CS 181AG
Lecture 12

Packet Classification Cont.

Arthi Padmanabhan

Oct 10, 2022

Feedback

- Common themes:
  - Office hours > lecture + activities > assignments > slides
  - Assignments reinforce material without taking 10+ hours a week
  - Having lecture slides beforehand (with room to annotate) is important
  - Slides are not a great study material on their own
    - Lecture recordings are available
    - Strongly encourage you to come to office hours
  - Starter code could be explained more
    - Will try!

Midterm

- Take-home midterm, released Oct 25 8am, due Oct 30 10pm (on Gradescope)
- Must be done in one sitting (no limit on time), though it will be written to take about 3 hrs
- Open: lecture slides, videos, textbook, assignments, NOT open: anything else on internet, conversations with classmates
- Will cover everything through 10/12
- For fairness, I will not respond to any questions about midterm content. If anything seems ambiguous, please clearly write down your assumptions and I will grade accordingly (within reason ☺)

Upcoming Schedule

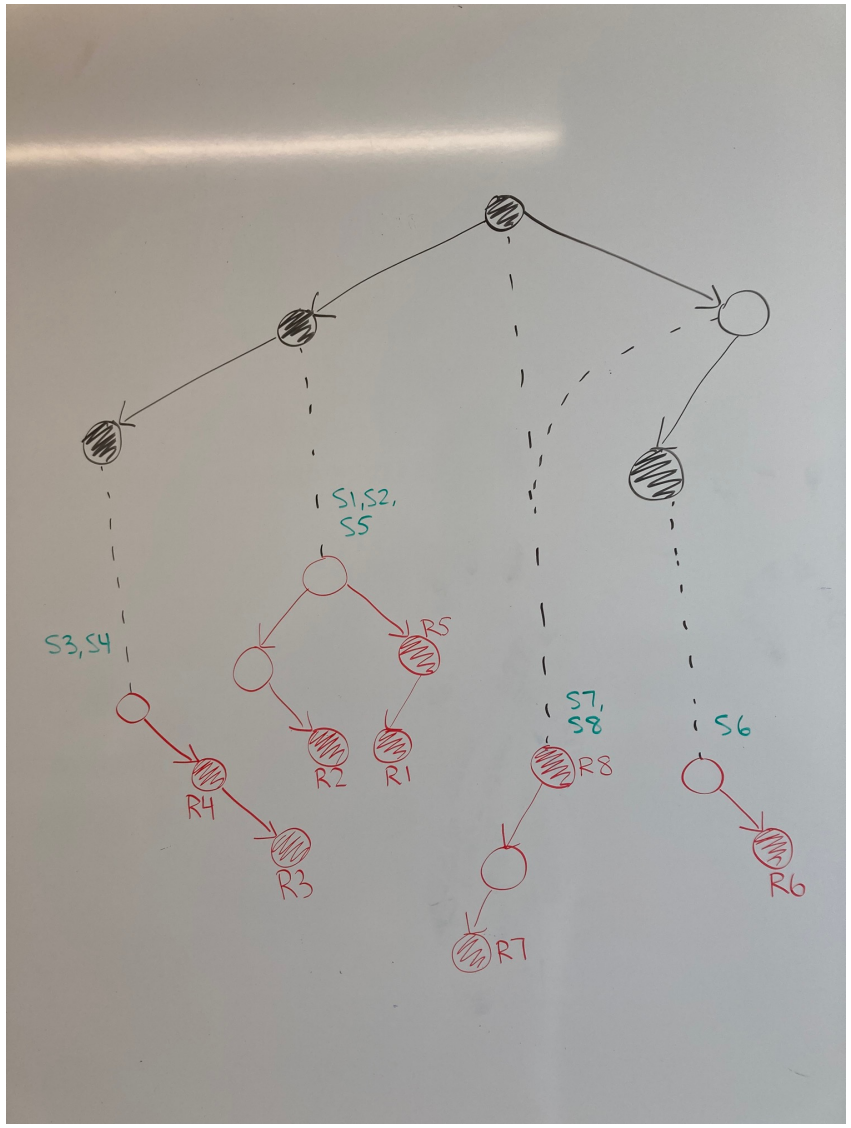
- 10/10, 10/12: Finish packet classification
- 10/17-10/18: Fall break!
  - No assignment due week of fall break
- 10/19: Start new topic (switching)
- 10/24: Midterm Review in class
  - No assignment due week of midterm
- 10/25 – 10/30: Midterm
  - No assignment due right after midterm
- Assignment 7 due Nov. 7 (Monday)

2D Schemes: Grid of Tries

Rule	Destination	Source
R1	D1 = 0*	S1 = 10*
R2	D2 = 0*	S2 = 01*
R3	D3 = 00*	S3 = 11*
R4	D4 = 00*	S4 = 1*
R5	D5 = 0*	S5 = 1*
R6	D6 = 10*	S6 = 1*
R7	D7 = *	S7 = 00*
R8	D8 = *	S8 = *

