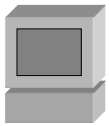
Arquitectura de routers

Jorge García Vidal

Intro

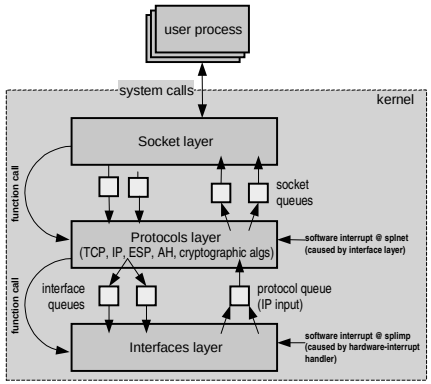


Networking equipment

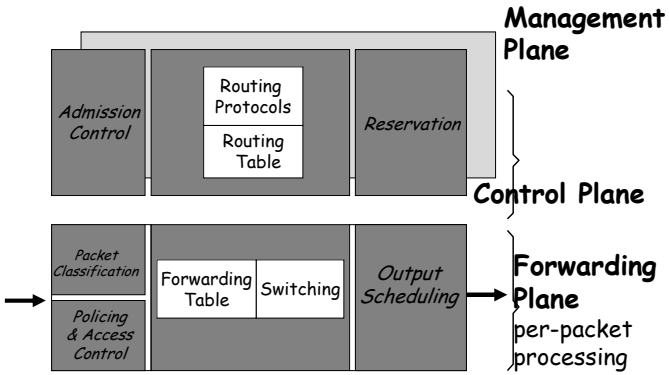
PC, ws Router, switch PDA

How can we implement networking software/hardware?

Intro: Hosts



Intro: Router



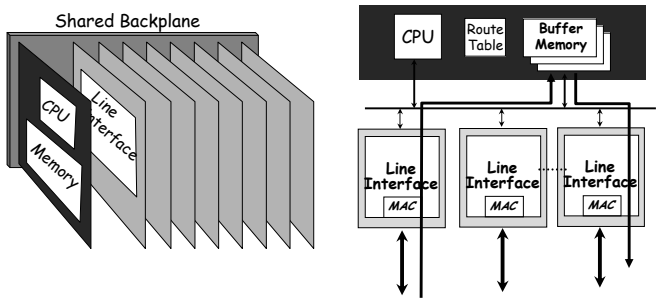
(De N. McKeown)

Intro Router

- "Wireline speed": el router puede encaminar un tren de paquetes consecutivos de tamaño mínimo (ej: 40 Bytes, 64 Bytes)

• 1 Mbps => 512/N μs	12 años
• 10 Mbps => 51.2/N μs	1 año
• 100 Mbps => 5.1/N μs	1 ½ mes
• 2.5 Gbps => 205 ns	2 días
• 10 Gbps => 51.2 ns	1/2 día
• 40 Gbps => 12.8 ns	2 ½ h
• DRAM (Random access time) = 60 ns	1/2 día
• SRAM = 5 ns	1 h

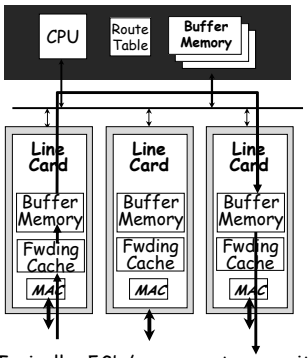
Router Structures



Typically <0.5Gb/s aggregate capacity

(De N. McKeown)

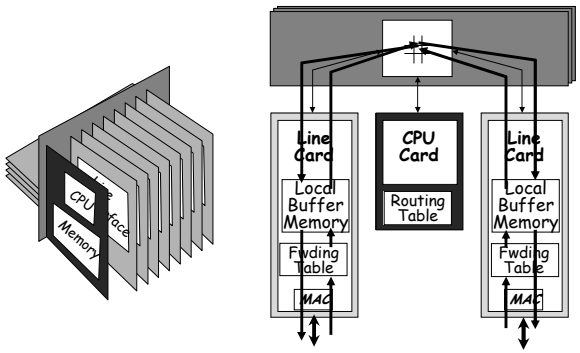
Router Structures



Typically <5Gb/s aggregate capacity

(De N. McKeown)

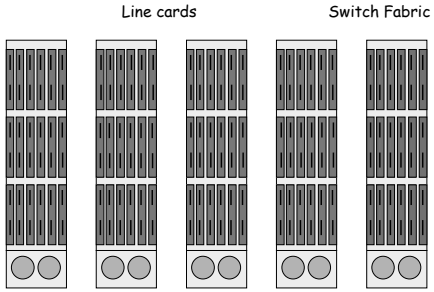
Router Structures



Typically <50Gb/s aggregate capacity

(De N. McKeown)

Router Structures



Next generation systems. Problem, how to integrate optic/electronic subsystems?

