Lesson 1 Introduction, History, and Internet Architecture

The first lesson will dive into the milestones and history of the Internet. Review the protocol, infrastructure, design choices, principles, etc. The lesson will also cover how the Internet would look like if it was **redesigned** today.

In this first lecture, we will start by looking at the devices (bridges and switches for example) that help provide connectivity between hosts in the same network.

Why Study Computer Networks?

Internet Growth

- Started as a research experiment in a lab. Escaped and evolved into a global communications infrastructure that has tremendous impact on human innovation
- Estimated 5.3 billion internet users

Networks Have an Instrumental Role in Society

Changes these things in pivotal ways: - Business - Ecommerce - Advertising - Could computing applications

Essential human connection and communication - Email - Instant messaging - Social networking - Virtual world applications

Fighting/Military - Large scale cyber attacks are related to war - Distribution of mis-information, fake news, etc.

Networking has a lot of opportunities for interdisciplinary research innovations

Ever-expanding/evolving collection of technologies, systems, protocols, architectures, algorithms, etc.

Topics often have ongoing cross-disciplinary innovations - Distributed systems - Operating systems - Computer architecture - Software engineering - Algorithms - Data structures - Graph theory - Queuing theory - etc.

Galactic network (1962)

Proposed by JCR Licklider when he was at MIT in 1962. He envisioned that everyone could quickly access data through a set of interconnected computers.

He was already well know/respected for leading the Defense Advanced Research Projects Agency.

He led a group of researchers to experiment with connecting two computers. MIT researcher, Lawrence G. Roberts connected one computer in MA. to another in CA. via low speed dial-up telephone line

The ARPANET (1969)

The results from the first experiment by Licklider showed that time-shared infrastructure was working sufficiently well.

Researchers indicated the need for packet-switching technology.

The ARPA: Advanced Research Projects Agency Network created a network interlinked 4 distinct notes at UCLA, Stanford, UCSB, and University of Utah

Network Control Protocol (NCP), initial APRANET host-to-host protocol (1970)

As the number of computers on ARPANET increased rapidly, protocol design was underway. ARPANET used host-to-host called NCP.

Allowed the network users to begin developing applications. Email was created from this in 1972.

Internetworking TCP/IP (1973)

DARPA team of researchers created the idea of open-architecture networking

Enabled individual networks to be independently designed and developed in accordance with specific environment and user requirements of that network.

New version of NCP protocol created. Coined: Transmission Control Protocol / Internet Protocol (TCP/IP). *Created by Bob Kahn and Vint Cerf at Stanford*

The first version of TCP later split its functionalities into two protocols.

IP was used only for addressing and forwarding individual packets, and the separate TCP was used fo service features such as flow control and recovery from lost packets.

The Domain Name System (DNS) and the World Wide Web WWW (1983, 1990 *respectively*)

The scale of the internet was increasing rapidly.

It was not feasible to have a single table of all hosts to store names and addresses.

Domain Name System (DNS) was designed to translate domain names to IP addresses by a scalable distributed mechanism. *introduced by Paul Mockapetris at USC 1983*

The World Wide Web (WWW) followed quickly after leveraging the new growth potential that DNS brought to the internet.

What is a Computer Network?

A network is built on top of many components that can vary in technologies, while at the same time it can offer different types of services.

Suppose two clients are communicating via a given protocol: Email. They can communicate despite using different tech stacks. WiFi, IP, Internet.

How do all these things interconnect to deliver content?

The designers of the Network Protocols provide structure to the network architecture by organizing each protocol into different layers. Which also allows the functionality of each protocol to be offered by different layers. Each layer offers different services.

Example: Airline System

DEPARTURE

- Ticket (purchase)
- Baggage (check)
- Gates (load)
- Runway takeoff

INTERMEDIATE AIR TRAFFIC CONTROL

- Airplane routing

ARRIVAL

- Runway landing
- Gates (un loading)
- Baggage (claim)
- Ticket (complain)

The OSI Model

Every layer provides service to something above

1. Physical Layer
2. Data Link Layer
3. Network Layer
4. Transport Layer
5. Session Layer
6. Presentation Layer
7. Application Layer

The traditional internet model only has 5 layers:

1. Physical Layer
2. Data Link Layer
3. Network Layer
4. Transport Layer
5. Application Layer

The interface between the application layer and the transport layer are known as **sockets**. It is up to the Engineer to design the functionality of the overall application and use of these connections.

