BITS Pilani
Pilani Campus

Advance Computer Networks (CS G525)

Virendra S Shekhawat
Department of Computer Science and Information Systems

innovate achieve lead

BITS Pilani
Pilani Campus

First Semester 2018-2019
Slide Deck-M1-1

Agenda

- Introduction with Students
- Course Overview
- Course Administration
- Let's Start Learning…

Goals and Course Objective

- To learn state-of-the-art about network architectures, protocols, systems, and network applications
- To learn "How to read research papers…?"
- To read others research work and critically analyze their work
- To engage ourselves in Networking research

Resources

- There is no prescribed Text Book !!!
- Want to brush-up basics of Computer Networks
  - *Computer Networking*: A Top Down Approach by Kurose & Ross
  - *Computer Networks: A Systems Approach* by L. Peterson Davie
- Course contents comprise
  - 30+ research papers (List is in the course handout)
  - Also will made available on the course page

Course Administration

- Class Timings
  - Mon, Wed, Fri @ 9:00 – 9:50 am
- Course page Information
  - **http://nalanda.bits-pilani.ac.in**
- Evaluation Plan (Pilani Campus)
  - Mid Semester Test @ 20% (Close Book)
  - Assignment @ 7% (Open Book)
  - *Lab Test @ 8% (Open Book)*
  - *Research Project @20% (Open Book)*
    - *Beginning of the semester: Research project problem formulation*
    - *Mid Semester assessment: Seminar/paper presentation [5%]*
    - *End Semester assessment: Documentation [3%], Design & Implementation [7%],* Result analysis & project outcome/contribution [5%]
  - Online Quizzes/Discussion Forum/Classroom Participation@10% (Open Book)
  - Comprehensive exam @35% (Close Book)
- Chamber (6121-R) consultation hour(s)
  - Tue, Thu, Sat @ 5:00 to 6:00 pm

Classroom Topics Coverage

- Little emphasis on undergraduate level stuff
  - Based on topics demand/requirement!
- Layer wise focus will be on the
  - Network Layer, Transport Layer, and Application layer
  - Less emphasis on link and physical layer
- What will be covered:
  - Protocols and Algorithms, Network Performance, Network Applications, Quality of Service
  - Focus will be on 3 Ws (What?, Why?, Why not?)

Course Plan

- Five Modules
  - Internet Architecture & Principles
  - Software Defined Networks (SDNs)
  - Network Traffic Control & Management
  - Wireless & Mobile Networks
  - Overlay & Data Center Networks

Objective of Networking?

- Communication between applications running on different hosts
- Should understand applications needs/requirements
  - Bandwidth requirement (i.e. bits/sec)
  - Traffic nature
    - Bursty or constant bit rate
  - Traffic target
    - Single or multiple, Fixed or mobile
  - Delay sensitive/tolerant
  - Loss sensitive/tolerant

Telephone network vs. Internet

- Telephone network
  - End to end connection establishment
  - Message switching
  - Resource utilization?

- Internet
  - Packet switching (packets are self contained unit!)
  - Use Store and forward
  - On demand resource allocation
  - Good for bursty traffic. Why???

Issues with Packet switching

- Reordering requires at destination
  - Packets can take different paths to reach destination
- Contention
  - Causes Packet loss
- Congestion
  - Causes Delay

What is Internet?

- Network of networks…
  - Multiple networks are interconnected
  - Key components
    - **Hosts** or end points
    - **Routers** to connect networks

Challenges

- Interconnected networks are different in various ways
  - Address formats
  - Performance – bandwidth/latency
  - Packet size
  - Loss rate/loss handling
  - Routing
- How to transfer the data from one network to another???

Network Functionalities

- Find the nodes in the Internet
- Route the packets
- Deal with the different packet size requirements from different networks
- Meet the application requirements
  - Reliability
    - Data loss and data corruption
  - Congestion and flow control
  - In order delivery
- **How to implement these functionalities?**

Layered Architecture

- Networking functionalities can be divided into modules/layers
- Good side of layered model
  - Each layer relies on services from layer below and exports services to layer above
  - Interface defines interaction
  - **Hides implementation** - layers can change without disturbing other layers (black box)

Example: OSI Layers

- Physical: how to transmit bits
- Data link: how to transmit frames
- Network: how to route packets
- Transport: how to send packets end to end
- Session: how to tie flows together
- Presentation: byte ordering, security
- Application: everything else
- Question…?
  - How TCP/IP is different from OSI?

