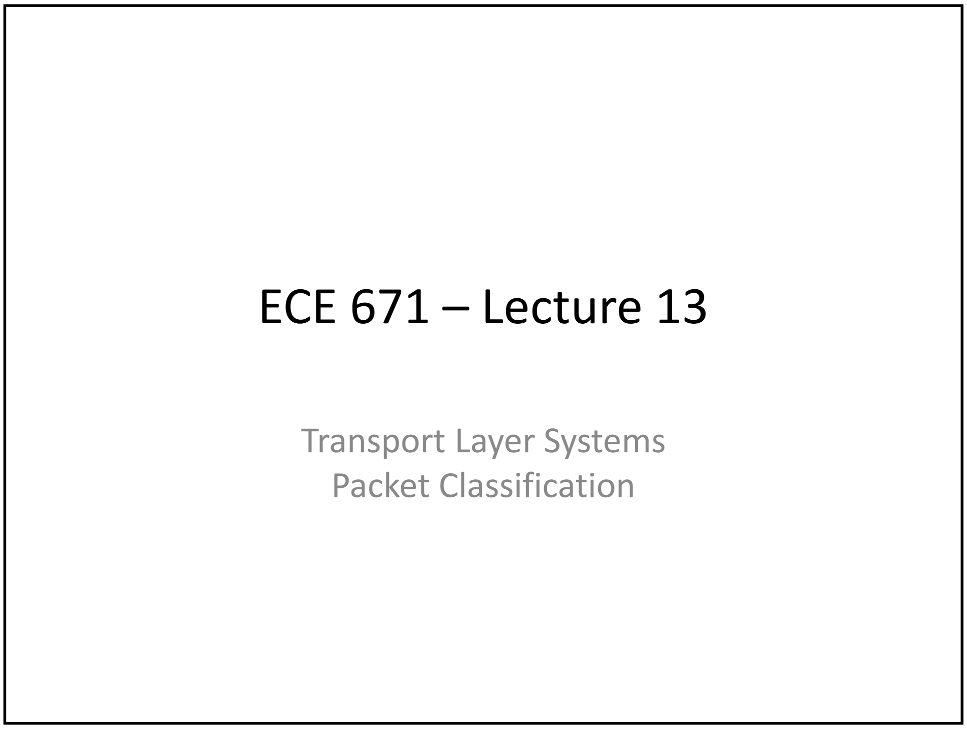
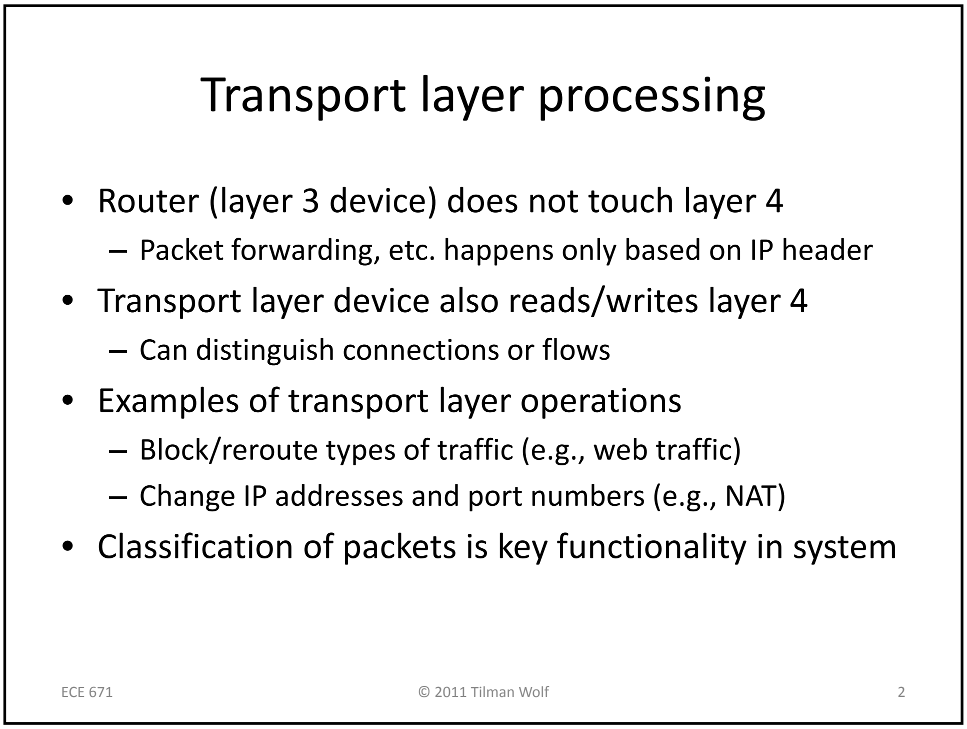


