Topic 2 – Internet and Architecture

- Protocol Standardization
- Internet Philosophy and Tensions
- The architects process
  - How to break system into modules
  - Where modules are implemented
  - Where is state stored

2

Recall What is a protocol?

human protocols:

- "what's the time?"
- "I have a question"
- introductions

… specific msgs sent

… specific actions taken when msgs received, or other events

network protocols:

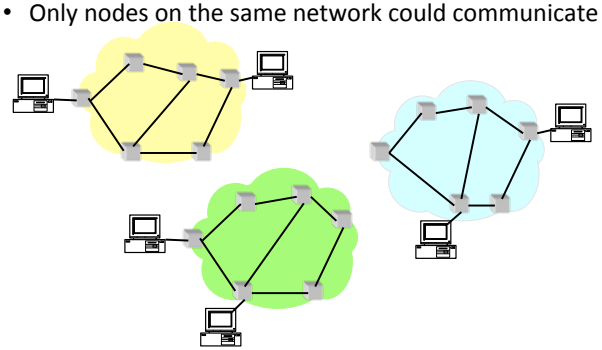
- machines rather than humans
- all communication activity in Internet governed by protocols

protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt

3

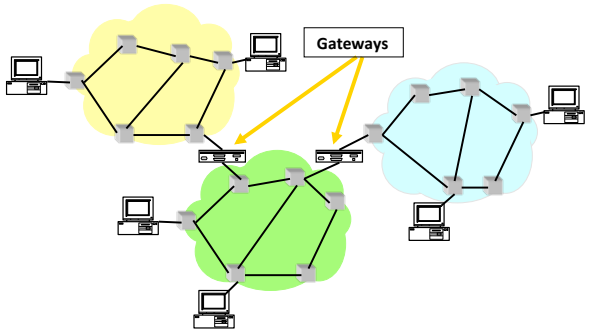
So many Standards Problem

- Many different packet-switching networks
- Each with its own Protocol
- Only nodes on the same network could communicate



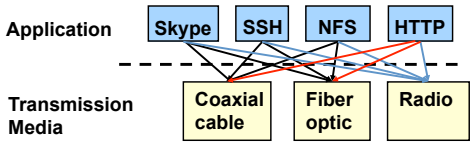
4

INTERnet Solution



5

A Multitude of Apps Problem

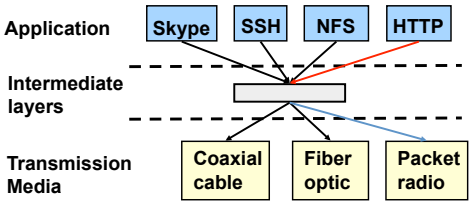


- Re-implement every application for every technology?
- No! But how does the Internet design avoid this?

6

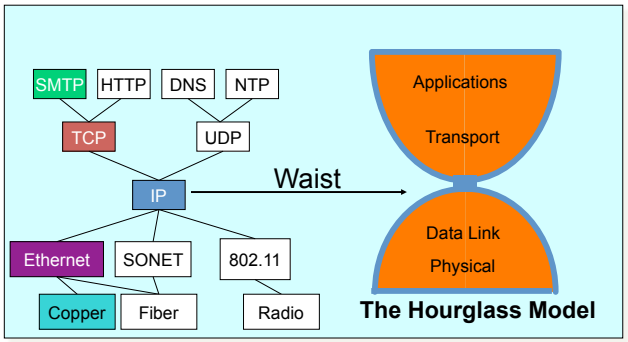
Solution: Intermediate Layers

- Introduce intermediate layers that provide set of abstractions for various network functionality and technologies
  - A new app/media implemented only once
  - Variation on "add another level of indirection"



7

The Internet *Hourglass*



There is just one network-layer protocol, **IP**.
The "narrow waist" facilitates interoperability.

8

Protocol Standardization

- All hosts must follow same protocol
  - Very small modifications can make a big difference
  - Or prevent it from working altogether
  - Cisco bug compatible!
- This is why we have standards
  - Can have multiple implementations of protocol
- Internet Engineering Task Force
  - Based on working groups that focus on specific issues
  - Produces "Request For Comments" (RFCs)
  - IETF Web site is ***http://www.ietf.org***
  - RFCs archived at ***http://www.rfc-editor.org***

9

Internet Motto

We reject kings, presidents, and voting. We believe in rough consensus and running code."

David Clark

D. Clark, "The Design Philosophy of the DARPA Internet Protocols", Sigcomm'88, 106-114, Palo Alto, CA, Sept 1988.

10

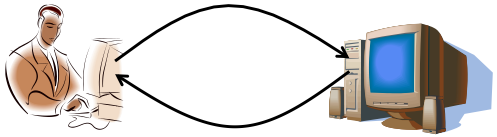
Alternative to Standardization?

- Have one implementation used by everyone
- Open-source projects
  - Which has had more impact, Linux or POSIX?
- Or just sole-sourced implementation
  - Skype, many P2P implementations, etc.

