

IP Router Architectures

TELCOM2321 – CS2520

Wide Area Networks

Dr. Walter Cerroni

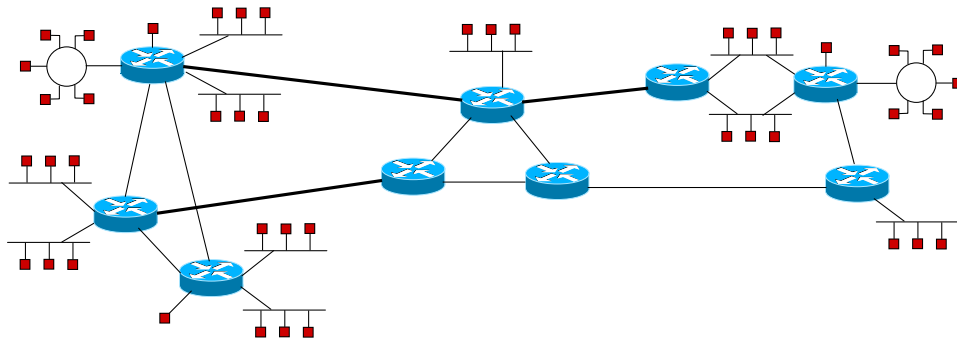
University of Bologna – Italy

Visiting Assistant Professor at SIS, Telecom Program

Reading

1. S. Keshav, R. Sharma
Issues and Trends in Router Design
IEEE Communications Magazine
Vol. 36, No. 5, May 1998, pp. 144-151
<http://ieeexplore.ieee.org>

Internetworking

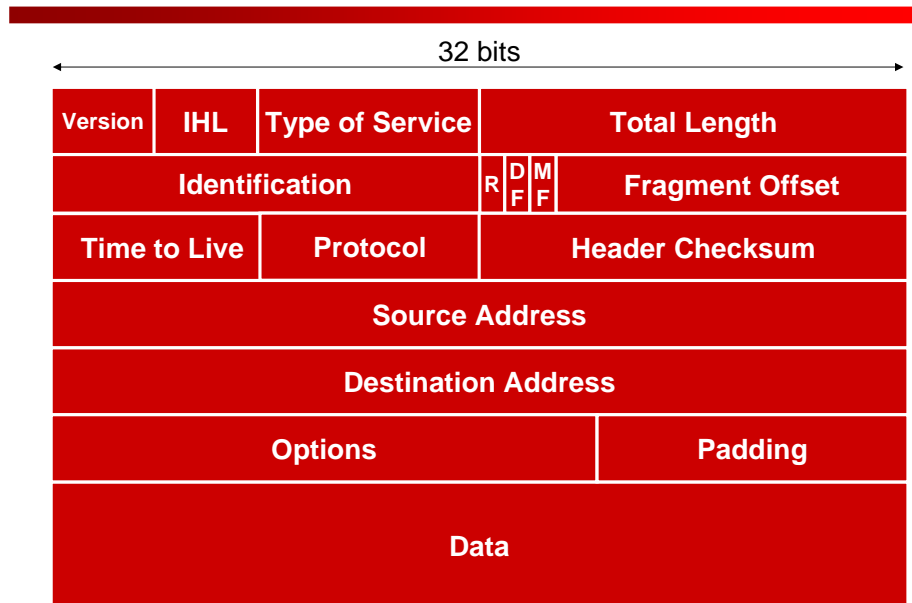


- Routers interconnect heterogeneous networks, each with their own protocols and data unit formats
- IP is the universal protocol that enables internetworking
- Routers cooperate to keep the Internet connected and to compute the best paths

Router classification

- **Access router**
 - used by ISPs to provide access to residential users or small business
 - large number of medium-low speed ports (50 kbps ÷ 10 Mbps)
 - capable of several protocols and access technologies (PPP, SLIP, ADSL, FTTx, ...)
- **Enterprise/campus router**
 - used to interconnect network infrastructures of medium to large sized enterprises and other institutions
 - limited number of high speed ports (10 Mbps ÷ 1 Gbps)
- **Backbone router**
 - used in carriers core networks and for inter-domain routing
 - small number of very high speed ports (1 Gbps ÷ 10 Gbps)
 - expensive but highly reliable