Any Drawbacks of Layering?

- Layer N may duplicate lower level functionality (e.g., error recovery)
- Layers may need same info (e.g., timestamp, MTU)
- Strict adherence to layering may hurt the performance. How?

Agenda

- Internet design philosophy
  - Compulsory Reading
    - The Design Philosophy of The DARPA Internet Protocols [Clark 1988]

Background

- Why do we need networking?
- Telephone network vs. Internet
- Why packet switching was considered for the Internet Architecture?

Goals for the Internet Architecture

0. Connect existing networks

Initially ARPANET and ARPA packet radio network

1. Survivability
2. Support multiple types of services

   Differ in Speed, Latency and Reliability

3. Must accommodate a variety of networks
4. Allow distributed management
5. Allow host attachment with a low level of effort
6. Be cost effective
7. Allow resource accountability

0. Connecting Networks

- How to internetwork various network technologies
- ARPANET, X.25 networks, LANs, satellite networks, packet networks, serial links…
- Many differences between networks
  – Address formats, Performance – bandwidth/latency, Packet size, Loss rate/pattern/handling and Routing

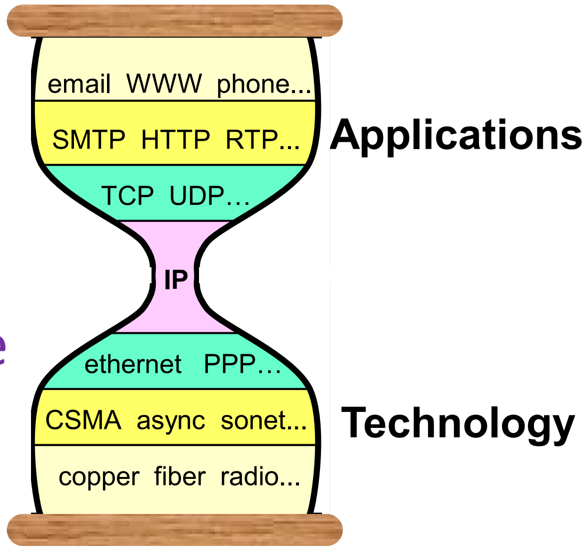
Standardization: Support Multiple Types of Networks

- How to provide Interconnectivity with wide variety of networks?
  - Minimal set of functionality for underlying network gives this flexibility!
    - Minimum packet size
    - Reasonable delivery odds, but not 100%
    - Some form of addressing unless point to point
- Important non-assumptions:
  - Perfect reliability
  - Broadcast, multicast
  - Priority handling of traffic
  - Internal knowledge of delays, speeds, failures, etc.

- Also achieves Goal 3 of supporting various networks

IP Hourglass

- Hide underlying technology from applications
- Decisions:
  - Network provides minimal functionality
  - "Narrow waist"
  - Tradeoff→ No assumptions no guarantee
- Question
  - If some applications want these services (e.g. reliability) then how to get it?



1. Survivability

- Ensure communication service even in the presence of network and router failures
- If network disrupted and reconfigured
  - Communicating entities should not care!
- How to achieve such reliability?
  - Where can communication state be stored?

	Network	**Host**
Failure handing	Replication	“Fate sharing”
Net Engineering	Tough	Simple
Trust	Less	More

Fate Sharing

- Lose state information for an entity if and only if the entity itself is lost.
- Examples:
  - OK to lose TCP state if one endpoint crashes
    - NOT okay to lose if an intermediate router reboots
  - **Is this still true in today's network?**

- Survivability compromise
  - Weak provisions for getting network failure information
  - Higher level protocols are used to detect the failure

2. Types of Service: Speed, Latency, Reliability

- Network layer provides one simple service: best effort datagram (packet) delivery
  - All packets are treated the same
- Relatively simple core network elements
- It is a building block from which other services (such as reliable data stream) can be built
- No QoS support assumed from below
  - Hard to implement without network support
  - QoS is an ongoing debate…

Two types of Service

- TCP vs. UDP
  - **Elastic apps** that need reliability: remote login, email, file transfer
  - **Inelastic**, loss-tolerant apps: real-time voice or video
  - Others in between, or with stronger requirements

- Original Internet model: "TCP/IP" one layer
  - First app was remote login…
  - But then came debugging, voice, etc.
  - These differences caused the layer split, added UDP

