Scaling IoT MUD Enforcement using Programmable Data Planes

Harish S A[1], Suvrima Datta[2], Hemanth Kothapalli[1], Praveen Tammana[1], Achmad Basuki[3], Kotaro Kataoka[1], Selvakumar Manickam[4], Venkanna U[2], Yung-Wey Chong[4]

[1] IIT Hyderabad
India

[2] IIIT Naya Raipur
India

[3] Universiti Sains
Malaysia

[4] Universitas Brawijaya
Indonesia

9th August 2022

IEEE/IFIP Network Operations and Management Symposium 2023

Internet of Things (IoT): Risks

Perpetrate attacks on critical Infrastructure

Malicious Botnets

DDoS attacks

Data exfiltration

Incapacitation

Mirai Botnet

BrickerBot

14 Billion IoT devices[1] 2022

27 Billion IoT devices[1] 2025

Key drivers

High competition: Less incentive for security

Patching vulnerabilities is difficult

Real time network based IoT security solution

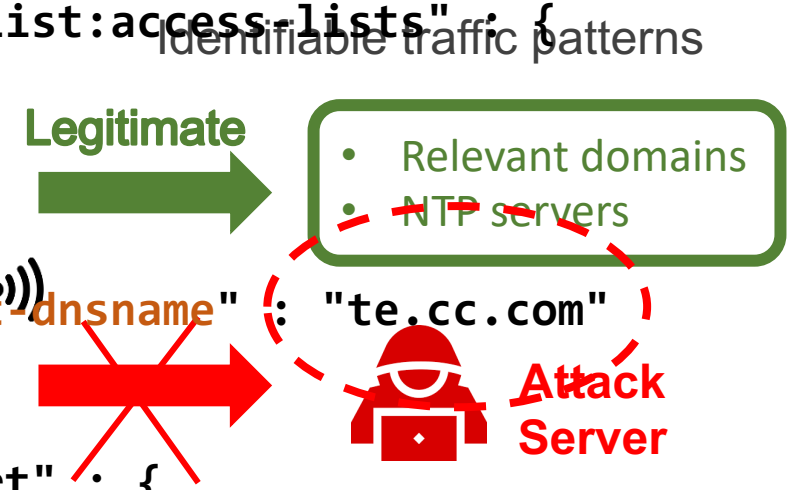
[1]IoT Analytics, Market insights for Internet of Things: State of IoT 2022

Manufacturer Usage Description (MUD)

MUD abstracts communication pattern of an IoT device to a MUD profile[1]

Example MUD profile

```
"ietf-access-control-list:access-lists" : {
   ...
  "matches" : {
    "ipv4" : {
      "protocol" : 6
      "ietf-acldns:dst-dnsname" : "te.cc.com"
    },
    "tcp" : {
      "destination-port" : {
        "operator" : "eq",
        "port" : 8777
      },
      "ietf-mud:direction-initiated" : "from-device"
    ...
```