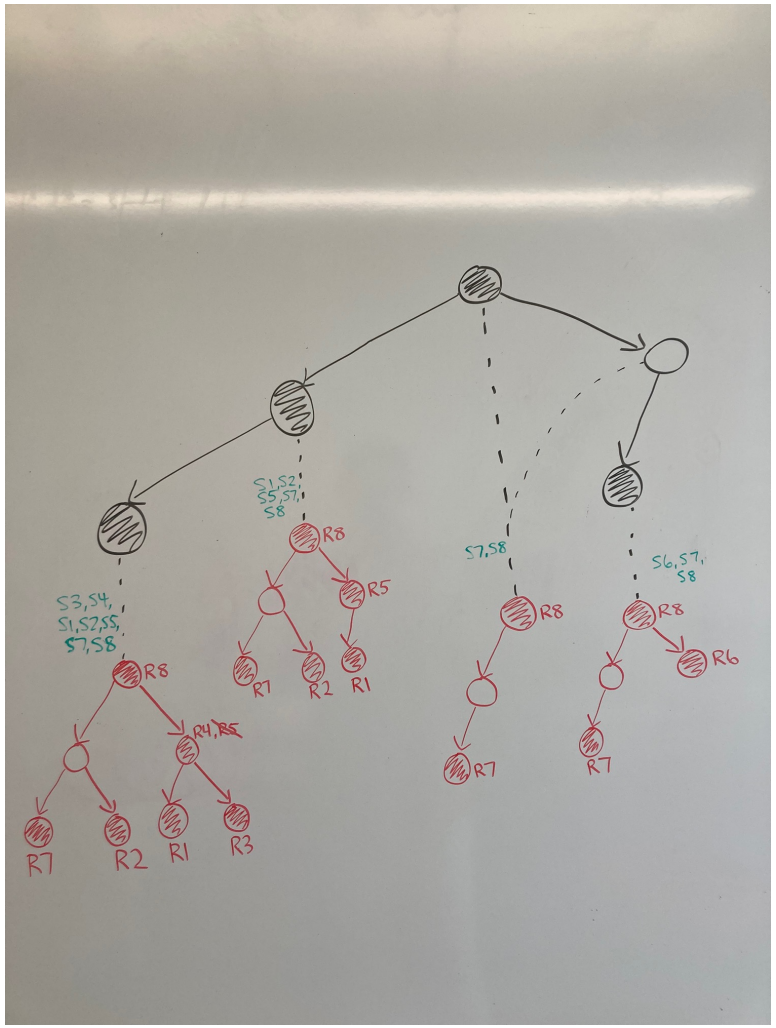
2D Schemes: Grid of Tries

- Grid of tries using backtracking (each source appears exactly once in entire grid of tries). Find longest destination D, check corresponding source trie, then backtrack to all prefixes of D, keeping track of lowest-cost rule so far



2D Schemes: Grid of Tries

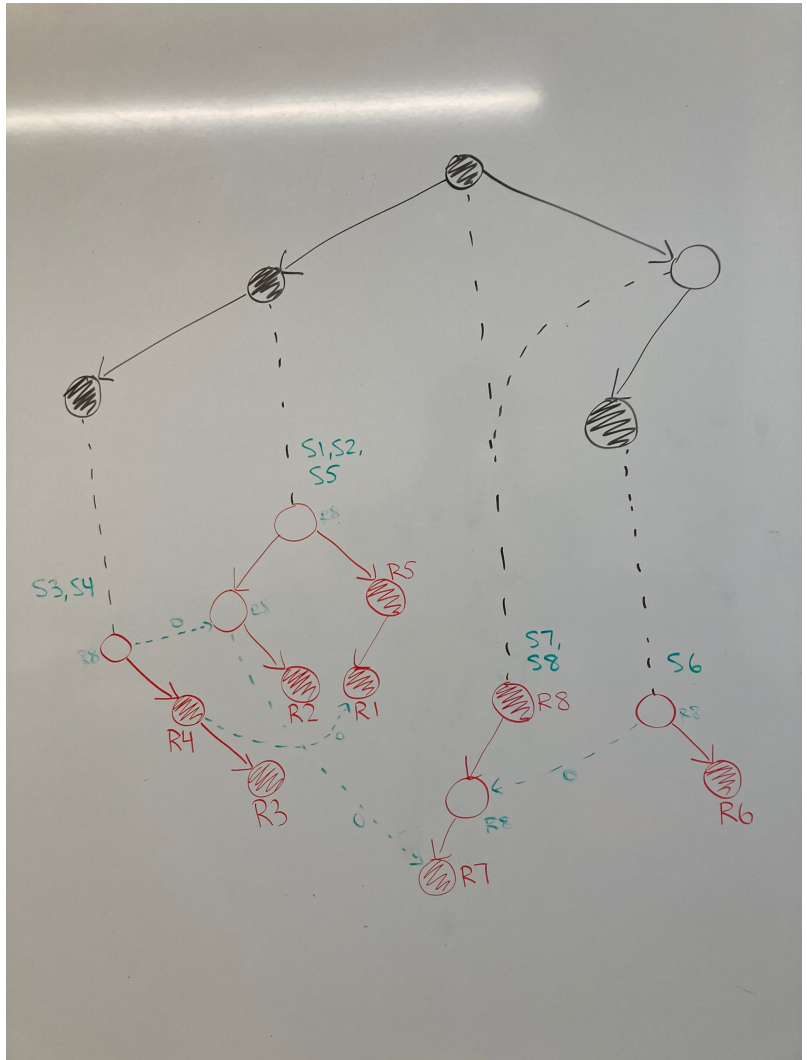
- Grid of tries without backtracking: once longest matching destination is found, all possible sources are contained in corresponding source trie



