Last Time: NAND Flash

Idea: Disk Replacement
Presents memory to the CPU as a set of pages.

Page format:

- 2048 Bytes + 64 Bytes
- (user data) + (meta data)

Note: NOR Flash is another flash product, for software code. NOR Flash read interface is just like SRAM.

Chip “remembers” for 10 years.

NAND Flash has better cost/bit than NOR.
Last Time: Making the Mac Mini G4

Size fixed by the “form factor” (physical size) of desktop DIMMS. Laptop DRAM is smaller, but too expensive for $499 price.
Why are networks different from buses?

**Serial:** Data is sent “bit by bit” over one logical wire.

**Network.** Primary purpose is to connect computers to computers.

**USB, FireWire.** Primary purpose is to connect devices to a computer.
Today: Networks

Link layers: Using physics to send bits from place to place.

Internet: A network of networks.

Routing: Inside the cloud.
Today: Router Design

Router architecture: What’s inside the box?

Forwarding engine: How a router knows the “next hop” for a packet.

Switch fabric: When buses are too slow ... replace it with a switch!
Networking bottom-up: Link two endpoints

Q1. How far away are the endpoints?

Japan-US undersea cable network

Physical media: optical fiber (photonics)

Distance + mobility + bandwidth influences choice of medium.

WiFi wireless from hotel bed to access point.

Physical media: unlicensed radio spectrum
Networking bottom-up: Link two endpoints

Q2. Initial investment cost for the link.

$1B USD. A ship lays cable on ocean floor.

The price of the WiFi laptop card + the base station.

"Unlicensed radio" -- no fee to the FCC

For expensive media, much of the "price" goes to pay off loans.
Q3. How is the link imperfect?

+++ A steady bitstream (“circuit”). No packets to lose.
+++ Only one bit flips per 1,000,000,000,000 sent.

--- Undersea failure is catastrophic

--- Someone walks by and the network stops working - “fading”.

Solution: Short packets spaced in time to escape the fade. If lost, do retransmits.
Q4. How does link perform?

**Latency:** % ping irt1-ge1-1.tdc.noc.sony.co.jp

PING irt1-ge1-1.tdc.noc.sony.co.jp (211.125.132.198): 56 data bytes
64 bytes from 211.125.132.198: icmp_seq=0 ttl=242 time=114.571 ms round-trip.

**Compare:**

Light speed in vacuum, SFO-Tokyo, 63ms RT.

In general, risky to halve the round-trip time for one-way latency: paths are often different each direction.

**BW:** In theory, 801.11b offers 11 Mb/s. Users are lucky to see 3-5 Mb/s in practice.

**Latency:** If there is no fading, quite good. I've measured <2 ms RTT on a short hop.
There are dozens of “link networks” ...

Diagram Credit: Steve Deering

Protocol Complexity

Link networks

The undersea cable, the hotel WiFi, and many others ... DSL, Ethernet, ...
Web browsers do not know about link nets

App authors do not want to add support for N different network types.

The undersea cable, the hotel WiFi, and many others ... DSL, Ethernet, ...

Diagram Credit: Steve Deering
Internet Protocol (IP): An abstraction for applications to target, and for link networks to support. Very simple, very successful.

IP presents link network errors/losses in an abstract way (not a link specific way).

Link layer is not expected to be perfect.

Diagram Credit: Steve Deering
The Internet interconnects “hosts” ...

IP4 number for this computer: 198.211.61.22

Every directly connected host has a unique IP number.

Upper limit of $2^{32}$ IP4 numbers (some are reserved for other purposes).

Next-generation IP (IP6) limit: $2^{128}$.

198.211.61.22 A user-friendly form of the 32-bit unsigned value 3335732502, which is:

$$198 \times 2^{24} + 211 \times 2^{16} + 61 \times 2^{8} + 22$$
Internet: Sends Packets Between Hosts

IP4, IP6, etc ...

How the destination should interpret the payload data.

From: IP number

To: IP number

Note: Could be a lie ...

IHL field: # of words in header. The typical header (IHL = 5 words) is shown. Longer headers code add extra fields after the destination address.