Abstracted a MUD pro

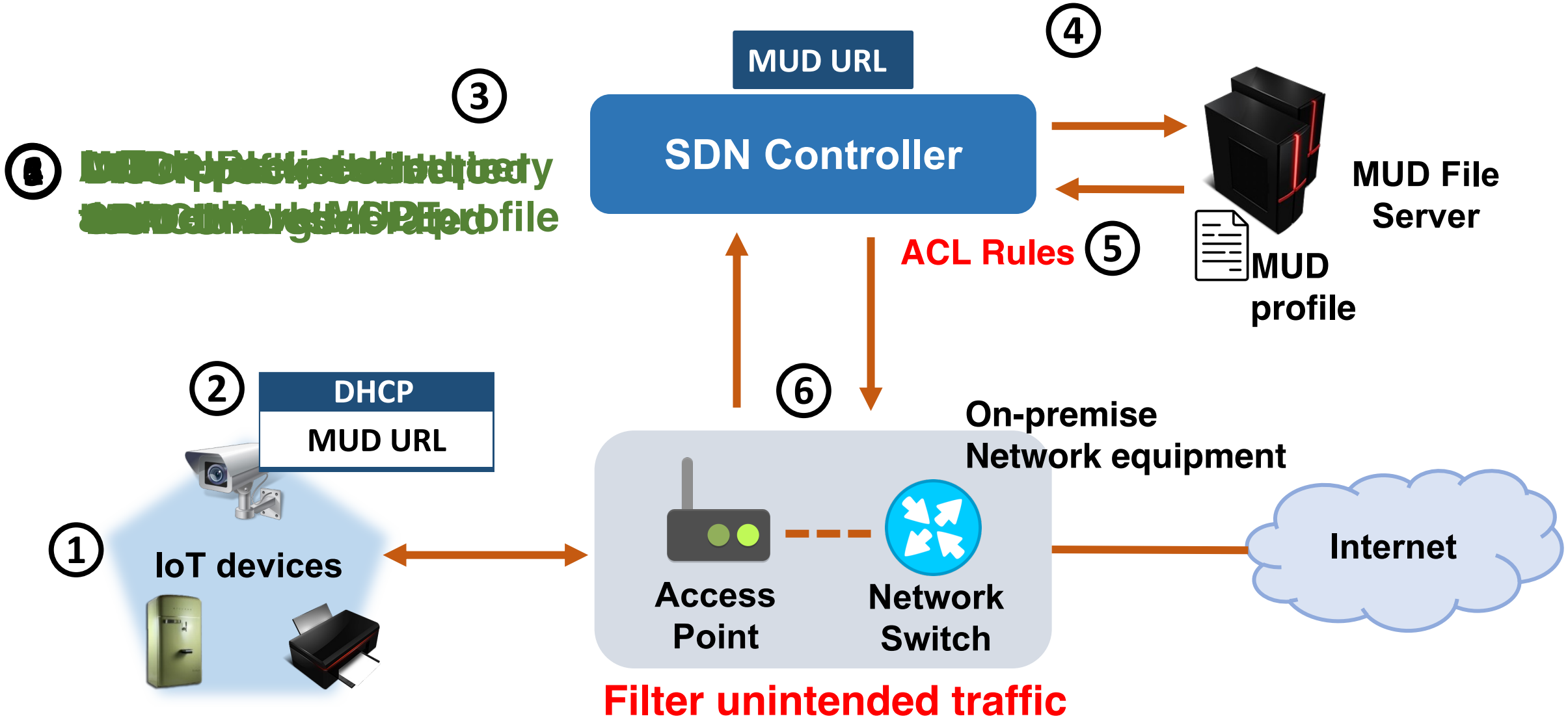
IoT Device ma

ACL rule

Type Eth	0x0800
Protocol	6
Src Port	*
Dst Port	8777
Src IP	*
Dst IP	te.cc.com 44.45.66.44

[1]National Cybersecurity Center of Excellence (NCCoE) : MUD related Resources

MUD enforcement



*Combining MUD policies for IDS [IoT S&P'18] | Volumetric attack detection using MUD [SOSR'19] | SoftMUD [NIST, ICN]

Existing Approaches

Enforced inside the local network

Clear as MUD [IoT S&P'18] | **Combining MUD policies for IDS** [IoT S&P'18] | **Volumetric attack detection using MUD** [SOSR'19] | **SoftMUD** [NIST, ICN]

