15-213

Internetworking I: Basics

April 13, 2000

Topics

- **Internetworking with repeaters, bridges and gateways**
- **Internetworking with routers**
  - the Internet Protocol (IP)
  - IP datagram delivery
  - IP addresses

The internetworking idea (Kahn, 1972)

Build a single network (an interconnected set of networks, or *internetwork*, or *internet*) out of a large collection of separate networks.

- **Each network must stand on its own, with no internal changes allowed to connect to the internet.**
- **Communications should be on a best-effort basis.**
- **"black boxes" (later called routers) should be used to connect the networks.**
- **No global control at the operations level.**

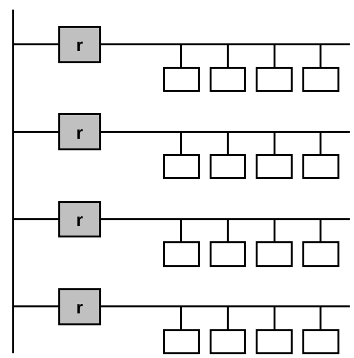
Internetworking challenges

Challenges:

- **heterogeneity**
  - lots of different kinds of networks (Ethernet, FDDI, ATM, wireless, point-to-point)
  - how to unify this hodgepodge?
- **scale**
  - how to provide uniques names for potentially billions of nodes? (naming)
  - how to find all these nodes? (forwarding and routing)

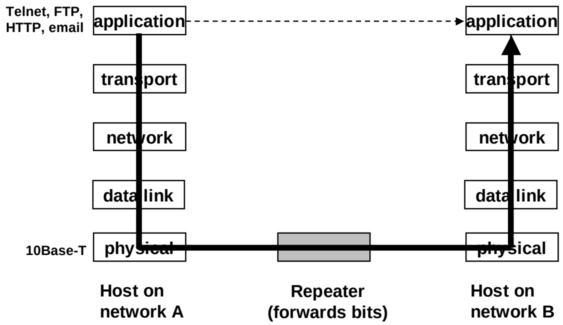
Note: *internet* refers to a general idea, *Internet* refers to a particular implementation of that idea (The global IP Internet).

Internetworking with repeaters



Repeaters (also called hubs) (r in the figure) directly transfer bits from their inputs to their outputs

Internetworking with repeaters



Internetworking with repeaters: Pros and cons

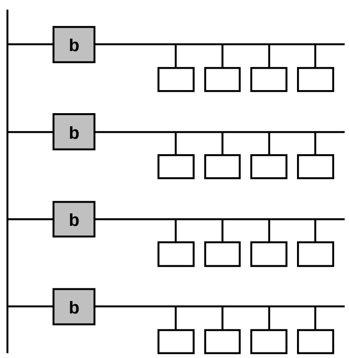
Pros

- **Transparency**
  - LANS can be connected without any awareness from the hosts.
- **Useful for serving multiple machines in an office from one ethernet outlet.**

Cons

- **Not scalable**
  - ethernet standard allows only 4 repeaters.
  - more than 4 would introduce delays that would break contention detection.
- **No heterogeneity**
  - Networks connected with repeaters must have identical electrical properties.

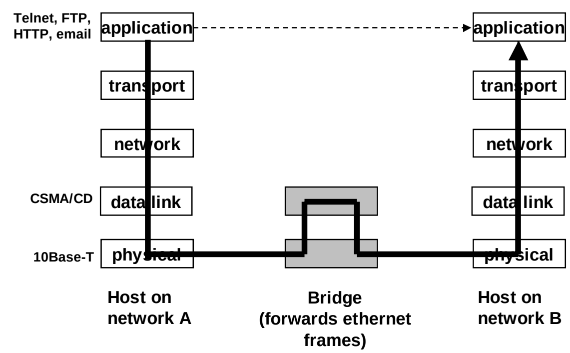
Internetworking with bridges



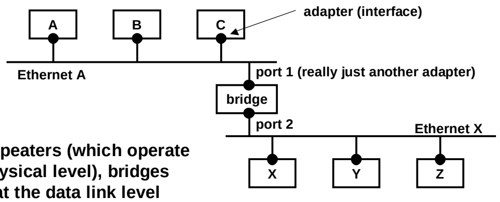
Bridges (b In the figure) maintain a cache of hosts on their input segments.