Algunos de los routers más rápidos…

Cisco CRS

Juniper T640

Los routers más comunes…

<>

J-series

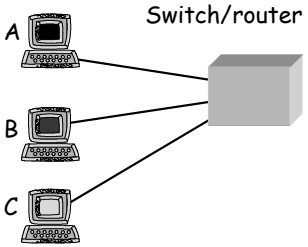
Router Structures

Growth in capacity of commercial routers:

- Capacity 1992 ~ 2Gb/s
- Capacity 1995 ~ 10Gb/s
- Capacity 1998 ~ 40Gb/s
- Capacity 2001 ~ 160Gb/s
- Capacity 2003 ~ 640Gb/s

Average growth rate: 2.2x / 18 months.

Routers, LAN switches, WAN switches



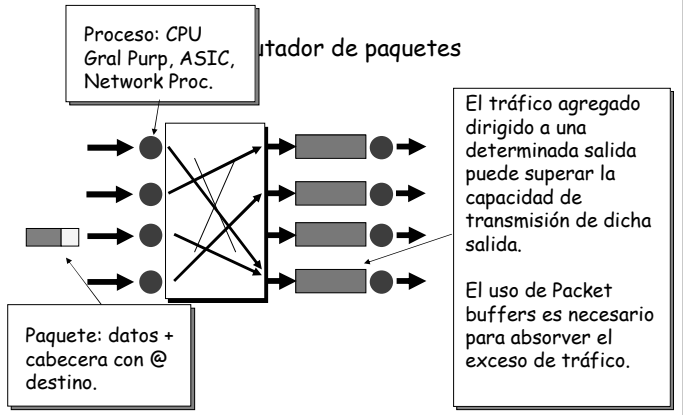
Ethernet switch: MAC addresses (48 bits, flat), Connectionless, L2, Exact Match

IPv4 routers: IP addresses, (32 bits, hierarchical) connectionless, L3, LPM

IPv6 routers: 128 bits, LPM

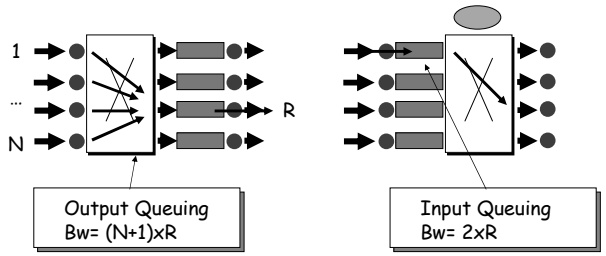
ATM switch: VCI/VPI (12 bits), connection oriented, L2 (?), Exact match (Direct lookup)

Routers



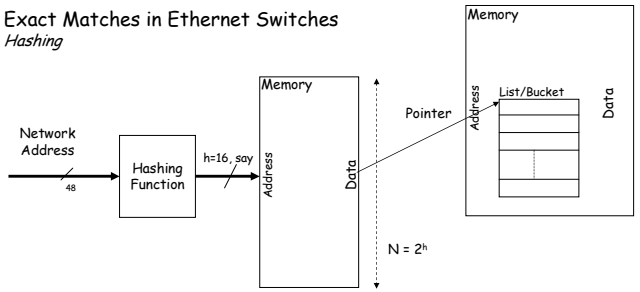
Packet Buffers

- El ancho de banda (Bw) necesario depende de la posición del packet buffer. En cualquier caso es proporcional a la velocidad de transmisión



L2 Address Lookup

Exact Matches in Ethernet Switches
Hashing



- Use a pseudo-random hash function (relatively insensitive to actual function)
- Bucket linearly searched (or could be binary search, etc.)
- Leads to unpredictable number of memory references

(De N. McKeown)

L2 Address lookup

Si la función de Hash da como resultado h bits (N = 2^h), y tenemos M direcciones guardadas, la probabilidad de colisión en el caso ideal es:

$P\{Colisión\} = 1 - [1 - \{(N-1)/N\}^M] \times N/M \approx M/2N = M/2^{h+1}$ si $N \gg M \gg 1$

