

Lecture 7 (Routing 4)

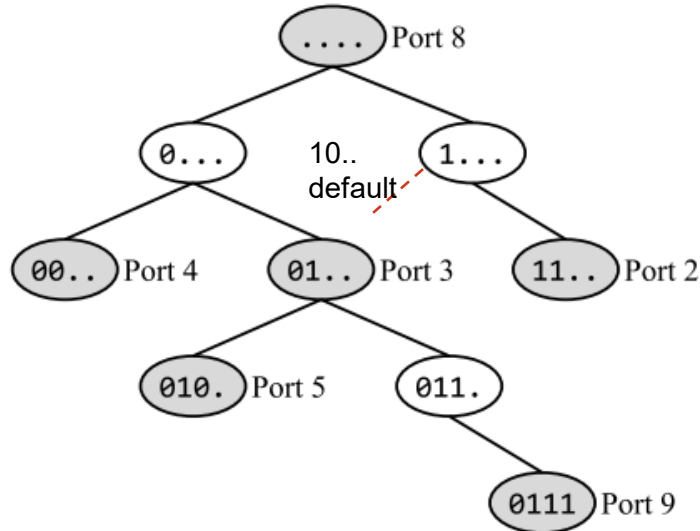
# IP Routers

# Exercises ANS

## Q1 Longest Prefix Matching

In this question, consider the longest-prefix-matching trie below. All subparts are independent.

- 1) How many of the 16 possible 4-bit IP prefixes are forwarded along the default route?
- 2) In this subpart only, suppose we delete the node labeled 010. (Port 5). How many of the 16 possible 4-bit IP prefixes are forwarded along a different port using the modified trie (compared to the original trie)?
- 3) Which of these IP prefixes represents exactly the set of IPv4 addresses that will get forwarded along Port 2?
- 4) How many 32-bit IPv4 addresses are forwarded along Port 3?



# Q1 Longest Prefix Matching ANS

1) How many of the 16 possible 4-bit IP prefixes are forwarded along the default route?

ANS: 4. All the 4-bit IP prefixes in the form 10.. will get forwarded along the default route, since it falls off the trie. There are 4 such prefixes: 1000, 1001, 1010, and 1011. The other 12 prefixes (with the form 00.. or 01.. or 11..) will not be forwarded along the default route

2) In this subpart only, suppose we delete the node labeled 010. (Port 5). How many of the 16 possible 4-bit IP prefixes are forwarded along a different port using the modified trie (compared to the original trie)?

ANS: 2. The prefix 010x, including 0100 and 0101, was forwarded along Port 5, but is now forwarded along Port 3. All other forwarding decisions remain unchanged.

3) Write the IP prefix that represents exactly the set of IPv4 addresses that will get forwarded along Port 2?

ANS: 192.0.0.0/2. IPv4 addresses are 32 bits, so the full prefix is: 11..... ..

Filling in zeros in all the unset bits: 11000000 00000000 00000000 00000000, which is 192.0.0.0

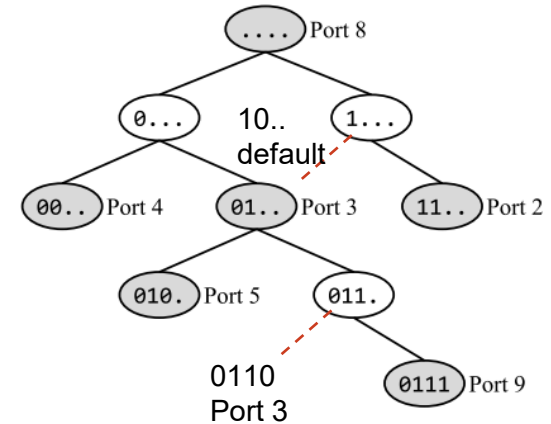
Add /2 to indicate that first two bits are fixed: 192.0.0.0/2

4) How many 32-bit IPv4 addresses are forwarded along Port 3?

