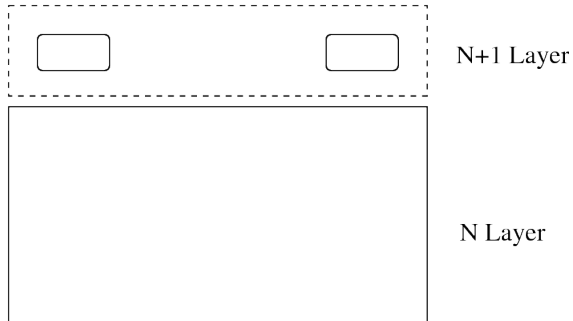
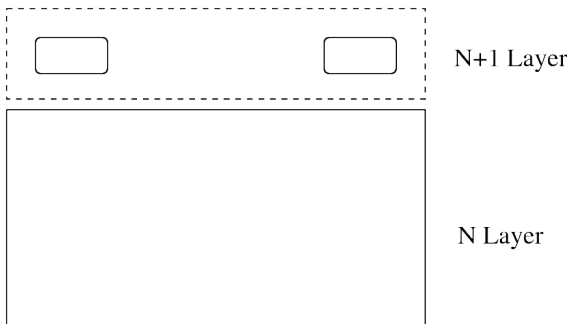


Layered Network Architecture



Layered Network Architecture



Question: How can we provide a reliable service on the top of a unreliable service?

ARQ: Automatic Repeat Request

- Can be used in every layer
- TCP uses ARQ

ARQ: Automatic Repeat Request

- Can be used in every layer
- TCP uses ARQ

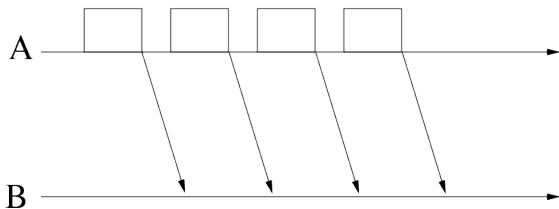
ARQ: Automatic Repeat Request

- Can be used in every layer
- TCP uses ARQ

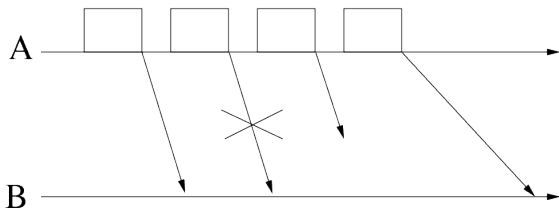
- Understand how ARQ works
- Understand approaches to improve the performance of ARQ

- Understand how ARQ works
- Understand approaches to improve the performance of ARQ

Ideally: Every Packet is Received without Errors



In Reality: Bad Things can Happen



ARQ: Retransmission Strategies

- **Objective:**

Design a protocol such that each packet is delivered once, and only once, and without errors.

- **Assumptions:**

- All packets with error are detected
- Delay can be arbitrarily long
- Some packet maybe lost
- Packets that arrive are in the same order

ARQ: Retransmission Strategies

- **Objective:**

Design a protocol such that each packet is delivered once, and only once, and without errors.

- **Assumptions:**

- All packets with error are detected
- Delay can be arbitrarily long
- Some packet maybe lost
- Packets that arrive are in the same order

ARQ: Retransmission Strategies

- **Objective:**

Design a protocol such that each packet is delivered once, and only once, and without errors.

- **Assumptions:**

- All packets with error are detected
- Delay can be arbitrarily long
- Some packet maybe lost
- Packets that arrive are in the same order

ARQ: Retransmission Strategies

- **Objective:**

Design a protocol such that each packet is delivered once, and only once, and without errors.

- **Assumptions:**

- All packets with error are detected
- Delay can be arbitrarily long
- Some packet maybe lost
- Packets that arrive are in the same order

ARQ: Retransmission Strategies

- **Objective:**

Design a protocol such that each packet is delivered once, and only once, and without errors.

- **Assumptions:**

- All packets with error are detected
- Delay can be arbitrarily long
- Some packet maybe lost
- Packets that arrive are in the same order

ARQ: Retransmission Strategies

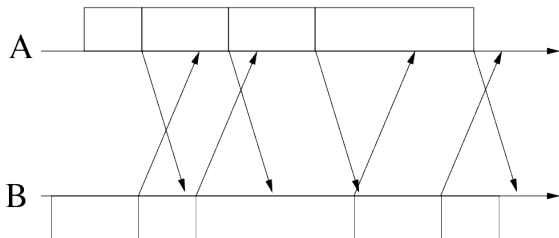
- **Objective:**

Design a protocol such that each packet is delivered once, and only once, and without errors.

- **Assumptions:**

- All packets with error are detected
- Delay can be arbitrarily long
- Some packet maybe lost
- Packets that arrive are in the same order

Focus on Communication from *A* to *B*



Stop-and-Wait ARQ: Sender

The algorithm at node A for A-to-B transmission:

- 1 Set the integer variable SN to 0.
- 2 Accept a packet from the next higher layer at A; if no packet is available, wait until it is; assign number SN to the new packet.
- 3 Transmit the SNth in a frame containing SN in the sequence number field.
- 4 If an error-free frame is received from B containing a request number RN greater than SN, increase SN to RN and go to step 2. If no such frame is received within some finite delay, go to step 3.

Stop-and-Wait ARQ: Receiver

The algorithm at node B for A-to-B transmission:

- 1 Set the integer variable RN to 0 and the repeat steps 2 and 3 forever.
- 2 Whenever an error-free frame is received from A containing a sequence number SN equal to RN, release the received packet to the higher layer and increment RN.
- 3 At arbitrary times, but within bounded delay after receiving any error-free data frame from A, transmit a frame to A containing RN in the request number field.