Packet Classification

Masoud Sabaei

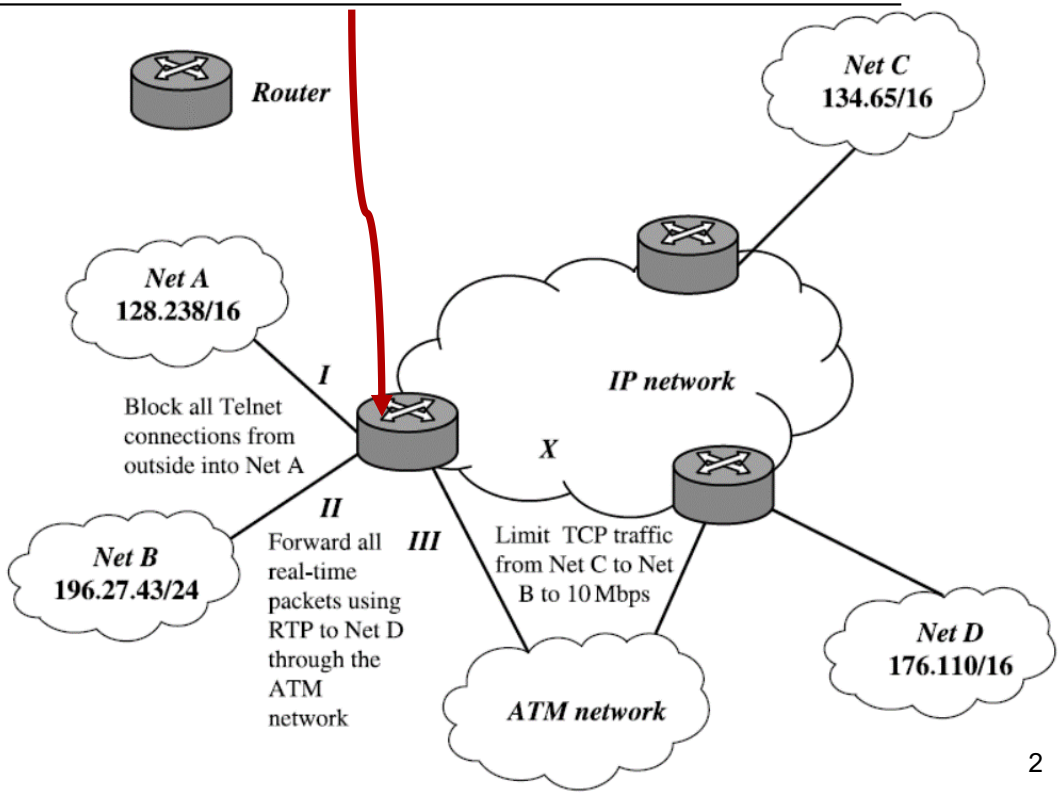
Associate professor

Department of Computer Engineering,
Amirkabir University of Technology

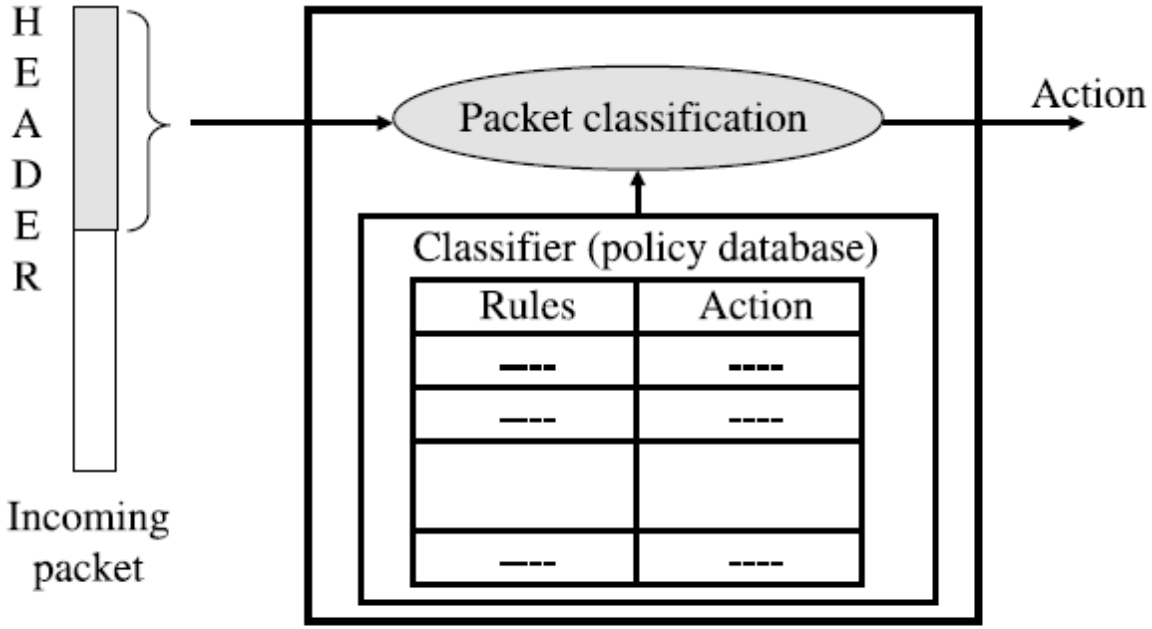
Classifier Example

	Network-Layer		Transport-Layer		Application-Layer	
Rule	Destination	Source	Protocol	Destination	Protocol	Action
R_1	128.238/16	*	TCP	= telnet	*	Deny
R_2	176.110/16	196.27.43/24	UDP	*	RTP	Send to port III
R_3	196.27.43/24	134.65/16	TCP	*	*	Drop traffic if rate > 10 Mbps
R_4	*	*	*	*	*	Permit

- **Packet Filtering.** Rule R1 blocks all telnet connections from outside into Net A, which may be a private research network.
- **Policy Routing.** Rule R2 enables the router to forward all real-time traffic using real-time transport protocol (RTP) in the application layer from Net B to Net D through the ATM network at the bottom of previous Figure.
- **Traffic Policing.** Rule R3 limits the total transmission control protocol (TCP) traffic rate from Net C to Net B up to 10 Mbps



Packet Classification



Matching the packet header to the rules in the classifier

Packet classification

1. A classifier C consists of N rules, Rj, $1 \leq j \leq N$, where Rj is composed of three entities:
   (a) A regular expression $Rj[i]$, $1 \leq i \leq d$, on each of the d header fields of a packet.
   (b) A number, *Pri(Rj)*, indicating the priority of the rule in the classifier.
   (c) An action, referred to as *Action(Rj)*.

2. An incoming packet P with the header considered as a d-tuple ($P1$, $P2$, . . . , Pd) is said to match Rj , if and only if, Pi matches $Rj[i]$, where $1 \leq i \leq d$.

3. Given an incoming packet P and thus the d-tuple, the d-dimensional packet classification problem is to find the rule Rm with the highest priority among all the rules Rj matching the d-tuple.

Example Classifier

Rule	F_1	F_2	F_3	F_4	Action
R_1	00*	110*	6	(10, 12)	Act_0
R_2	00*	11*	(4, 8)	15	Act_1
R_3	10*	1*	7	9	Act_2
R_4	0*	01*	10	(10, 12)	Act_1
R_5	0*	10*	(4, 8)	15	Act_0
R_6	0*	1*	10	(10, 12)	Act_3
R_7	*	00*	7	15	Act_1

Performance Metrics

- **Search Speed,**
- **Storage Requirement,**
- **Scalability in Classifier Size,**
- **Scalability in the Number of Header Fields,**
- **Update Time,**
- **Flexibility in Specification.**