Selectively transfer ethernet frames from their inputs to their outputs.

Internetworking with bridges



Bridges



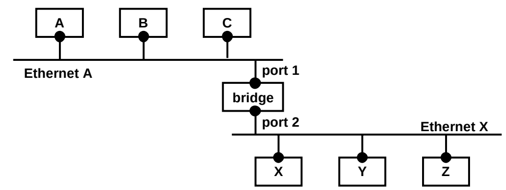
Unlike repeaters (which operate at the physical level), bridges operate at the data link level (or link level).

By link level, we mean that they can parse and understand e.g. ethernet frames (as opposed to IP packets).

Basic forwarding algorithm (*flooding*): copy each received frame to all other ports.

Learning bridges

Problem: Flooding is wasteful



Optimization: Forward packets only when necessary by learning and remembering which hosts are connected to which bridge ports.

Learning bridges (cont)

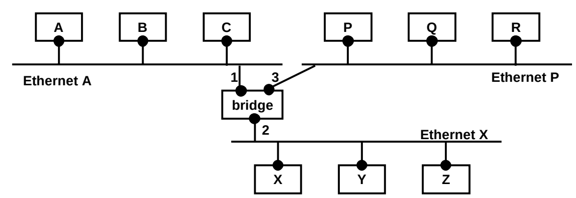
***Learning algorithm*:**

1. **start with empty hash table *T* that maps hosts to ports**
2. **receive frame from host *src* on port *p***
3. **add *(src,p)* to *T***
4. **delete *old* entries**

***Forwarding algorithm*:**

1. **receive frame *f* from host src to host *dst* on port *p***
2. **if *T(dst) = n/a* then flood *f*.**
   else if *T(dst) = p* then discard *f*
   else forward *f* on port *T(dst).*

Learning bridges (example)



A -> C

host	port
A	1

flood 2 & 3

B -> A

host	port
A	1
B	1

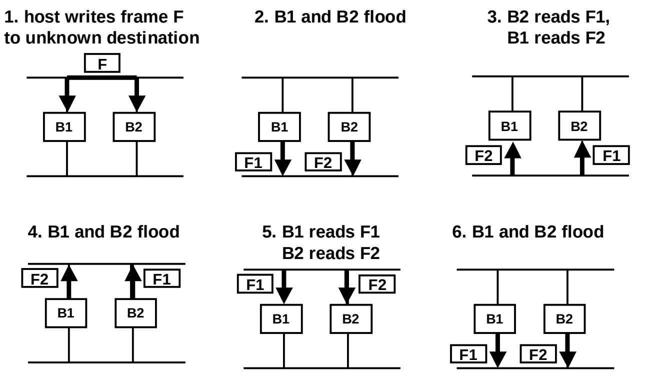
discard

X -> A

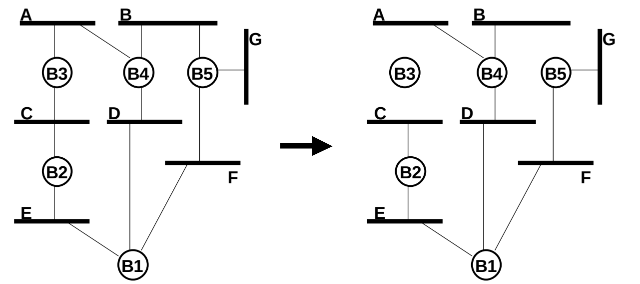
host	port
A	1
B	1
X	2

forward on 1

Cycles in bridged networks

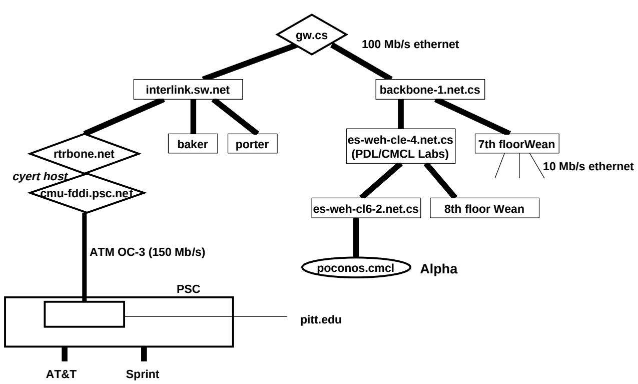


Spanning tree bridges



- Networks are graph nodes, ports are graph edges
- Tree is constructed dynamically by a distributed "diffusing computation" that prunes ports.
- "spanning" refers only to networks, not bridges