ECE 671 – Lecture 13

Transport Layer Systems
Packet Classification

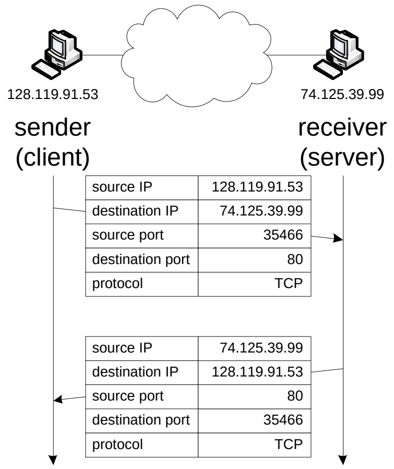


Transport layer processing

- Router (layer 3 device) does not touch layer 4
  - Packet forwarding, etc. happens only based on IP header
- Transport layer device also reads/writes layer 4
  - Can distinguish connections or flows
- Examples of transport layer operations
  - Block/reroute types of traffic (e.g., web traffic)
  - Change IP addresses and port numbers (e.g., NAT)
- Classification of packets is key functionality in system

5-Tuple

- 5-tuple identifies traffic
  - IP addresses (src and dst)
  - Port numbers (src and dst)
  - Layer 4 protocol (e.g., TCP)
- Single connection
  - 5-tuple fully specified
  - “Flow classification”
- Classes of traffic
  - 5-tuple partially specified
  - “Matching”



source IP	128.119.91.53
destination IP	74.125.39.99
source port	35466
destination port	80
protocol	TCP

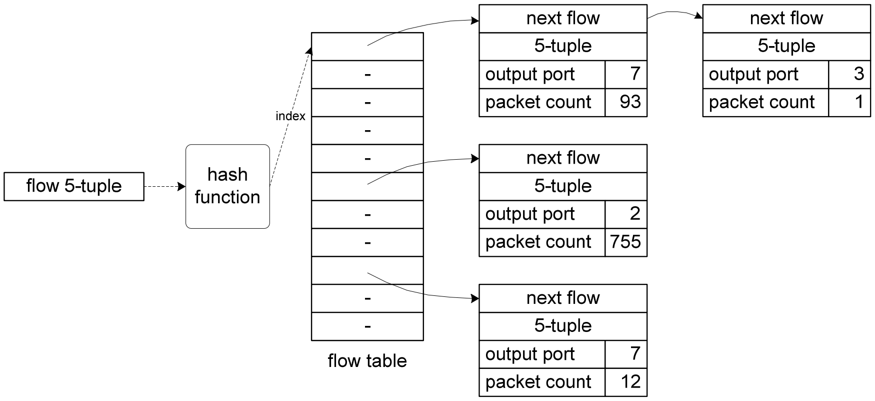
source IP	74.125.39.99
destination IP	128.119.91.53
source port	80
destination port	35466
protocol	TCP

Flow classification

- How to keep track of all (active) flows in system?

Flow classification

- Data structure for flow records
  - Hash function reduces 5-tuple space to size of flow table



Matching problem

- Example set of matching rules:

Source IP	Destination IP	Source port	Destination port	Protocol	Action
128.252.*	*	*	*	TCP	permit
*	128.252.*	*	80	TCP	permit
128.252.*	129.69.8.*	*	554	UDP	permit
150.140.129.*	128.252.*	[1024-65535]	*	*	permit
*	*	*	*	*	deny

- Need to determine what rule applies to a packet
- What are the challenges?

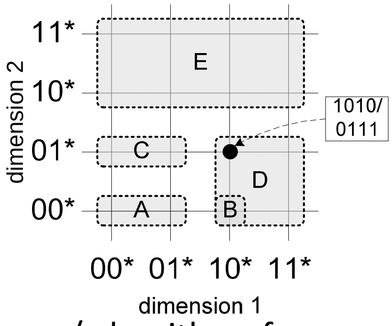
Matching problem

- Challenges
  - Very large space of potential rules
  - Wildcards cause rules to overlap
  - Potentially conflicting actions
- Assumption:
  - Priority order of rules (lower rule index gets priority)
- Maintenance of rule set very difficult in practice
  - Manual verification of “correctness”