Trie of Tries

- Construct a trie of destination prefixes
- Each valid destination prefix (D) points to a trie of source prefixes
  - The source trie contains source prefixes for all rules with a destination field exactly equal to D
- Instead of backtracking, let’s add switch pointers, pointing us to the next location we would backtrack to if it exists

Switch Pointers



One More Example

- Draw all switch pointers for correctness, but remember: it is very important that when looking up the source trie, we keep track of the best rule we've seen so far
  - i.e., it is possible for switch pointers to point us to a worse rule, but we need to remember the best we've seen so far

Rule	Destination	Source
R1	D1 = 101*	S1 = 0*
R2	D2 = 10*	S2 = 1*
R3	D3 = 0*	S3 = 10*
R4	D4 = 0*	S4 = 00*
R5	D5 = *	S5 = 00*
R6	D6 = *	S6 = 01*
R7	D7 = *	S7 = *

Geometrical View of Classification

- Each combination of fields represents a single point in space
- Can assess complexity of classifier by number of disjoint regions
- Observations:
  - Number of disjoint regions is small compared to worst case (N rules, K fields -> worst case of N^k. Reality is closer to N*k
  - Once source and destination are matched, less than 20 remaining matching rules for all other fields combined

Divide and Conquer Algorithms

- Solve best match by field, then efficiently put results together

Using Bitmaps

- Key observation: operations on bitmaps can be sped up by hardware

Rule	Destination	Source	Dest. Port	Src Port	Flags
R1	D1 = 0*	S1 = 10*	25	*	*
R2	D2 = 0*	S2 = 01*	25	123	*
R3	D3 = 00*	S3 = 11*	53	*	UDP
R4	D4 = 00*	S4 = 1*	*	*	*
R5	D5 = 0*	S5 = 1*	20	*	*
R6	D6 = 10*	S6 = 1*	*	*	TCP_ack
R7	D7 = *	S7 = 00*	*	*	UDP
R8	D8 = *	S8 = *	*	*	*

Using Bitmaps

- Find least cost rule using bitmaps for the following (for simplicity, IPs will be 4 bits):
  - (0001, 0011, 53, 35, UDP)
  - (0111, 1011, 25, 30, -)
  - (0011, 1000, 20, 30, UDP)

Rule	Destination	Source	Dest. Port	Src Port	Flags
R1	D1 = 0*	S1 = 10*	25	*	*
R2	D2 = 0*	S2 = 01*	25	123	*
R3	D3 = 00*	S3 = 11*	53	*	UDP
R4	D4 = 00*	S4 = 1*	*	*	*
R5	D5 = 0*	S5 = 1*	20	*	*
R6	D6 = 10*	S6 = 1*	*	*	TCP_ack
R7	D7 = *	S7 = 00*	*	*	UDP
R8	D8 = *	S8 = *	*	*	*

Using Bitmaps

- Does require linear search
  - Summary bits can help, but still requires linear search of summary bits

Rule	Destination	Source	Dest. Port	Src Port	Flags
R1	D1 = 0*	S1 = 10*	25	*	*
R2	D2 = 0*	S2 = 01*	25	123	*
R3	D3 = 00*	S3 = 11*	53	*	UDP
R4	D4 = 00*	S4 = 1*	*	*	*
R5	D5 = 0*	S5 = 1*	20	*	*
R6	D6 = 10*	S6 = 1*	*	*	TCP_ack
R7	D7 = *	S7 = 00*	*	*	UDP
R8	D8 = *	S8 = *	*	*	*

Cross Producting

- Divide and Conquer algorithm (solve best match by field, then put results together)
- Precompute result for each possible best match
- Problem?