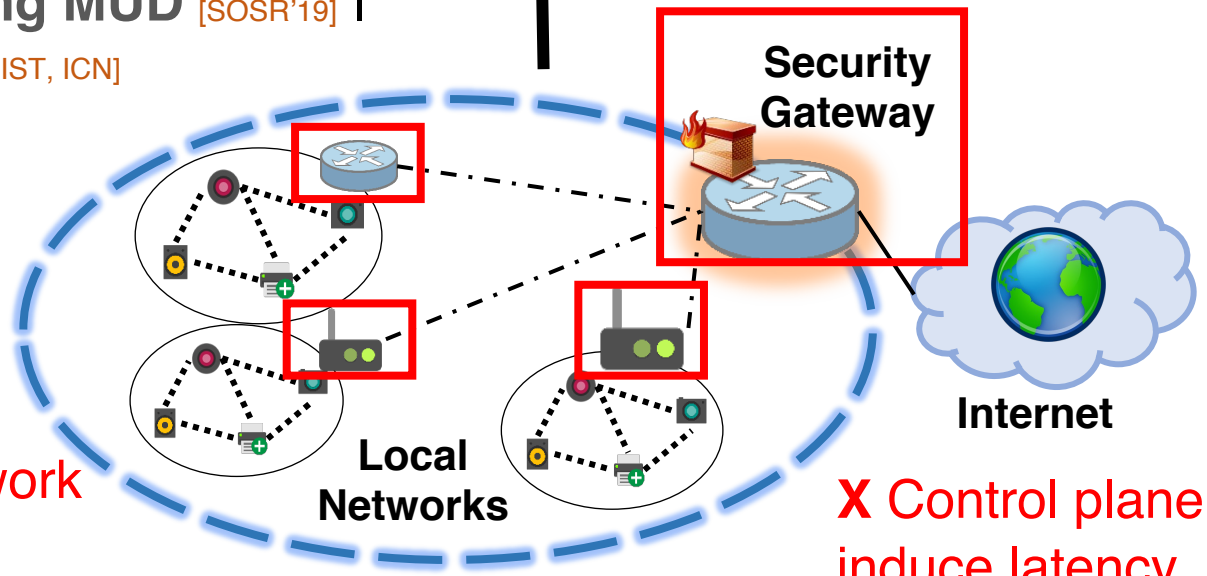
Enforce MUD rules in local access points and switches

X Heterogeneity of network devices

X Mobility and rule migration

Enforced at the network edge

IoT security at ISP using NFV [NOMS'20]



Enforce MUD rules at the network edge gateway infrastructure

X Control plane software modules induce latency

X High CP-DP bandwidth overhead

Not suitable for large IoT networks

In this work..