Application Layer

The application layer includes multiple protocols here are some:

- HTTP (web)
- SMTP (email)
- FTP (transfers files between hosts)
- DNS (translates domain names to IP addresses)

At the application layer, **packets of information are known as messages**.

Presentation Layer

Plays the intermediate role of formatting the information that it receives from the layer below (session and transport layers) and delivering it to the application layer.

examples: video stream, translating integers from big edian to little edian.

Session Layer

Responsible for the mechanism that manages the different transport streams that belong to the same session between end-user processes.

example: teleconference applications, the session layer ties the audio stream to the video stream (syncs)

Transport Layer

Responsible for end-to-end communication between hosts. Supports two protocols:

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

TCP offers *connection-oriented service* to the applications on the layer above. Provides **guaranteed delivery of the application-layer messages, flow control, congestion control**

TCP Flow Control: matches the sender's and receiver's speed

Congestion Control: sender slows its transmission rate when it perceives the network to be congested.

UDP Provides a connectionless, best effort service to the applications that are running in the layer above

There is no reliability, flow control, or congestion control.

Packets of information are known as **segments**.

Network Layer

Packets of information are known as **datagrams**.

Responsible for **moving datagrams from one internet host to another**. [Source Host] sends the segment (along with destination address) from the transport layer ot the network layer. Delivers the datagram to the **transport layer** at the [Destination Host].

This layer has Internet Protocol: IP and the Routing Protocols.

IP is **the glue** that binds the internet together. **All internet hosts** and devices that have a network layer **must run the IP protocol**.

This defines the fields in the datagram and how the Source/Destination hosts and the intermediate routers use these fields so that the datagrams that a Source Internet Host sends reach their destination.

It is a routing protocol that determines the routes that the datagrams can take between sources and destinations.

Data Link Layer

Refers to packets of information as **frames**.

examples of Data Link Layer protocols: - WiFi - Bluetooth - Ethernet - Point-to-Point Protocol (PPP)

Responsible for moving the frames from one node (or host/router on network) to the next node.

Assuming there is a SENDER and RECEIVER host, the network layer will route datagrams through multiple routers across the path between the SENDER and RECEIVER hosts.

At each node in the network, the Network Layer, passes the datagram to the Data Link Layer, which delivers the datagram to the next node. At the node, the Link Layer passes the datagram up to the Network Layer.

Data Link Layer offers services that depend on the data link layer protocol.

examples: - Reliable delivery - converts data transmission from one node across one link to a receiving node - *this reliable delivery service differs from TCP because TCP offers reliability from source host to destination end host*

Quiz

> [TRUE or FALSE] Some data link layer protocols, such 802.11 (WiFi), implement some basic error correction as the physical medium used is easily prone to interference and noise (such as a nearby running microwave).
> **Is this a violation of the end-to-end principle?**

Answer

false

> No, because violations of the e2e principle typically refer to scenarios where it is not possible to implement a functionality entirely at the end hosts, such as NAT and firewalls.
> In this question, we have a lower level protocol implementing error checking.

Physical Layer

Facilitates the interaction with hardware.

Responsible for transferring bits within a frame between two nodes that are connected through a physical link.

example: Ethernet, a main protocol in Data Link layer, has different physical layer protocols for **twisted-pair copper wire, coaxial cable, and single-mode fiber optics.**

Layers Encapsulation

How do the layers and protocols that run on each layer communicate with each other?

Encapsulation and De-encapsulation

HOST sends an application layer message M to the Transport Layer.

Transport Layer appends the Transport Layer Header Information (HT)

The Application Message and Transport Layer Header Information together is called a **segment** also known as the **Transport Layer Segment**.

The segment **encapsulates** the Application Layer message. This information helps the RECEIVING HOST: - inform the RECEIVER-side Transport layer which application to deliver the message to - perform error detection - determine whether bits in the message have changed during transmission

Basically, **data integrity** and **who should receive the message**.

Intermediate Devices and Encapsulation

This is the path that connects the SENDING and RECEIVING hosts.

