Layering and End-to-End Argument

Wireless Networks Lecture 2

Dr. Ahmed Mahdi Jubair

Today

1. Layering and **the End-to-End Argument**

2. Transmission Control Protocol (TCP) primer

3. Split Connection TCP over wireless

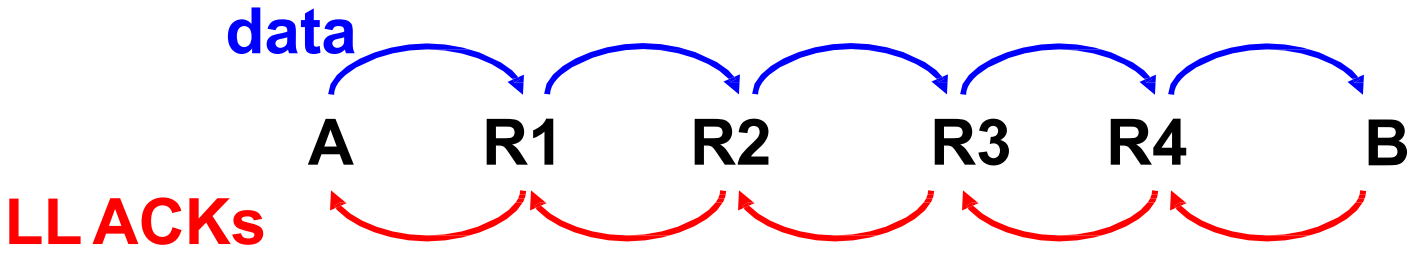
Motivation: End-to-End Argument

- **Five layers** in the Internet architecture model

- **Five places** to solve many of same problems:
  - In-order delivery
  - Duplicate-free delivery
  - Reliable delivery after corruption, loss
  - Encryption
  - Authentication

- ***In which layer(s) should a particular function be implemented?***

Example: Careful file transfer from A to B



- **Goal: Accurately copy file on A's disk to B's disk**

- Straw man design:
  - Read file from **A**'s disk
  - **A** sends stream of packets containing file data to **B**
    - L2 retransmission of lost or corrupted packets at each hop
  - **B** writes file data to disk

- ***Does this system meet the design goal?***
  - Bit errors on links not a problem

Where might errors happen?

- On **A**'s or **B**'s disk
- In **A**'s or **B**'s RAM or CPU
- In **A**'s or **B**'s software
- In the RAM, CPU, or **software** of **any router** that forwards packet

- Why might errors be **likely?**
  - Drive for CPU speed and storage density: pushes hardware to EE limits, engineered to tight tolerances
    - *e.g.*, today's disks return data that are the output of an maximum-likelihood estimation!
  - Bugs abound!

Solution: End-to-End verification

1. **A** keeps a **checksum** with the on-disk data
   - *Why not compute checksum at start of transfer?*
2. **B** computes checksum over received data, sends to **A**
3. **A** compares the two checksums and resends if not equal

- Can we eliminate hop-by-hop error detection?

- Is a whole-file checksum, **alone,** enough?

End-to-End Principle

- Only the **application at communication endpoints can completely and correctly** implement a function

- Processing in **middle alone cannot** provide function
  - Processing in **middle may,** however, be an **important performance optimization**

- Engineering middle hops to provide guaranteed functionality is *often* **wasteful of effort, inefficient**

Perils of lower-layer implementation

- **Entangles** application behavior with network internals

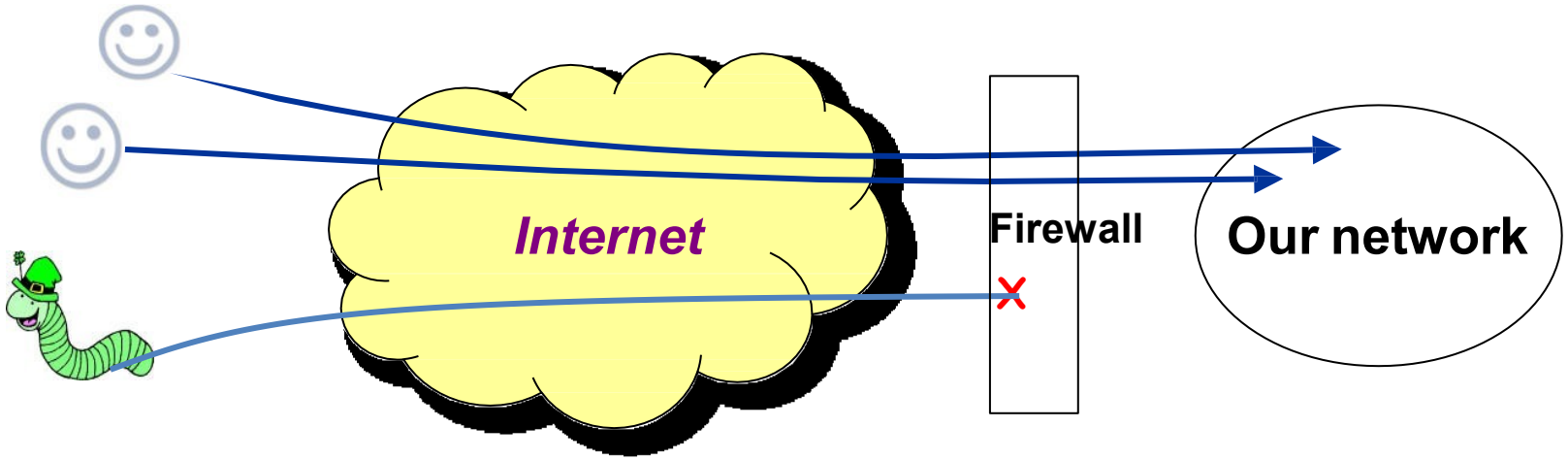
- **Suppose** each IP router **reliably transmitted** to next hop
  - **Result:** Lossless delivery, but **variable delay**
    - ftp: **Okay,** move huge file reliably (just end-to-end TCP works fine, too, though)
    - Skype: **Terrible,** jitter packets when a few corruptions or drops not a problem anyway

- **Complicates deployment** of innovative applications
  - Example: Phone network *v.* the Internet

Advantages of lower-layer implementation

- Can improve **end-to-end system performance**
- Each application author **needn't recode a shared function**
- Overlapping error checks (*e.g.*, checksums) at all layers invaluable in **debugging and fault diagnosis**
- If end systems not cooperative (increasingly the case), **only way to enforce resource allocation!**

End-to-end violation: Firewalls



- Firewalls clearly **violate the e2e principle**
  - **Endpoints** are capable of deciding what traffic to ignore
  - Firewall **entangled** with design of network and higher protocol layers and apps, and vice-versa
    - ***e.g.:*** New ECN bit to improve TCP (wireless) congestion control; many firewalls **filter all such packets!**
    - **ECN** (explicit congestion notification: is a mechanism in TCP/IP where routers can signal that they are almost overloaded)
- **Yet, we probably do need firewalls**

Summary: End-to-End principle

- Many functions **must** be implemented **at application endpoints** to provide desired behavior
  - Even if implemented in "middle" of network

- End-to-end approach **decouples design** of components in network interior from design of applications at edges
  - Some functions still **benefit** from implementation in **network interior** at cost of entangling interior, edges

- End-to-end principle is **not sacred;** it's just a way to think critically about design choices in communication systems