Packet Classification Schemes

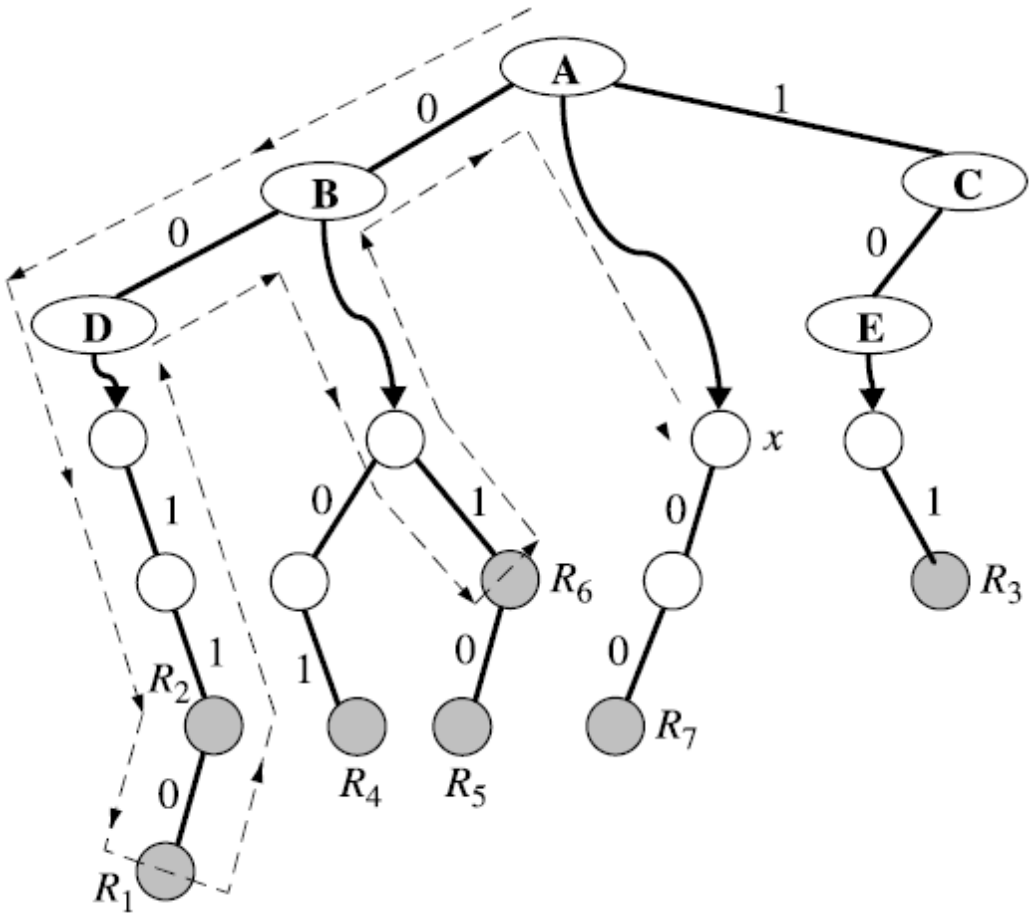
- Trie-based Classifications,
- Geometric Algorithms,
- Heuristic Algorithms,
- TCAM-based Algorithms.

Trie-based Classifications

- Hierarchical Tries
- Set-Pruning Trie
- Grid of Tries
- Extending Two-Dimensional Schemes
- Field-Level Trie Classification (FLTC)

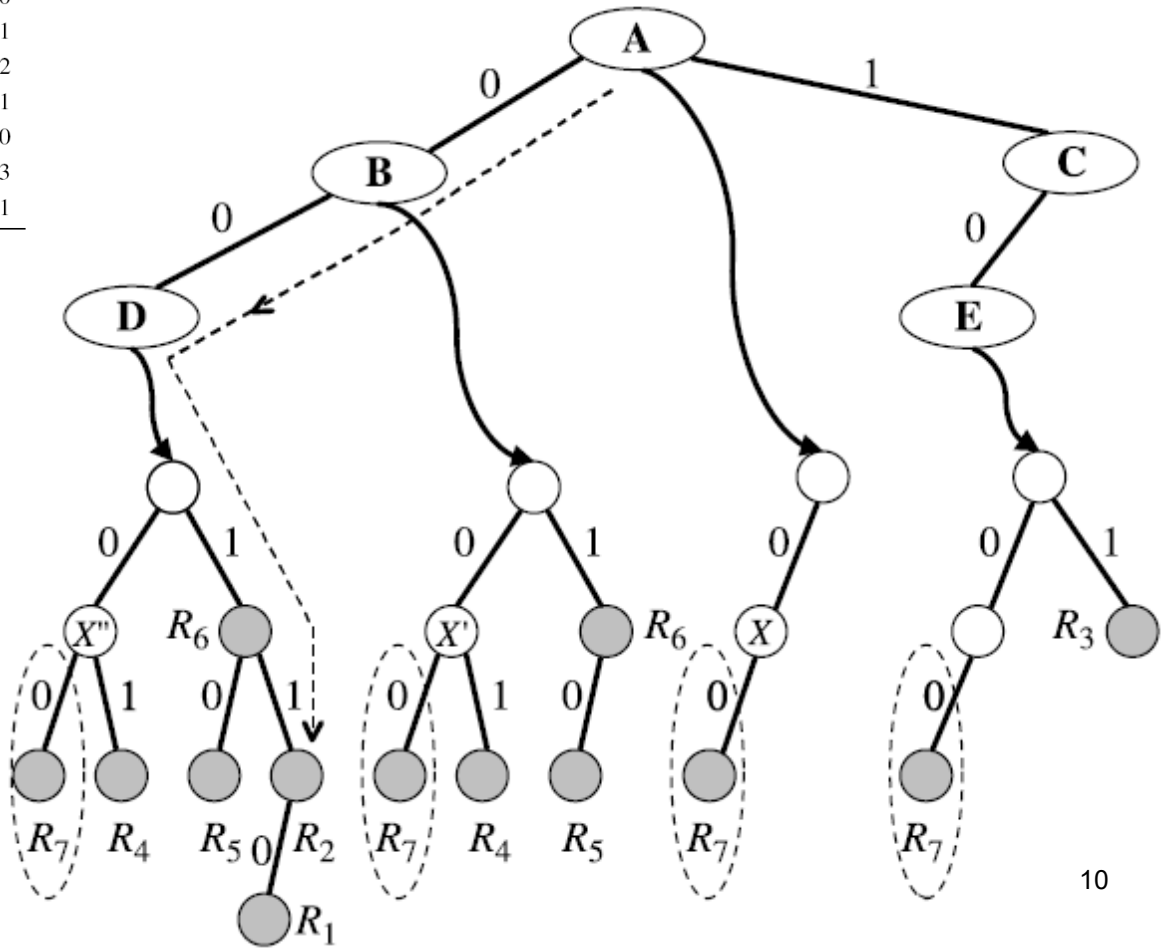
Hierarchical Tries

Rule	F_1	F_2	F_3	F_4	Action
R_1	00*	110*	6	(10, 12)	Act_0
R_2	00*	11*	(4, 8)	15	Act_1
R_3	10*	1*	7	9	Act_2
R_4	0*	01*	10	(10, 12)	Act_1
R_5	0*	10*	(4, 8)	15	Act_0
R_6	0*	1*	10	(10, 12)	Act_3
R_7	*	00*	7	15	Act_1



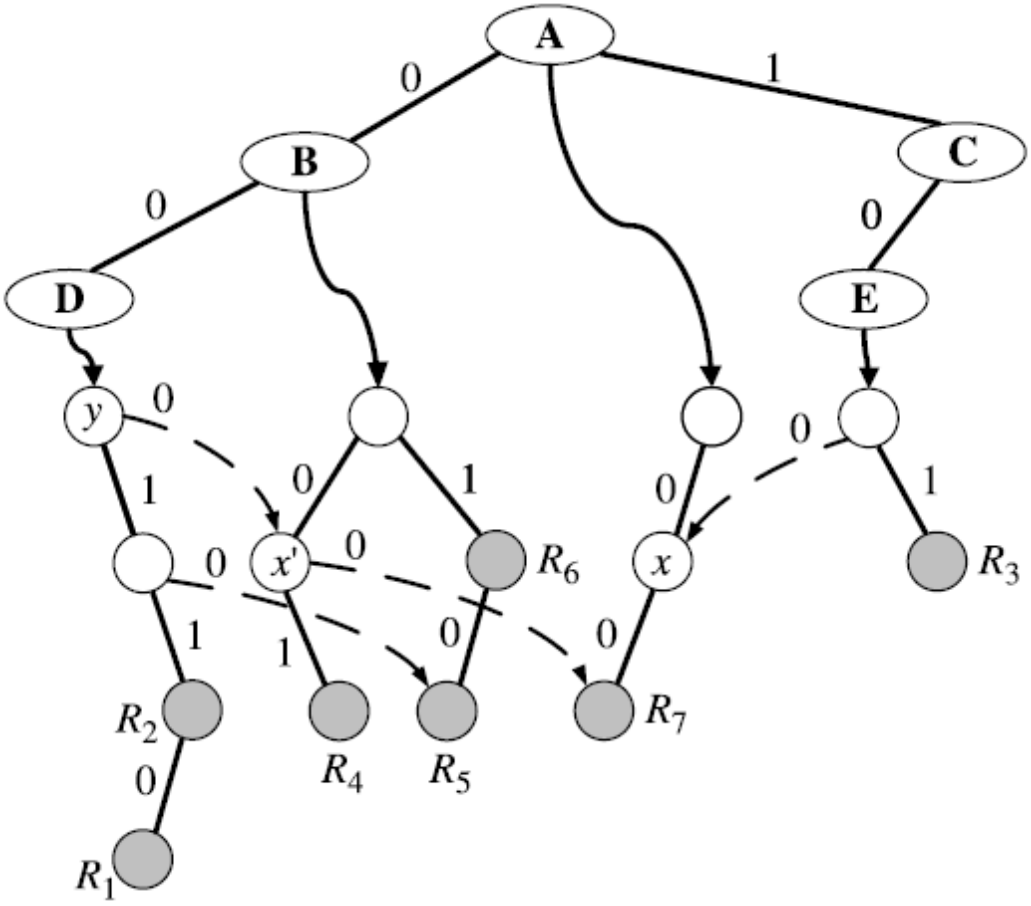
Set-Pruning Trie

Rule	F_1	F_2	F_3	F_4	Action
R_1	00*	110*	6	(10, 12)	Act_0
R_2	00*	11*	(4, 8)	15	Act_1
R_3	10*	1*	7	9	Act_2
R_4	0*	01*	10	(10, 12)	Act_1
R_5	0*	10*	(4, 8)	15	Act_0
R_6	0*	1*	10	(10, 12)	Act_3
R_7	*	00*	7	15	Act_1



