TCP window size of 0 will effectively deny processing of segments, with exception to segments where the ACK, RST and URG code bits are set for incoming segments. Where a window size of 0 exists, the sender must still periodically check the window size status of receiving TCP to ensure any change in the window size is effectively reported, the period for retransmission is generally two minutes. When a sender sends periodic segments, the receiving TCP must still acknowledge with a sequence number announcement of the current window size of 0.

TCP Round Trip Delay e Timeout (1/2)

- How to estimate RTD ?
- To sample the Round Trip Delay (`sampleRTD`): time from segment transmission to ack reception
  - Pay attention to retransmission and cumulative acks
- Estimation of the `sampleRTD` mean

TCP Round Trip Delay e Timeout (2/2)

```
EstimatedRTD = (1-x)*EstimatedRTD + x*SampleRTD
```

- Typical x values in the range 0.1 - 0.125

```
Timeout = EstimatedRTD + 4*Deviation
```

- Deviation = (1-x)*Deviation +x*|SampleRTD-EstimatedRTD|

TCP transmission rate

- Two processes determine:
  - Congestion Control (Congwin)
  - Flow control (RcvWindow)

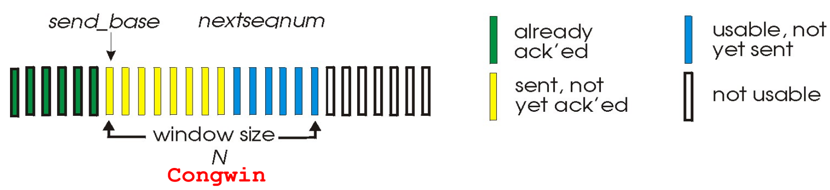
```
TrWindow = min (Congwin, RcvWindow)
```

➢ **TrWindow bytes can be sent during 1 RTD →**

$$\text{max througput} = \frac{\text{TrWindow}}{\text{RTD}} \quad \text{byte/sec}$$

Congestion Control

- The transmission rate il determined by the TrWindow size. Assume TrWindow=`Congwin < RcvWindow`:



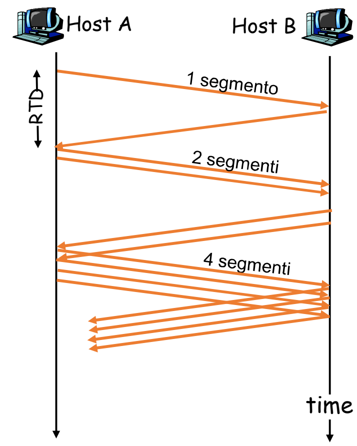
Congestion Control

- A time-out event is considered due to a congestion event
- Algorithms used in differnet TCP versions:
  - Slow start +Congestion avoidance
  - Fast restransmit + Fast recovery
- "Test" the network to discover the usable bandwidth
- Two main phases
  - slow start
  - congestion avoidance
- Used varaibles:
  - `Congwin`
  - `threshold:` between slow start phase and congestion control phase

TCP Slowstart

Slowstart Algorithm

initialization: `Congwin= 1`
for (each acked segment)
```
    Congwin++
```
until (loss event) OR
```
    (CongWin > threshold)
```

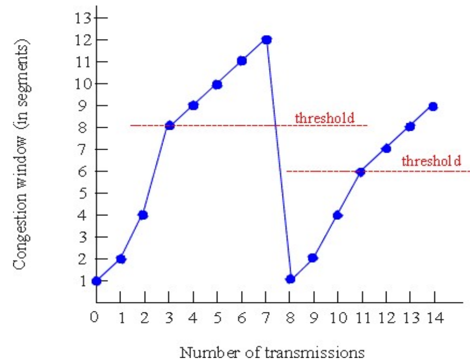


- Exponential growth of the window size for each RTD
- Loss event: timeout (Tahoe TCP) and/or 3 duplicated ACKs (Reno TCP)