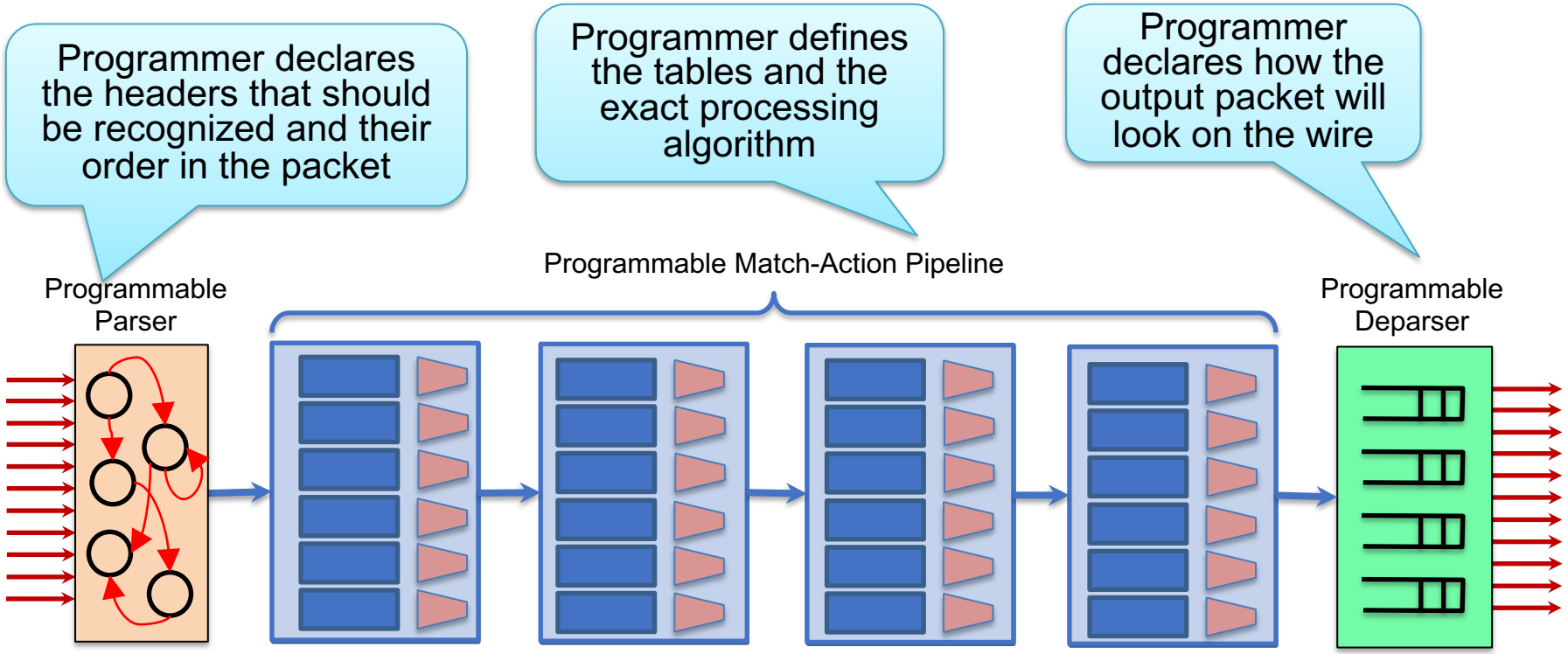
Question

How to scale MUD enforcement to support large IoT networks?

Key contribution

We design a **data plane primitive** to scale MUD enforcement using P4-based programmable switch

Protocol Independent Switch Architecture (PISA)[1]



Line speed packet-processing guarantees

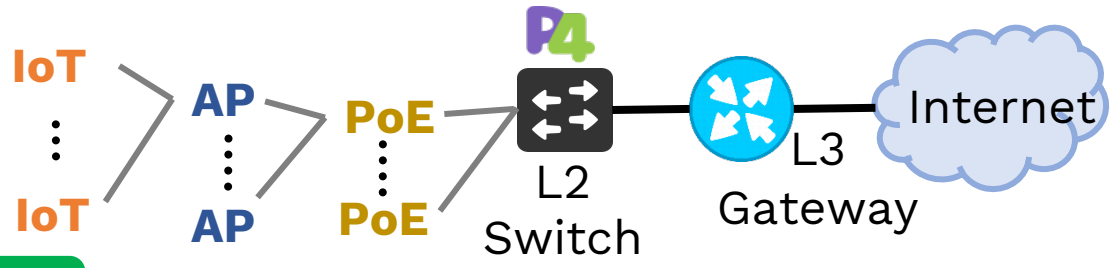
Sufficient programmability

Variety of Targets[2]