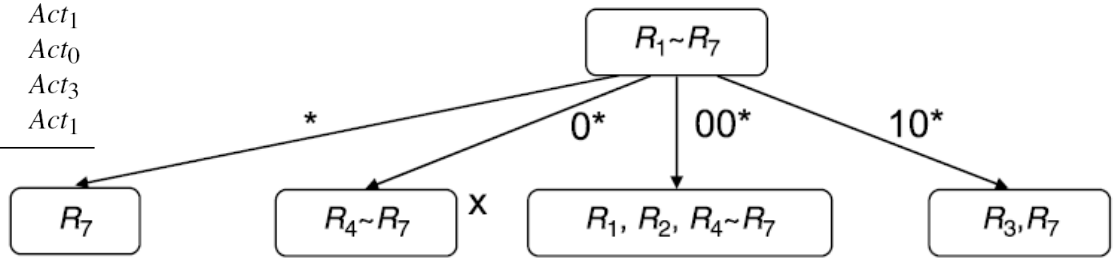
Grid of Tries

Rule	F_1	F_2	F_3	F_4	Action
R_1	00*	110*	6	(10, 12)	Act_0
R_2	00*	11*	(4, 8)	15	Act_1
R_3	10*	1*	7	9	Act_2
R_4	0*	01*	10	(10, 12)	Act_1
R_5	0*	10*	(4, 8)	15	Act_0
R_6	0*	1*	10	(10, 12)	Act_3
R_7	*	00*	7	15	Act_1

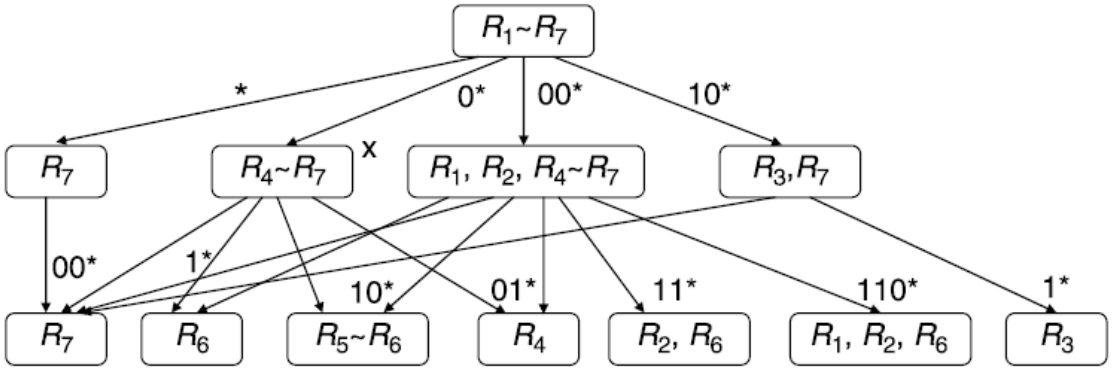


Field-Level Trie Classification (FLTC)

Rule	F_1	F_2	F_3	F_4	Action
R_1	00*	110*	6	(10, 12)	Act_0
R_2	00*	11*	(4, 8)	15	Act_1
R_3	10*	1*	7	9	Act_2
R_4	0*	01*	10	(10, 12)	Act_1
R_5	0*	10*	(4, 8)	15	Act_0
R_6	0*	1*	10	(10, 12)	Act_3
R_7	*	00*	7	15	Act_1



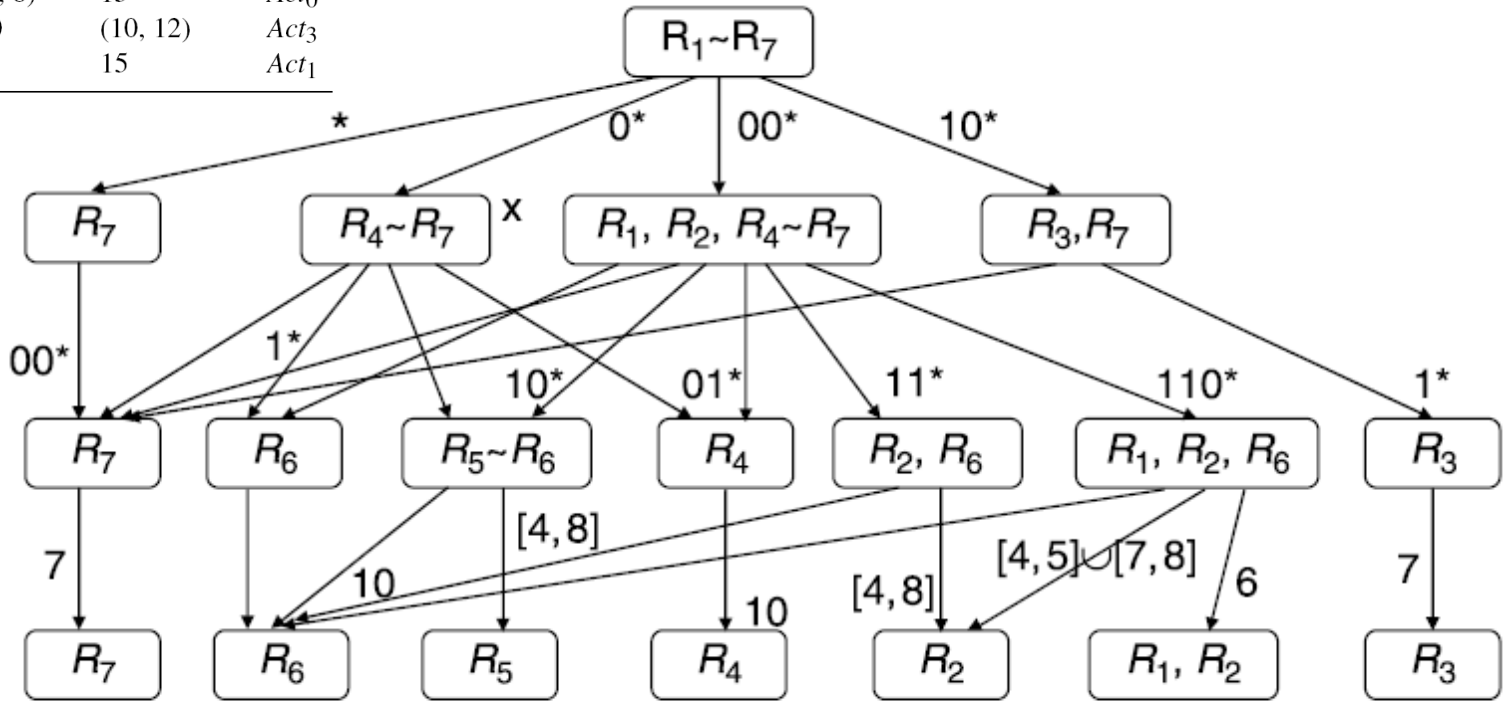
(a)



(b)