IPv4 packet format



Basic IPv4 router actions at packet arrival

1. Header extraction
2. Header error control
 - if wrong checksum, discard packet
3. TTL decrement
 - if TTL = 0, discard packet
4. Destination address lookup in the forwarding table
 - if destination is unreachable, discard the packet
5. Fragmentation, if required
 - if DF = 1, discard packet
 - otherwise update fragment offset
6. Update header checksum
7. Forward packet to the output
 - queue packet if channel is unavailable

IP datagram transfer

Performed at each router through two functions

1. Routing

- Information exchange with other routers
 - routing protocols
- Local processing
 - routing algorithms
- Routing table population

2. Forwarding

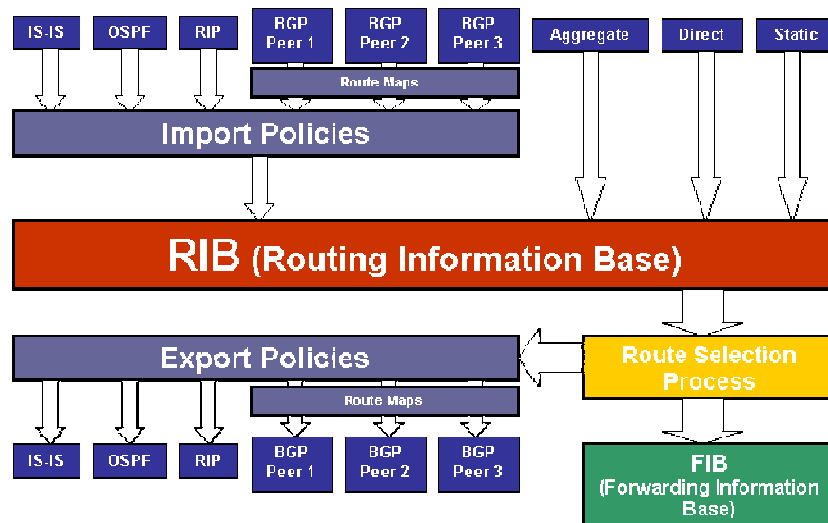
- IP header processing
- Forwarding table lookup
- Header update
- Datagram switching
 - physical transfer of datagrams to the output

Routing vs. forwarding table

- Routing table
 - result of the routing protocols and algorithms
 - each entry includes route prefix, next hop and metric
 - also called Routing Information Base (RIB)

- Forwarding table
 - built upon routing table content (complete or partial)
 - each entry includes also the output interface
 - used to actually forward datagram
 - optimized for fast table lookup
 - also called Forward Information Base (FIB)

Routing vs. forwarding table



Forwarding table lookup

- What is needed
 - datagram destination address
 - route prefix from each entry
- Which entry is chosen
 - prefix including the destination address
 - longest prefix match in case of multiple matches (i.e. the most specific one)
- How it could be implemented
 - bit-wise AND operation of the destination address with each entry's prefix netmask
 - result compared with prefix
 - starting with the longest prefix

Lookup table example (1)

	Prefix	Next-hop	Etc.
1	0.0.0.0/0	A	...
2	137.204.0.0/16	B	...
3	137.204.57.0/24	A	...
4	137.204.57.128/25	C	...

- Incoming datagram destination address: 137.204.57.174

137.204. 57.174
255.255.255.128
137.204. 57.128 = 137.204.57.128

bit-wise AND

- Entry no. 4 is chosen

Lookup table example (2)

	Prefix	Next-hop	Etc.
1	0.0.0.0/0	A	...
2	137.204.0.0/16	B	...
3	137.204.57.0/24	A	...
4	137.204.57.128/25	C	...