IHL field: # of words in header. The typical header (IHL = 5 words) is shown. Longer headers code add extra fields after the destination address.
Link networks transport IP packets

ISO Layer Names:
- IP packet: "Layer 3"
- WiFi and Cable Modem packets: "Layer 2"
- Radio/cable waveforms: "Layer 1"

For this "hop", IP packet sent "inside" of a wireless 801.11b packet.

For this "hop", IP packet sent "inside" of a cable modem DOCSIS packet.
Link layers “maximum packet size” vary.

Maximum IP packet size 64K bytes. Maximum Transmission Unit (MTU -- generalized “packet size”) of link networks may be much less – often 2K bytes or less. Efficient uses of IP sense MTU.

Fragment fields: Link layer splits up big IP packets into many link-layer packets, reassembles IP packet on arrival.

Maximum IP packet size 64K bytes. Maximum Transmission Unit (MTU -- generalized “packet size”) of link networks may be much less – often 2K bytes or less. Efficient uses of IP sense MTU.

Fragment fields: Link layer splits up big IP packets into many link-layer packets, reassembles IP packet on arrival.
IP abstraction of non-ideal link networks:

- A sent packet may never arrive ("lost").
- If packets sent P1/P2/P3, they may arrive P2/P1/P3 ("out of order").

**Best Effort:** The link networks, and other parts of the "cloud", do their best to meet the ideal. But, no promises.

- Relative timing of packet stream not necessarily preserved ("late" packets).
- IP payload bits received may not match payload bits sent. IP header protected by checksum (almost always correct).
How do apps deal with this abstraction?

"Computing" apps use the TCP (Transmission Control Protocol).

TCP lets host A send a reliable byte stream to host B. TCP works by retransmitting lost IP packets. Timing is uncertain.

Retransmission is bad for IP telephony: resent packets arrive too late.

IP telephony uses packets, not TCP. Parity codes, audio tricks used for lost packets.

Diagram Credit: Steve Deering
Routing
Undersea cables meet in Hawaii ...
In Makaha, a router takes each Layer 2 packet off the San Luis Obispo (CA) cable, examines the IP packet destination field, and forwards to Japan cable, Fiji cable, or to Kahe Point (and onto big island cables).
Example: \texttt{berkeley.edu to sony.co.jp}

Passes through 21 routers ...

\begin{verbatim}
% traceroute irtl-gel-1.tdc.noc.sony.co.jp
traceroute to irtl-gel-1.tdc.noc.sony.co.jp (211.125.132.198), 30 hops max, 40 byte packets
1  soda3a-gw.eecs.berkeley.edu (128.32.34.1)  20.581 ms  0.875 ms  1.381 ms
2  soda-cr-1-1-soda-br-6-2.eecs.berkeley.edu (169.229.59.225)  1.354 ms  3.097 ms  1.028 ms
3  vlan242.inr-202-doecev.berkeley.edu (128.32.255.169)  1.753 ms  1.454 ms  1.138 ms
4  ge-1-3-0.inr-001-eva.berkeley.edu (128.32.0.34)  1.746 ms  1.174 ms  2.22 ms
5  svl-dc1--ucb-egm.cenic.net (137.164.23.65)  2.653 ms  2.72 ms  12.031 ms
6  dc-svl-dc2--svl-dc1-df-icomm-2.cenic.net (137.164.22.209)  2.478 ms  2.451 ms
7  dc-sol-dc1--svl-dc1-pos.cenic.net (137.164.22.28)  4.509 ms  95.013 ms  7.7
8  dc-sol-dc2--sol-dc1-df-icomm-1.cenic.net (137.164.22.211)  18.319 ms  4.324 ms
9  dc-slo-dc1--sol-dc2-pos.cenic.net (137.164.22.26)  19.403 ms  10.077 ms  13.232 ms
10 dc-slo-dc2--dcl-df-icomm-1.cenic.net (137.164.22.123)  8.049 ms  20.653 ms  8.993 ms
11 dc-lax-dc1--slo-dc2-pos.cenic.net (137.164.22.24)  94.579 ms  14.52 ms  21.745 ms
12 rtrisi.ultradns.net (198.32.146.38)  25.48 ms  12.432 ms  17.837 ms
13 lax001bb00.iij.net (216.98.96.176)  11.623 ms  25.698 ms  11.382 ms
14 tky002bb01.iij.net (216.98.96.178)  168.082 ms  196.26 ms  121.914 ms
15 tky002bb00.iij.net (202.232.0.149)  144.592 ms  208.622 ms  121.801 ms
16 tky001bb01.iij.net (202.232.0.70)  153.757 ms  110.29 ms  184.985 ms
17 tky001bb00.iij.net (210.130.130.100)  114.234 ms  110.095 ms  169.692 ms
18 210.138.131.198 (210.138.131.198)  113.893 ms  113.665 ms  114.22 ms
19 ertl-gel000.tdc.noc.ssd.ad.jp (211.125.132.69)  114.758 ms  138.327 ms  113
20 211.125.133.86 (211.125.133.86)  113.956 ms  113.73 ms  113.965 ms
21 irtl-gel-1.tdc.noc.sony.co.jp (211.125.132.198)  145.247 ms  *  136.884 ms
\end{verbatim}