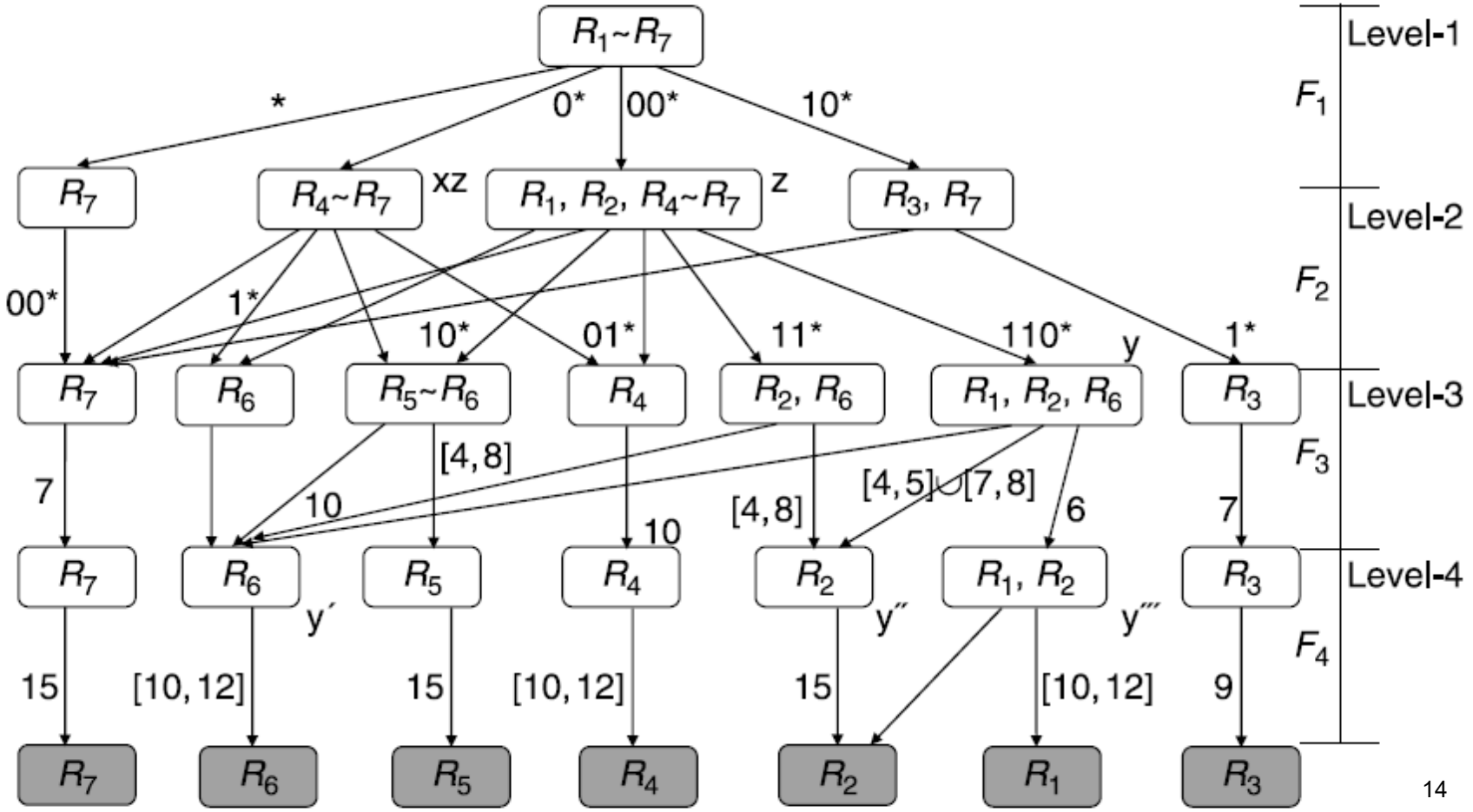
Field-Level Trie Classification (FLTC)

Rule	F_1	F_2	F_3	F_4	Action
R_1	00*	110*	6	(10, 12)	Act_0
R_2	00*	11*	(4, 8)	15	Act_1
R_3	10*	1*	7	9	Act_2
R_4	0*	01*	10	(10, 12)	Act_1
R_5	0*	10*	(4, 8)	15	Act_0
R_6	0*	1*	10	(10, 12)	Act_3
R_7	*	00*	7	15	Act_1



(c)

Field-Level Trie Classification (FLTC)



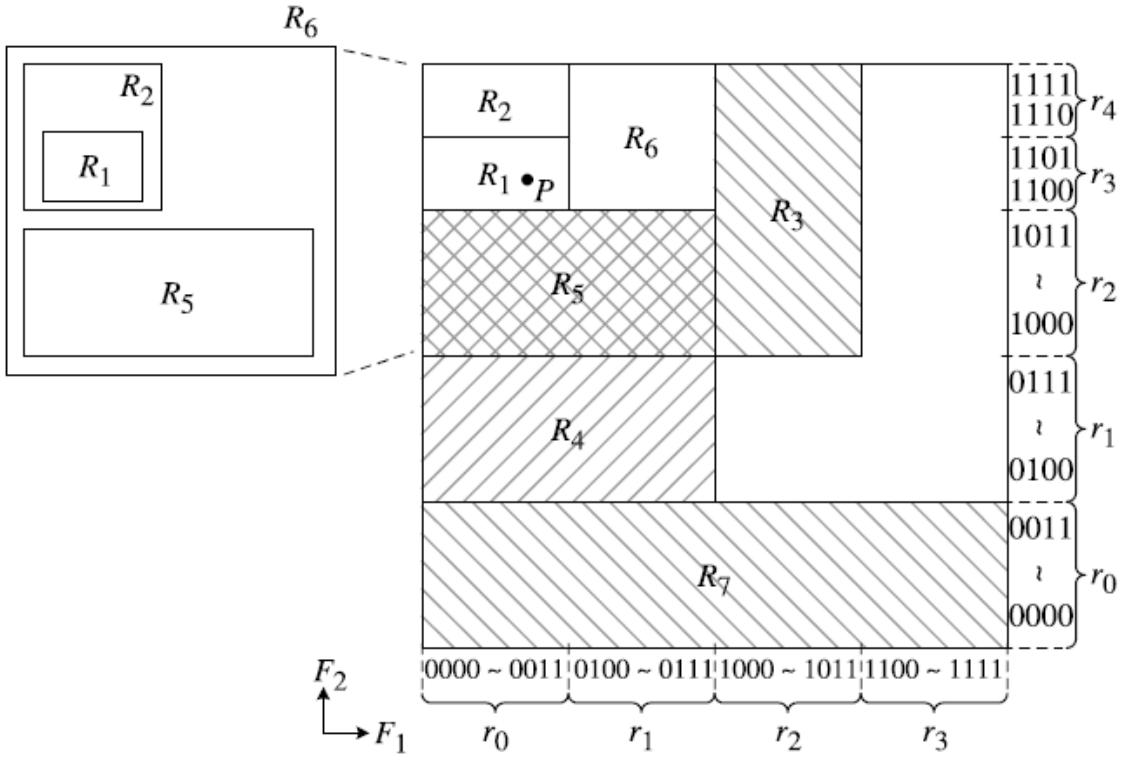
(*d*)

GEOMETRIC ALGORITHMS

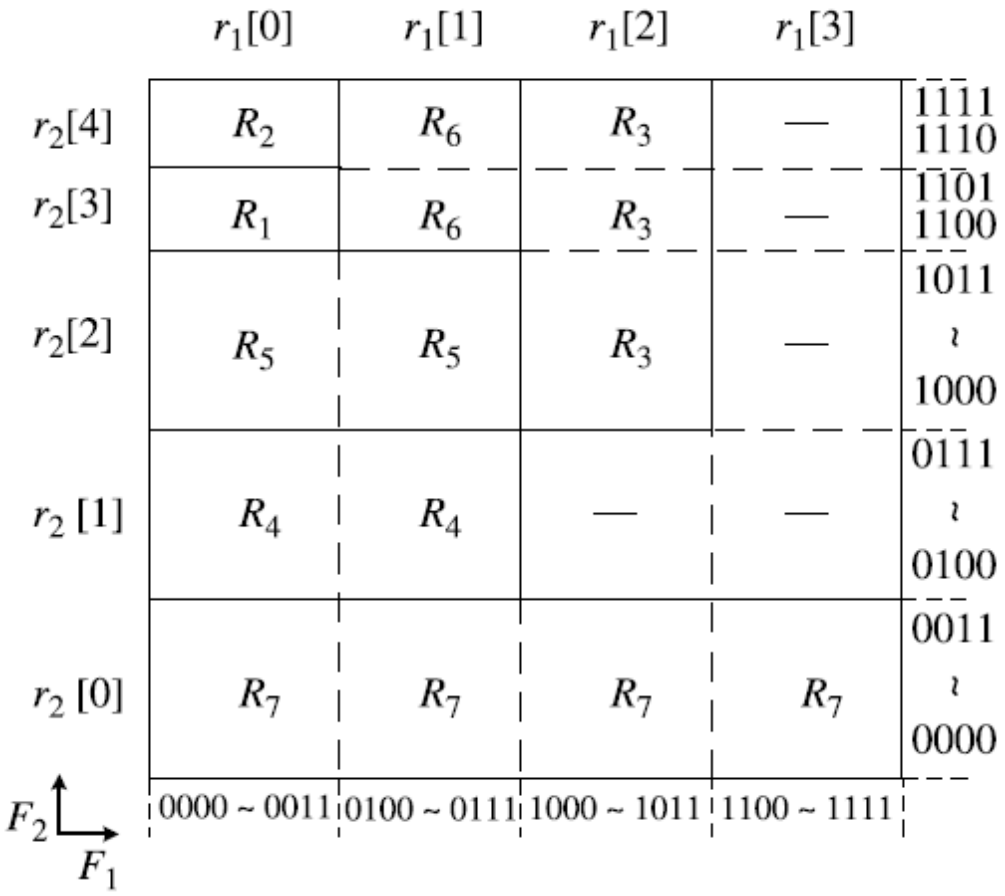
- Cross-Producting Scheme
- Bitmap-Intersection
- Parallel Packet Classification *(*P^2C*)*
- Area-Based Quadtree
- Hierarchical Intelligent Cuttings
- HyperCuts