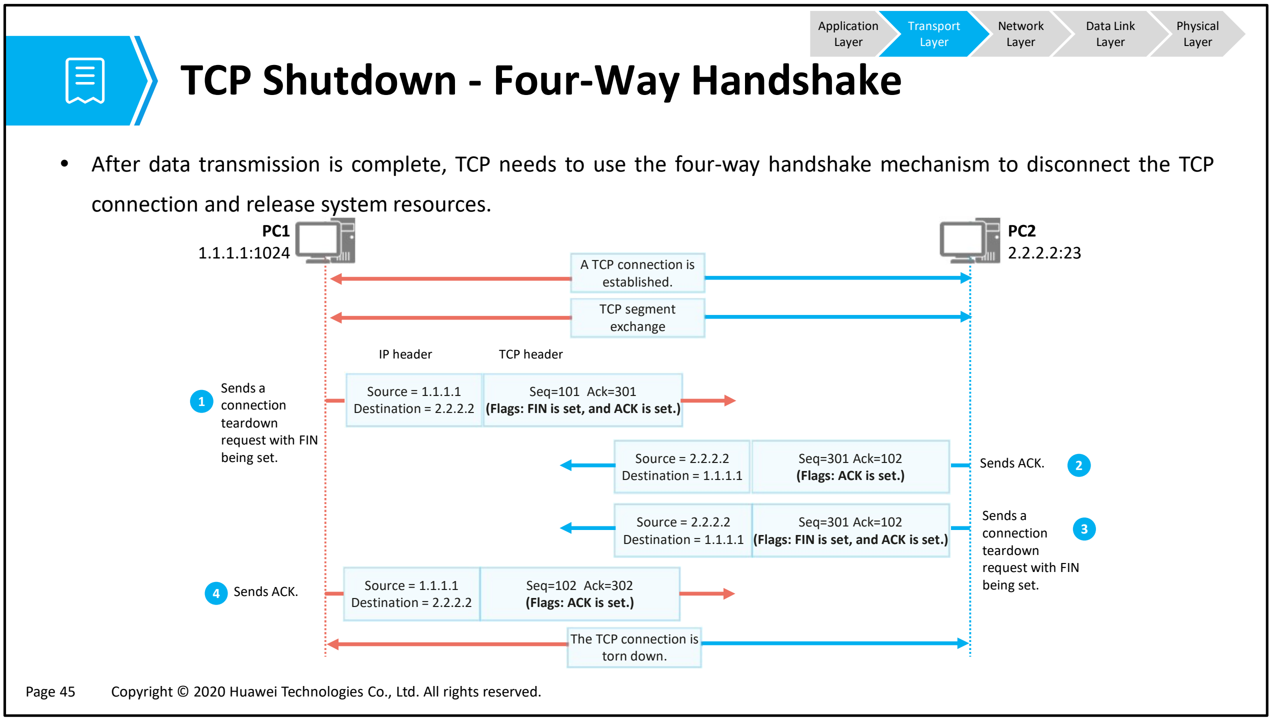
TCP Congestion Avoidance

Congestion avoidance

```
/* Slow Start ended*/
/* Congwin > threshold */
Until (loss event) {
  each w acked segments :
    Congwin++
  }
threshold = Congwin/2
Congwin = 1
Start slowstart
```



The TCP Reno does not use slowstart (fast recovery) in case of 3 duplicated ACKs: fast retransmit => start the procedure before



- TCP supports data transmission in full-duplex mode, which means that data can be transmitted in both directions at the same time. Before data is transmitted, TCP sets up a connection in both directions through three-way handshake. Therefore, after data transmission is complete, the connection must be closed in both directions. This is shown in the figure.
  1. PC1 sends a TCP segment with FIN being set. The segment does not carry data.
  2. After receiving the TCP segment from PC1, PC2 replies with a TCP segment with ACK being set.
  3. PC2 checks whether data needs to be sent. If so, PC2 sends the data, and then a TCP segment with FIN being set to close the connection. Otherwise, PC2 directly sends a TCP segment with FIN being set.
  4. After receiving the TCP segment with FIN being set, PC1 replies with an ACK segment. The TCP connection is then torn down in both directions.

