CS 7260 – Internet Architecture and Protocols
Scribe notes for 21-Sep-2012
Taken by Ajay Raghav, Gurubabu

Extended grid of tries

Destination	Source	Destination Port	Source Port	Flags	Comments
M	*	25	*	*	Allow inbound mail
M	*	53	*	UDP	Allow DNS access
M	S	53	*	*	Secondary access
M	*	23	*	*	Incoming telnet
TI	*TO*	123	123	UDP	NTP time info
*	Net	*	*	*	Outgoing packets
Net	*	*	*	TCP ack	Return ACKs OK
*	*	*	*	*	Block everything!

The Worst case complexity for packet classification on "k" fields can be quite bad when the value of "k" is high and the number of rules are more. Hence we formulate the next few algorithms.

1st algorithm – Extended grid of trie.

Grid of trie is a solution for packet classification over 2 fields. What if there is a classification of 5 fields ?

Build a trie as usual, destination IP and then the source IP and when the matching node is found at the source trie, decide on which other packet classification you would need to take on next.

Data Structure for Algo 1

At each node "A", store all the rules set that "A" match with in the rule set

Eg – DST is M and SRC is S. From the table, this pair matches with R1, R2, R3, R3 and R7.
(if you are wondering how R7 is in the list, it is because the destination M is inside the bigger Net cloud)

Working of Algo 1

Walk down the Destination Trie as far as you can
Then walk down the Source Trie as far as you can
and see the set of Matching Rules

But there is a catch with algo 1. Because the number of matching rules for a Destination and Source Pair is not going to be large, we can find the exact rule, in smaller number of checks. But since each such node has to house the set of rules that match that node, the storage cost can be exorbitant for more packet classification fields and rules count.

Hence algorithm 2 – using switch pointer.

Trick – in Packet Classification over 2 fields, we store one value which is the highest priority rule that matches the node. But in a Packet Classification over "k" fields, we store more than one such rule in each and every node.

Recap – switch pointers allows you to walk through the skyline points.

=> We can pre compute in a Packet Classification over 2 fields. But it is hard to pre compute in a Packet Classification over "k" fields <=

Next algorithm – Bit vector Linear Search

All till now, we have looked at finding LPM for exact IPs. But what if the given IP set is a range of IP? Given an IP packet, how do we perform a LPM on all 5 fields ?

- We can do it using Bit Vector Linear Search.

Index	DST	SRC	DSTPort	SRCPort	Flags
1	M	*	25	*	*
2	T1	S	53	123	UDP
3	*	T0	23		TCP-ACK
4	NET	NET	*		

=> The above table is got by taking the unique entries in each column from the rule set shared at the beginning of this scribe <=

On demand cross producting

Instead of precomputing the indices, we compute this at run time. Every time a packet is processed, check the cache of if the matching entry is found. If found, proceed. Else, do the LPM and get the 5 tuple index and add it to the cache.

But what if the packet is classified across multiple rules ? consider,

R1	**M**	*	**25**	*****	*****
R2	T1	**S**	53	123	UDP

What if a packet is classified across both R1 and R2 where the RED fields show the matching fields, spawn across two rules. ?

In this case, we precompute on columns, using bit vectors.

If LPM(Dst) is M, then the rules which M matches are made '1' and rest as '0'
-> 1 1 1 1 0 1 1

If LPM(dst) is T1, then
-> 0 0 0 0 1 1 1

Similarly,
If LPM(Dst) = * , then -> 0 0 0 0 0 0 0 1 0
And if LPM(Dst) = NET, then - > 0 0 0 0 0 1 1

These are called the Bit Vectors.

So, for the packet given above, the bit vectors are,

LPM(Dst) = M	->	1 1 1 1 0 1 1
LPM(Src) = S	->	1 1 1 1 0 0 1
LPM(Dstport) = 25	->	1 0 0 0 0 1 1
LPM(SrcPort) = *	->	1 1 1 1 0 1 1
LPM(Flags) = *	->	1 0 1 1 0 1 0
BIT WISE AND	->	1 0 0 0 0 0 0 (Highest Priority Rule to Lowest Priority Rule)

Here, there is only one bit which is set to 1 and Rule 1 is chosen for the packet and actions are performed accordingly. But when there are multiple bits set to 1, we chose the highest priority rule. This is taken by taking the left most rule which matches.

If you are wondering about the efficiency and time taken when the packet classifications are over many fields and when there are a million rules, surprisingly, it is not a big issue as the present hardware are capable of performing more than 1000 bit wise operations per second.

>> END <<