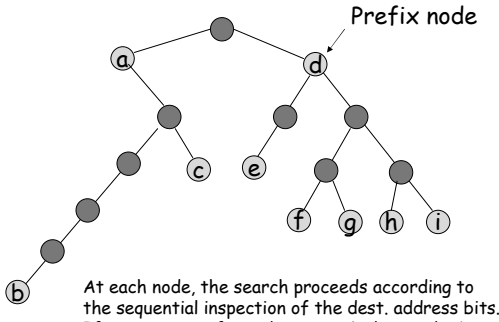
Ejemplo: M= 1024, h= 16, P{Colisión} = 0.008 (1% aprox)
M = 64536, h=16, P{Colisión} = 0.373 (37% aprox)

Ejemplo:
Para tener una P{Colisión} < 1% cuando M= 100.000, deberíamos usar
Un valor de h ≈ log2(M/P{Colisión}} -1 ≈ 3 x log10{105/10-2} -1 = 20,
Por lo que la memoria intermedia debería tener un tamaño de 1 M x n

IP lookup

Binary Trie

a : 0*
b : 01000*
c : 011*
d : 1*
e : 100*
f : 1100*
g : 1101*
h : 1110*
i : 1111*

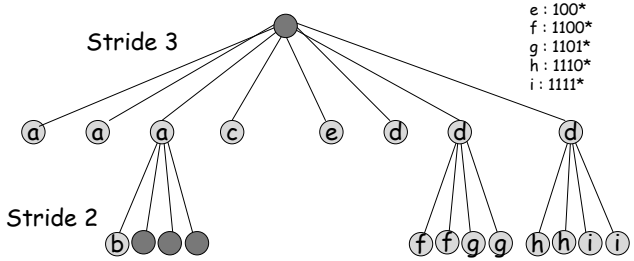


At each node, the search proceeds according to the sequential inspection of the dest. address bits.
If we visit a prefix node, we mark this as the best matching prefix so far (BMP).
Search ends when there are no more branches to take
Worst case : 32 memory access

IP lookup

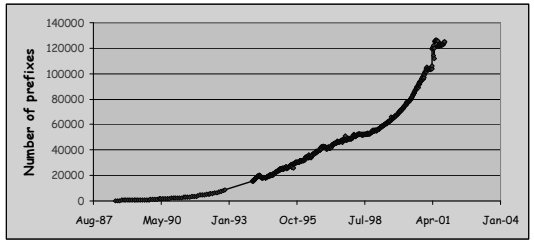
Multibit Trie

a : 0*
b : 01000*
c : 011*
d : 1*
e : 100*
f : 1100*
g : 1101*
h : 1110*
i : 1111*



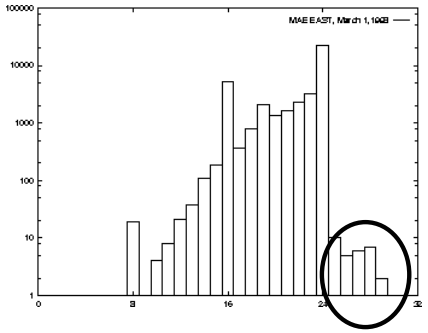
Multibit tries reduce memory access, although required memory increases as sometimes we need to expand prefixes (ej: case of a)
The search alg. is the same as in the case of binary tries, but now we inspect several bits simultaneously

IP lookup



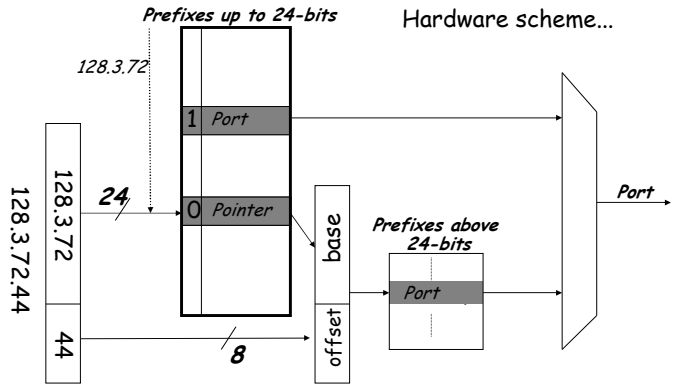
(De N. McKeown)