GEOMETRIC ALGORITHMS

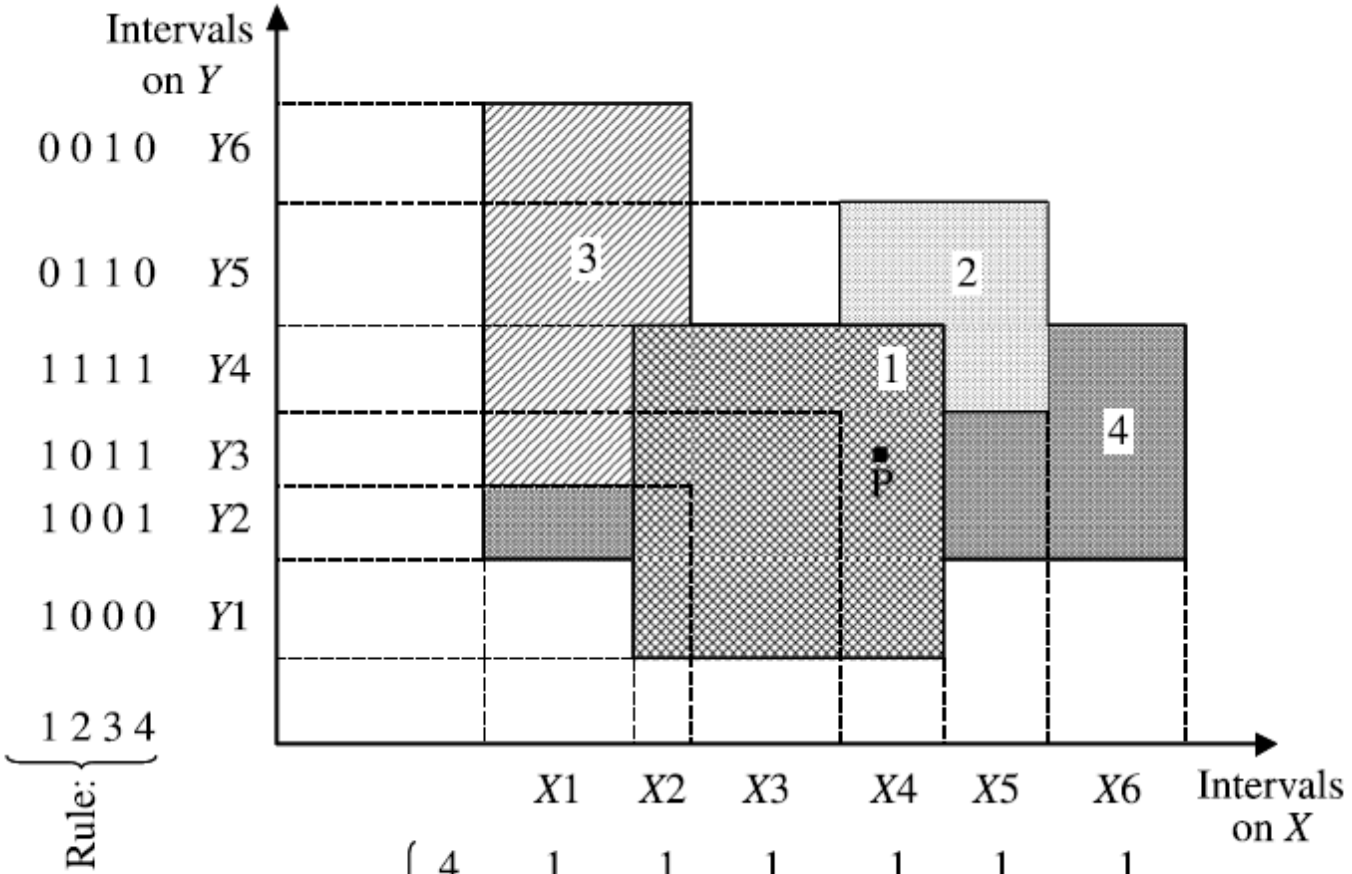
Rule	F_1	F_2	F_3	F_4	Action
R_1	00*	110*	6	(10, 12)	Act_0
R_2	00*	11*	(4, 8)	15	Act_1
R_3	10*	1*	7	9	Act_2
R_4	0*	01*	10	(10, 12)	Act_1
R_5	0*	10*	(4, 8)	15	Act_0
R_6	0*	1*	10	(10, 12)	Act_3
R_7	*	00*	7	15	Act_1



Cross-Producting Scheme

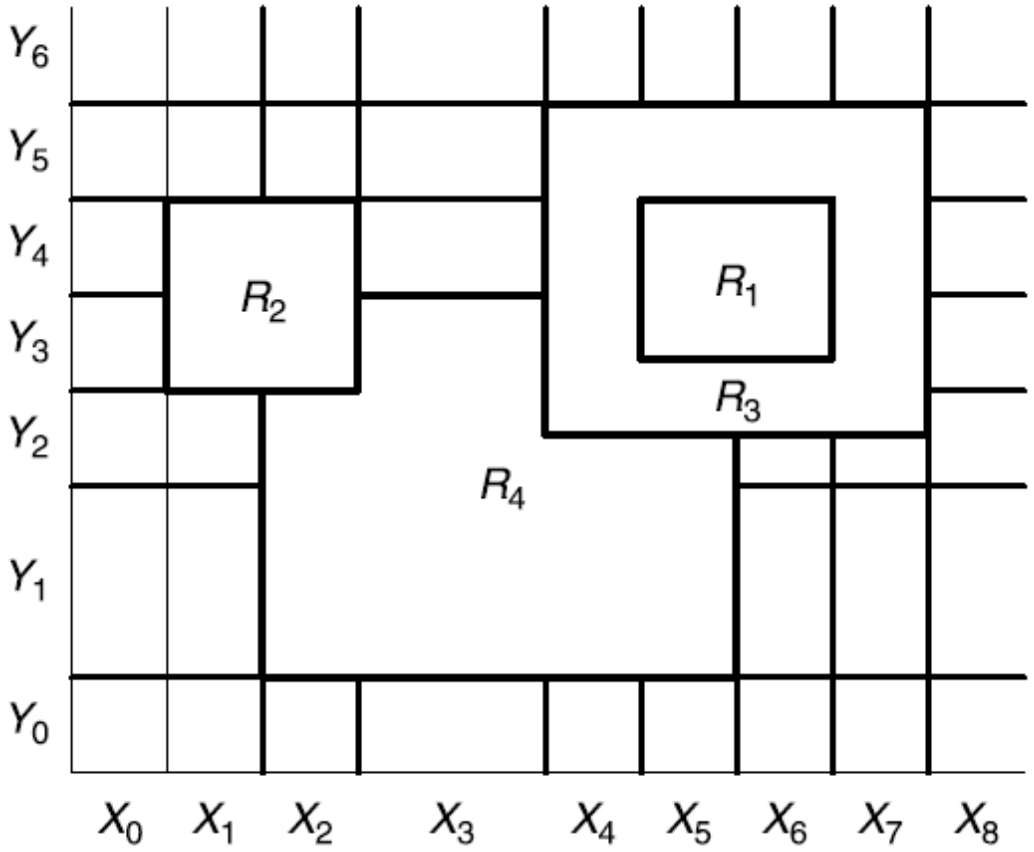


Bitmap-Intersection

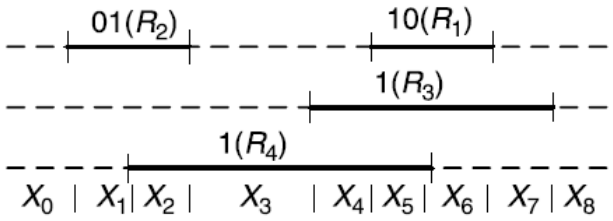


Rule:	X1	X2	X3	X4	X5	X6
4	1	1	1	1	1	1
3	1	1	0	0	0	0
2	0	0	0	1	1	0
1	0	1	1	1	0	0

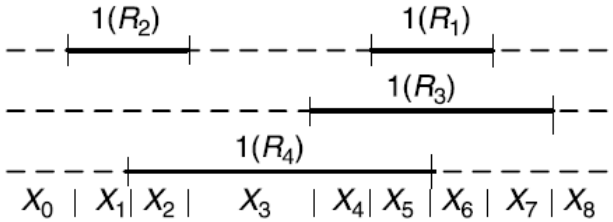
Parallel Packet Classification (P^2C)