4. Distributed Management

- Different agencies are involved in managing gateways and networks
- Each organization running its own routing algorithms
- Lack of tools availability for distributed applications, especially in the area of routing
- Requirement
  - Tools for resource management in the context of multiple administration

5. Host attachment cost for Internet

- Costlier Stuff. Why?
  - All important functions are implemented at host
  - Benefit
    - Host engineering is easy compared to network engineering!
  - Harm
    - Bad implementation can hurt network as well as host!
    - Host misbehave becomes a bigger problem as Internet is growing

6. Is Internet Architecture cost effective?

- Overhead of packet headers (40 Bytes)
  - More significant for small size packets
  - Worst case: single character remote login packet
- Retransmission of lost packets
  - Retransmitted packets cross the net twice
  - No mechanism to recover lost packets from network!
- Inference
  - Reliability enhancement to be added to the network for loss sensitive applications

Question

- What are the key observations about Internet architecture design?

Summary: Minimalist Approach

- Dumb network
  - IP provide minimal functionalities to support connectivity
    - Addressing, forwarding, routing
- Smart end system
  - Transport layer or application performs more sophisticated functionalities
    - Flow control, error control, congestion control
- Advantages
  - Accommodate heterogeneous technologies (Ethernet, modem, satellite, wireless)
  - Support diverse applications (telnet, ftp, Web, X windows)
  - Decentralized network administration

Agenda

- End to End Argument in System Design
  - Reading References
    - End-to-End Argument in System Design [J H Saltzer 1984]

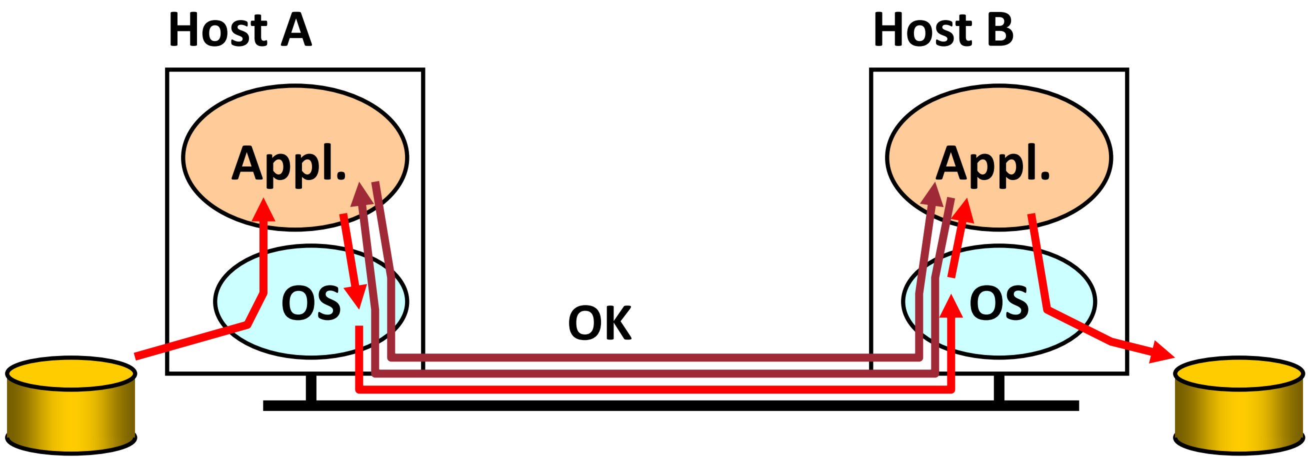
Least focused goals in DARPA Architecture

- Allow distributed management
- Allow host attachment with a low level of effort
- Be cost effective
- Allow resource accountability

Network Functionalities: Implementation

- Network functionalities examples
  - Reliable and sequenced delivery, Addressing and routing, Security, Ethernet collision detection, Multicast, Real-time guarantees, error recovery, duplicate message suppression etc.
- **Where** to place a functionality…?
  - Inside the network (in switching elements)
  - At the edges (i.e. hosts)
  - As a joint venture
  - Redundantly at both places