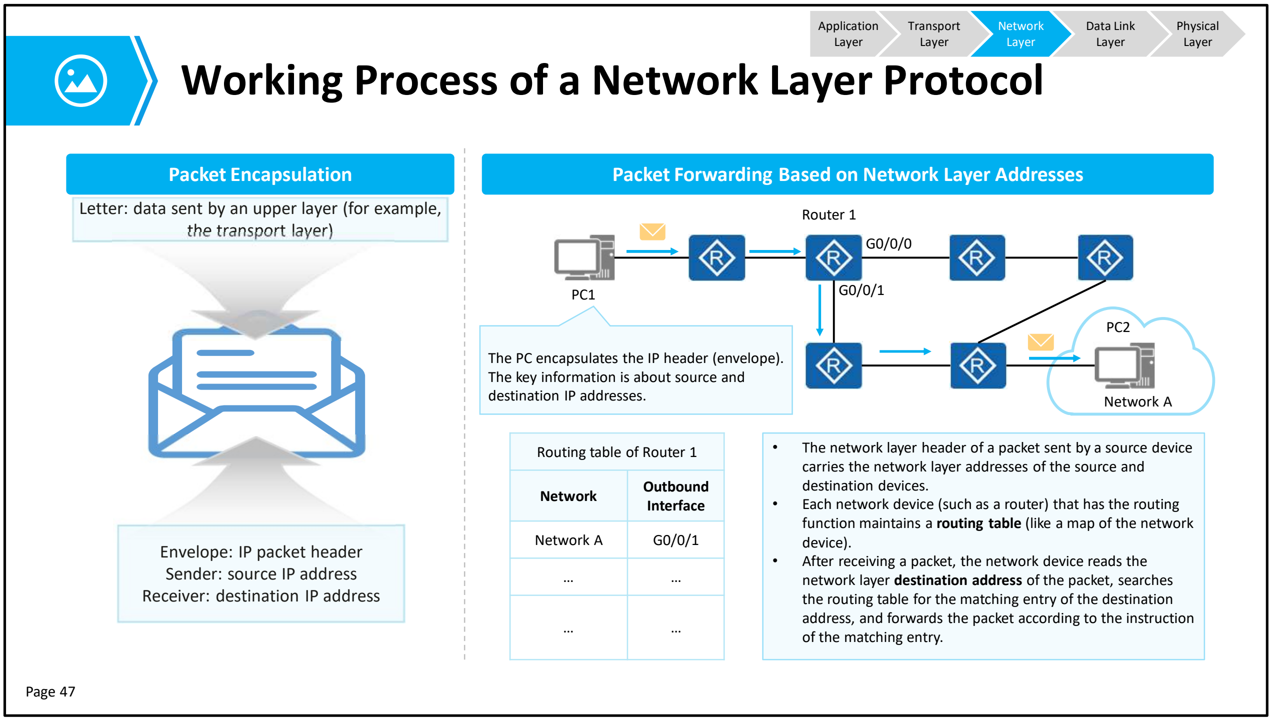
Network Layer

- The transport layer is responsible for establishing connections between processes on hosts, and the network layer is responsible for transmitting data from one host to another.
- PDUs transmitted at the network layer are called packets.

Application Layer
Transport Layer
Network Layer (Packet)
Data Link Layer
Physical Layer

- The network layer is also called the Internet layer.
  - It sends packets from source hosts to destination hosts.
- Functions of the network layer:
  - Provides logical addresses for network devices.
  - Routes and forwards data packets.
  - Common network layer protocols include IPv4, IPv6, ICMP, and IGMP.

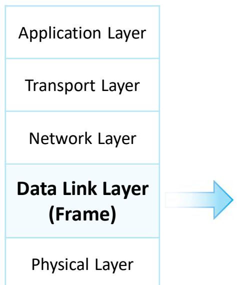
- Internet Protocol Version 4 (IPv4) is the most widely used network layer protocol.