- Incoming datagram destination address: 137.204.57.24

137.204. 57. 24
255.255.255.128
137.204. 57. 0 \neq 137.204.57.128

137.204. 57. 24
255.255.255. 0
137.204. 57. 0 = 137.204.57.0

- Entry no. 3 is chosen

Lookup table example (3)

	Prefix	Next-hop	Etc.
1	0.0.0.0/0	A	...
2	137.204.0.0/16	B	...
3	137.204.57.0/24	A	...
4	137.204.57.128/25	C	...

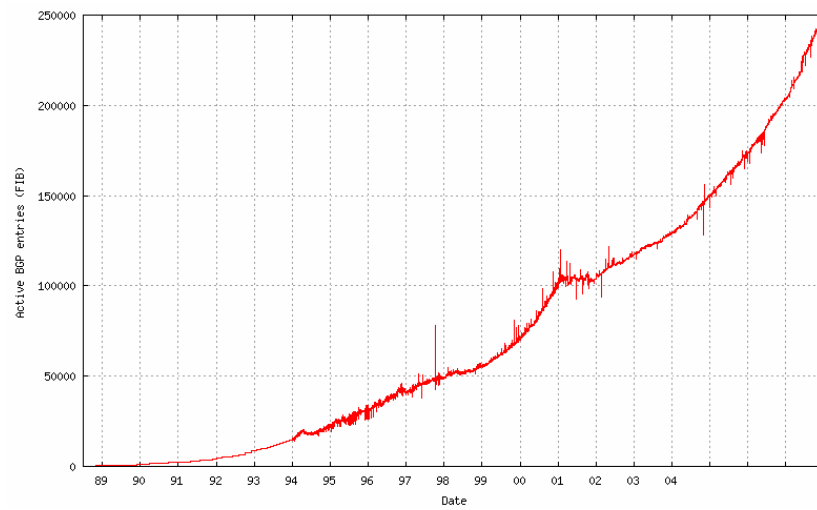
- Incoming datagram destination address: 80.48.15.170

80. 48. 15.170	80. 48. 15.170
<u>255.255.255.128</u>	<u>255.255.255. 0</u>
80. 48. 15.128	80. 48. 15. 0

80. 48. 15.170	80. 48. 15.170
<u>255.255. 0. 0</u>	<u>0. 0. 0. 0</u>
80. 48. 0. 0	0. 0. 0. 0

- Entry no. 1 is chosen (default route)

BGP FIB size since 1994



Source: <http://bgp.potaroo.net/>

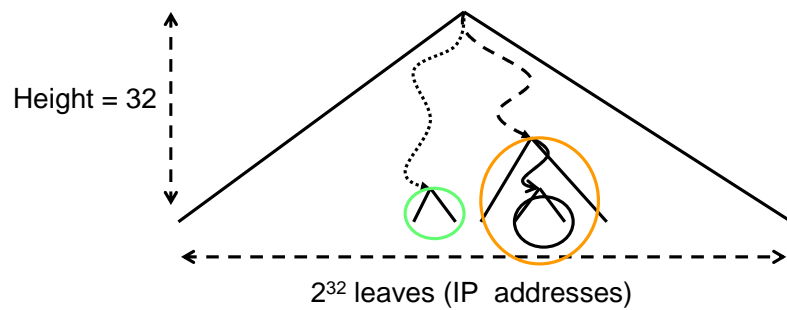
<http://bgp.potaroo.net/as2.0/bgptable.txt> → 43 MB text file

High-speed route lookup

- Very fast route lookups are required at gigabit speed
 - lookups/sec as high as packets/sec
- Route lookup time depends on
 - forwarding table size
 - number of memory accesses required
 - memory access time
- Forwarding table changes are much slower
 - in the order of the minute
- Store routing entries in data structures such that
 - access time is optimized with respect to update time

