Reducing TCAM Power Consumption and Increasing Throughput

Rina Panigrahy, Samar Sharma
(Cisco Systems)

Outline

- Problem
- Solution strategy
- Algorithms
- Further results

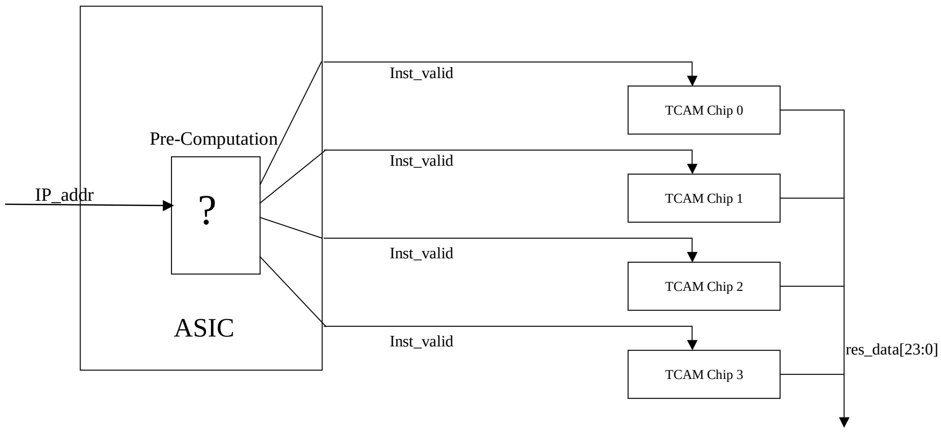
TCAM

- IP forwarding : core operation
- TCAMs are commonly used
- Easy to use, but lot of power (~15.3W)
- Multiple chips (4 to 8)
  - power = 8*15.3W = 123W
- Doesn’t scale in terms of power

Solution Strategy

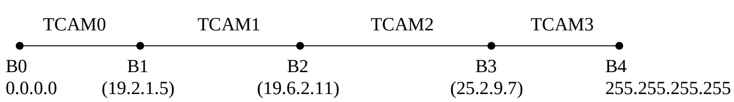
- Current method looks up every TCAM chip
- Idea : prune the search to one TCAM
  - pre-compute by looking at DestIP which TCAM chip to search
- What is such a pre-computation ?

Strategy



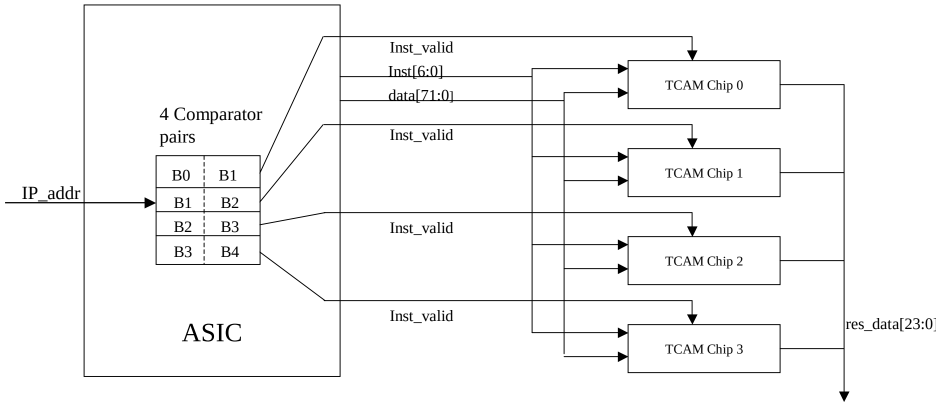
Solution

- IP address space : 0 to 2^{32}-1
- Divide this into 4 parts



- Given DestIP
  - Find which range it falls in
  - Lookup in ***only*** that TCAM