Portion of the bridged CMU internet



Internetworking with bridges: Pros and cons

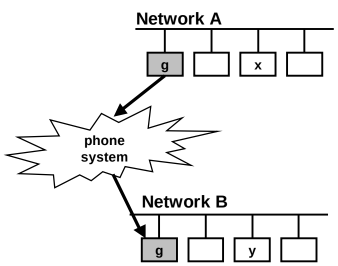
Pros

- **Transparency**
  - LANS can be connected without any awareness from the hosts
  - popular solution for campus-size networks

Cons

- **Transparency can be misleading**
  - looks like a single Ethernet segment, but really isn't
  - packets can be dropped, latencies vary
- **Homogeneity**
  - can only support networks with identical frame headers (e.g., Ethernet/FDDI)
  - however, can connect different speed Ethernets
- **Scalability**
  - tens of networks only
    - » bridges forward all broadcast frames
    - » increased latency

Internetworking with application gateways

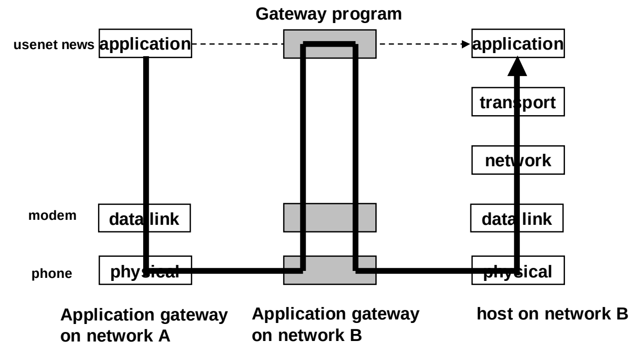


application gateways (g in the figure) connect different networks for particular applications.

Example:

- User on host x posts news item to gateway machine on network A.
- Gateway on A passes item (along with others) to gateway B.
- User on host y reads message from gateway on B.

Internetworking with application gateways



Internetworking with application gateways: Pros and cons

Pros

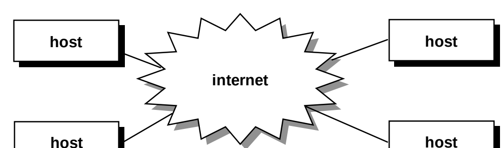
- **Heterogeneous**
  - can connect different types of networks
- **Simple**
  - modems + gateway software

Cons

- **Not general-purpose**
  - each solution is application-specific

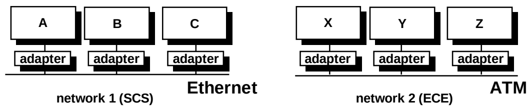
Internetworking with routers

Def: An *internetwork* (internet for short) is an arbitrary collection of *physical networks* interconnected by routers to provide some sort of host-to-host packet delivery service.



Building an internet

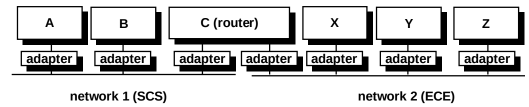
We start with two separate, unconnected computer networks (subnets), which are at different locations, and possibly built by different vendors.



Question: How to present the illusion of one network?

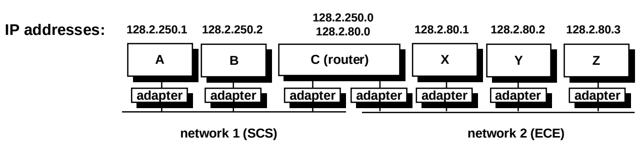
Building an internet (cont)

Next we physically connect one of the computers, called a router (in this case computer C), to each of the networks.



