Internet: The Big Picture

Prof. Anja Feldmann, Ph.D.

Internet design philosophy (Clark '88)

In order of importance

- *Connect existing networks*
  - *Initially ARPANET and ARPA packet radio network*
- *Survivability*
  - *Ensure communication service even under network/router failures*
- *Support multiple types of services*
- *Must accommodate a variety of networks*
- *Allow distributed management*
- *Allow host attachment with a low level of effort*
- *Be cost effective*
- *Allow resource accountability*

Internet design philosophy (Clark ’88)

In order of importance

- *Connect existing networks*
  - *Initially ARPANET and ARPA packet radio network*
- *Survivability*
  - *Ensure communication service even under network/router failures*
- *Support multiple types of services*
- *Must accommodate a variety of networks*
- *Allow distributed management*
- *Allow host attachment with a low level of effort*
- *Be cost effective*
- *Allow resource accountability*

Different ordering of priorities may make a different architecture!

Survivability

Continue to operate even under network failures (e.g., link/router failures)

- As long as network is not partitioned, two endpoints *should* be able to communicate
- Any other failure (*except* network partition) should be ***transparent*** to endpoints

Decision

- *Maintain end-to-end transport state only at end-points!*
- Eliminates problem of handling state inconsistency and performing state restoration when router fails

Internet

- ***Stateless network architecture***
- No notion of a session/call at network layer

Survivability

Continue to operate even under network failures (e.g., link/router failures)

- As long as network is not partitioned, two endpoints *should* be able to communicate
- Any other failure (*except* network partition) should be ***transparent*** to endpoints

Decision

- *Maintain e*
- Eliminates restoration
  when rout

Grade? *A-*

Because convergence times are relatively slow!

- *BGP takes minutes to converge.*
- *IS-IS OSPF takes ~10 seconds.*

Internet

- ***Stateless network architecture***
- No notion of a session/call at network layer

Types of Services

Add UDP to TCP to better support apps

- e.g., “real-time” applications
- Arguably main reason for separating TCP, IP

- Datagram abstraction
  - Lowest common denominator on which services can be built
  - ***Service differentiation*** *was* considered (remember ***ToS***?), but it never happened on a large scale (Why?)

Types of Services

Add UDP to TCP to better support apps

- e.g., “real-time” applications
- Arguably main reason for separating TCP, IP

- Datagram ab
    - Lowest co
    - ***Service dif*** ened on a large scale (Why

Grade? A-

Proven to allows lots of application to be invented and flourish.

- *Except multimedia; perhaps that is not a transport service issue!*

Variety of Networks

Very successful !?

- Because of the minimalist service:
- It requires from underlying network only to deliver a packet with a “reasonable” probability of success

Does not require …

- *Reliability*
- *In-order delivery*

The mantra: IP over everything

- *Then: ARPANET, X.25, DARPA satellite network, ...*
- *Now: ATM, SONET, WDM, …*

Variety of Networks

Very successful !?

- Because of the minimalist service:
- It requires from underlying network only to deliver a packet with a "reasonable" probability of success

Does not requ

- *Reliability*
- *In-order de*

Grade? *A*

Cannot name a link layer technology that IP does not run over!

- *Carrier pigeon RFC*

The mantra: IP over everything

- *Then: ARPANET, X.25, DARPA satellite network, …*
- *Now: ATM, SONET, WDM, …*

Distributed Management

Administrative autonomy

- IP interconnects networks

- Each network can be managed by a different organization
- Different organizations need to interact only at the boundaries

- … but this model *complicates* routing

Distributed Management

Administrative autonomy

- IP interconnects networks

- Each network can be managed by a different organization
- Different c

- … but this

Grade? A/B

***A** for implementation; **B** for concept (disagreement)*

Cost Effectiveness

Sources of inefficiency

- *Header overhead*
- *Retransmissions*
- *Routing*

- … but *“optimal”* performance has *never* been top priority!

Cost Effectiveness

Sources of inefficiency

- *Header overhead*
- *Retransmissions*
- *Routing*

Grade? ***A***

… … …

- … but ‘ iority!

Low cost of attaching a new host

Not a strong point!

  - Higher than other architecture because the *intelligence is in hosts* (e.g., telephone vs. computer)

- Bad implementations or malicious users can produce considerably harm (remember *fate-sharing*?)

Low cost of attaching a new host

Not a strong point!