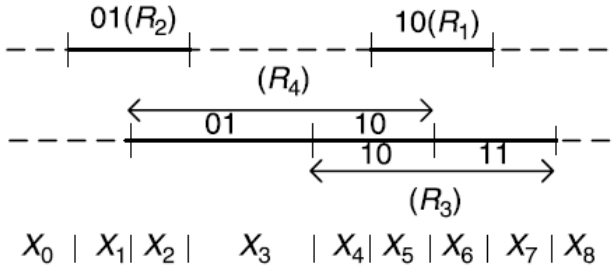
Parallel Packet Classification (P^2C)



(*a*) Style I



(*b*) Style II



(c) Style III

Parallel Packet Classification (P^2C)

Intermediate Result Vectors for the Range Hierarchies

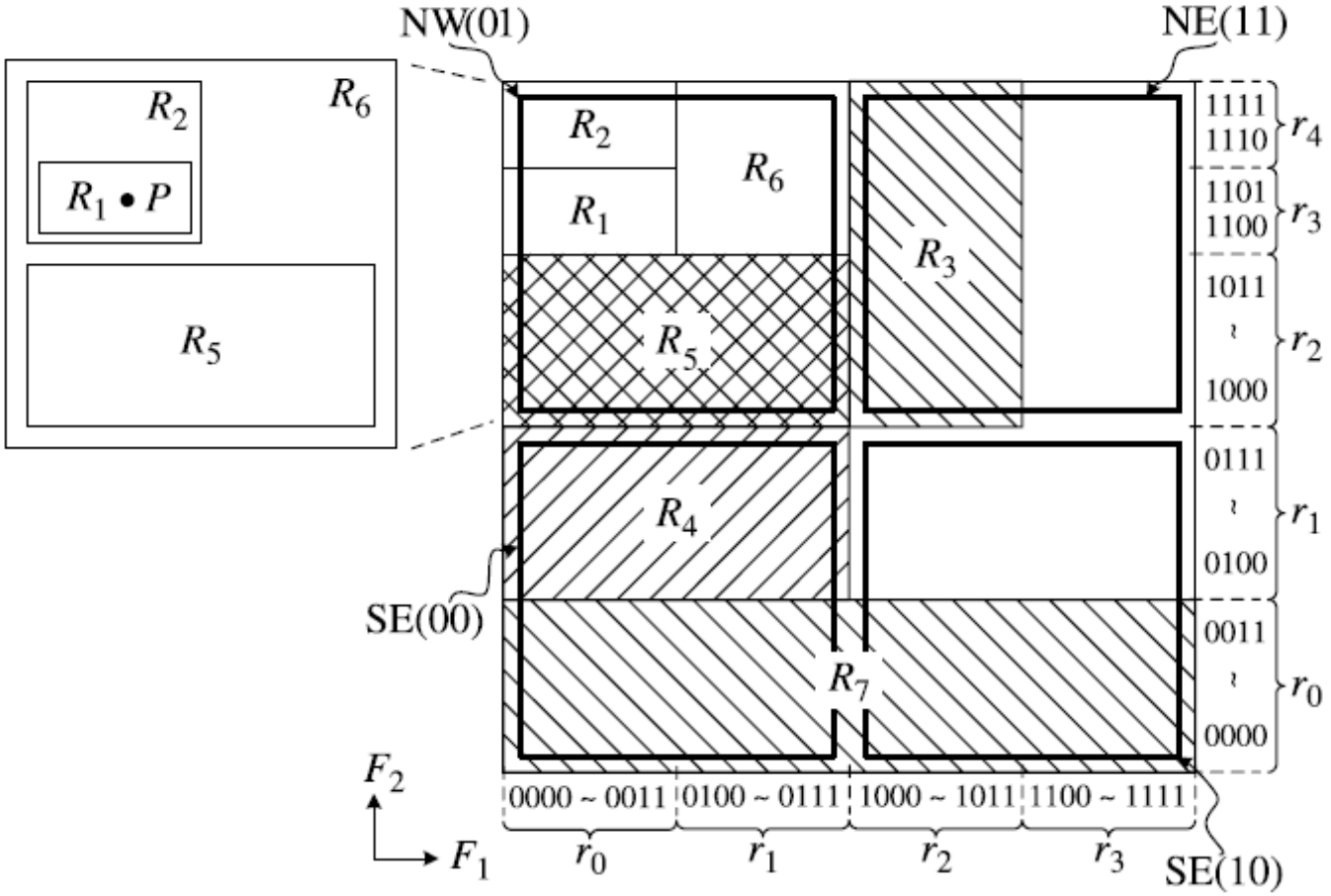
	Ranges						
	X_1	X_2	X_3	X_4	X_5	X_6	X_7
Style I	0100	0101	0001	0011	1011	1010	0010
Style II	100	101	001	011	111	110	010
Style III	0100	0101	0001	0010	1010	1011	0011

Parallel Packet Classification (P^2C)

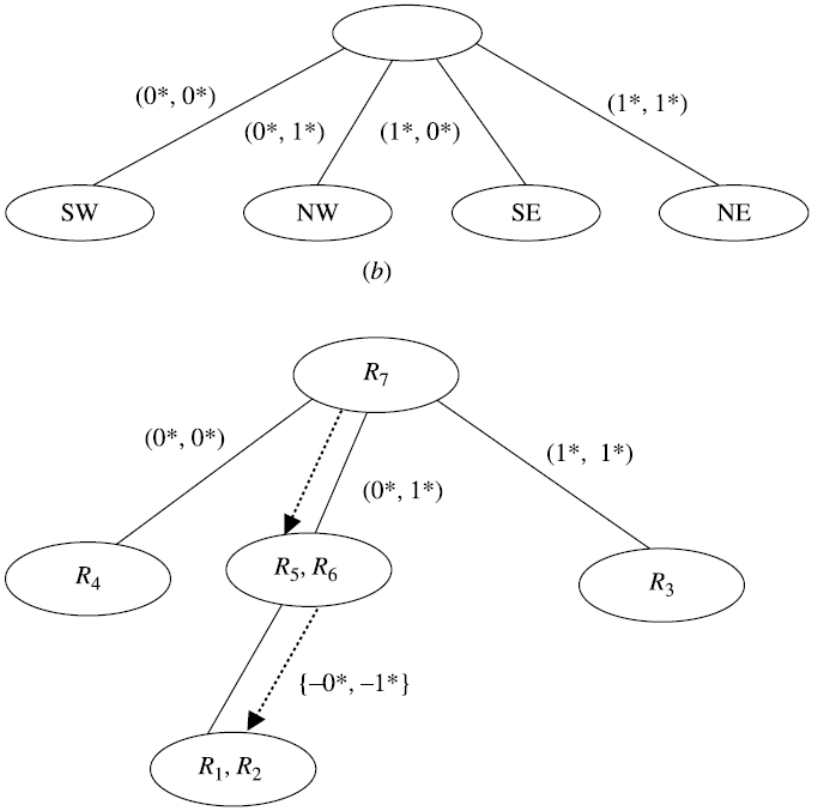
Ternary-Match Conditions for the Range Hierarchies

	R_4	R_3	R_2	R_1
Style I	xxx1	xx1x	01xx	10xx
Style II	xx1	x1x	10x	11x
Style III	xx01,xx10	xx10,xx11	01xx	10xx

