Computer Networks and reference models

Chapter 2

1. List of Problems (so far)

- How to ensure connectivity between users?
- How to share a wire?
- How to pass a message through the network?
- How to build Scalable Networks?
- How to ensure the reliability of transmission?
- How to avoid deadlock and congestion situations?
- How to ensure the security during transitions?

Designer must solve these problems when designing a network

2. Overview of the Principles of Computer Networks

1. Physical communication and data transmission,
2. Shared Medium Access,
3. Data switching and Routing,
4. Framing (segmentation and reassembly),
5. Error and Flow control,
6. Congestion Control,
7. Authentication,
8. Protocols…

3. Some Definitions

- **Network**: Collection of interconnected machines
- **Host**: Machine running user application
- **Subnet**: Subset of the network, responsible for carrying messages between hosts
- **Channel**: Logical line of communication
- **Topology**: Network configuration
- **IMP**: Intermediate Processing Node (Router)
- **Protocol**: Rules of communication

4. How Do Computers Communicate? (1)

- With 1's and 0's
  - Computers only deal with 1's and 0's
  - So do networks
- How do we transmit 1's and 0's in a network?

4. How Do Computers Communicate? (2)

Essentials of Data Communication

- Message
- Sender
- Receiver
- Medium and channel
- Understandability
- Error detection
- Security

5. Network Architecture Models

- A reference model that describes the layers of hardware and software necessary to transmit data between two points.
- Reference models are necessary to increase the likelihood that different components from different manufacturers will converse.
- There are two standard models that are important: The OSI Model, and the Internet Model.

6. Standards (1)

- ***Standards*** are rules of operation that are followed by most or all vendors.
- Standards allow hardware and software from different vendors to work together.
- Competition among vendors brings lower prices and feature-rich products.

6. Standards Making Organizations (2)

ISO = International Standards Organization

ITU = International Teletraffic Union (formerly CCITT)

ANSI = American National Standards Institute

IEEE = Institute of Electrical and Electronic Engineers

IETF = Internet Engineering Task Force

ATM Forum = ATM standards-making body

...and many more

7. Different Layering Architectures (1)

- ISO OSI 7-Layers Architecture
- TCP/IP 5-Layers Architecture
- Novell NetWare IPX/SPX 5-Layers Architecture

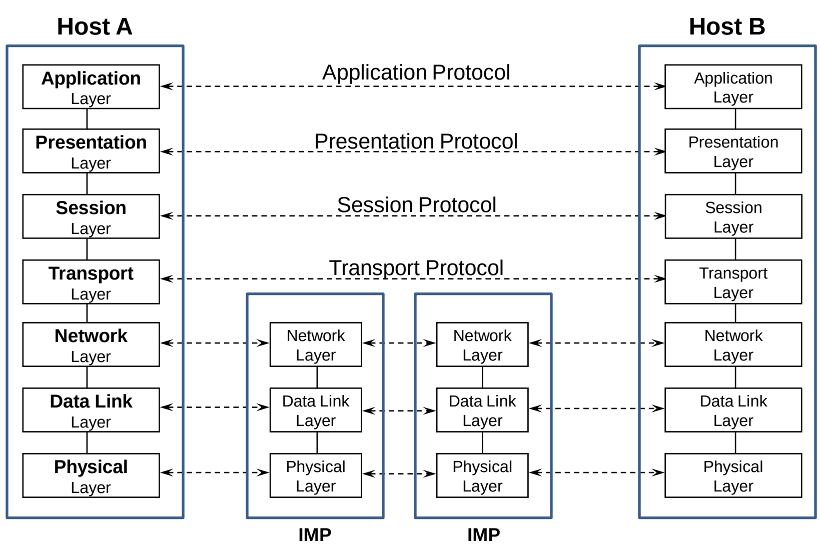
7. ISO OSI Layering Architecture (2)

- 1977 : ISO initiated a layered network architecture,
- 1983 : Definition of the OSI model

  (Open Systems Interconnection)

  - **Open** : open systems to communicate with other systems
  - **Systems** : all the resources (hardware and software) contributing to the processing and transfer of information
  - **Interconnection**