Example: Reliable File Transfer

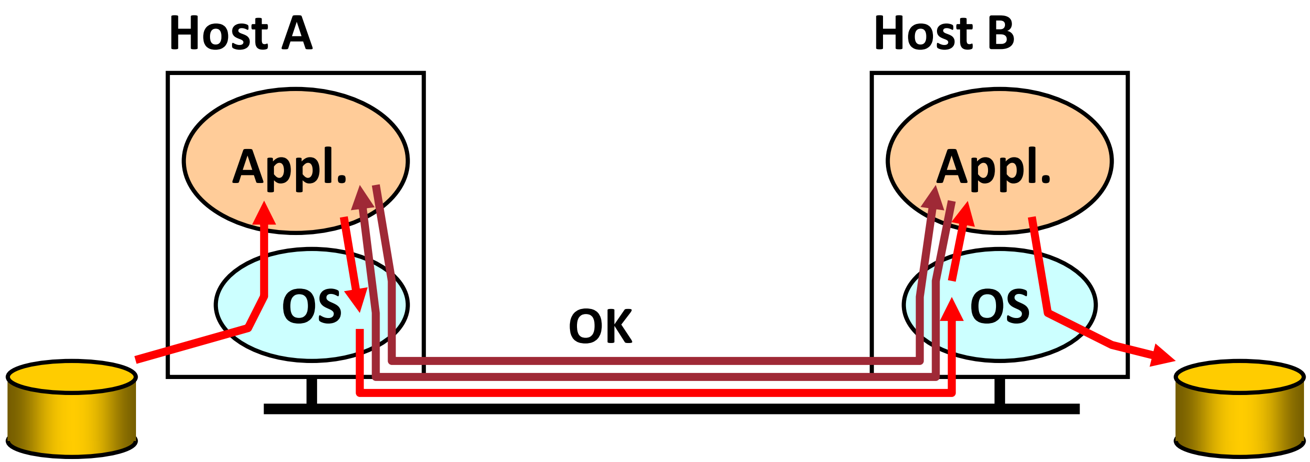


Threats Involved in File Transfer

- File read may contain incorrect data due to hardware faults in the disk storage system
- The FS, FT program, DC system might make a mistake in buffering and copying the data of the file, either at A or B (*writing correct program can avoid this!!!*)
- The processor or its memory might have transient error while doing buffering and copying
- The communication system might drop/flip some bits in a packet or drop a packet
- Either of the host may crash at any point of time

How to cope up with threats…

- **Solution 1**: make each step reliable, and then concatenate them
  - By Duplicate copies, timeout retry, redundancy
- **Solution 2**: end-to-end check and retry



E2E Example: File Transfer

- Even if network guaranteed reliable delivery
  - Need to provide end-to-end checks
    - e.g., buffering errors, h/w fault in disk storage system, network card may malfunction
  - The receiver has to do the check anyway!

- Full functionality (i.e. reliability) can only be entirely implemented at application layer; no need for reliability from lower layers
- Argument
  - There are functions that can only be ***"correctly"*** implemented by the end hosts – do not try to completely implement these elsewhere
  - It is a guideline not a law!

Quiz

- Does FTP look like E2E file transfer?
  - *Hint:* TCP provides reliability between kernels not between disks

Other e2e Examples

- Secure Transmission of Data
- Duplicate Message Suppression
- Delivery Guarantees
- Guaranteeing FIFO Message Delivery
- Transaction Management

Secure Transmission of Data

- If **data transmission system** performs encryption and decryption
  - it must be trusted to manage securely the keys used
  - data passed into the node is in clear form hence it becomes vulnerable
  - The authenticity of data must still be checked by the application

- Question
  - If Application does this….. then…?

Duplicate Message Suppression

- A duplicate message may be delivered due to -
  - Time-out triggered failure detection and retry mechanisms
- What is expected from the network?
  - Either it should detect and suppress such duplicate messages or deliver them as it is for the application to handle it.
- What is the correct approach?
  - Should network do this or application do this?
  - Think about the duplicate messages originated by the application itself!!!
    - e.g., A remote system user may initiates a new login to a time sharing system

How to Decide the End Points…?

- End points vary from application to application
  - e.g. Real time voice transmission vs. non-real time voice transmission
- Other Similar Examples
  - RISC chips vs. CISC chips
  - Feature Implementation at OS level vs. Application level

Discussion

- Is there any need to implement reliability at lower layers?
  - Yes, but only to improve performance
- If network is highly unreliable
  - Adding some level of reliability helps performance, not correctness
  - Don’t try to achieve perfect reliability!
- Implementing a functionality at a lower level should have minimum performance impact on the applications that do not use the functionality

Quiz

- What should be done at the end points, and what by the network?
  - Reliable/sequenced delivery?
  - Addressing/routing?
  - Security?
  - Ethernet collision detection?
  - Multicast?
  - Real-time guarantees?