- When IP is used as the network layer protocol, both communication parties are assigned a unique IP address to identify themselves. An IP address can be written as a 32-bit binary integer. To facilitate reading and analysis, an IP address is usually represented in dot-decimal notation, consisting of four decimal numbers, each ranging from 0 to 255, separated by dots, such as, 192.168.1.1.
- Encapsulation and forwarding of IP data packets:
  - When receiving data from an upper layer (such as the transport layer), the network layer encapsulates an IP packet header and adds the source and destination IP addresses to the header.
  - Each intermediate network device (such as a router) maintains a routing table that guides IP packet forwarding like a map. After receiving a packet, the intermediate network device reads the destination address of the packet, searches the local routing table for a matching entry, and forwards the IP packet according to the instruction of the matching entry.
  - When the IP packet reaches the destination host, the destination host determines whether to accept the packet based on the destination IP address and then processes the packet accordingly.
- When the IP protocol is running, routing protocols such as OSPF, IS-IS, and BGP are required to help routers build routing tables, and ICMP is required to help control networks and diagnose network status.

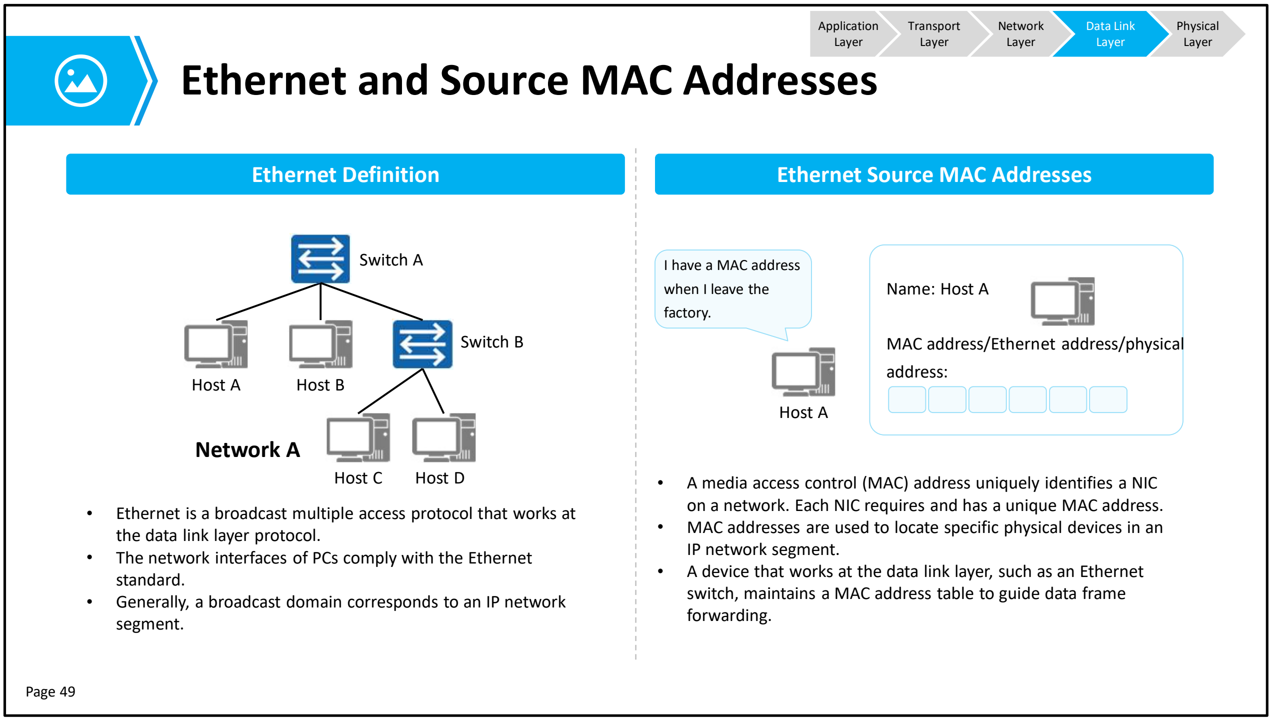
Data Link Layer

- The data link layer is located between the network layer and the physical layer and provides services for protocols such as IP and IPv6 at the network layer. PDUs transmitted at the data link layer are called frames.
- Ethernet is the most common data link layer protocol.



The data link layer is located between the network layer and the physical layer.

- The data link layer provides intra-segment communication for the network layer.
- The functions of the data link layer include framing, physical addressing, and error control.
- Common data link layer protocols include Ethernet, PPPoE, and PPP.