Cross ocean in 1 hop - link about 175 ms round-trip

CS 152 L21: Networks and Routers
Left on Internet Initiative Japan (IIJ) in LA

lax001bb00.iij.net (216.98.96.176)
Arrived IIJ in Ariake

tky002bb01.iij.net (216.98.96.178)
A-to-B packet path may differ from B-to-A

Different paths: Different network properties (latency, bandwidth, etc)

Diagram Credit: Van Jacobsen

Economics: A and B use different network carriers ... carriers route data onto their networks ASAP.
A 50-Gb/s IP Router

Craig Partridge, Senior Member, IEEE, Philip P. Carvey, Member, IEEE, Ed Burgess, Isidro Castineyra, Tom Clarke, Lise Graham, Michael Hathaway, Phil Herman, Allen King, Steve Kohalmi, Tracy Ma, John Mcallen, Trevor Mendez, Walter C. Milliken, Member, IEEE, Ronald Pettyjohn, Member, IEEE, John Rokosz, Member, IEEE, Joshua Seeger, Michael Sollins, Steve Storch, Benjamin Tober, Gregory D. Troxel, David Waitzman, and Scott Winterble

How to Design a Router

IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 6, NO. 3, JUNE 1998

Fig. 1. MGR outline.
A. Design Summary
A simplified ... Because the forwarding engines are separate from
the line cards, they may receive packets from line cards that
Recall: Routers are like hub airports

In Makaha, a router takes each Layer 2 packet off the San Luis Obispo (CA) cable, examines the IP packet destination field, and forwards to Japan cable, Fiji cable, or to Kahe Point (and onto big island cables).
The Oahu router ...

Assume each “line” is 160 Gbits/sec each way.

IP packets are **forwarded** from each inbound Layer 2 line to one of the four outbound Layer 2 lines, based on the destination IP number in the IP packet.
Challenge 1: Switching bandwidth

At line rate: $5 \times 160$ Gb/s = 100 GB/s switch! Latency not an issue ... wide, slow bus OK.

FIFOs (first-in first-out packet buffers) help if an output is sent more bits than it can transmit. If buffers "overflow", packets are discarded.
Challenge 2: Packet forwarding speed

For each packet delivered by each inbound line, the router must decide which outbound line to forward it to. Also, update IP header.

Line rate: 160 Gb/s
Average packet size: 400 bits
Packets per second per line: 400 Million
Packets per second (5 lines): 2 Billion

Thankfully, this is trivial to parallelize ...
Challenge 3: Obeying the routing “ISA”

Internet Engineering Task Force (IETF) “Request for Comments” (RFC) memos act as the “Instruction Set Architecture” for routers.

RFC 1812 (above) is 175 pages, and has 100 references which also define rules...
The MGR Router: A case study ...

A 50-Gb/s IP Router

Craig Partridge, Senior Member, IEEE, Philip P. Carvey, Member, IEEE, Ed Burgess, Isidro Castineyra, Tom Clarke, Lise Graham, Michael Hathaway, Phil Herman, Allen King, Steve Kohalmi, Tracy Ma, John Mcallen, Trevor Mendez, Walter C. Milliken, Member, IEEE, Ronald Pettyjohn, Member, IEEE, John Rokosz, Member, IEEE, Joshua Seeger, Michael Sollins, Steve Storch, Benjamin Tober, Gregory D. Troxel, David Waitzman, and Scott Winterble

The “MGR” Router was a research project in late 1990’s. Kept up with “line rate” of the fastest links of its day (OC-48c, 24 Gb/s optical).