- Higher than other architecture because the *intelligence is in hosts* (e.g., telephone vs. computer)

- Bad imple duce considera

Grade? C

But things are improving with DHCP, autoconfigurations.

- *A higher grade may be possible some time in the future*

Accountability

… !? …

Accountability

… !? …

Grade? F

… !? … !? … !? …

What about the future?

Datagram *not* the best abstraction for …

  - Resource management, accountability, QoS

New abstraction: ***Flow*** (see OpenFlow, IPv6)

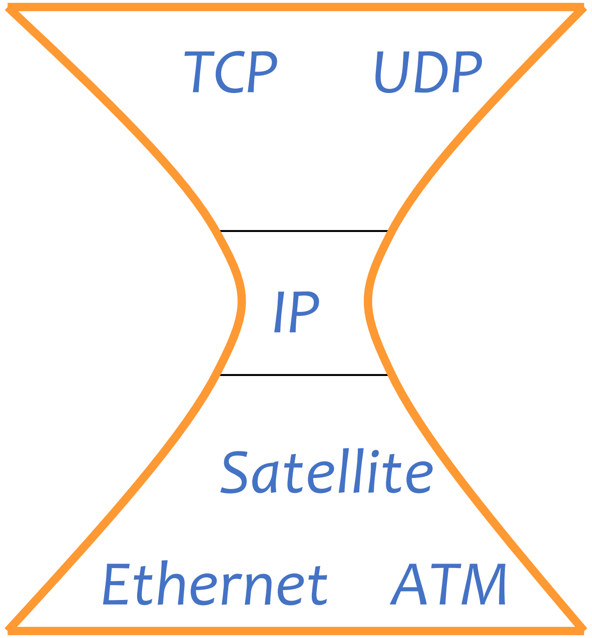
  - But no one knows what a flow is!

- Routers require to maintain per-flow state

- State management:
  - Recall: Recovering lost state is hard
  - Here we see proposals for "soft state"!
  - *Soft-state*: End hosts responsible to maintain the state

Summary: Internet architecture

- ***Packet-switched*** datagram network
- IP is the ***glue*** (network layer overlay)
- IP ***hourglass*** architecture
  - All hosts and routers run IP
- ***Stateless*** architecture
  - No per-flow state *(except for multicast)*



IP hourglass

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

But that was ***yesterday***!

… *what about* **tomorrow**?

Rethinking Internet design

What has changed?

- ***Operation in untrustworthy world***
  - Endpoints can be malicious
  - If endpoint not trustworthy, but want trustworthy network → more mechanism in network core
- ***More demanding applications***
  - End-end best effort service not enough
  - New service models in network (Intserv, diffserv)?
  - New application-level service architecture built on top of network core (e.g., CDN, VPNs)?

Rethinking Internet design

What has changed?

- ***ISP service differentiation***
  - ISP doing more (than other ISPs) in core maybe a competitive advantage
- ***Rise of third-party involvement***
  - Interposed between endpoints (even against will)
  - e.g., Chinese Gov't, US recording industry
- ***Less sophisticated users***

Rethinking Internet design

- *Operation in untrustworthy world*
- *More demanding applications*
- *ISP service differentiation*
- *Rise of third-party involvement*
- *Less sophisticated users*

All five changes may motivate shift away from end-end!

What’s at stake?

“At issue is the conventional understanding of the ‘Internet philosophy’

- Freedom of action
- User empowerment
- End-user responsibility for actions taken
- Lack of control “in” the net that limit or regulate what users can do

The end-end argument fostered that philosophy because they enable the freedom to innovate, install new software at will, and run applications of the users’ choice.”

[Blumenthal and Clark, 2001]

Technical response to changes

- ***Trust:*** *Emerging distinction between what is "in" network (us, trusted) and what is not (them, untrusted).*
  - Ingress filtering
  - Emergence of Internet UNI (user network interface, as in ATM)?