7. ISO OSI Layering Architecture (3)

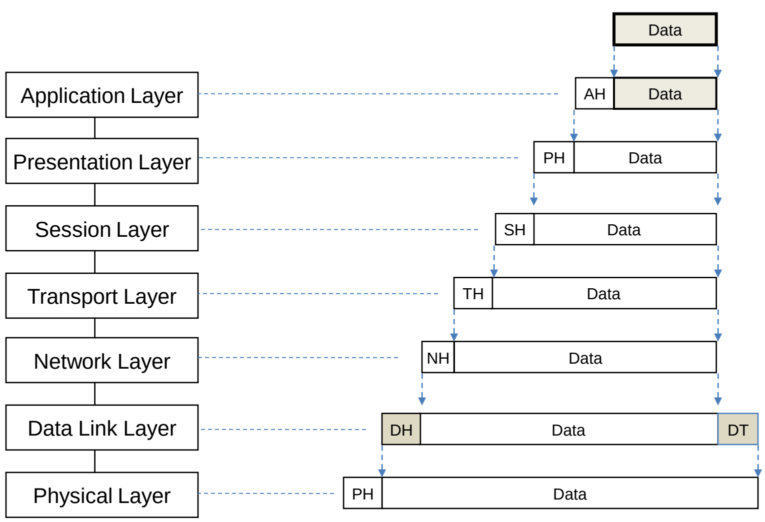


8. ISO’s Design Principles

1. A layer should be created where a different level of abstraction is needed,
2. Each layer should perform a well-defined function,
3. The layer boundaries should be chosen to minimize information flow across the interfaces,
4. The number of layers should be:
   - **Large** enough that distinct functions need not be thrown together in the same layer out of necessity,
   - and **small** enough that the architecture does not become unwieldy.

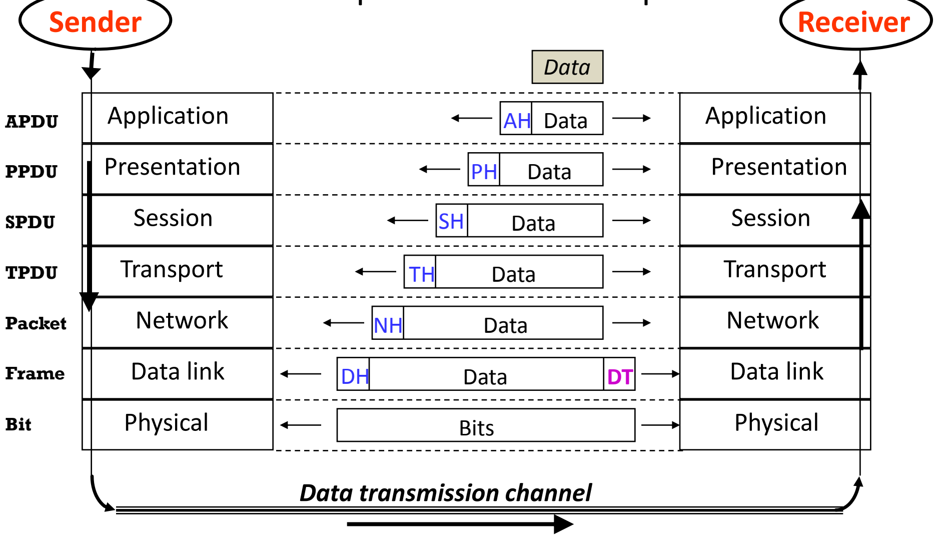
9. General Protocol Functions (1)