IP lookup



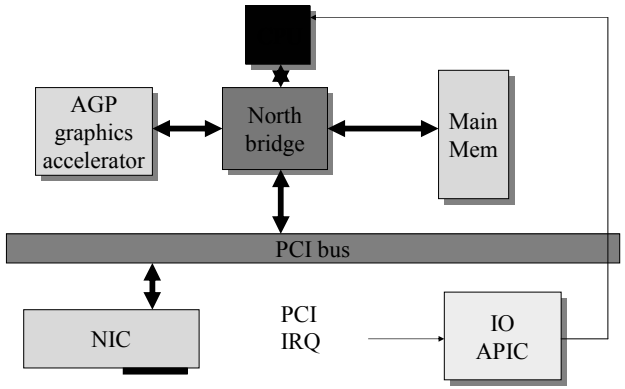
Most prefixes are 24-bits or shorter

IP lookup

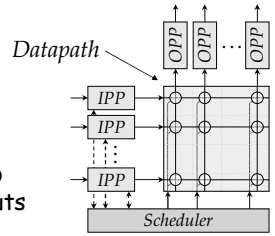


(De N. McKeown)

I/O Bus



Crossbars & Switch Fabric Scheduling

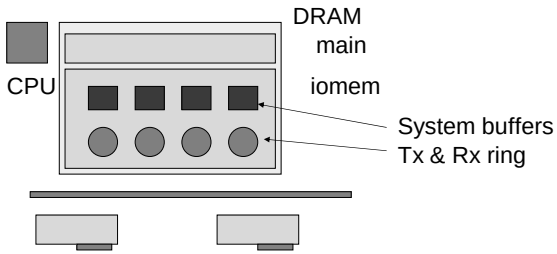


- Crossbar allows multiple cells to pass in parallel to distinct outputs
- Datapath is the *easy* part
- Scheduler arbitrates access to outputs

(De J. Turner)

Shared Mem Routers (Cisco terminology)

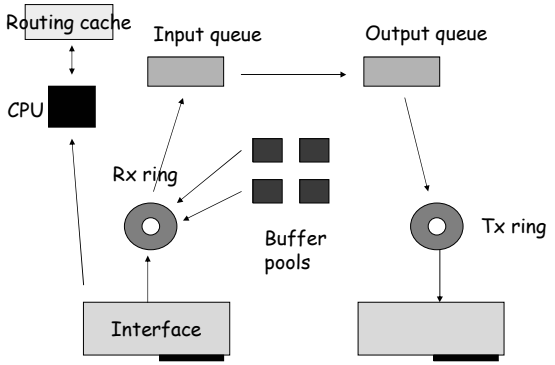
- Cisco 1600, 2500 , 4000, 4500 and 4700
- Architecture: CPU, Main Memory and Interfaces
- Use System buffers for packet buffering



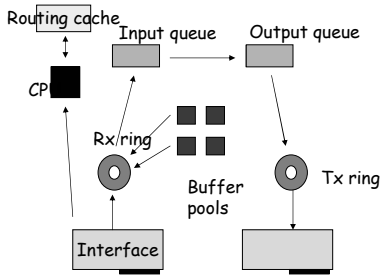
Shared Memory Routers

- The CPU is responsible for executing all the switching methods; there are no offload processors involved
- Cisco 1600, 2500: Motorola 68000
- Cisco 4500, 4700: RISC MIP
- "Shared Memory" is not a convenient name.

Shared Memory Routers



SMR: Receiving the pckt

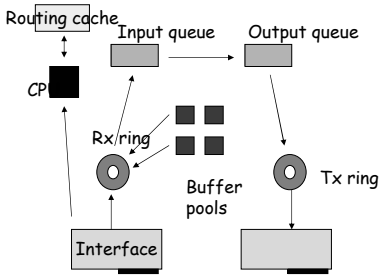


The interface copies the packet into a buffer in the receive ring using DMA

The media controller changes ownership of the pckt buffer to the processor and issues an interrupt to the CPU

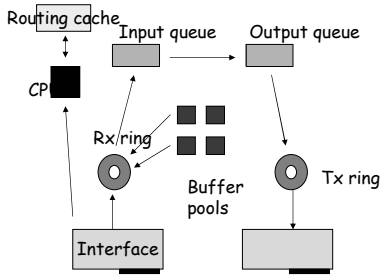
The CPU removes the incoming pckt and removes the pckt buffer of the receive ring

SMR: Switching the pckt



After the receive ring is replenished, the CPU begins switching the pckt. IOS attempts to switch using the fastest method configured on the interface: CEF, FS or PS

SMR: Transmitting the pckt



The pckt is placed in the Tx ring.

The Output interface periodically polls its tx ring. As soon as the media controller detects a pckt, it copies the pckt onto the network media and raises a Tx interrupt to the processor.

The CPU unlinks the pckt buffer from the transmit ring and returns the buffer to the pool of buffers from which it came