- ***Modify endpoints***
  - Harden endpoints against attack
  - Endpoints do content filtering: Net-nanny
  - *CDN, ASPs:* Rise of structured, distributed applications in response to inability to send content (e.g., multimedia, high bandwidth) at high quality

Technical response to changes

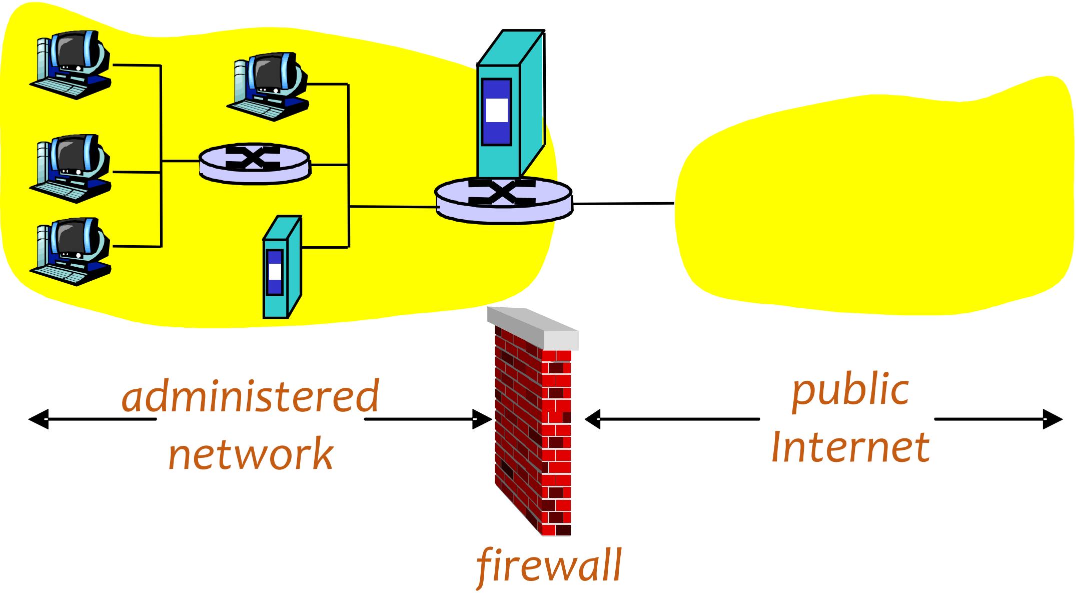
- ***Add functions to the network core***
  - Filtering firewalls
  - Application-level firewalls
  - NAT boxes
  - Network virtualization

… all operate within network, making use of application-level information

- Which addresses can do what at application level?
- If addresses have meaning to applications, NAT must “understand” that meaning

Firewalls

- Isolates organization's internal network from larger Internet, allowing some packets to pass, blocking others.



Firewalls: Why?

Prevent denial of service attacks

- ***SYN flooding***: Attacker establishes many bogus TCP connections, no resources left for “real” connections

Prevent illegal modification/access of internal data

- E.g., attacker replaces CIA’s homepage with something else

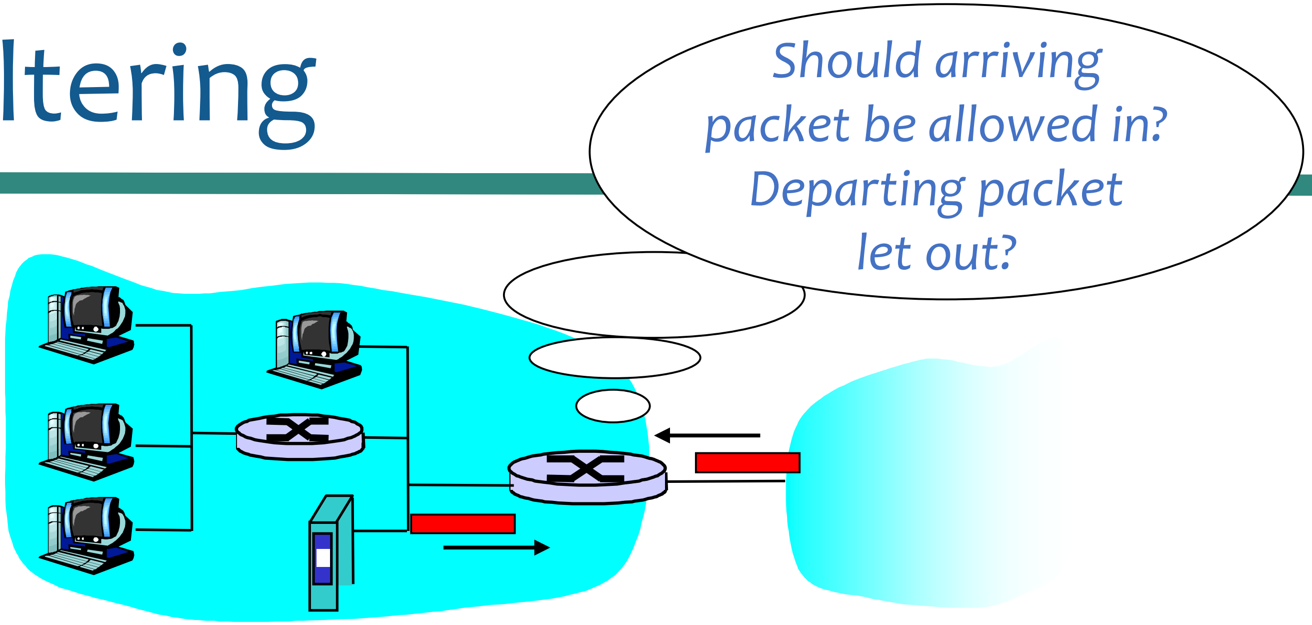
Allow only authorized access to inside network