Header encapsulation and decapsulation

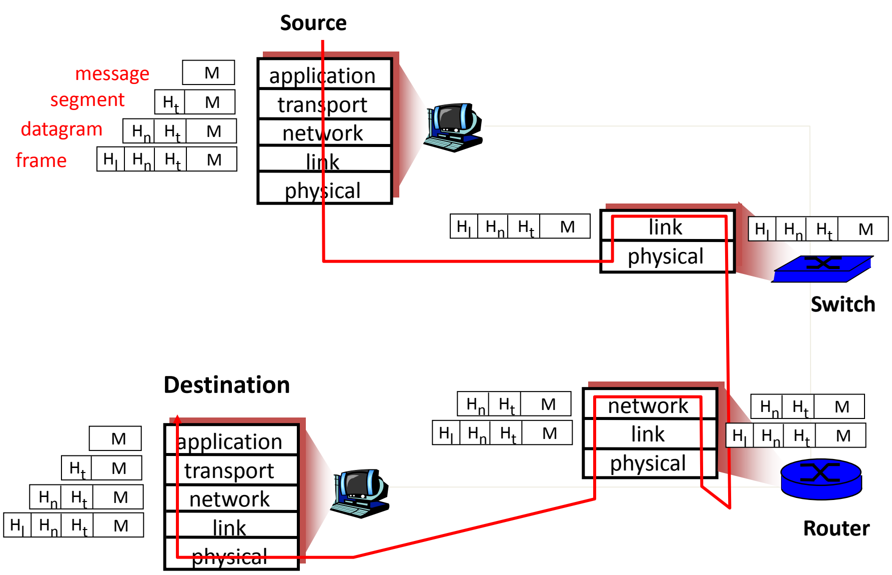


9. General Protocol Functions (2)

Header encapsulation and decapsulation



9. General Protocol Functions (3)



9. General Protocol Functions (4)

Application	**FTP, Video, …**
Presentation	**Presentation of the information : syntax**
Session	**Open a session between S/D**
Transport	**Flow Control of packets (transmitted/received)**
Network	**Addressing and routing …**
Data link	**Restitute a frame from binary sequence (10110), point-to-point data link management, error control (detection/correction)**
Physical	**analog/digital conversion (encoding, modulation) …**

Physical medium

10. Layer 1: Physical Layer

- Functions:
  - Transmission of a raw bit stream
  - Forms the physical interface between devices
- Issues:
  - Which modulation technique (bits to pulse)?
  - How long will a bit last?
  - Bit-serial or parallel transmission?
  - Half- or Full-duplex transmission?
  - How many pins does the network connector have?
  - How is a connection set up or torn down?

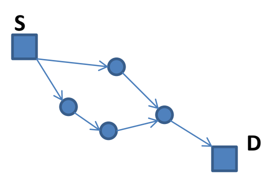
The design issues deal with mechanical, electrical, and procedural interfaces.

11. Layer 2: Data Link Layer

- Functions:
  - Provides reliable transfer of information between two adjacent nodes,
  - Creates frames, or packets, from bits and vice versa,
  - Provides frame-level error control,
  - Provides flow control,
  - Establishes, maintains & releases a point-to-point connection.
- In summary, the data link layer provides the network layer with what appears to be an error-free link for packets.
- Broadcast networks have an additional issue in the data link layer: how to control access to the shared channel. A special sublayer of the data link layer (MAC: Medium Access Control) deals with the problem.

12. Layer 3: Network Layer

The main task of the network layer is to determine how data can be delivered from source to destination.



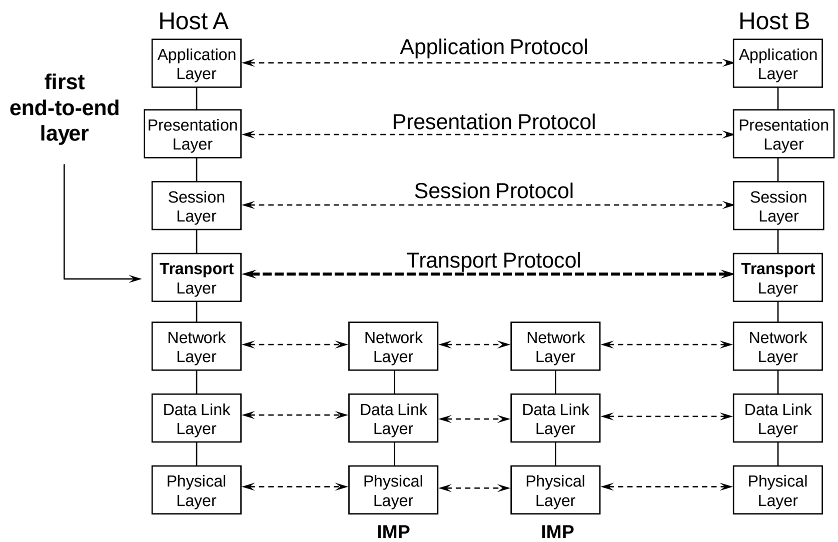
- Functions:
  - Responsible for routing decisions,
    - ✓ Dynamic routing
    - ✓ Fixed (static) routing
  - Performs congestion control,
  - Allows interconnection of heterogeneous networks.