Hosts may be: - Layer 3 devices: **Routers** - Layer 2 devices: **Switches**

Design Choice

END HOSTS implement all 5 layers while intermediate devices (routers, switches) do not. *This ensures that the Internet architecture puts most of the complexity and at the edges of the network, which keeps the CORE simple*

End to End Principle

Specific application level functions **ARE NOT** built into lower levels of the system/at the core of the network.

E2E Principle: The *network core* should be simple and minimal. End systems should carry the intelligence.

Reasoning: Most functions/features can only be completely implemented at the endpoints of networks. This makes it trivial to try and include these extra complexities within the CORE NETWORK infrastructure.

This design decision allowed the Internet to grow rapidly. Since the edge systems were able to innovate and create numerous applications/services, rather than be limited by the middle of the network which is harder to modify.

Moving functions and services closer to the applications that use them increases the flexibility and autonomy of the application designer to offer these services to the needs of the specific application.

Higher level protocol/OSI layers are free to organize the lower-level network resources to achieve application design

Violations of the E2E Principle

E2E has been adopted to help the internet grow, but there are some edge cases that break this rule.

Examples

- Firewalls

- Traffic Filters

Firewalls usually exist on the edges of a network. They monitor network traffic as it enters the local network. The *firewall* will allow traffic through, or drop traffic flagged/deemed malicious.

Firewalls violate E2E because they are an intermediate device operated between two end hosts, but possess the ability to dop connection between hosts.

Network Address Translation (NAT) helps mitigate issues with shortage of internet addresses. An ISP assigns a single IP address to the entire network to interface with the public internet. *[example: 52.125.24.9]*

The hosts/nodes behind the router communicating with the ISP/Public Internet receive a **Private IP Address** to address the device within the private subnet. *[example: 10.0.0/8, 192.168.0.0/24]* This addressing technique allows for hundreds or thousands of hosts behind a single public IP (within a private network).

The private hosts are always behind a NAT, the NAT handles communication between the hosts on the private network and the hosts on the public Internet.

All traffic that leaves the home router/IP and is destined to hosts on the Public Internet must have the source address set to the IP of the Public Facing Interface of the NAT-enabled router.

The router is a translator, it maintains a NAT translation table and records SOURCE and DESTINATION IP addresses and ports. The **Translation Table (NAT Table)** provides a mapping between *public facing IP address/ports* and the *private address/ports*.

Why do NAT Boxes Violate E2E?

Hosts behind NAT boxes are not globally addressable/route-able. It is not possible for other hosts on the public-facing Internet to initiate connections to these devices.

Hosts behind a NAT and host on the public internet CANNOT communicate be default without the intervention/intermediary of a NAT box.

Other Workarounds:

- Session Traversal Utilities for NAT (STUN)
  - enables hosts to discover NATs and the public IP address and port number that the NAT has allocated for the applications
- UDP Hold Punching
  - establishes bidirectional UDP connections between hosts behind NATs

Hourglass Internet Shape/Architecture

Currently, there is only a few protocols at the Network Layer, while each layer higher AND lower in the OSI model have many protocols. This creates an "Hourglass" shape based on the number of protocols per OSI layer

In the early 1990's, there were several other network layer protocols that competed with IPV4 - Novell's IPX - X.25

New protocols in the Network Layer would have to have significant advantages over the existing [IPV4, TCP, UDP].

Researchers have suggested a model called **Evolutionary Architecture Model (EvoArch)** which can help to study layered architectures and their evolution in a quantitative manner. Researchers explain the hierarchical structure of the layered architecture eventually led to the hourglass shape.

EvoArch

Layers A protocol stack is modeled as a directed, acyclic network with L layers.

Nodes Each network protocol is represented as a node. The layer of a Node u is denoted by `l(u)`.

Edges Dependencies between protocols are represented as directed edges.

Node incoming edges If a protocol u at layer L uses the service provided by a protocol w at the lower layer L-1, then this is represented by an "upwards" edge from w to u.

Node substrates We refer to substrates of a node u, `S(u)`, as a set of nodes that u is using their services. Every node has at least one substrate, except the nodes at the bottom layer.

Node outgoing edges The outgoing edges from a anode U terminate at the products of u. The products of a node u are represented by `P(u)`

Layer generality Each layer is associated with a certain probability `s(L)`. A node u at layer L+1 independently selects each node of layer L as the substrate with probability `s(l)`.

example: Internet protocol stack, layer 1 is highly general, and the protocols at this layer offer a general bit transfer service between two connected points, which most higher-layer protocols would use.