- A MAC address is recognizable as six groups of two hexadecimal digits, separated by hyphens, colons, or without a separator. Example: 48-A4-72-1C-8F-4F

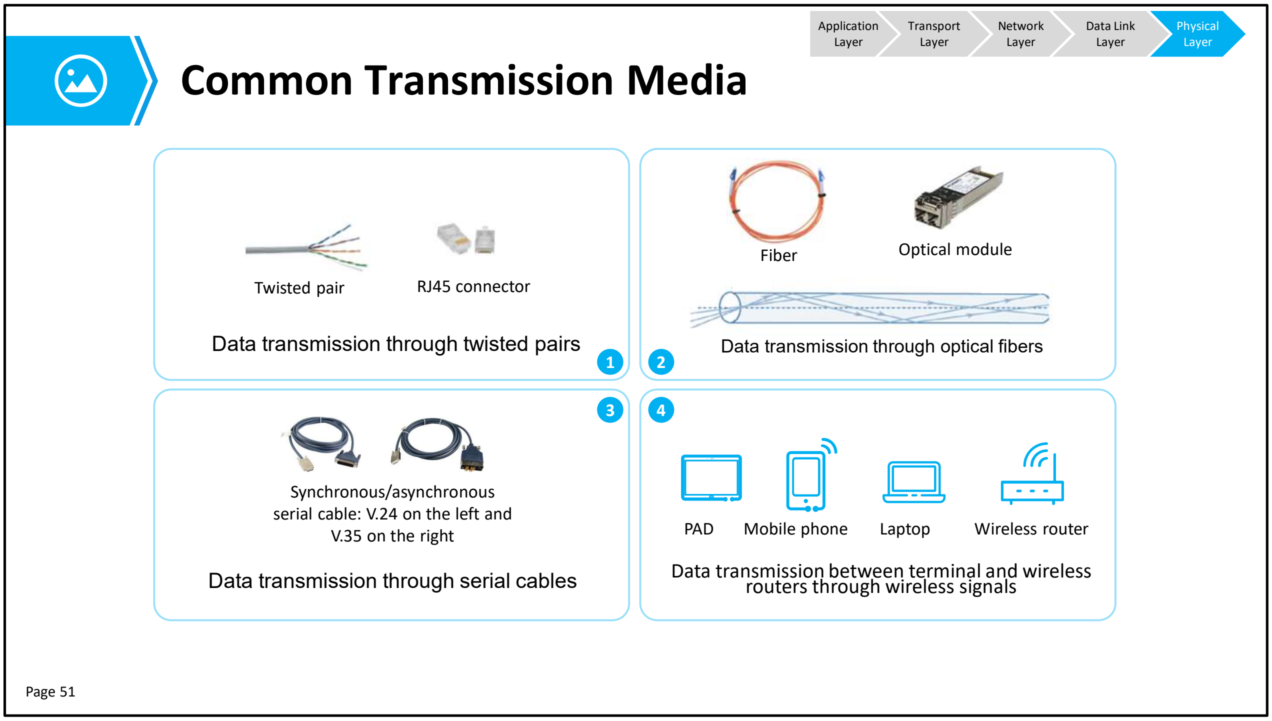
Physical Layer

- After data arrives at the physical layer, the physical layer converts a digital signal into an optical signal, an electrical signal, or an electromagnetic wave signal based on the physical media.
- PDUs transmitted at the physical layer are called bitstreams.

Application Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer (Bitstream)

The physical layer is at the bottom of the model.