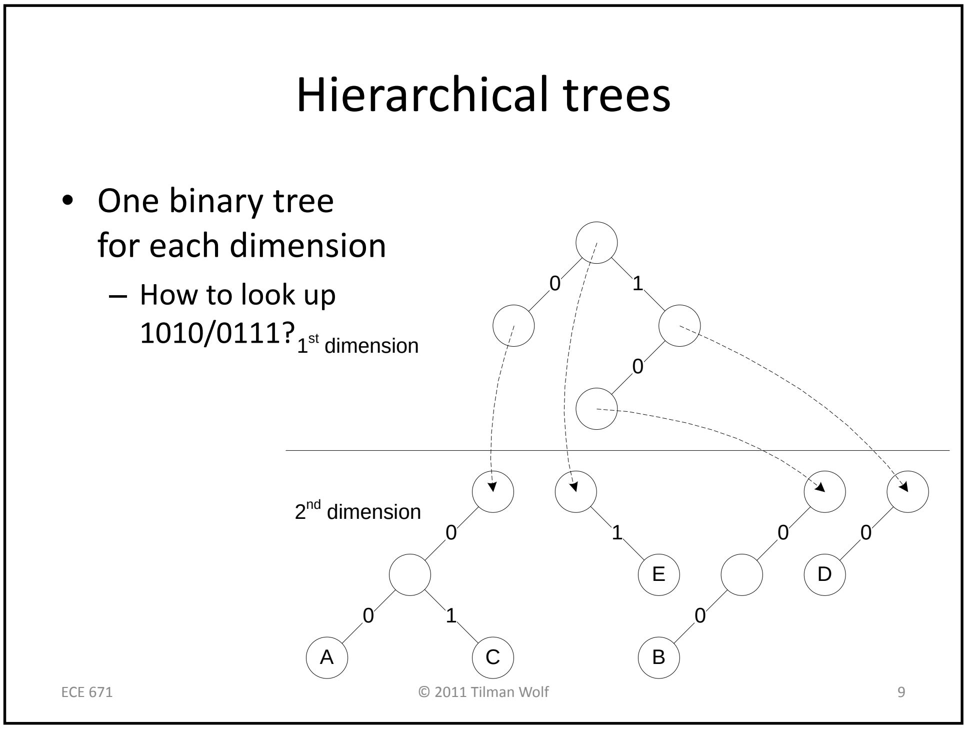
Matching algorithms

- Example rules for algorithms:
  - Only 2 dimensions

Rule	1st field	2nd field
A	0*	00*
B	10*	00*
C	0*	01*
D	1*	0*
E	*	1*

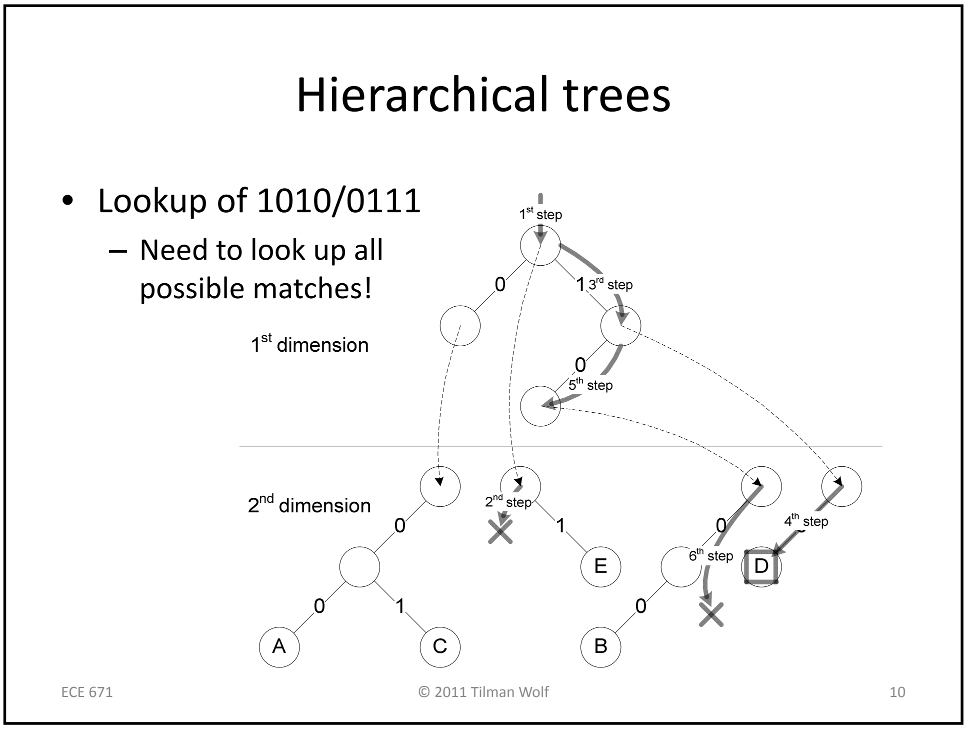


- What are suitable data structures / algorithms for matching?



Hierarchical trees

- One binary tree for each dimension
  - How to look up 1010/0111?