Solution

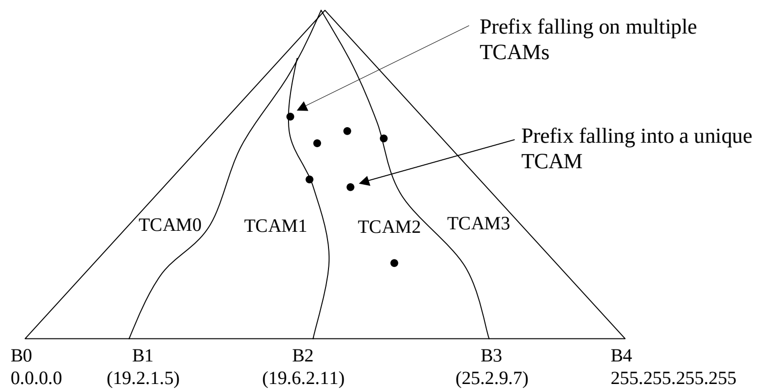


- TCAM0 contains all prefixes that can match IP addresses from B0 to B1; and so on

Solution Details

- $TCAM_i$ contains all prefixes that can match IP addresses from B_i to B_{i+1}.
- Look at the ranges [B0, B1], [B1, B2], …
- A prefix P belongs to $[B_i, B_{i+1}]$ if P can match some address in that range
- Some prefixes can belong to multiple ranges
- Most belong to a unique range.

Patricia Tree of IP prefixes



Prefixes in TCAM_i

- Look at the Patricia tree of the prefixes
- Look at paths
  - B0 to root, B1 to root, ….
  - Call these paths BP0, BP1, ….
- Look at the regions carved by these paths.
  - Call these regions R0, R1, ….
  - R_i has boundaries BP_i and BP_{i+1}
- i^{th} TCAM contains prefixes in R_i and in the boundary paths BP_i and BP_{i+1}

Choosing Boundaries

- Choose B_i's so that each region contains about equal number of prefixes.
- Each boundary path has at most 32 prefixes.
- Each TCAM has at most 64 additional shared prefixes.

Lookup

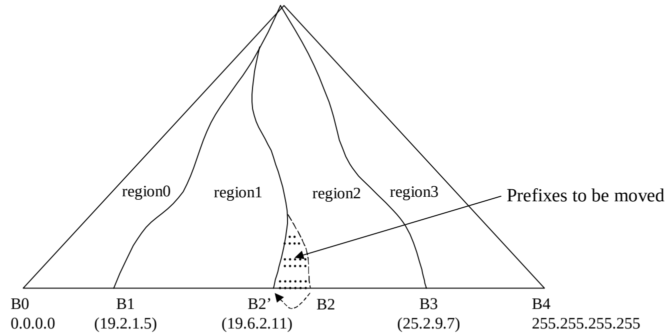
- The DestIP is sent to the ASIC
- The ASIC compares it with the Boundaries
- Now, the ASIC knows which TCAM to lookup
- Only that TCAM is looked up.

Inserts

- Divide the initial set of prefixes in equal sized regions.
  - Can be done by maintaining counts for each region
- During insert,
  - If there is free entry in the TCAM, use it
  - Else (rare) borrow a free entry from next TCAM by changing boundary.

Inserts (details)

Patricia Tree of IP prefixes



Insert (contd.)

- Say insert into a full $TCAM_i$
- Look at the boundary BP_{i+1}
- Let P be the first prefix strictly left of BP_{i+1}
- Set B_i = P11..1
- Insert all prefixes on the new boundary path into $TCAM_{i+1}$
- Good amortized cost. Worst case 32 per boundary

Insert (better worst case)

- Use “diffused” boundaries
  - A TCAM has some prefixes just outside its boundaries
- Worst Case: 1 insert per boundary

Further Results

- Can be used to construct a ~3W TCAM for forwarding.
- Gives worst case power and throughput bounds
- Increase throughput for given traffic distribution
  - Use all TCAMs concurrently for different lookups
- Scales for IPv6