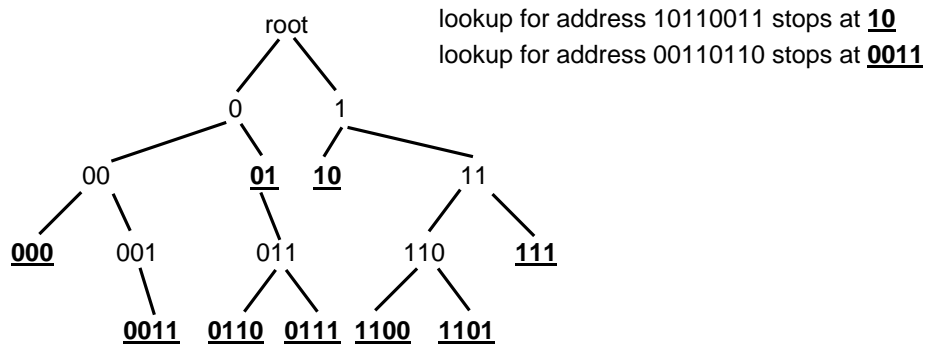
Lookup in a binary tree

- Routes are stored in a binary tree structure
 - each bit of the destination address selects the left or right child
 - starting from the most significant bit
- Each path in the tree represents a prefix
 - all leaves in the sub-tree are addresses included in the prefix
 - more in-depth paths are longest prefix



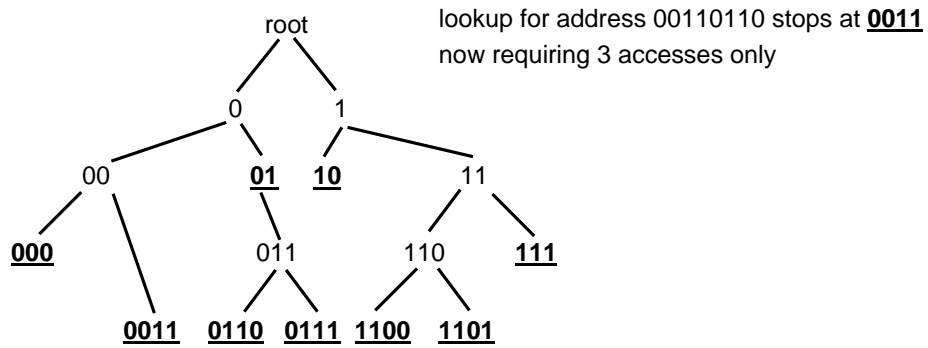
From tree to trie

- No need to store all the possible addresses
- Boldface underlined nodes correspond to prefix entries in the forwarding table
- Worst case (/32 prefix) requires 32 memory accesses



Patricia trie

- A node with a single child and not representing a prefix entry can be skipped
- Memory accesses can be reduced

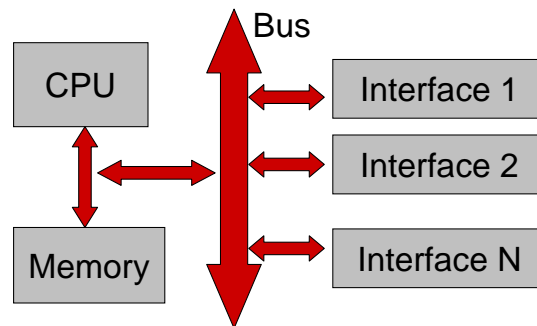


High-speed route lookup

- Goals
 - to reduce the number of accesses to memory
 - to reduce the trie size so that it can be stored in a fast cache memory
- Patricia trie can help in achieving short lookup times
- Other solutions have been proposed
 - content-addressable memories (CAMs)
 - table compaction techniques
 - hashing techniques