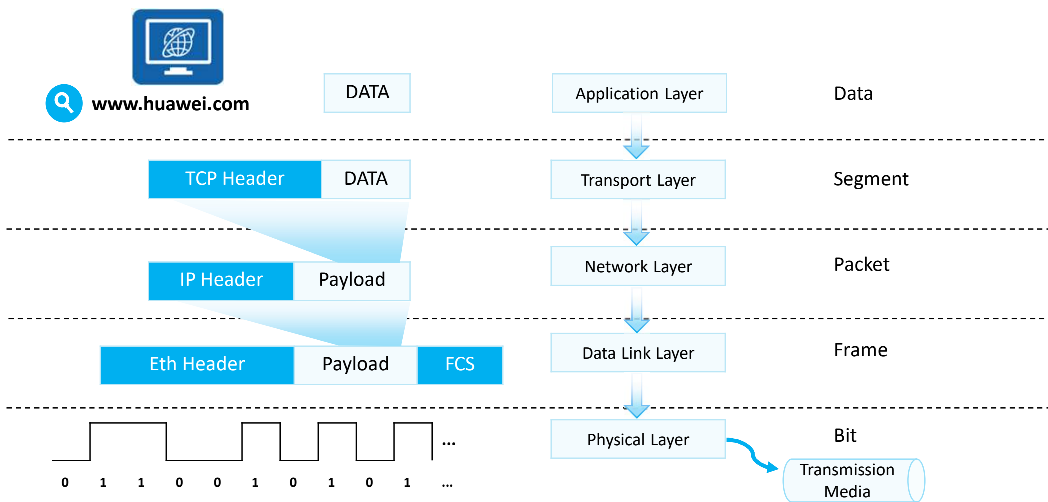
- This layer transmits bitstreams on media.
- It standardizes physical features such as cables, pins, voltages, and interfaces.
- Common transmission media include twisted pairs, optical fibers, and electromagnetic waves.



- Twisted pairs: most common transmission media used on Ethernet networks. Twisted pairs can be classified into the following types based on their anti-electromagnetic interference capabilities:
  - STP: shielded twisted pairs
  - UTP: unshielded twisted pairs
- Optical fiber transmission can be classified into the following types based on functional components:
  - Fibers: optical transmission media, which are glass fibers, used to restrict optical transmission channels.
  - Optical modules: convert electrical signals into optical signals to generate optical signals.
- Serial cables are widely used on wide area networks (WANs). The types of interfaces connected to serial cables vary according to WAN line types. The interfaces include synchronous/synchronous serial interfaces, ATM interfaces, POS interfaces, and CE1/PRI interfaces.
- Wireless signals may be transmitted by using electromagnetic waves. For example, a wireless router modulates data and sends the data by using electromagnetic waves, and a wireless network interface card of a mobile terminal demodulates the electromagnetic waves to obtain data. Data transmission from the wireless router to the mobile terminal is then complete.