13. Layer 4: Transport Layer (1)

- Functions:
  - accepts data from the session layer, split it up into smaller units, pass them to the network layer, and ensure that the pieces all arrive correctly at the other end.
  - Hide the details of the network from the session layer.

    Example: If we want to replace a point-to-point link with a satellite link, this change should not affect the behavior of the upper layers
  - Provides reliable end-to-end communication.

13. Layer 4: Transport Layer (2)



13. Layer 4: Transport Layer (3)

- Functions (cont'd):
  - Performs end-to-end flow control (available resources, packets, acknowledgments …).
  - Makes sure all information is accounted for:
    - ✓ Missing information
    - ✓ Duplicated information
  - Performs packet retransmission when packets are lost by the network.
  - Ensures that packets are delivered error free, in sequence with no losses or duplications.

14. Layer 5: Session Layer

The session layer allows users on different machines to establish sessions (conversations) between them

- May perform synchronization between several communicating applications (by inserting checkpoints into the data stream. if session fails, only data after the most recent checkpoint need to be retransmitted),
- Groups several user-level connections into a single "session",
- Establishes connection between applications,
- Reestablishes connection,
- Manages who can transmit data at a time and for how long (Token management).

15. Layer 6: Presentation Layer

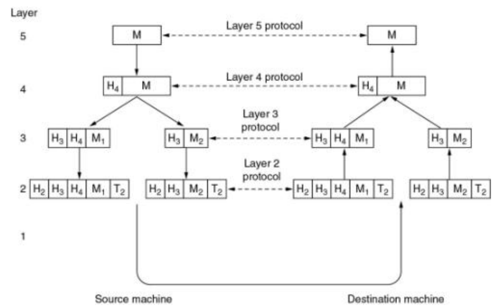
Performs specific functions that are requested regularly by applications

- Masks the differences of data formats between dissimilar systems
- Examples:
  - Encodes and decodes data (ex. MPEG2,3…)
  - ASCII to Unicode, Unicode to ASCII
  - Encryption (decryption)
  - Compression (decompression)
  - LSB-first representations to MSB-first representations

16. Layer 7: Application Layer

- Application layer protocols are application-dependent
- Implements communication between two applications of the same type
- Examples:
  - FTP
  - SMTP (email)

17. Protocol Hierarchies

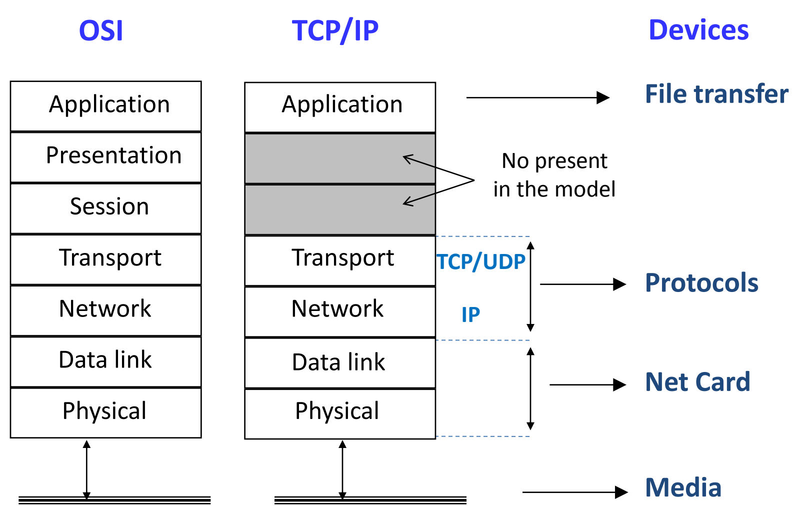


- Each layer takes data from above
  - adds header information to create new data unit (“encapsulation”),
  - passes new data unit to layer below.

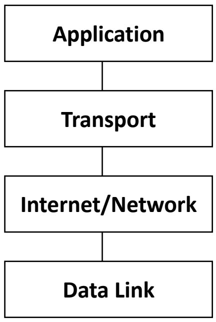
- In a network architecture model, only the lowest layer contains a physical connection, while all higher layers contain logical connections.