Architectural approach is still valid today ...
MGR top-level architecture

A 50 Gb/s switch is the centerpiece of the design. Cards plug into the switch.

In best case, on each switch “epoch” (transaction), each card can send and receive 1024 bits to/from one other card.
MGR cards come in two flavors....

**Line card**: A card that connects to Layer 2 line. Different version of card for each Layer 2 type.

**Forwarding engine**: Receives IP headers over the switch from line cards, and returns forwarding directions and modified headers to line card.
A control processor for housekeeping

Forwarding engine handles fast path: the “common case” of unicast packets w/o options. Unusual packets are sent to the control processor.

Control processor
The life of a packet in a router ...

1. Packet arrives in line card. Line card sends the packet header to a forwarding engine for processing.

Note: We can balance the number of line cards and forwarding engines for efficiency: this is how packet routing parallelizes.
The life of a packet in a router ...

2. Forwarding engine determines the next hop for the packet, and returns next-hop data to the line card, together with an updated header.
The life of a packet in a router ...

3. Line card uses forwarding information, and sends the packet to another line card via the switch.

Recall: Each line card can receive a packet from the switch at the same time -- a switch is not like a bus!
The life of a packet in a router ...

4. Outbound line card sends packet on its way ...

Backpressure: A mechanism some Layer 2 links have to tell the sender to stop sending for a while ...
Packet Forwarding

A. Design Summary

A simplified version of the MGR (Management and Routing) design has been shown. The solution consists of separate processors for the card’s forwarding functions, which are a subset of the processor's responsibilities. The forwarding engine is essentially a table that maps packet addresses to different destinations. Each card can have a single table, or the tables can be distributed across multiple cards. The forwarding functions of the card were historically implemented on a single processor, but now they are separate from the line cards.

Innovations such as the MGR design have allowed routing table sizes to be very modest compared to traditional routers. They have enabled faster routing and switching times, and have improved network performance by allowing more parallelism.

In the past, finding the appropriate routing path from a source to a destination often involved confusion, but modern designs like MGR have simplified these processes. The nightmares of previous table generation and memory management are a thing of the past.

There are advantages to using separate forwarding engines, as they can be dedicated processors that don’t have to handle line cards. This allows for faster and more efficient forwarding. In the past, we had to consider the overhead of multiple path determination, but MGR has simplified this process. Additionally, the use of engines with dedicated hardware has led to greater flexibility and reduced link costs.
### Forwarding engine computes “next-hop”

The forwarding engine looks at the destination address, and decides which outbound line card will get the packet closest to its destination. How?

```plaintext
<table>
<thead>
<tr>
<th>Version</th>
<th>IHL</th>
<th>Type of Service</th>
<th>Total Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Identification</td>
<td>Flags</td>
<td>Fragment Offset</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Time to Live</td>
<td>Protocol</td>
<td>Header Checksum</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Source Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Destination Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Payload data (size implied by Total Length header field)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
```

---

**Bitfield numbers**

**Header**

**Data**
Recall: Internet IP numbers ...

**IP4 number for this computer:** 198.211.61.22
198.211.61.22 == 3335732502 (32-bit unsigned)

Every directly connected host has a unique IP number.

Upper limit of $2^{32}$ IP4 numbers (some are reserved for other purposes).
Routers use BGP to exchange routing tables. Tables code if it is possible to reach an IP number from the router, and if so, how “desirable” it is to take that route.

Routers use BGP tables to construct a “next-hop” table. Conceptually, forwarding is a table lookup: IP number as index, table holds outbound line card.

A table with 4 billion entries???
Tables do not code every host ...

Routers route to a “network”, not a “host”. /xx means the top xx bits of the 32-bit address identify a single network.