ANS:  $2^{28}$ . Port 3 corresponds to the 01.. node. In this prefix, there are 30 bits unset, so there are  $2^{30}$  IPv4

addresses in this range. However, we have to subtract all the addresses in the 010. range ( $2^{29}$  addresses), and the 0111 range ( $2^{28}$  addresses), since they get forwarded out of ports 5 and 9, respectively (based on longest-prefix matching). This gives us a total of  $2^{30} - 2^{29} - 2^{28} = 2^{28}$  addresses forwarded along Port 3.

Another method is by realizing that only IPv4 addresses with 0110 as its prefix will be forwarded along Port 3 (based on longest-prefix matching). This means the next 28 bits can each be either 0 or 1, with a total of  $2^{28}$  possible addresses

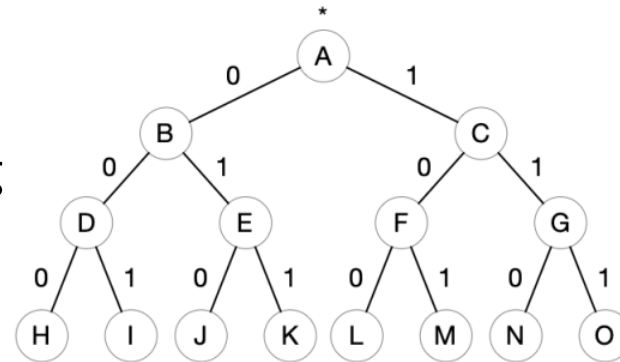


Red dashed lines are for illustration purposes, indicating what happens when falling off the trie. not part of the trie.

## Q2 Longest Prefix Matching (not covered in midterm)

- ▶ Consider a router with the following forwarding table. Fill out the table below with the port value that should be associated with each node in the prefix tree to implement the forwarding table of the router.
- ▶ Note that some rows may not be used; put NA on any unused rows. Your tree must use the minimum number of nodes possible (meaning the fewest nodes in the tree to reach all labeled nodes)

Destination	Port
0.0.0.0/3	1
32.0.0.0/3	2
64.0.0.0/3	2
96.0.0.0/3	3
128.0.0.0/3	4
Default	1