[1]P4.org
[2]OpenNetworking.org: link

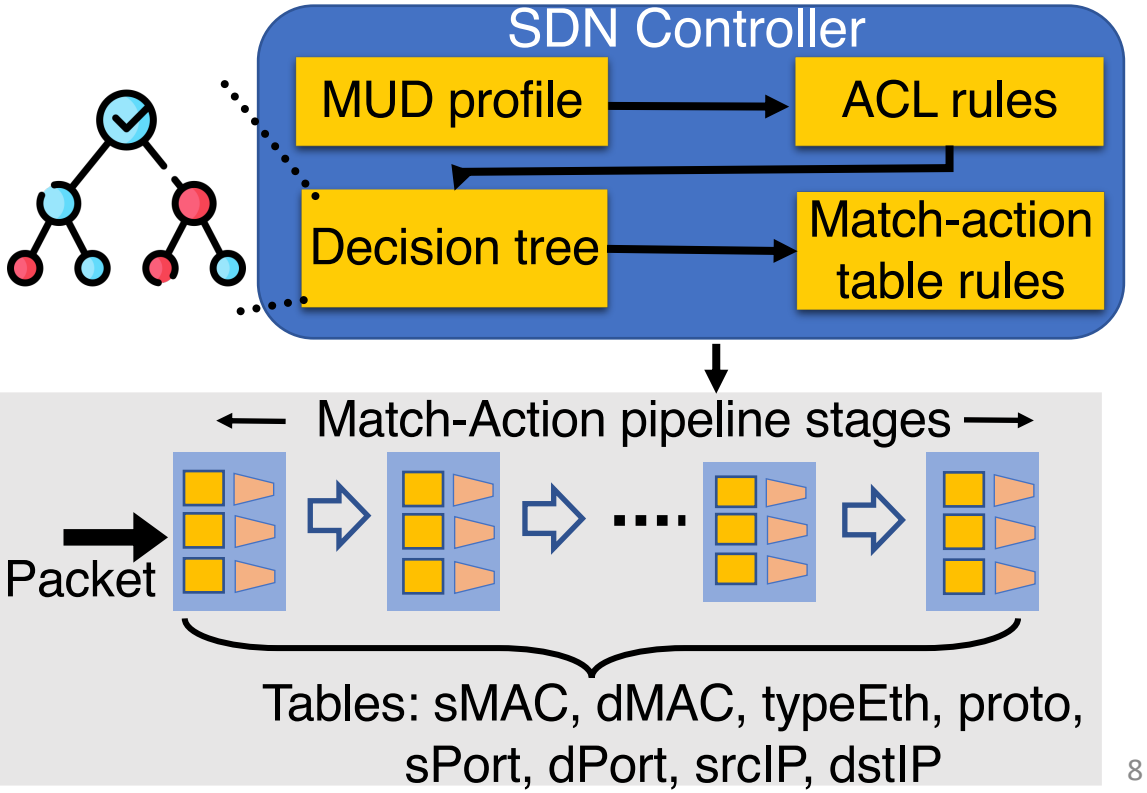
Our approach in a nutshell

- Enforce MUD higher up in the network hierarchy.

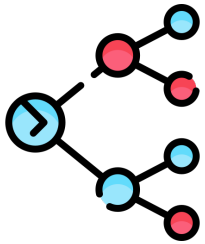


Handles management and mobility overhead

- We represent MUD rules as decision trees
- Translated to the switch match action pipeline



Addresses scaling of MUD ACL rules to thousands of IoT devicesd



Scaling on the switch is challenging

MUD ACL rules are stored in **switch memory**

TCAM

Costly | Power Hungry | Limited

Matches supported

- Exact
- Longest prefix
- Ternary

TypeEth	0x0800
Protocol	6
Src Port	*
Dst Port	8777
Src IP	*
Dst IP	44.45.66.44

SRAM

Relatively abundant

~~MUD ACL~~

Matches supported

- Exact

Problem: MUD Rules contain ternary matches

Key implementation question

Can we design an efficient SRAM-based classification algorithm that runs entirely in the P4 data plane so that we can scale MUD enforcement to thousands of IoT devices?

Key Ideas

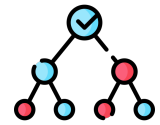
1. Partition and place MUD rules in SRAM

2. Decision tree representation of MUD rules

Observations

Observation 1: MUD based ACL rules contain repetitions in header values (e.g., MAC address of the same IoT)

→ Decision tree-based approach to avoid re-encoding the repetitions



Observation 2: MUD ACL rules contain only exact or ternary matches (*).

→ We emulate ternary matching (*) using SRAM based operations

Rule No.	sMAC	protocol	sPort	dPort	srcIP	dstIP
1	0D:E8:49:46:8E:4D	6	*	8777	*	te.cc.com
2	0D:E8:49:46:8E:4D	6	*	80	*	www.e.org
3	0D:E8:49:46:8E:4D	6	*	80	*	www.e1.org
:		:	:	:	:	:

Exact Matches

Ternary Match

We develop an SRAM-based decision tree packet classification algorithm that runs entirely in the switch data plane.

