

# Efficient Prefix Cache for Network Processors

*M. J. Akhbarizadeh and M. Nourani*  
{eazadeh, nourani}@utdallas.edu

Department of Electrical Engineering  
University of Texas at Dallas



System-on-Chip Design & Test Group

## Overview



- Use of address cache for IP packet forwarding
- Caching address ranges instead of IP addresses to reduce cache size
- Caching route prefixes directly
  - *Encompassing (Parent) prefixes* are the problem
  - *RRC-PR* avoids the parent prefixes and only places *disjoint* prefixes into the cache.
  - *RRC-ME* uses *Minimal Expansion* algorithm to handle parent prefixes.
- Reverse Routing-Cache (RRC) can reduce the cache size over 33 times.



System-on-Chip Design & Test Group

## Background and motivation

- IP addresses in a packet trace show strong locality of references.
  - But the resulting cache can be large and not affordable for many applications.
  - Such cache structure is called the *IP cache* in this presentation.
- Some designs place fixed size disjoint address ranges into cache to reduce size [Chiueh and Pradham, IEEE Micro, Feb 2000].
  - The FIB needs preprocessing.
  - Incremental updates to the FIB are not straightforward.

UTD

System-on-Chip Design & Test Group

## Statement of the problem

- Q: why don't they place prefixes directly into the cache?
- A: Parent prefixes are the main problem.
  - A hit in the cache with a parent prefix does not guarantee that there isn't a more specific match in the FIB.
- We offer two simple heuristics to handle the parent prefixes smoothly.

UTD

System-on-Chip Design & Test Group

## Fundamental approach



- Our contribution is the *Reverse Routing-Cache* (RRC) method of prefix caching.
  - Use ternary CAM to implement a fully associative cache that can store variable length prefixes.
  - When an address matches a prefix in the FIB, place that prefix in the cache.
  - FIB needs to be able to distinguish parent prefixes.
- LRU replacement policy is employed.
- Two heuristics to handle the problem of parent prefixes:
  - RRC with Parent prefix Restriction (**RRC-PR**)
  - RRC with Minimal Expansion of parent prefixes (**RRC-ME**)

UTD

System-on-Chip Design & Test Group

## RRC with Prefix Restriction (RRC-PR)



- The population of parent prefixes is small (less than 8%).
- Their share of LPM matches is usually below 25%.
- Follow the common-case-fast principal
  - Avoid placing the parent prefixes into the cache.
  - The majority of the traffic still takes advantage of the small and efficient route cache.

UTD

System-on-Chip Design & Test Group

## Statistical data to support RRC-PR

- In all the studied forwarding tables, the population of parent prefixes never passed 8%.
- In practice, ISPs resent issuing more specific IP addresses to their client networks.
- Even though the popularity of parent prefixes is more than average, it never passed 25% in our studies.

Forwarding table	Table size	Max parents population	Parents max share of traffic
AS1221	172k	6.7%	21.5%
AS4637	140k	7.2%	23.2%
AS6447	165k	6.3%	20.8%
MAE-West	28k	6.5%	19.7%
AADS	27k	6.8%	22.3%

(a) AADS Site

Sampling Date	Table Size	Number of Parents	Parent Percentage
03/14/02	21649	1453	6.7%
03/15/02	21604	1468	6.8%
03/19/02	21744	1483	6.8%

(b) MAE-WEST Site

Sampling Date	Table Size	Number of Parents	Parent Percentage
03/14/02	18619	1209	6.5%
03/15/02	18681	1211	6.5%
03/19/02	18768	1232	6.5%

(c) PacBell Site

Sampling Date	Table Size	Number of Parents	Parents Percentage
03/14/02	5222	379	7.2%
03/15/02	5239	379	7.2%
03/19/02	3875	233	6.0%

UTD

System-on-Chip Design & Test Group

## RRC with Minimal Expansion (RRC-ME)

- A parent prefix often indicates the aggregation of several networks under an ISP (roughly speaking)
- A match can be interpreted as the popularity of one of the networks/servers within the realm of that ISP.
  - So, caching a sub-range of the parent prefix that matches the given IP address, in the form of a more specific prefix, is usually effective.
- Use the *Minimal Expansion (ME)* algorithm to adaptively and incrementally place sub-ranges of the parent prefixes into the prefix cache.

UTD

System-on-Chip Design & Test Group



## Characteristics of the ME algorithm

- MEP is the shortest disjoint child of P that matches  $lp$ .
- The ME algorithm does not modify FIB
  - MEP is only added to the cache.
- For a trie-based FIB the ME algorithm does not introduce considerable overhead.
  - MEP runs in  $O(1)$  time after the search is done.
- For other FIB structures, for instance, when the search engine is TCAM:
  - A copy of the FIB might be placed in a trie structure for ME operation.
  - ME algorithm runs in  $O(W)$  time after the search is done.

UTD

System-on-Chip Design & Test Group

## Average search times

- Suppose  $h = \text{hit ratio}$ ; and  $m = 1 - h = \text{miss ratio}$ .
- Suppose that  $T$  is the miss penalty (FIB search time) and that the hit time is 1.
- For RRC-PR:  $T_{\text{srch}} = (1 - m) + mT$ .
- For RRC\_ME:  $T_{\text{srch-ME}} = (1 - m) + (1 + a_p - a_p K_p)T$ .
  - Where,  $a_p$  is the relative ratio of FIB matches with parent prefixes, and  $K_p$  shows how slower a parent miss is than a disjoint miss.
- For a trie-based FIB, capable of ME algorithm,  $K_p = 1$  and  $T_{\text{srch-ME}} = T_{\text{srch}}$ .