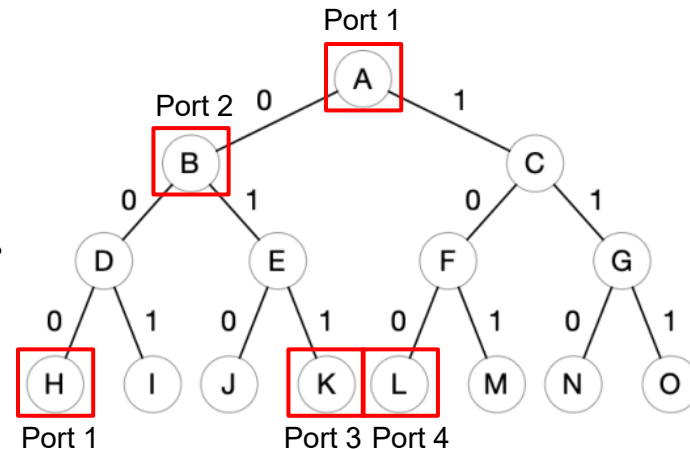


Node	Port
A	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	

# Q2 Longest Prefix Matching ANS

- ▶ We leverage **Longest Prefix Match (LPM)** to group routes together and only create nodes for the exceptions. First, let's look at the first 3 bits (the /3 subnet) of the destination networks:
  - ▶ **0.0.0.0/3:** Starts with 000 -> Port 1
  - ▶ **32.0.0.0/3:** Starts with 001 -> Port 2
  - ▶ **64.0.0.0/3:** Starts with 010 -> Port 2
  - ▶ **96.0.0.0/3:** Starts with 011 -> Port 3
  - ▶ **128.0.0.0/3:** Starts with 100 -> Port 4
- ▶ In the provided tree, the nodes represent binary paths:
  - ▶ **Level 0:** Node A (Root, \*)
  - ▶ **Level 1 (1 bit):** Node B (0), Node C (1)
  - ▶ **Level 2 (2 bits):** Node D (00), Node E (01), Node F (10), Node G (11)
  - ▶ **Level 3 (3 bits):** Node H (000), Node I (001), Node J (010), Node K (011), Node L (100), etc.
- ▶ Instead of explicitly creating individual entries for every /3 path, the solution aggressively compresses the rules:
  - ▶ **Node A = Port 1 (The Global Default):** The root node acts as the catch-all. Anything that doesn't match a more specific rule falls back to Port 1.
  - ▶ **Node B = Port 2 (The Grouping):** Both the 32.x (001) and 64.x (010) networks go to Port 2. Instead of defining Nodes I and J individually, we assign the *entire* 0 branch (Node B) to Port 2.
  - ▶ **Node H = Port 1 (Exception for 000):** Because we assigned everything starting with 0 to Port 2, the 000 block (0.0.0.0/3) would incorrectly be routed to Port 2. We add Node H (000) and map it back to Port 1. Because H is deeper in the tree, **LPM** ensures it overrides Node B.
  - ▶ **Node K = Port 3 (Exception for 011):** The 96.x block (011) also falls under the 0 branch, but it needs to go to Port 3. We explicitly map Node K to Port 3.
  - ▶ **Node L = Port 4 (Standalone Rule):** The 128.x block (100) falls under the 1 branch. Since the 1 branch inherits Port 1 from the root (Node A), we explicitly define Node L (100) as Port 4.
- ▶ With route aggregation, the router only needs to store 5 routing entries (A, B, H, K, L) rather than defining all paths, leaving the rest of the nodes as "NA".

Destination	Port
0.0.0.0/3	1
32.0.0.0/3	2
64.0.0.0/3	2
96.0.0.0/3	3
128.0.0.0/3	4
Default	1

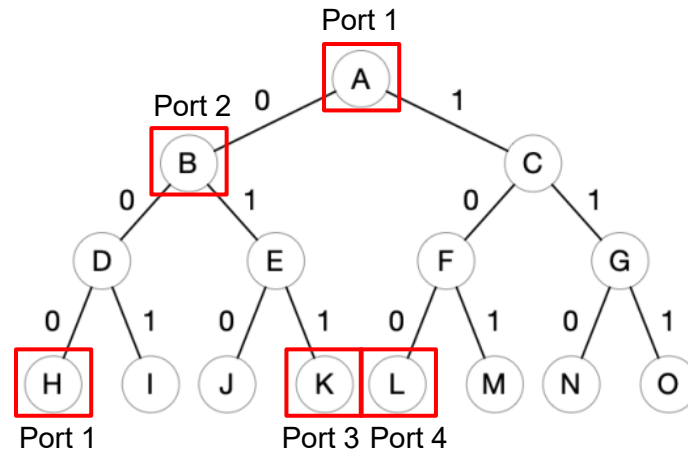


Node	Port
A	Port 1
B	Port 2
C	NA
D	NA
E	NA
F	NA
G	NA
H	Port 1
I	NA
J	NA
K	Port 3
L	Port 4
M	NA
N	NA
O	NA

# Q Routing 2 ANS

- ▶ 1. Packet: 10.0.0.1 (binary: 00001010...)
- ▶ Matches Root: Node A = Port 1 (LPM: /0)
- ▶ 2. Packet: 40.0.0.1 (binary: 00101000... = 32.0.0.0/3)
- ▶ Matches Node B = Port 2
- ▶ 3. Packet: 70.0.0.1 (binary: 01000110... = 64.0.0.0/3)
- ▶ Matches Node B = Port 2
- ▶ 4. Packet: 100.0.0.1 (binary: 01100100... = 96.0.0.0/3)
- ▶ Matches Node K = Port 3
- ▶ 5. Packet: 140.0.0.1 (binary: 10001100... = 128.0.0.0/3)
- ▶ Matches Node L = Port 4
- ▶ 6. Packet: 0.0.0.1 (binary: 00000000... = 0.0.0.0/3)
- ▶ Matches Node B and Node H. LPM means it matches Node H = Port 1

Destination	Port
0.0.0.0/3	1
32.0.0.0/3	2
64.0.0.0/3	2
96.0.0.0/3	3
128.0.0.0/3	4
Default	1



Node	Port
A	Port 1
B	Port 2
C	NA
D	NA
E	NA
F	NA
G	NA
H	Port 1
I	NA
J	NA
K	Port 3
L	Port 4
M	NA
N	NA
O	NA