11

Client-Server Communication

- Client "sometimes on"
  - Initiates a request to the server when interested
  - E.g., Web browser on your laptop or cell phone
  - Doesn't communicate directly with other clients
  - Needs to know the server's address
- Server is "always on"
  - Services requests from many client hosts
  - E.g., Web server for the *www.cnn.com* Web site
  - Doesn't initiate contact with the clients
  - Needs a fixed, well-known address



12

Peer-to-Peer Designs

- No always-on server at the center of it all
  - Hosts can come and go, and change addresses
  - Hosts may have a different address each time
- Example: peer-to-peer file sharing
  - All hosts are both servers and clients!
  - Scalability by harnessing millions of peers
  - "self-scaling"
- Not just for file sharing!
  - This is how many datacenter applications are built
  - Better reliability, scalability, less management…
    - Sound familiar?

13

Internet Design Goals (Clark '88)

- **Connect existing networks**
- Robust in face of failures
- Support multiple types of delivery services
- Accommodate a variety of networks
- Allow distributed management
- Easy host attachment
- Cost effective
- Allow resource accountability

14

Connect Existing Networks

- Internet (e.g., IP) should be designed such that all current networks could support IP.

15

Robust

- As long as the network is not partitioned, two endpoints should be able to communicate
- Failures (excepting network partition) should not interfere with endpoint semantics

- *Very successful, not clear how relevant now*
- *Second notion of robustness is underappreciated*

16

Types of Delivery Services

- Use of the term "communication services" already implied an application-neutral network
- Built lowest common denominator service
  - Allow end-based protocols to provide better service
- Example: recognition that TCP wasn't needed (or wanted) by some applications
  - Separated TCP from IP, and introduced UDP

17

Variety of Networks

- Incredibly successful!
  - Minimal requirements on networks
  - No need for reliability, in-order, fixed size packets, etc.
  - A result of aiming for lowest common denominator
- IP over everything
  - Then: ARPANET, X.25, DARPA satellite network..
  - Now: ATM, SONET, WDM…

18

Decentralized Management

- Both a curse and a blessing
  - Important for easy deployment
  - Makes management hard today

19

Host Attachment

- Clark observes that cost of host attachment may be higher because hosts have to be smart
- But the administrative cost of adding hosts is very low, which is probably more important

20

Cost Effective

- Cheaper than telephone network
- But much more expensive than circuit switching
- Perhaps it is cheap where it counts (low-end) and more expensive for those who can pay….

21

Resource Accountability

- Failure!
  - No coordinated resource accounting
  - No coordinated resource management
  - No coordinated resource control
  - No coordinated resource ….

BUT Failure is information too

22

Real Goals

Internet Motto

We reject kings , presidents, and voting. We believe in rough consensus and running code." – David Clark

- **Build something that works!**
- Connect existing networks
- Robust in face of failures
- Support multiple types of delivery services
- Accommodate a variety of networks
- Allow distributed management
- Easy host attachment
- Cost effective
- Allow resource accountability

23

Questions to think about….

- What priorities would a commercial design have?
- What would the resulting design look like?
- What goals are missing from this list?
- Which goals led to the success of the Internet?

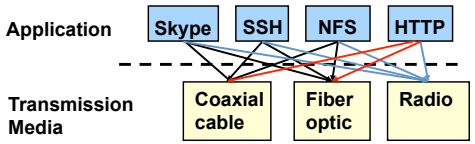
24

The Networking Dilemma

- Many different networking technologies
- Many different network applications
- How do you prevent incompatibilities?

25

The Problem

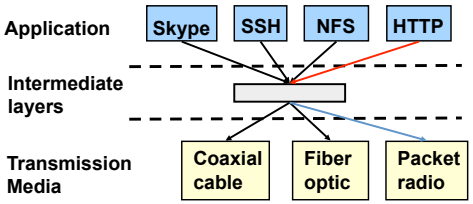


- Re-implement every application for every technology?
- No! But how does the Internet design avoid this?

26

Solution: Intermediate Layers

- Introduce intermediate layers that provide set of abstractions for various network functionality and technologies
  - A new app/media implemented only once
  - Variation on "add another level of indirection"



27

Network Architecture

- Architecture is not the implementation itself
- Architecture is how to organize/structure the elements of the system and their implementation
- What *interfaces* are supported?
  - Using what sort of abstractions
- *Where* functionality is implemented?
  - The modular design of the network

28

Computer System Modularity

Partition system into modules & abstractions:

- Well-defined interfaces give flexibility
  - ***Hides*** implementation - can be freely changed
  - Extend functionality of system by adding new modules
- E.g., libraries encapsulating set of functionality
- E.g., programming language + compiler abstracts away how the particular CPU works …

29

Computer System Modularity (cnt'd)

- Well-defined interfaces hide information
  - Isolate assumptions
  - Present high-level abstractions
- **But can impair performance!**
- Ease of implementation vs worse performance

30

Network System Modularity

Like software modularity, but:

- Implementation is distributed across many machines (routers and hosts)
- Must decide:
  - How to break system into modules
    - **Layering**
  - Where modules are implemented
    - **End-to-End Principle**
  - Where state is stored
    - **Fate-sharing**

31

Remember that slide!

- The relationship between architectural principles and architectural decisions is crucial to understand

32