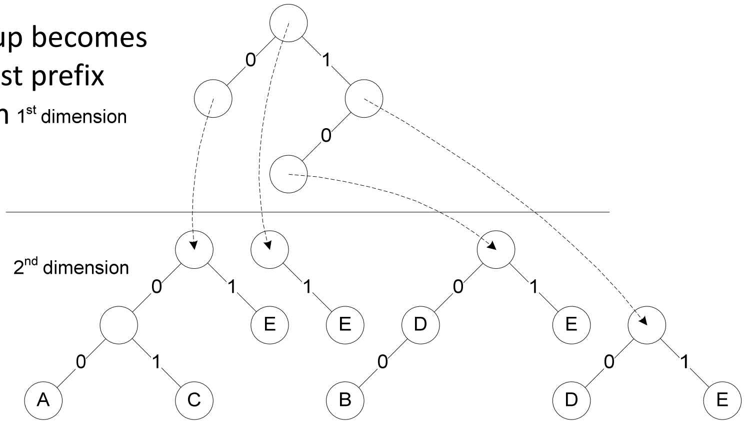


Hierarchical trees

- Lookup of 1010/0111
  - Need to look up all possible matches!

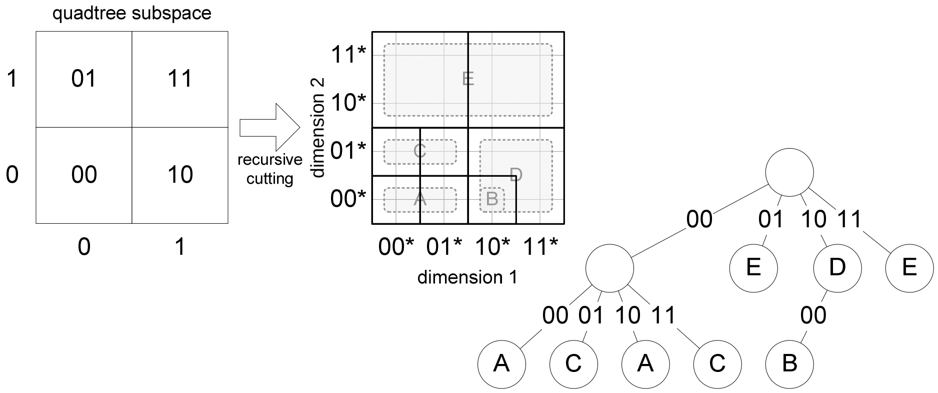
Set-pruning trees

- Second dimension includes all rules for shorter prefixes in first dimension
  - Lookup becomes longest prefix match



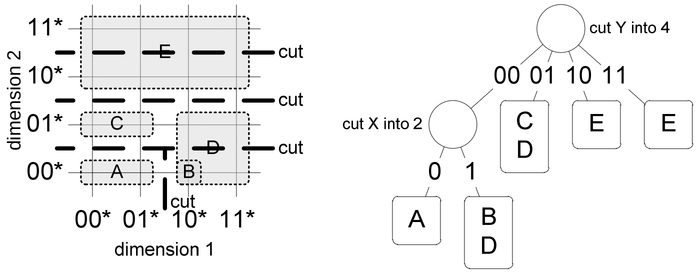
Area-based quadtree

- Look up one bit from each dimension in one step
  - Recursive cutting of areas as necessary



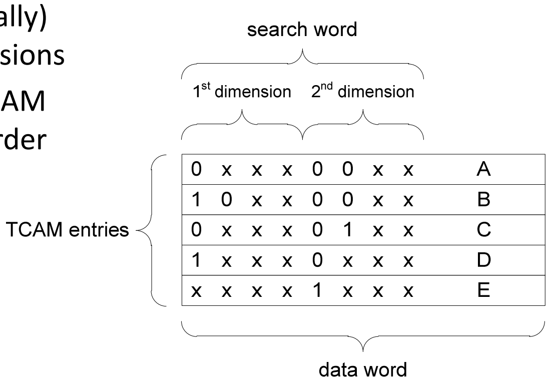
Hierarchical Intelligent Cuttings

- Heuristically divide space by cuttings
  - Goal is to have small set of rules in remaining area
  - Linear search within remaining rule set



TCAM

- Ternary content-addressable memory
  - Ideal hardware component for lookups
  - Search word (logically) divided into dimensions
  - Priority order in TCAM matches priority order of rules



0	x	x	x	0	0	x	x	A
1	0	x	x	0	0	x	x	B
0	x	x	x	0	1	x	x	C
1	x	x	x	0	x	x	x	D
x	x	x	x	1	x	x	x	E

Transport layer systems

- We can now perform flow classification or matching
  - Identify connections or flows
- Next lecture
  - Firewalls
  - Network address translation