Agenda

- Tussle in Cyber space
  - Compulsory Reading
    - Tussle in Cyberspace: Defining Tomorrow's Internet [Clark 2003]

Requirement in Today's Communication

- User's communicate but don't trust
  - User desire anonymity
- End-Parties Distrust Their Software and Hardware
  - E.g. Cookies, collect consumer details for marketing goals
- Third Party asserts it's right to interpose communication
  - Private ISPs and Govt. agencies wants to monitor traffic
- One party forces Interaction on Another
  - E.g. Email Spam
- Multi-way Communication (*Internet has been designed for one to one communication only…* **end to end argument**)
  - e.g. Teleconferencing, Broadcasting

Where are we moving…

- Operation in Untrustworthy World
- Demanding Applications
  - Audio/Video Streaming
    - Uses intermediate nodes (Violates end to end argument)
- ISP Service Differentiation
  - Application specific services are offered by some ISPs
- Third Party Involvement
  - Ex. Govt. agencies wants to monitor the traffic

Changes Over The Time

- Internet developed in simpler times
  - Common goals, consistent vision
  - Research curiosity → component of mainstream society
- With success came multiple goals – examples:
  - ISPs must talk to provide connectivity but are fierce competitors
  - Privacy of users vs. government's need to monitor
  - Music lovers wants to exchange recordings with each other but the rights holder wants to stops them
- Argument
  - Must deal with the tussle between concerns in networks' technical architecture

Natures of Engineering and Society

- Engineers: Solve the problems by designing mechanisms with predictable consequences.
- Society: Dynamic management of evolving and conflicting interests.

Tussle Spaces [1]

- Economics
  - Providers tussles as they compete to get customers
  - Customers tussle with providers to get the service at a low price

Solution

- ***Principle of design of choice*** into mechanism is the building block of competition
  - Customers must have the ability to choose (switch) providers freely.

Examples

- Provider lock-in from IP addressing
  - Incorporate mechanisms that make it easy for a host to change address
  - Like you can change cell phone carrier without changing your cell phone number

- Value pricing to improve revenues
  - Divide customers based on their willingness to pay
    - Pay higher rate to run a server at home

Tussle Spaces [2]

- Trust
  - Users do not trust each other
  - Users don't trust parties they actually want to talk to
    - Stealing /gathering information
    - Explicit choice of trusted 3rd party
  - Less and less trust to their own software
    - Browsers gather the information without user's knowledge
  - Host security is questionable? Hence Firewalls are introduced.
- Design for choice: privacy vs. security
  - Users should be able to choose-
    - With whom they interact (Identity….?)
    - What level of transparency they offer to other users

Tussle “Trust”: Examples

- How to prevent DoS attacks?

- How to deal with the tussle about “Firewalls”?

Tussle Spaces [3]

- Openness
  - The openness to innovation that permits a new application to be deployed
  - But economical motivations are against openness
    - Proprietary interfaces give market power
- Vertical integration by ISPs
  - Bundling infrastructure and services
  - Some what restricted but better QoS
- Solution
  - Separate the tussle of vertical integration and sustaining innovation
- Network or Internet neutrality principle. Good or Bad?

New Design Principles?

- Design for variation in outcome
  - Allow design to be flexible to different uses/results
  - Tussle in the design, not by violating the design

- Two specific principles:
  - Modularize the design along tussle boundaries
  - Design for choice to express preferences

Example: Modularization along Tussle boundaries

- Isolate tussles
  - QoS designs uses separate ToS bits instead of overloading other parts of packet like port number
  - Separate QoS decisions from application/protocol design
- Provide choice → allow all parties to make choices on interactions
  - Creates competition
  - Fear between providers helps shape the tussle
  - Example: mail system

Challenges…

- Flexible designs will be complex
  - Applications should be written to deal with this complexity
  - Innovations will be slow

- Flexibility may decrease efficiency
  - Not optimized for all cases

Summary

- Should we rethink about end to end argument?
- Should we rethink about Internet as a transparent network…?
- Can we design a system without any predefined outcome…?
- Can modularization alone solve all problems?

Thank You!