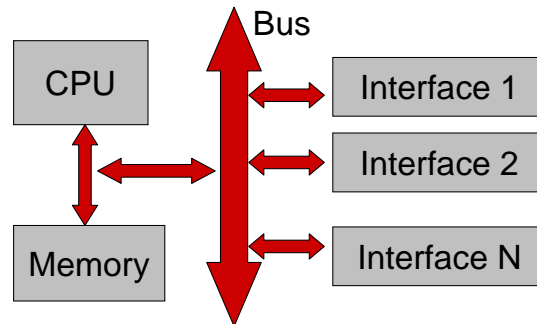
First generation router architecture

- Workstations with network interfaces, bus and CPU
 - incoming packets are transferred through the bus and stored in the central memory
 - header processing and table lookup performed by CPU
 - CPU takes care of the routing table also
 - processed packets are forwarded to the outputs via the bus



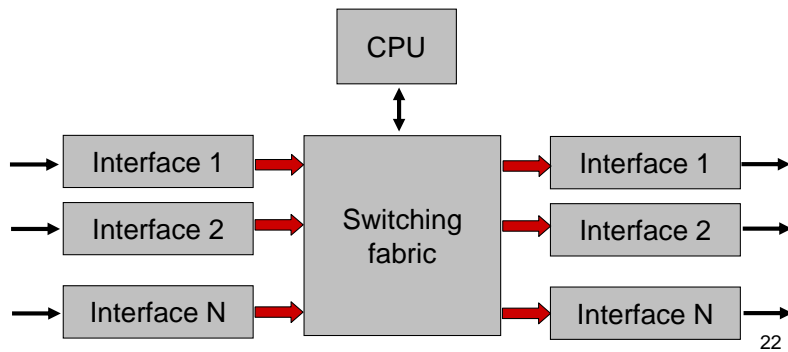
Performance limitations

- The bus is the bottleneck
 - shared by all interfaces
 - each packet must cross it twice
 - 1 Gbps bus allows up to 5 links at 100 Mbps
- The central memory is another bottleneck
 - slow table lookup



High-performance router architecture

- The main CPU executes the routing protocols, computes the routing table and manage the whole system
- The input interfaces are equipped with local CPUs and memories and perform the forwarding table lookup
- The switching fabric allows direct transfer of multiple packets at the same time
- Processing implemented by specialized hardware



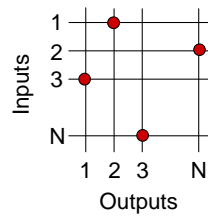
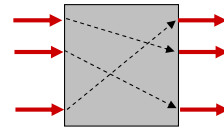
NxN switching fabric

- Multiple parallel transfers

- Example:

- **crossbar**

- N^2 crossing points open/closed by a controller device (**scheduler**)
 - single stage switching matrix
 - non-blocking: it is always possible to set up a path between an unused input port and an unused output port
 - switching speed limited by the scheduler speed



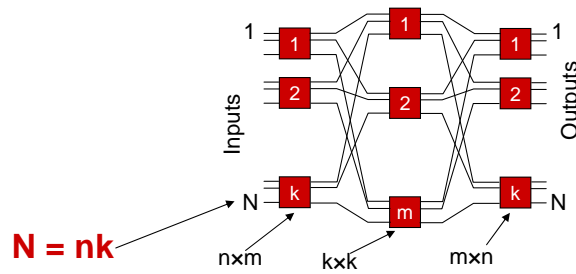
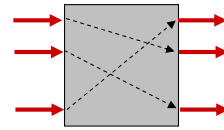
NxN switching fabric

- Multiple parallel transfers

- Example:

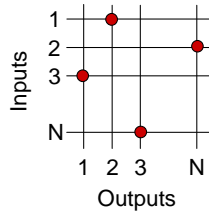
- **Clos network**

- multi-stage switching matrix
- basic switching elements are non-blocking
- reduced complexity
- non-blocking under given conditions



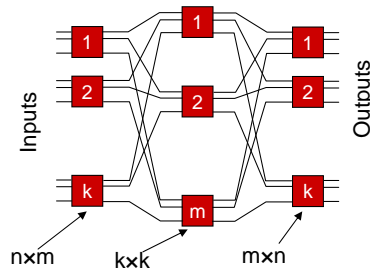
Crossbar vs. Clos network: complexity

Crossbar



complexity = N^2

3-stage symmetric Clos network



- $N = nk$
- non blocking when $m \geq 2n - 1$
- complexity = $2knm + mk^2$
- choosing n and m appropriately
→ complexity = $O(N^{3/2})$

