CSE123A discussion session

2007/02/16  Ryo Sugihara

Topics

• HW2?

• Review
  – Data Link Layer (5): Error recovery sublayer
    • Sliding window
      – Go-back n / Selective reject
    • Initialization issues
  – Data Link Layer (6): Media access sublayer
    • Statistical multiplexing
    • ALOHA
    • Ethernet
Where are we now?

Today's topic

Sliding window protocol

• **Why?**
  – For more efficient use of bandwidth
    • cf.) Stop and Wait protocol

• **How?**
  – Use “window”
    • e.g.) window = (1,3): “sender can send frames 1,2,3 without waiting for Acks”

• **Two variants**
  – “Go-back-n”
  – “Selective reject”
Sliding window: Example

- Buffer out-of-order frames, or not

Initialization issue

- Difficult, in the face of link/node failure

- Impossible, without any one of them
  - Non-volatile memory
  - Timer
  - Random number
**Media access**

- “Who sends next?”
  - Mediation, but in a distributed manner
  - Only necessary on “broadcast link”

**Statistical multiplexing**

- “Static” multiplexing
  - TDM, FDM
  - Wasteful for “bursty” traffic
    - def) High peak/avg ratio
    - Each user uses a lot of bandwidth, but only infrequently

- Idea of statistical multiplexing
  - Each user tries to access when necessary
  - When collision happens, retry later
    - Randomness takes place here
ALOHA

- Predecessor of Ethernet
- Multiplexing for wireless network
- On collision, randomized backoff
  - Collision detection based on ACK
- 2 variants: unslotted/slotted
  - Slotted ALOHA is more efficient, but needs synchronized clock

---

Ethernet

- ALOHA + improvements
  - Carrier sense
  - Collision detection
  - Adaptive backoff time

- "CSMA/CD"
**Ethernet: Carrier sense**

- “Before you send, check if nobody is sending data”
- Does not guarantee collision-free
  - Time for signal propagation

  ![Diagram](image1)

- (Off-topic) Carrier sense is not always possible in wireless network
  - “hidden terminal”

**Ethernet: Collision detection (1/2)**

- Collision is detected by monitoring the line while sending data
  - This method is related to the necessity of “minimal frame size” (& max link length)

  ![Diagram](image2)

- (Off-topic) Collision detection (in this way) is usually impossible in wireless network
  - ALOHA uses ACK
Ethernet: Collision detection (2/2)

- **Minimum frame size** (& max link length)
  - **Why?**
    - We want **every** involved party to detect collision

![Diagram showing collision and propagation delay]

Ethernet: Adaptive backoff time

- **ALOHA**: random, but static
  - like “Randomly pick 1T or 2T”
  - What if there are multiple contenders?

- **Ethernet**:
  - “Binary exponential backoff”
    - 1<sup>st</sup> time: 1-2T
    - 2<sup>nd</sup> time: 1-4T
    - 3<sup>rd</sup> time: 1-8T
    - ...

  - Prob(failure) exponentially drops with the num of trials