Advanced Topics in IP Networks
Lecture 03

Lecture by Dr. Anat Bremler-Barr
Typeset by Steven Karas

2018-11-01
Last edited 20:57:40 2018-11-01

1 Recap

Last week we talked about the structure of routers and introduced the basics of packet forwarding.

2 Packet Forwarding

At the link level, we need to know who we are. The ethernet layer has a hard-coded MAC address (typically burned into the chip, but can usually be configured). Then we either assume we have an IP address (static configuration), or get it dynamically (DHCP). To know how to forward a destination, we discover a neighboring IP's ethernet address using ARP. If the IP is not local, then we forward to our default gateway. DNS is used to translate a domain name into an IP address.

In 1994, classful IP allocations were replaced by CIDR. Note that IPv4 uses 32 bit addresses, whereas IPv6 uses 128 bit addresses.

Because line speeds are increasing, the lookup time for a forwarding strategy must be absolutely minimal.

2.1 Longest Prefix Match

Longest prefix match is the standard strategy for packet forwarding, where packets are forwarded according to the longest matching prefix. This is a harder problem than exact match.

2.2 Radix Trie

Give a W-bit prefix and N prefixes in a table, we have $O(W)$ lookup, $O(NW)$ storage, and $O(W)$ update complexities. This approach is simple and reuses mature algorithms with existing open implementations. However, the worst case lookup can be slow, and has the potential for wasting space.

If we compress prefixes, then we can reduce the space requirement to $O(N)$. However, this still requires many memory accesses.

To solve this problem, we can use a k-ary trie, with depth $\frac{W}{k}$ and degree 2^k. An example of the actual structure of this trie can be found on slide 64. The time complexity of lookup for this approach is $O\left(\frac{W}{k}\right)$. The space complexity is $\frac{NW}{k} \cdot 2^k$. This approach was presented in 1998[4].

2.3 Binary Search on Prefix Intervals

Presented in 1999[3].

In this approach, we perform a binary search on the prefixes, using the covering interval as a sort key. There are at most $2N + 1$ disjoint intervals. Prefix matching has complexity $\log_2 N$. However, updating this structure may cost $O(N)$. B-trees can be used to minimize memory accesses.

The principle here is to exploit the wide word width of the hardware.

2.4 32 Exact Matches

A naive approach to this is to simply run 32 exact match searches and pick the most specific one.

2.5 Binary Search on Prefix Lengths

Presented in 1997[6].

The principle here is to design a better algorithm that is tailored to the specific problem. Lookup here is $O(\log W)$. By adding markers to aid the binary search, we precompute the best matching prefix (bmp) to avoid backtracking. [1]

2.6 Multi Protocol Label Switching (MPLS)

MPLS attaches an extra header for forwarding inside a network. The header is an unstructured 32 bit field, with only link-local significance. The idea being to pass hints along in the header.

The connection table contains mappings from a input tuple of (port, label) to an output tuple (port, label) and a label operation. The lookup is by exact match. This is the same forwarding algorithm as Frame Relay and ATM.

This is used in practice by many networks, in particular by backbone providers to ensure load balancing.

2.7 Current State of the Art

For 40Gbps speeds, memory pipelining and SRAM hardware works well enough. This is tricker than it sounds because the memory allocations among stages is nontrivial.

For 10Gbps speeds, multibit tries with DRAM, RAMBUS, and pipelines work. Compression algorithms can be used to fit tries into SRAM.

[1]The HW questions are about this approach. There is an example on slide 70.

3 Classification

There are many motivations for packet classification. Accounting, billing, rate limiting, quality of service, packet filtering, and policy based routing are a few examples.

However, because IP is a stateless/sessionless protocol, we need to identify flows. Flows are typically defined as a 5-tuple (source IP, destination IP, source port, destination port, transport protocol). A router then needs to apply special processing based on policy.

Formal definition Given a classifier C with N rules R_j with $1 \leq j \leq N$, where R_j is a 3-tuple:

1. A regular expression $R_j[i]$, $1 \leq i \leq d$ on each of the d header fields
2. A number pri(R_j), indicating the priority of the rule in the classifier
3. An action, referred to as action(R_j)

For example, we can consider routing lookup as a one dimensional classifier with routing table entries as the rules, the outgoing interface as the action, and the prefix length as the priority.

3.1 Classifier algorithms

3.1.1 Assessment Metrics

- Speed
- Storage complexity
- Scalability w.r.t. number of rules
- Scalability w.r.t. number of dimensions
- Preprocessing time
- Update time
- Flexibility in rule specification

For example, in micro-flow recognition, we can expect approximately 1M flows, with a very high update rate, but with very few wildcards. On the other extreme, firewall applications will have few rules with many wildcards and human-scale update rates.

3.1.2 Linear Search

Store rules in a linked list. $O(N)$ storage and lookup time, but $O(1)$ update.

3.1.3 Ternary Content Addressable Memory (TCAM)

Memory device with fixed width arrays. Each bit is 0, 1, or x (don't care). Lookup is done against all entries in parallel, and the first result is returned.

Common widths in contemporary hardware are 80 to 144 bits. The major disadvantage of this is the high cost and energy consumption. However, the advantage is $O(1)$ search for $O(2^{15})$ rules. This also does not map well to matching ranges.

Prefix Expansion Expressing ranges as a set of prefixes with separate TCAM entries for each prefix. Worst case, this can give us $2(W-2)$ entries for a set of rules. For practical applications, we typically see x6 space expansion.

References

[1] Mark Crovella and Balachander Krishnamurthy. *Internet Measurement: Infrastructure, Traffic and Applications.* John Wiley & Sons, Inc., New York, NY, USA, 2006.

[2] James F. Kurose and Keith Ross. *Computer Networking: A Top-Down Approach Featuring the Internet.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2002.

[3] Butler Lampson, Venkatachary Srinivasan, and George Varghese. Ip lookups using multiway and multicolumn search. *IEEE/ACM Trans. Netw.*, 7(3):324–334, June 1999.

[4] V. Srinivasan and George Varghese. Faster ip lookups using controlled prefix expansion. *SIGMETRICS Perform. Eval. Rev.*, 26(1):1–10, June 1998.

[5] George Varghese. *Network Algorithmics,: An Interdisciplinary Approach to Designing Fast Networked Devices (The Morgan Kaufmann Series in Networking)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[6] Marcel Waldvogel, George Varghese, Jon Turner, and Bernhard Plattner. Scalable high speed ip routing lookups. In *Proceedings of the ACM SIGCOMM '97 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '97, pages 25–36, New York, NY, USA, 1997. ACM.