CS 838 Algorithmic design of fast network systems

Quiz 2 solution

Page 300 exercise 1. After you're done, answer the same question if routers are allowed to have negated port ranges. Discuss the encoding of the following two ranges: !25 matching all source ports except 25 and !1024-4095 matching all destination ports except those between 1024 and 4095.

Extra credit: You can encode port numbers before you send them to the TCAM (and of course also set the corresponding bits in the TCAM keys accordingly) using a mapping of your choice (you need not use one of the mappings discussed in class, you can give any mapping that is correct). **Only for the port ranges below**, come up with a compact encoding (as few bits per port number as possible) that ensures that the number of TCAM entries is exactly the number of rules in the database. Give the TCAM entries for all the ranges below, and the encodings for ports 1000, 2000, 3000, 5000, and 10000.

Ranges		Ports	
80	0001		
25	0010		
53	0011		
119	0100		
23	0101		
0-1,023	0***	1,000	0000
1,024-65,535	1***	2,000	1100
1,024-9,999	11**	3,000	1100
4,096-65,535	1*1*	5,000	1110
1,214	1101	10,000	1010
6,667	1111		

Ranges: 80, 25, 53, 119, 23, 0-1023, 1024-65535, 1024-9999, 4096-65535, 1214, 6667

The worst case prefix expansion for a word size of w (16) is 2w-2 prefixes (30). We achieve this worst case for $[1,2^{16}-2]=[0000000000000001_2,1111111111111110_2]$ which is represented by the following prefixes:
$0000000000000001=[1,1] =[2^0, 2^1-1]$
$000000000000001*=[2,3] =[2^1, 2^2-1]$
$00000000000001**=[4,7] =[2^2, 2^3-1]$
…
$01*************=[2^{14},2^{15}-1]$
$10*************=[2^{16}-2^{15}, 2^{16}- 2^{14}-1]$
…
$11111111111110**=[2^{16}-2^3, 2^{16}- 2^2-1]$
$111111111111110*=[2^{16}-2^2, 2^{16}- 2^1-1]$
$1111111111111110=[2^{16}-2^1, 2^{16}- 2^0-1]$

The range $[1024,65535] =[2^{10},2^{16}-1]$ is represented by the following 6 prefixes:
$000001**********=[2^{10},2^{11}-1]$
$00001***********=[2^{11},2^{12}-1]$
$0001************=[2^{12},2^{13}-1]$
$001*************=[2^{13},2^{14}-1]$
$01**************=[2^{14},2^{15}-1]$
$1***************=[2^{15},2^{16}-1]$

The worst case prefix expansion for negated port ranges is also 2w-2=30 and it is achieved by the range [32768,32769]= ![2^{15},2^{15}+1]

To represent !25=!0000000000011001$_2$=[0,24]U[26,65535] we use these 16 prefixes:
000000000000****=[0,15]
0000000000010***=[16,23]
0000000000011000=[24,24]
000000000001101*=[26,27]
0000000000111**=[28,31]
00000000001*****=[32,63]=[2^5,2^6-1]
0000000001******=[64,127]=[2^6,2^7-1]
…
01**************=[2^{14},2^{15}-1]
1***************=[2^{15},2^{16}-1]

To represent ![1024,4095]=![2^{10},2^{12}-1]= [0,2^{10}-1] U[2^{12},2^{16}-1] we use these 5 prefixes:
000000**********=[0,1023] =[0,2^{10}-1]
0001************=[2^{12},2^{13}-1]
001*************=[2^{13},2^{14}-1]
01**************=[2^{14},2^{15}-1]
1***************=[2^{15},2^{16}-1]

```
BuildPrefixes(Min,Max){
  if(Min==Max) return Max
  LCP = LongestCommonPrefix(Min,Max)
  RMin = remove LCP followed by 1 from Max
  RMax = remove LCP followed by 0 from Min
  Result = prepend LCP followed by 1 to PrefixesUpTo(RMax)
    U prepend LCP followed by 0 to PrefixesDownTo(RMin)
  return Result
}
PrefixesUpTo(RMax){
  Prefix = RMax with 1s right of rightmost 0 replaced by *s
  First = RMax with 1s right of rightmost 0 replaced by 0s
  if (First == all 0s) return {Prefix}
  NewRMax = First – 1          // arithmetic subtraction
  return {Prefix} U PrefixesUpTo(NewRMax)
}
PrefixesDownTo(RMin){
  Prefix = RMin with 0s right of rightmost 1 replaces by *s
  Last = RMin with 0s right of rightmost 1 replaced by 1s
  if (Last == all 1s) return {Prefix}
  NewRMin = Last + 1           // arithmetic addition
  Return {Prefix} U PrefixesDownTo(NewRMin)
}
```