UTD

System-on-Chip Design & Test Group

## Coherence: insertion into FIB

- When a new route (say  $P_1$ ) is added to the FIB, it can turn a mirrored prefix into a parent.
- A parent prefix cannot reside in cache. It must be removed in order for the system to perform correctly.
- To find out the mirror of a new parent upon insertion of  $P_1$ :
  - Lookup the zero-padded  $P_1$  ( $P_10..0$ ) in cache.
  - Lookup the one-padded  $P_1$  ( $P_11..1$ ) in cache.
  - If both match the same cache entry, then that entry must be removed.
- Three memory accesses are necessary, hence  $O(1)$ .

UTD

System-on-Chip Design & Test Group

## Coherence: deletion from FIB

- When a route is being deleted from the FIB, its mirror in RRC should also be eliminated.
- For disjoint prefixes, a single lookup will find the possible mirror in the cache.
  - For parent prefix  $P_1$  that have MEPs in RRC-ME:
    - All MEPs of  $P_1$  must be eliminated.
    - MEPs are those prefixes of  $P_1$  that have the same data field (egress number or index) as  $P_1$ . Thus, associative search on both fields is needed.
  - That search can further be reduced to looking for all prefixes of  $P_1$ .
  - Less hardware but some good entries get deleted from the cache.

UTD

System-on-Chip Design & Test Group

## Finding MEPs in RRC-ME

- To find MEPs, the TCAM's capability for searching a masked key is exploited.
- In this mode of search, the TCAM circuit accepts an input mask together with the searchable key.
- The input mask is applied to all the TCAM words, while the TCAM internal masks are applied to the searchable key.
- Any match can be the input key's prefix (parent), or exact equal, as well as its child.
- Since parents are not allowed in RRC-ME, any match in this mode will be a child prefix.
  - MEPs are included within those matches

UTD

System-on-Chip Design & Test Group

## Complexity of delete coherence

- When a disjoint prefix is deleted from FIB, cache coherence takes only one cycle.
- When deleting parent prefixes from FIB, their possible MEs must be eliminated from the cache:
  - if *search and destroy* operation is supported by the TCAM then one lookup does the job.
  - Otherwise,  $N_{RRC} - 1$  cycles in the worst case, where  $N_{RRC}$  is the RRC size.
  - In practice the maximum number of children of a parent prefix in RRC-ME was 17.
  - The average cycles for coherence with a deletion was 1.25.

UTD

System-on-Chip Design & Test Group

## Use of RRC with network processors

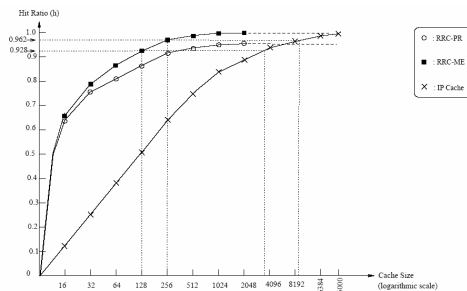
- RRC makes an efficient level 1 cache.
  - To significantly reduce the bus transaction with the external search engine.
  - To improve the performance of a multithreaded environment.
- Since it is small, multiple copies of such level one cache are affordable.
  - Maybe, one copy for each pipeline.
- A smaller and more efficient cache brings about the possibility of more sophisticated implementations

UTD

System-on-Chip Design & Test Group

## Performance evaluation

- RRC-ME, RRC-PR, and IP Cache were simulated in similar manners:
  - Fully associative,
  - LRU replacement policy.
- Real-life data used for simulation (only one example shown here and on the paper):
  - Sample of MAE-West routing table [[www.merit.edu/ipma](http://www.merit.edu/ipma)],
  - Traffic trace from the main router of a national laboratory.
- RRC-ME, and RRC-PR were sizes between 16 and 2048.
  - AT size 2048, the hit ratio is 0.998 for RRC-ME, 0.956 for RRC-PR, and only 0.917 for IP cache.



UTD

System-on-Chip Design & Test Group

## Cache sizes for two typical hit ratios

- For two typical hit ratios the necessary sizes are shown in the table.
- For hit ratios over 0.96, a size reduction more than 33 times can be achieved.
- The average latencies for a miss penalty of 120 ( $K_p=1$ ) are:
  - 9.52 for  $h=0.952$ ,
  - 5.52 for  $h=0.962$ .

Methods	Size for $h=0.928$	Size for $h=0.962$
RRC-ME	128	256
IP cache	3500	8600
Size improvement	27.3	33.6

UTD

System-on-Chip Design & Test Group

## Summary and conclusion

- Parent prefixes make prefix caching a difficult task.
- Previous art needs preprocessing of the FIB.
- Reverse Routing-Cache (RRC) was introduced.
  - It is a method to cache route prefixes efficiently.
- Two flavors of the proposed design:
  - RRC-PR avoids caching parent prefixes all together.
  - RRC-ME employs the minimal expansion algorithm to partially represent parent prefixes by their disjoint expansions.
- RRC-ME approaches the expansion task incrementally and adaptively.
- Upon FIB updates, cache coherence is efficiently possible through limited cache lookups.
- A size reduction of 33 times and more over a conventional IP address cache is possible.
- RRC-ME is a viable choice for level-1 network processor cache.

UTD

System-on-Chip Design & Test Group

# Thank You

*((Questions please))*

*Email: eazadeh@utdallas.edu*

Department of Electrical Engineering  
University of Texas at Dallas



System-on-Chip Design & Test Group