Node evolutionary value The value of a protocol node, `v(u)`, is computed recursively based on the products of u. The model captures the value of protocol u, which is driven by values of the protocols that depend on it.

example: TCP has a high evolutionary value because it is used by many higher-layer protocols
If a new protocol was introduced at the Network Layer (same as TCP) that has better performance, the **evolutionary value** of the new protocol will be low if it is not used by important/popular higher-layer protocols.

This metric predicts how well a given protocol will *survive* in regards to the competition with other protocols at the same layer.

Node competitors and competition threshold Competitors of node u are known as `C(u)`, nodes at layer L, share at least a fraction c of node u's products. The fraction c is the competition threshold.
New nodes w compete with node u if w shares at least a fraction c of u's productions.

Node death rate The model tracks a "death rate" for protocols that cease, or are introduced.

Competition among nodes become more intense, it is more likely that a protocol u does if at least one of its competitors has a higher value that itself.

When a node u does, its products also die if their only substrate is u.

Node basic birth process A new node is assigned randomly to a layer. The number of new nodes at a given time is set to a small faction, *e.g. 1% - 10%*, of the total number of nodes in the network at a given time.
So, the larger a protocol stack is, the faster it grows.

EvoArch Example (Toy)

An example L has 4 layers. The evolutionary value of each node is shown inside each circle. The generality probability ofr each layer is shown at the left of each layer, denoted at `s(L)`. *The generality of each layer decreases as we move to higher layers*, on average, the number of products per node decreases.

Assume, competition threshold: `c = 3/5`. Nodes u, q, and w, compete in layer 2. Nodes u and q compete, but its unlikely to cause q to die because u and q have comparable evolutionary values. In contrast, it is likely that w will die because its value is much less that the maximum value of its competitor: u.

EvoArch iterations

read further here:
https://gatech.instructure.com/courses/356044/pages/evolutionary-architecture-

model?module_item_id=3650132

Implications for Internet Architecture and Future Internet Architecture

EvoArch explains the survival of the TCP/IP stack through the period of rapid development [70's and 80's]. TCP/IP was not competing with the Telephone network, instead it was used to support popular early internet services like: - FTP - Email - Telnet

This distinction helped it grow and increase its value without competing with the telephone network. As additional services were created, they also leveraged TCP/IP protocols, further strengthening the TCP/IP protocols.

IPV4, TCP, UDP provided a stable framework as protocols in the lower layers (Data Link, Physical Layers) as well as new applications and services being created in the higher layers.

EvoArch outlines why an *explosion* of new protocols/services at a higher layer inherently kills/destroys protocols at a lower layers

Quiz

Which of the following are ramifications of the "hourglass shape of the internet"?

- A. Many technologies that were not originally designed for the internet have been modified so that they have versions that can communicate over the internet (such as Radio over IP).

- B. It has been a difficult and slow process to transition to IPv6, despite the shortage of public IPv4 addresses.

- C. Applications like BitTorrent leverage peer-to-peer networking instead of a more traditional client-server model for better performance.

Answer

A, B

A is correct
Modifying a technology so that it is compatible with the rest of the internet (i.e., by making it compatible with IP) greatly enhances market penetration (from the vendor's perspective), and/or decreases the amount of extra development that would need to happen.

B is correct
A big part of the Internet infrastructure uses IPV4 while the cost of transitioning is high. This reflects as a consequence of the narrow waist.

C is not relevant here
The hourglass shape of the Internet refers to Internet architecture in terms of protocols available at the different layers.

[OPTIONAL] Architecture Redesign

Read more here:
https://gatech.instructure.com/courses/356044/pages/optional-reading-architecture-redesign?module_item_id=3650136

Interconnecting Hosts and Networks

There are numerous devices that provided connectivity between hosts within the same network. These devices offer different services and operate on different OSI layers

Repeaters and Hubs

Operate on Physical Layer (L1)

They receive and forward digital signals to connect different Ethernet segments.

Advantage

- Simple and inexpensive devices
- Can be arranged hieratically

Disadvantages

- Hosts that are connected through *Repeaters and Hubs* belong to the same collision domain
  - they compete for access to the same link

Bridges and L2 Switches

Enable communication between hosts that are not directly connected.

Operate on Data Link Layer (L2) based on MAC addresses. They receive packets and forward them to the appropriate destination.