Area-Based Quadtree



Area-Based Quadtree



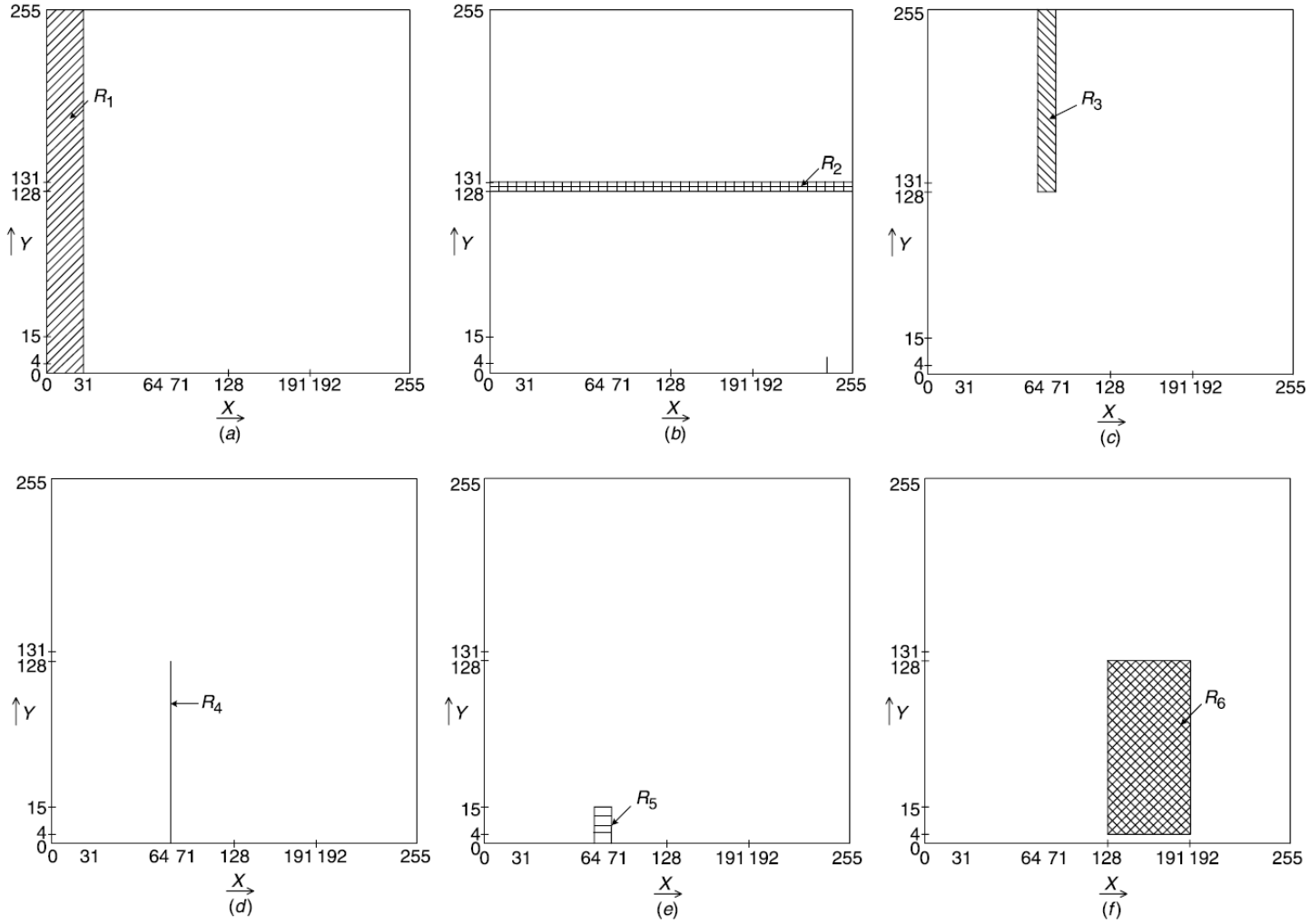
Hierarchical Intelligent Cuttings

Rule	X Range	Y Range
R_1	0–31	0–255
R_2	0–255	128–131
R_3	64–71	128–255
R_4	67–67	0–127
R_5	64–71	0–15
R_6	128–191	4–131
R_7	192–192	0–255

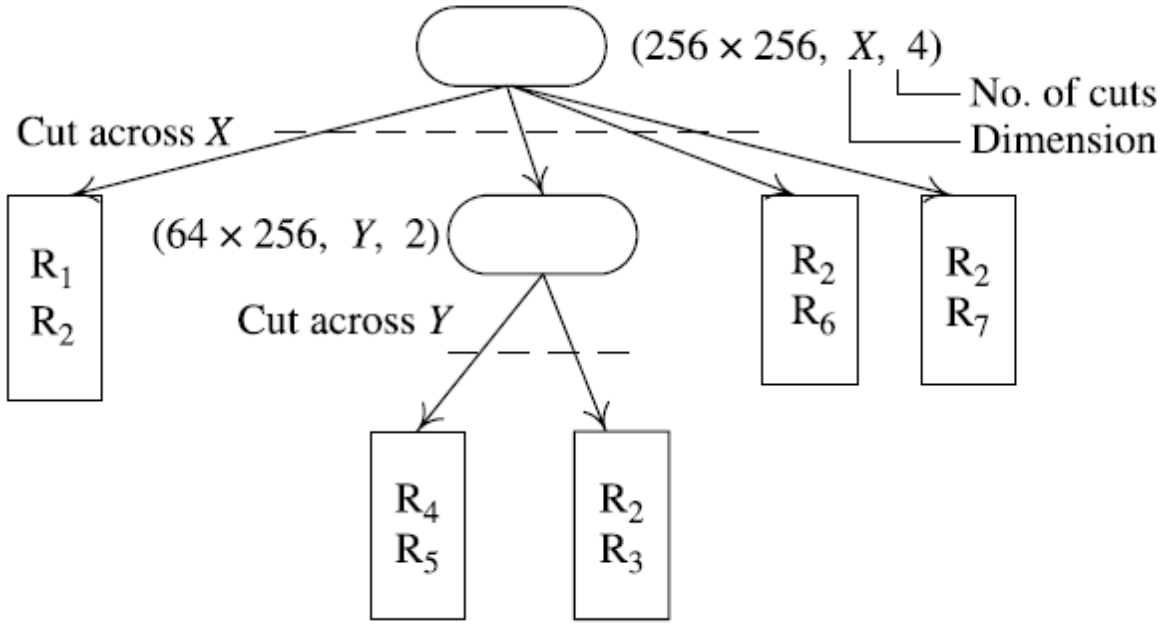
Rule Set Example with Two Dimensions in Ranges

Hierarchical Intelligent Cuttings

Geometrical representation of the seven rules in the table (*a*) *R*1; (*b*) *R*2; (*c*) *R*3; (*d*) *R*4; (*e*) *R*5; (*f*) *R*6.