- Set of authenticated users/hosts

Two types of firewalls

- *Application-level*
- *Packet-filtering (stateless/stateful)*

Packet filtering


Should arriving packet be allowed in? Departing packet let out?

- Internal network connected to Internet via ***router firewall***

- Router ***filters packet-by-packet***, decision to forward/drop packet based on:
  - Source IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - ICMP message type
  - TCP SYN and ACK bits

Packet filtering

Example 1: Block incoming and outgoing datagrams with IP protocol field = 17 and with either src or dst port = 23

- All incoming and outgoing UDP flows and telnet connections are blocked

Example 2: Block inbound TCP segments with ACK=0

- Prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside

KISS principle

- Success of LAN protocols, RISC architecture: *KISS!*
  - “Building complex functions into network optimizes network for small number of services, while substantially increasing cost for uses unknown at design time”
  - “End-end argument does not oppose active networks per se but instead strongly suggests that enthusiasm for the benefits of optimizing current application needs by making the network more complex may be misplaced”

Will IP take over the world?

Reasons for success of IP

Reachability

- Reach every host, adapts topology when links fail

Heterogeneity

- Single service abstraction (best effort) regardless of physical link topology

Many other claimed (or commonly accepted) reasons for IP's success may not be true!

… let's take a closer look

IP already dominates global communication?

Business revenues

- ISPs: *13B*
- Broadcast TV: *29B*
- Cable TV: *29.8B*
- Radio broadcast: *10.6B*
- Phone industry: *268B*

Router/telco switch markets

- Core router: *1.7B*; edge routers: *2.4B*
- SONET/SDH/WDM: *28B*, Telecom MSS: *4.5B*

IP is more efficient?

- *Statistical multiplexing VS circuit switching!*
- *Link utilization*
  - Avg. link utilization in Internet core: *3% to 30%*
  - Avg. utilization of Ethernet is currently: *1%*
  - Avg. link utilization of long-distance phone lines: *33%*
- *Low IP link utilization:* ***On purpose!***
  - Predictability, stability, low delay, resilience to failure

At low utilization, we forfeit benefits of statistical multiplexing!

IP is more robust?

- Median IP network availability: *(Downtime)* *471 min/yr*
- Avg. phone network downtime: *5 min/yr*

- Convergence time with link failures:
  - SONET: *50 ms*
  - BGP: *3 – 15 minutes*

- *Inconsistent routing state*
  - Human misconfigurations
  - In-band signaling (signaling and data share same network)
  - Routing computation “complex”

IP is simpler?

Intelligence at edge, simplicity in core

- Telephone switch: *3M lines of code*
- Cisco IOS: *8M lines of code*

Line-card complexity

- *Router:* 30M gates in ASICs, 1 CPU, 300M packet buffers
- *Switch:* 25% of gates, no CPU, no packet buffers

Support for real-time apps.?

Examples

- Telephony over IP
- Real-time video over IP

Not really yet!

Summary: Benefits of IP?

- IP *supports many different types of data applications* at a wide range of data rates
- *Phone network: 1 of many services (voice, fax, touch-tone service, 800 numbers, teletype, hearing impaired services, lots of enhanced voice services, voicemail, …)*
- IP *traffic, services more diverse* (?). IP *works at higher bandwidths* (factually true for end applications, but cores are both high speed)
- *Claim:* IP supports short bursty connections "better" (implicit: Less setup cost, less resources used – not that important given utilization figures)
- IP has 1-RTT transaction times, phone network is at least 2-RTTs (setup plus transaction)