18. TCP/IP Layering Architecture (1)

Comparison OSI, TCP/IP ➔ devices (H/S)

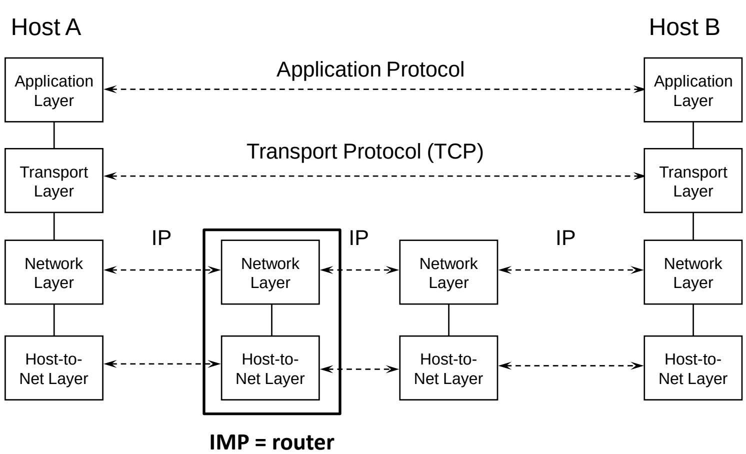


18. TCP/IP Layering Architecture (2)



- A simplified model
- The network layer
  - Hosts drop packets into this layer,
  - routes towards destination- only promise- try my best
- The transport layer
  - reliable byte-oriented stream

18. TCP/IP Layering Architecture (3)



19. The Internet’s Design Principles

- Protocols came first, and the layering model came later.
  the model was just a description of the existing protocols.
- TCP/IP specifically designed for the Internet.
- TCP/IP model doesn’t describe other protocols well.
  As a consequence, the model was not useful for describing other non-TCP/IP networks.

20. Types of servers (1)

Different types of servers exist. Most popular of which are:

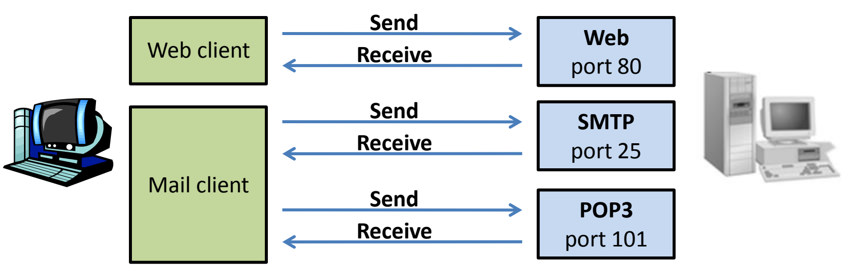
- **Web**. World Wide Web (WWW)
- **Mail send**. Simple Mail Transfer Protocol (SMTP)
- **Mail receive**. Post Office Protocol v.3 (POP3)
- **File transfer**. File transfer protocol (FTP)

20. Types of servers (2)

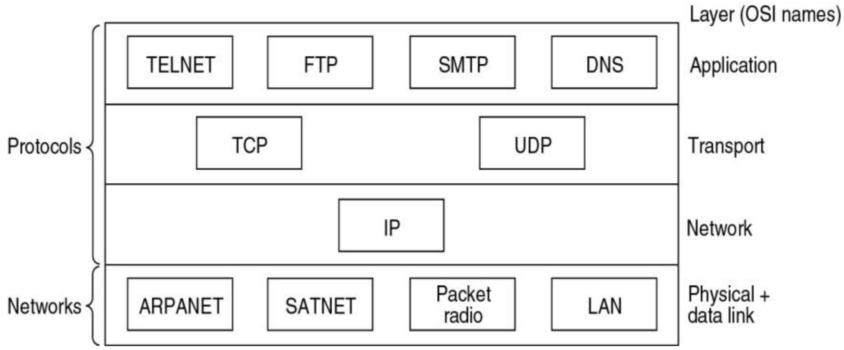
- Servers are software generally referred to as services.
- Multiple servers may run on the same machine.
- Needed. A unique way of identifying that server.
- Solution. Use a unique identification number for every server (PORT NUMBER).

20. Client/server communication

- Diagram of a client connecting to multiple servers on a single remote machine.



21. Communication profiles



22. QoS metrics (1)

Different applications have different requirements regarding the handling of their traffic in the network. These requirements are expressed using the following QoS-related parameters:

- **Bandwidth.** The amount of data that can be transmitted in a fixed amount of time. For digital devices, the bandwidth is usually expressed in bits per second (**bps**) or Bytes per second (**Bps**). For analog devices, the bandwidth is expressed in cycles per second, or Hertz (**Hz**).
- **Throughput.** The average rate of successful message delivery over a communication channel. The throughput is usually measured in bits per second (bit/s or bps).