Advantages source

- MAC Address Leading
  - When a bridge receives a frame from a device on one of its connected segments, it examines the source MAC address of the frame and associates it with the port on which it was received. This process allows the bridge to build a table of MAC addresses and their corresponding port locations.
- Filtering and Forwarding
  - When a frame is received, the bridge checks the destination MAC address against its MAC address table. If the destination address is known and located on the same segment as the source, the bridge filters the frame and does not forward it. If the destination address is on a different segment, the bridge forwards the frame only to the appropriate segment.
- Segment Isolation
  - Bridges help in segmenting a network into smaller collision domains. In traditional Ethernet networks, all devices on the same segment share the available bandwidth, and collisions can occur. By dividing a network into multiple segments connected by bridges, collisions are reduced, improving overall network performance.
- Loop Prevention
  - One significant challenge in bridged networks is the potential for loops. Loops can cause broadcast storms and negatively impact network performance. To prevent loops, many bridges implement the Spanning Tree Protocol (STP), which dynamically detects and eliminates redundant paths.
- Transparent Operation
  - Bridges operate transparently to connected devices. Devices on the network are unaware that a bridge is present, as the bridge operates at a lower layer of the OSI model and does not impact the network's IP addressing or higher-layer protocols.

Disadvantages

- Finite bandwidth
  - if the arrival rate of the traffic is higher than the capacity of the unit, packets are temporarily stored in buffers
  - if the buffer space gets full, packets are dropped

Routers and L3 Switches

Operate on Network Layer L3. Discussed more in upcoming lecture on Routing Protocols.

Learning Bridges

Bridge

Device with multiple inputs and outputs. Transfers frames from ONE input to ONE OR MORE outputs.

Does NOT forward all frames

Learning Bridge

Learns, populates, maintains a forwarding table.

The forwarding table is consulted/referenced when forwarding frames to specific ports so that traffic is not spammed to all ports.

How does this work? Each frame received is a **learning opportunity** to know which hosts are reachable through which ports.

The bridge can view the port over which a frame arrives and the source host. The bridge can construct a **forwarding table** mapping each host to a port.

Looping Problem in Bridges and The Spanning Tree Algorithm

Using bridges to connect to LANs fails if the network has loops/cycles. This means that bridges would loop through packets *infinitely*

Spanning Tree Algorithm solves this problem by identifying links that lead to loops and excluding them from the topology.

Construct a network as a graph where *Bridges* are **nodes** and *Links* between bridges are **edges**, **Spanning Tree Algorithm** is used to have the *bridges* select which *links (ports)* to use for forwarding without enuring loops.

The Algorithm

Every **node/bridge** in the graph has an ID

Base Case: Select a single **node/bridge** as the root

For each iteration, each node sends all neighboring nodes a configuration message with 3 fields:

1. The sending node's ID
2. The ID of the root node
3. The distance between the root and the current node

During the first iteration, each node will think it is the root node. *This is because the configuration message should have a distance of 0.*

Each message received from a new iteration will be compared to the existing message. The message will be overridden based on these cases:

1. The root of the configuration has a smaller ID
2. The roots have equal IDs, but one configuration indicates a smaller distance to the root
3. Both root IDs and distances to the root are equal, the node breaks the tie by selecting the configuration of the sending node that has the smallest ID

The node stops sending configuration messages over a link/port if:

1. Either closer to the root
2. Has the same distance from the root, but smaller ID

Quiz

Which of the following statements are correct?

- A. The Spanning Tree Algorithm helps to prevent broadcast storms
- B. The Spanning Tree Algorithm presented in this lecture always results in a spanning tree that places the root in a topologically central location, so that all the nodes are as "close" as possible to the root.
- C. Network traffic cannot traverse an inactive link

Answer

A

A is correct. That is the purpose of the Spanning Tree Algorithm. Although it is still possible to have broadcast storms on the network (such as from a bad network card), STP prevents broadcast storms that result from having loops present in the network topology.

B is incorrect. The Spanning Tree Algorithm presented guarantees a unique spanning tree that all the nodes will agree to, but sometimes this isn't the most "optimal" tree possible. Network administrators can configure the switch ID if they want to have a specific spanning tree.

C is incorrect. Traffic can still reach the link, but the link is not used to forward traffic.