<table>
<thead>
<tr>
<th>Network</th>
<th>IP Address Range</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.32.0.0/16</td>
<td>128.32.0.0 - 128.32.255.255</td>
<td>UCB Local Area Networks *</td>
</tr>
<tr>
<td>136.152.0.0/16</td>
<td>136.152.0.0 - 136.152.255.255</td>
<td>UCB Local Area Networks and Home IP Service #</td>
</tr>
<tr>
<td>169.229.0.0/16</td>
<td>169.229.0.0 - 169.229.255.255</td>
<td>UCB Local Area Networks</td>
</tr>
<tr>
<td>131.243.52.0/24</td>
<td>131.243.52.0 - 131.243.52.255</td>
<td>UCB Melvin Calvin Lab. building</td>
</tr>
<tr>
<td>192.101.42.0/24</td>
<td>192.101.42.0 - 192.101.42.255</td>
<td>UCB Local Area Networks</td>
</tr>
<tr>
<td>199.133.139.0/24</td>
<td>199.133.139.0 - 199.133.139.255</td>
<td>USDA/UCB Joint Local Area Network</td>
</tr>
</tbody>
</table>

Thus, all of UCB only needs 6 routing table entries. Today, Internet routing table has about 100,000 entries.
Forwarding engine: Also updates header

**Time to live.** Sender sets to a high value. Each router decrements it by one, discards if 0. Prevents a packet from remaining in the network forever.

Checksum. Protects IP header. Forwarding engine updates it to reflect the new Time to Live value.
MGR forwarding engine: a RISC CPU

Off-chip memory in two 8MB banks: one holds the current routing table, the other is being written by the router’s control processor with an updated routing table. Why?? So that the router can switch to a new table without packet loss.

85 instructions in “fast path”, executes in about 42 cycles. Fits in 8KB I-cache

Performance: 9.8 million packet forwards per second. To handle more packets, add forwarding engines. Or use a special-purpose CPU.
Switch Architecture

Switch

Line

Line

Line

Engine

Engine

Line

Line
What if two inputs want the same output?

A pipelined arbitration system decides how to connect up the switch. The connections for the transfer at epoch N are computed in epochs N-3, N-2 and N-1, using dedicated switch allocation wires.
A complete switch transfer (4 epochs)

Epoch 1: All input ports (that are ready to send data) request an output port.

Epoch 2: Allocation algorithm decides which inputs get to write.

Epoch 3: Allocation system informs the winning inputs and outputs.

Epoch 4: Actual data transfer takes place.

Allocation is pipelined: a data transfer happens on every cycle, as does the three allocation stages, for different sets of requests.
Epoch 3: The Allocation Problem

### Output Ports (A, B, C, D)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

A 1 codes that an input has a packet ready to send to an output. Note an input may have several packets ready.

### Allocator

Allocates returns a matrix with one 1 in each row and column to set switches. Algorithm should be "fair", so no port always loses ... should also "scale" to run large matrices fast.
“Best-effort” and Routers

Network Working Group
Request for Comments: 1812
Obsoletes: 1716, 1009
Category: Standards Track

F. Baker, Editor
Cisco Systems
June 1995

Requirements for IP Version 4 Routers
Recall: The IP “non-ideal” abstraction

- A sent packet may never arrive (“lost”).
- Router drops packets if too much traffic destined for one port, or if Time to Live hits 0, or checksum failure.
- If packets sent P1/P2/P3, they may arrive P2/P1/P3 (“out of order”).
- Relative timing of packet stream not necessarily preserved (“late” packets).
- This happens when the packet’s header forces the forwarding processor out of the “fast path”, etc.
- IP payload bits received may not match payload bits sent.
- Usually happens “on the wire”, not in router.
Conclusions: Router Design

**Router architecture:** The “ISA” for routing was written with failure in mind -- unlike CPUs.

**Forwarding engine:** The computational bottleneck, many startups target silicon to improve it.

**Switch fabric:** Switch fabrics have high latency, but that’s OK: routing is more about bandwidth than latency.
Reminder: No Checkoff this Friday!

Final checkoff the following Friday ...

TAs will provide “secret” MIPS machine code tests.

Bonus points if these tests run by end of section. If not, TAs give you test code to use over weekend.

Final report due following Monday, 11:59 PM