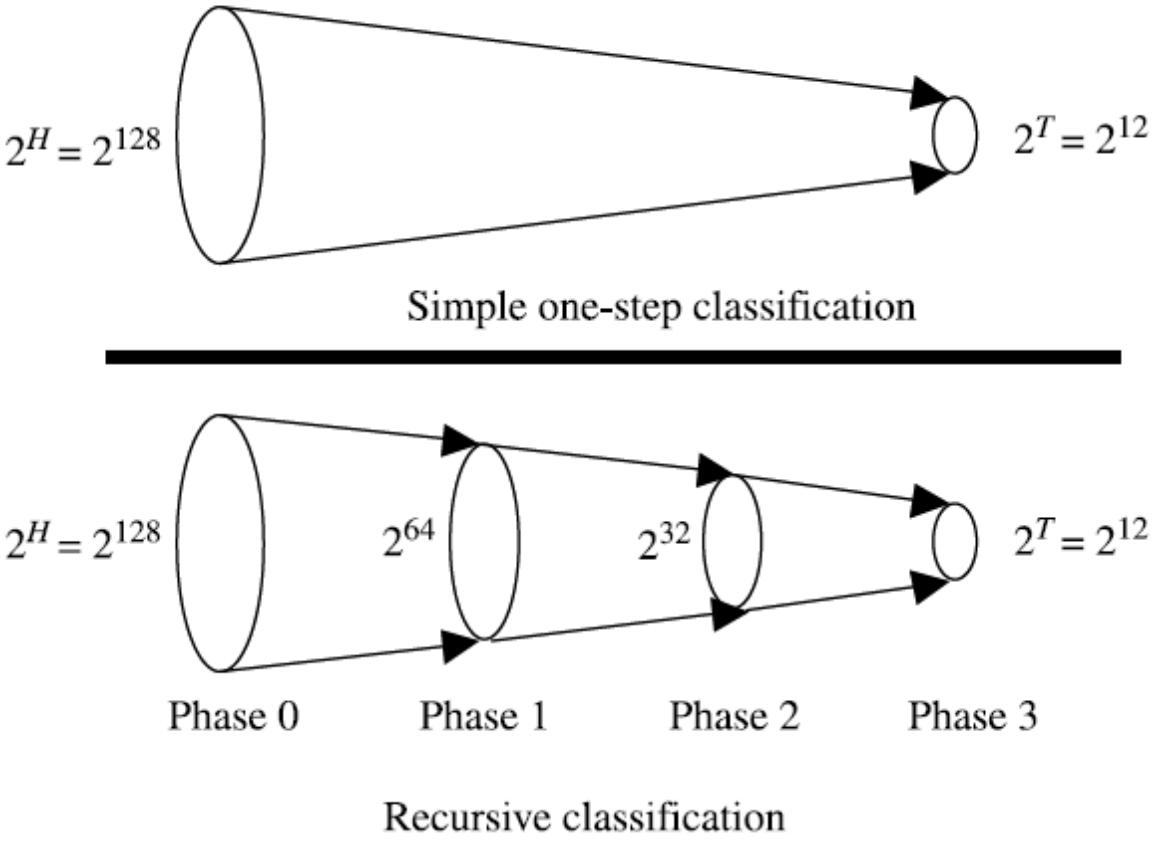
Hierarchical Intelligent Cuttings



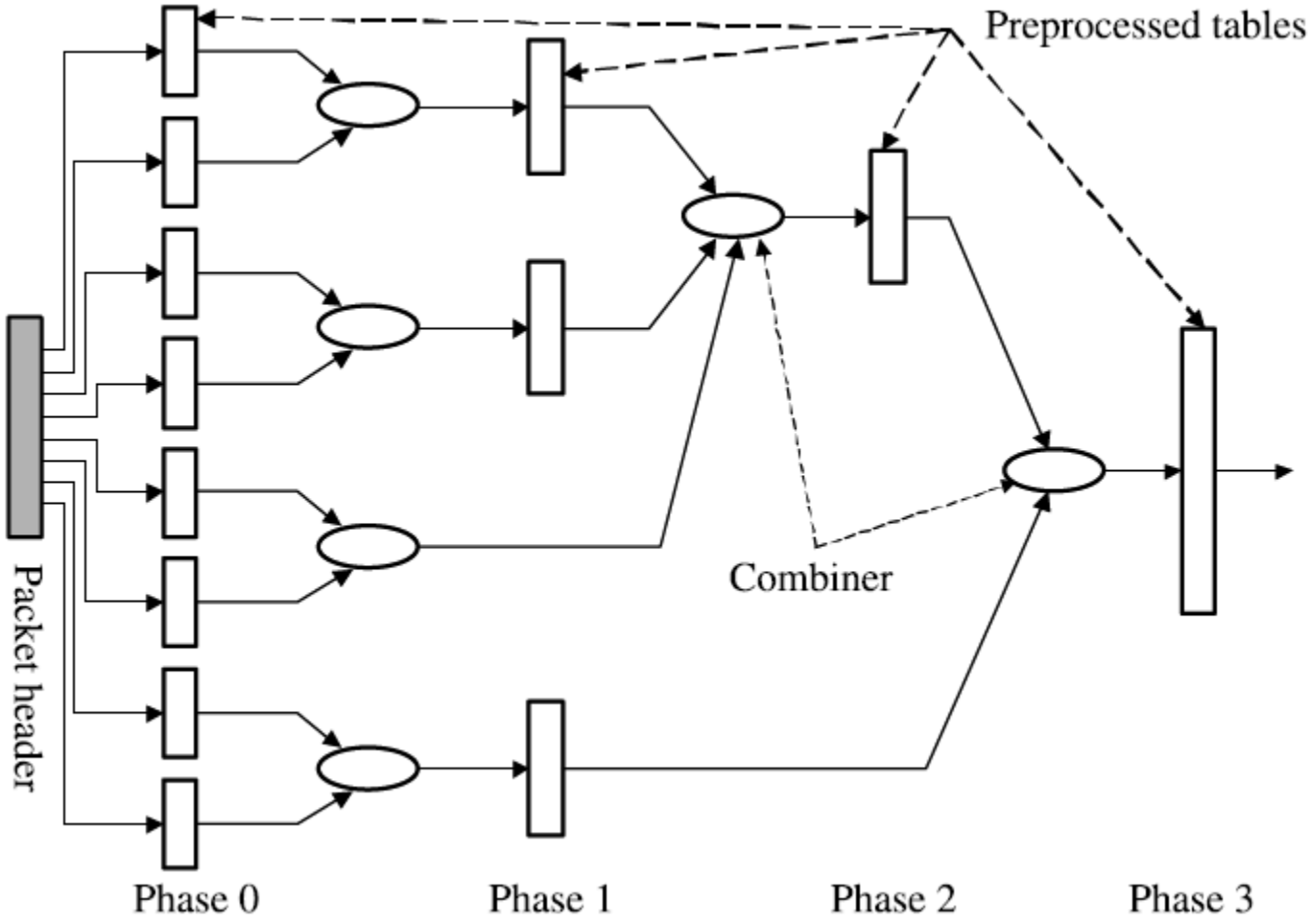
Heuristic Algorithms

- Recursive Flow Classification,
- Tuple Space Search.

Recursive Flow Classification

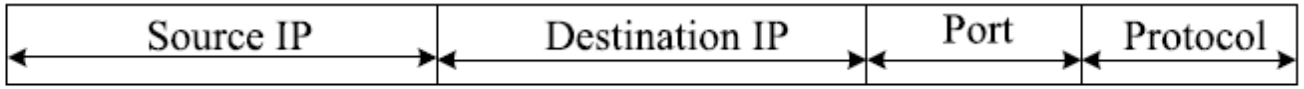


Recursive Flow Classification



Recursive Flow Classification Rule Set Example

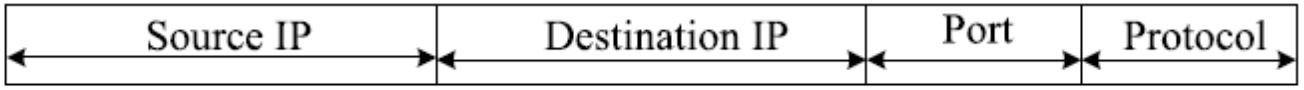
Destination IP (addr/mask)	Source IP (addr/mask)	Port Number	Protocol
152.163.190.69/255.255.255.255	152.163.80.11/0.0.0.0	*	*
152.168.3.0/255.255.255.0	152.163.200.157/0.0.0.0	eq www	UDP
152.168.3.0/255.255.255.0	152.163.200.157/0.0.0.0	range 20–21	UDP
152.168.3.0/255.255.255.0	152.163.200.157/0.0.0.0	eq www	TCP
152.163.198.4/255.255.255.255	152.163.160.0/255.255.252.0	gt 1023	TCP
152.163.198.4/255.255.255.255	152.163.36.0/0.0.0.255	gt 1023	TCP



Width (bits)	32	32	16	8
Chunks #	0	1	2	3

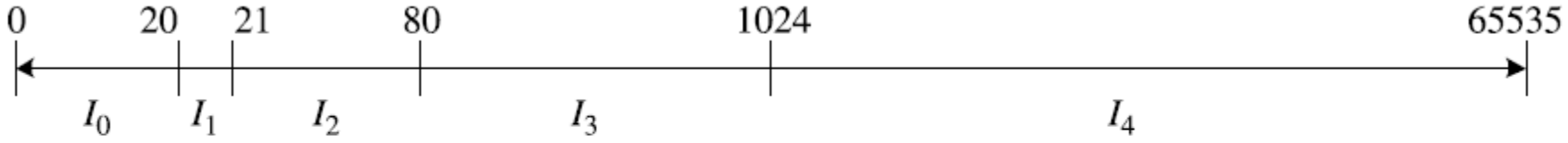
Recursive Flow Classification

Chopping of packet header into chunks for rule set *C* in the first phase of RFC.



Width (bits)	32	32	16	8
Chunks #	0	1	2	3

Example of computing the four equivalence classes *E*0 . . . *E*3 for chunk 2 (corresponding to the 16-bit transport-layer destination port number) in the rule set of the classifier.



$E_0 = \{20, 21\}$

$E_1 = 80$

$E_2 = \{1024–65535\}$

$E_3 = \{0\text{-}19, 22–79, 81–1023\}$

Rule storing organization for RFC for the rule set in the classifier table.

Destination IP (addr/mask)	Chunk Number	eqID (only 2 bits required)
152.163.190.69/255.255.255.255	0	00
152.168.3.0/255.255.255.0	1	01
152.163.198.4/255.255.255.255	2	10

(*a*)

Source IP (addr/mask)	Chunk Number	eqID (only 2 bits required)
152.163.80.11/0.0.0.0	0	00
152.163.200.157/0.0.0.0	1	01
152.163.160.0/255.255.252.0	2	10
152.163.36.0/0.0.0.255	3	11

(*b*)

Port Number	Chunk Number	eqID (only 2 bits required)
Range 20-21	0	00
eq www	1	01
gt 1023	2	10
0-19,22-79,81-1023	3	11

(*c*)

Protocol	Chunk Number	eqID (only 2 bits required)
tcp	0	00
udp	1	01
all remaining protocols	2	10

(*d*)

Port Number and Protocol	Chunk Number	eqID (only 3 bits required)
eq www & udp	0	000
Range 20-21 & udp	1	001
eq www & tcp	2	010
gt 1023 & tcp	3	011
all remaining crossproducts	4	100

(*e*)

(a) Destination IP field made into chunks and epIDs.

(b) Source IP field made into chunks and eqIDs.

(c) Port number field made into chunks and eqIDs.

(d) Protocol field made into chunks and eqIDs.

(e) Port number and protocol fields combined and made into chunks and eqIDs.