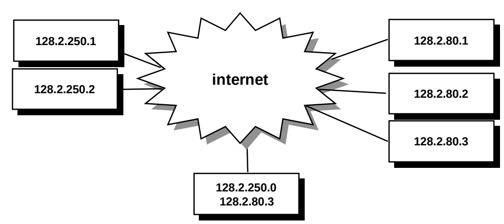
Building an internet (cont)

Finally, we run a software implementation of the Internet Protocol (IP) on each host and router. IP provides a global name space for the hosts, routing messages between network1 and network 2 if necessary.

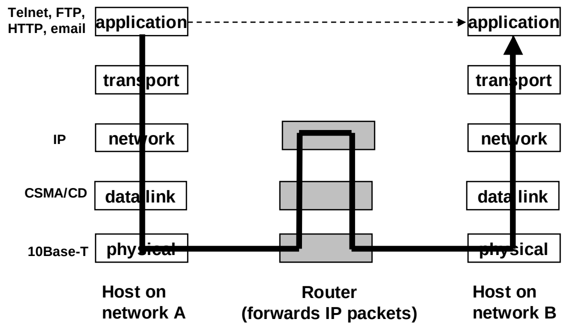


Building an internet (cont)

At this point we have an internet consisting of 6 computers built from 2 original networks. Each computer on our internet can communicate with any other computer. IP provides the illusion that there is just one network.



Internetworking with routers

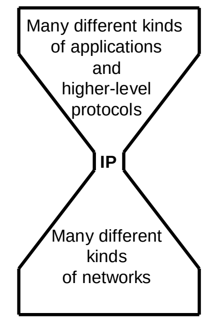


IP: Internetworking with routers

IP is the most successful protocol ever developed

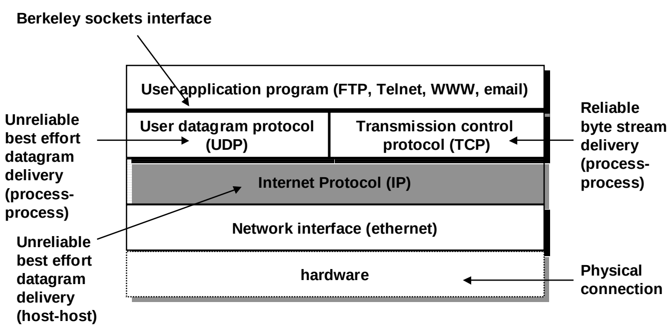
Keys to success:

- **simple enough to implement on top of any physical network**
  - e.g., two tin cans and a string.
- **rich enough to serve as the base for implementations of more complicated protocols and applications.**
  - The IP designers never dreamed of something like the Web.
- **"rough consensus and working code"**
  - resulted in solid implementable specs.



The "Hourglass Model", Dave Clark, MIT

Internet protocol stack



IP service model

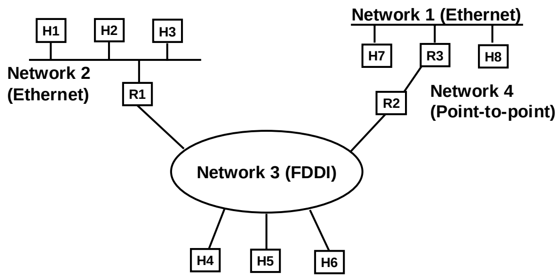
IP service model:

- **Delivery model: IP provides best-effort delivery of datagram (connectionless) packets between two hosts.**
  - IP tries but doesn't guarantee that packets will arrive (best effort)
  - packets can be lost or duplicated (unreliable)
  - ordering of datagrams not guaranteed (connectionless)
- **Naming scheme: IP provides a unique address (name) for each host in the Internet.**

Why would such a limited delivery model be useful?