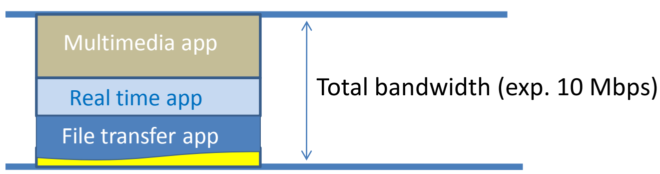
22. QoS metrics (2)

- **Latency.** The delay that an application can tolerate in delivering a packet of data. The latency is usually expressed in second.
- **Jitter.** the variation in latency (expressed in second).
- **Loss.** the percentage of lost data.

QoS mechanisms work by controlling the allocation of network resources to application traffic in a manner that meets the application's service requirements.

22. QoS metrics (3)

Bandwidth



Bandwidth amount required by applications:

- Large,
- Fixed,
- Variable,
- Available,
- Not specified.

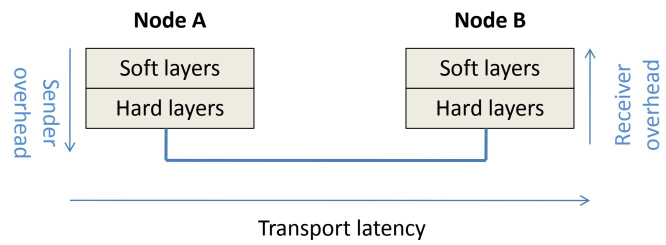
Network protocols implement bandwidth allocation in different ways:

- **Static**. Reserves fixed bandwidth amount during connection.
- **Dynamic**. Adjusts bandwidth amounts for the supported applications.

22. QoS metrics (4)

Latency

The time from when a bit is sent until it is received at the other end.

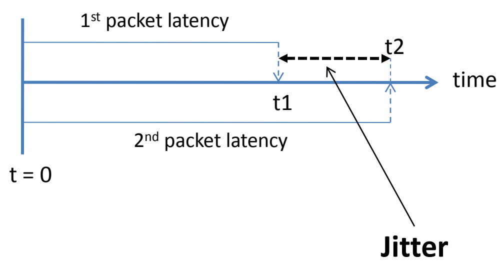


Total latency = Sender overhead + Transport latency + Receiver overhead

22. QoS metrics (5)

Jitter

Packets from the source will reach the destination with different delays. This variation in delay is known as **jitter** and can seriously affect the quality of streaming audio and/or video.



Jitter may have a limited value, otherwise QoS is degraded.

22. QoS metrics (6)

Loss

- Packet is lost (NACK, time out).

$$Loss.rate = \frac{Lost\,pckts}{Total\,Nb\,pckts}$$

23. More Definitions

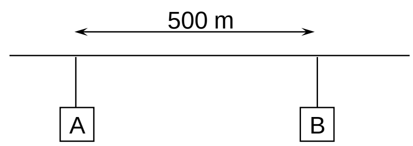
- **Packet length**: size of a packet (units = bits)
- **Channel speed:** How fast the channel can transmit bits (units = bits/second, Mbits/sec, Gbits/sec)
- **Packet transmission time:** amount of time to transmit an entire packet (units = seconds, msec, µsec)

$$\boxed{Pckt.tr.time = \frac{Pckt.Length}{Chanl.Speed}}$$

- **Propagation delay:** Delay imposed by the transmission medium, depends on distance (units = µsec/km)

$$\boxed{prop.delay.A \rightarrow B = prop.delay * Dist.A \rightarrow B}$$

24. Some Math



packet length - 1500 bytes
channel speed - 10 Mbits/sec
propagation delay - 5 μsec/km $\equiv$ 200.10^3 km/s

Packet transmission time = packet length / channel speed

= (1500 bytes x 8 bits/byte) / (10 x 10^6 bits/sec)

= 1.2 x 10^{-3} sec <u>or</u> 1.2 msec

Prop. delay between A & B = (500 m x 1km/1000 m) x (5 x 10^{-6} sec/km)

= 2.5 x 10^{-6} sec or 2.5 μsec