Contents

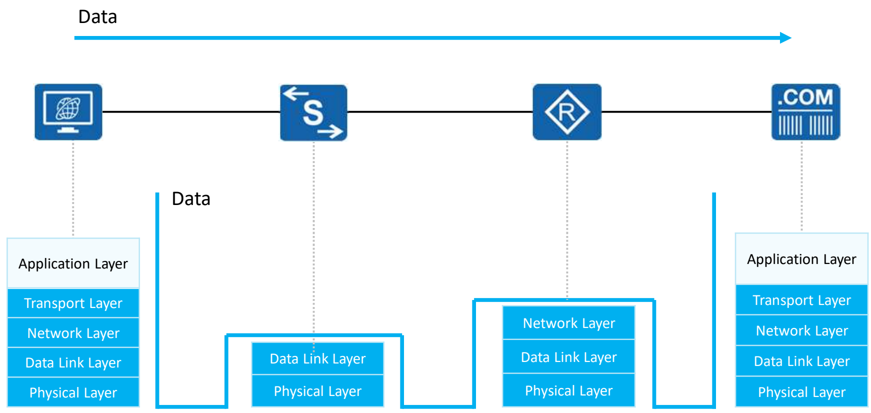
Data Encapsulation on the Sender



- Assume that you are using a web browser to access Huawei's official website. After you enter the website address and press Enter, the following events occur on your computer:
  1. The browser (application program) invokes HTTP (application layer protocol) to encapsulate the application layer data. (The DATA in the figure should also include the HTTP header, which is not shown here.)
  2. HTTP uses TCP to ensure reliable data transmission and transmits encapsulated data to the TCP module.
  3. The TCP module adds the corresponding TCP header information (such as the source and destination port numbers) to the data transmitted from the application layer. At the transport layer, the PDU is called a segment.
  4. On an IPv4 network, the TCP module sends the encapsulated segment to the IPv4 module at the network layer. (On an IPv6 network, the segment is sent to the IPv6 module for processing.)
  5. After receiving the segment from the TCP module, the IPv4 module encapsulates the IPv4 header. At this layer, the PDU is called a packet.
  - Ethernet is used as the data link layer protocol. Therefore, after the IPv4 module completes encapsulation, it sends the packet to the Ethernet module (such as the Ethernet NIC) at the data link layer for processing.
  - After receiving the packet from the IPv4 module, the Ethernet module adds the corresponding Ethernet header and FCS frame trailer to the packet. At this layer, the PDU is called a frame.
  - After the Ethernet module completes encapsulation, it sends the data to the physical layer.
  - Based on the physical media, the physical layer converts digital signals into electrical signals, optical signals, or electromagnetic (wireless) signals.
  - The converted signals start to be transmitted on the network.

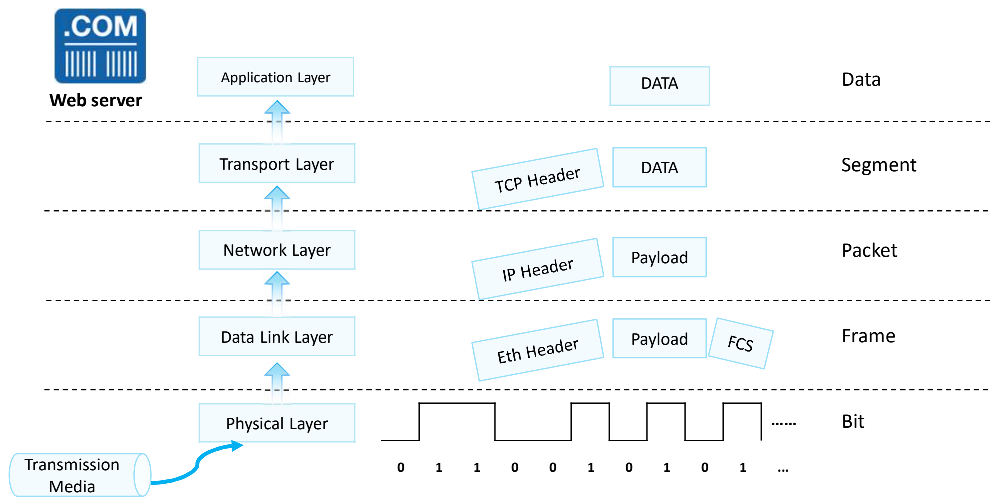
Data Transmission on the Intermediate Network

- Encapsulated data is transmitted on the network.



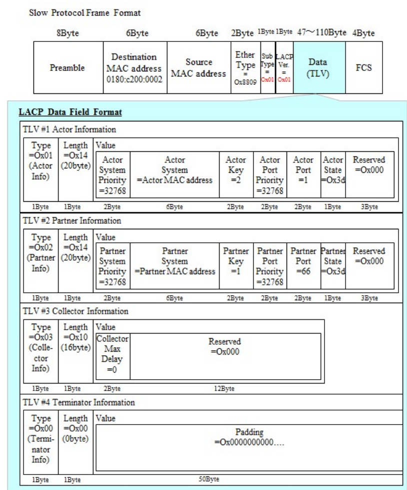
- In most cases:
  - A Layer 2 device (such as an Ethernet switch) only decapsulates the Layer 2 header of the data and performs the corresponding switching operation according to the information in the Layer 2 header.
  - A Layer 3 device (such as a router) decapsulates the Layer 3 header and performs routing operations based on the Layer 3 header information.
  - Note: The details and principles of switching and routing will be described in subsequent courses.

Data Decapsulation on the Receiver



- After being transmitted over the intermediate network, the data finally reaches the destination server. Based on the information in different protocol headers, the data is decapsulated layer by layer, processed, transmitted, and finally sent to the application on the web server for processing.

Spare slide



- https://milestone-of-se.nesuke.com/en/nw-basic/link-aggregation/lacp-decides-active-standby/