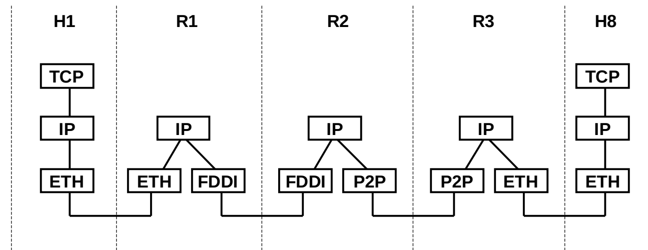
- **simple, so it runs on any kind of network**
- **provides a basis for building more sophisticated and user-friendly protocols like TCP and UDP**

IP datagram delivery: Example internet

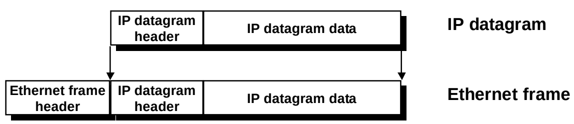


IP layering

Protocol layers used to connect host H1 to host H8 in example internet.

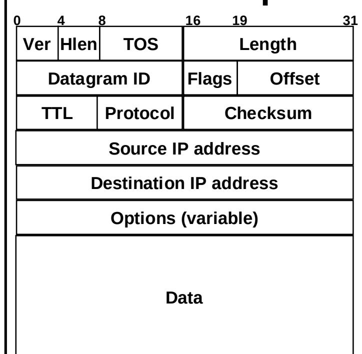


Encapsulating IP datagrams in Ethernet



The same idea is used for other types of physical networks

IP packet format



VER	IP version
HL	Header length (in 32-bit words)
TOS	Type of service (unused)
Length	Datagram length (max 64K B)
ID	Unique datagram identifier
Flags	xxM (more fragmented packets)
Offset	Fragment offset
TTL	Time to Live
Protocol	Higher level protocol (e.g., TCP)

Fragmentation and reassembly

Different networks types have different maximum transfer units (MTU).

A problem can occur if packet is routed onto network with a smaller MTU.

- e.g. FDDI (4,500B) onto Ethernet (1,500B)

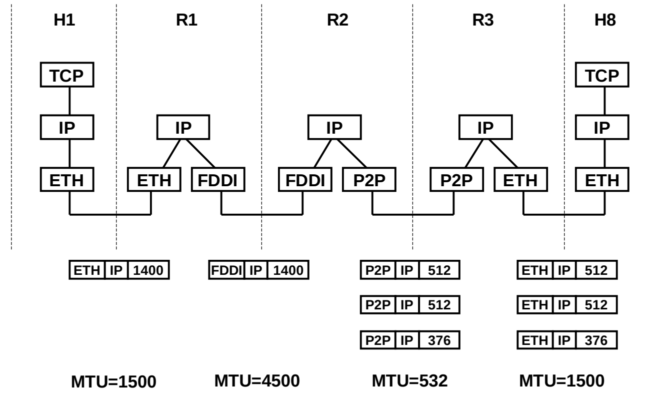
Solution: break packet into smaller fragments.

- each fragment has identifier and sequence number

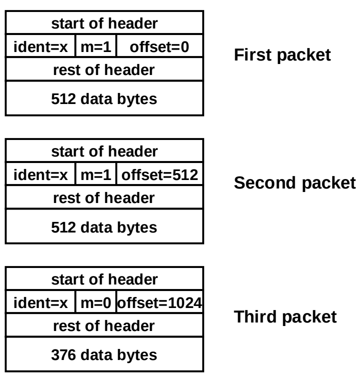
Destination reassembles packet before handing it up in the stack.

- alternative would be to reassemble when entering network with larger MTU

Fragmentation example



Fragmentation example (cont)

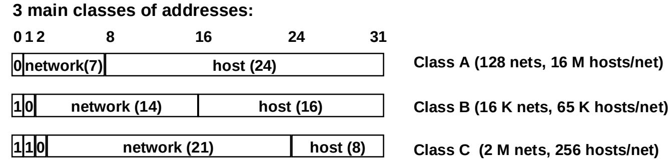


Internet addresses

Each host h has a physical address P(h) and a unique IP address I(h).

IP addresses contain a network part and a host part:

3 main classes of addresses:



Note: this simple A, B, C scheme has been largely replaced by the CIDR (classless interdomain routing) technique allows for variable bit length network numbers.

Example Internet addresses

Host	IP Number	Class	Network
cs.cmu.edu	128.2.222.173	B	0x0002
cmu.edu	128.2.35.186	B	0x0000
cs.stanford.edu	171.64.64.64	B	0x2640
	att.com	192.128.133.151 C	
	0x008085