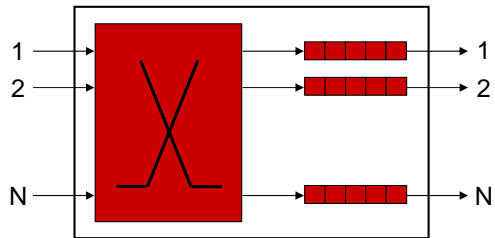
Clos network: non-blocking conditions

- Strict-sense non-blocking
 - a given unused input port on an ingress element must be connected to a given unused output port on an egress element
 - worst case scenario
 - all the $n - 1$ other inputs to the same ingress element are used and each is connected to a different middle-stage element
 - all the $n - 1$ other outputs of the same egress element are used and each is connected to a different middle-stage element
 - to be strictly non-blocking, at least an additional middle-stage element is required
 - $m \geq 2(n - 1) + 1$
- Rearrangeably non-blocking
 - it is always possible to set up a path between an unused input port and an unused output port by rearranging existing connections
 - $m \geq n$

Queue positioning within routers

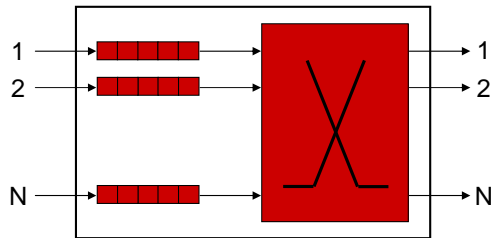
- Routers need to store packets before forwarding
 - even with non-blocking switching fabrics
 - contention resolution for packets directed to the same output port
- Alternatives:
 - Output Queuing – OQ
 - Input Queuing – IQ
 - Virtual Output Queuing – VOQ
 - Shared Buffer queuing – SB

Output queuing



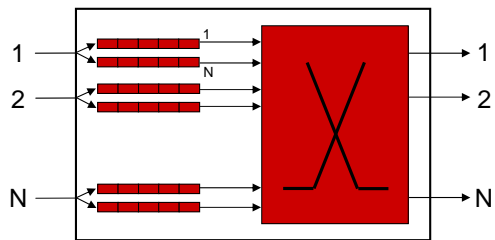
- Packets are queued after being switched
- They wait for the output link to become available (FCFS)
- If N packets arrive at the same time destined to the same output port, the switching fabric must forward all of them
- Switching speed must be at least equal to the sum of the input link speeds (e.g. $N \times$ input rate)
- Problems with high-speed input links

Input queuing



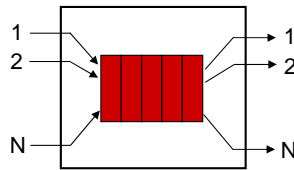
- Packets are queued before being switched
- No switching fabric speed-up
- Head-Of-Line (HOL) blocking
 - first packet in line is forwarded when the required output is available (FCFS)
 - meanwhile, following packets in the queue are not able to be forwarded, even though their required output port is available
 - maximum throughput achievable = 58.6%

Virtual output queuing



- Each input has a separate queue dedicated to each output
- Packets are queued before being switched, but after knowing the output port
- HOL blocking problem solved
 - 100% throughput attainable
- No switching fabric speed-up
- N^2 queues are needed
- More complex scheduler is required

Shared buffer queuing



- Packets are stored in a single shared memory, arranged in N logical output queues
- Better utilization of the shared buffering space
- A packet is dropped only when the shared memory is full
- More complex memory management
 - must be random access (no shift register)
 - must be fast (N packets inserted/extracted at a time)

Summary

- Router functions
- Forwarding table lookup
- Router architectures
- Buffer positioning