Technical challenges

Challenges	Our approach
1. Emulate '*' behavior using purely SRAM-based operations • A decision tree approach requires '*' values to be unfurled.	Partition the MUD ACLs to reduce impact of '*' values on the size of the ruleset
2. MUD rules of same device types are repeated. • Same IoT device types have similar MUD profiles	Identify and encode rules of an IoT device type only once in the decision tree
3. Map decision tree to the SRAM match action pipeline in the switch data plane	Introduce state values to drive packets through switch pipeline stages

Technical challenges

Challenges	Our approach
1. Emulate '*' behavior using purely SRAM-based operations • A decision tree approach requires '*' values to be unfurled.	Partition the MUD ACLs to reduce impact of '*' values on the size of the ruleset
2. MUD rules of same device types are repeated. • Same IoT device types have similar MUD profiles	Identify and encode rules of an IoT device type only once in the decision tree
3. Map decision tree to the SRAM match action pipeline in the switch data plane	Introduce state values to drive packets through switch pipeline stages

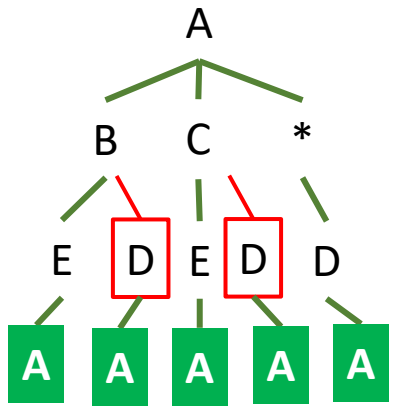
Challenge 1: Rule unfurling

Challenge: When using decision tree-based approach using SRAM based operations, unfurling wildcard values (*) lead to rule explosion

Original MUD ACL Rules

	H1	H2	H3	Action
Rule 1	A	B	E	Allow
Rule 2	A	C	E	Allow
Rule 3	A	*	D	Allow

Decision tree



Scenarios

[A,B,E] Allowed
[A,C,E] Allowed
[A,U,D] Allowed
[A,B,D] not allowed

Modified Rules

	H1	H2	H3	Action
Rule 1	A	B	E	Allow
Rule 2	A	C	E	Allow
Rule 3	A	*	D	Allow
Unfurled Rule 1	A	B	D	Allow
Unfurled Rule 2	A	C	D	Allow

Added rules

Our approach: Partition the MUD ACL ruleset to reduce the number of * values and avoid explosion of added rules.

Solution: Rule partitioning[1]

	Rule	sMAC	dMAC	typEth	Proto	sPort	dPort	srcIP	dstIP
Forward Traffic	1	94:16:3e:3b:41:cf	*	0x0800	6	*	8899	*	52.23.33.88
Backward Traffic	2	*	94:16:3e:3b:41:cf	0x0800	6	8899	*	52.23.33.88	*
Forward Traffic	3	94:16:3e:3b:41:cf	*	0x0800	17	*	53	*	*
Backward Traffic	4	*	94:16:3e:3b:41:cf	0x0800	17	53	*	*	*
Forward Traffic	5	94:16:3e:3b:41:cf	*	0x0800	6	*	9207	*	*
Backward Traffic	6	*	94:16:3e:3b:41:cf	0x0800	6	9207	*	*	*
Local Rules	7	94:16:3e:3b:41:cf	*	0x0800	6	49153	*	*	*
Local Rules	8	*	94:16:3e:3b:41:cf	0x0800	6	*	49153	*	*

FT (Forward Traffic)

Represents traffic originating from the device

- Represented as a separate decision tree on SRAM

BT (Backward Traffic)

Represents traffic incoming to the device

- Represented as a separate decision tree on SRAM

LCL (local rules)

Represents traffic that occur between the IoT devices

- Stored without modification in the TCAM

SRAM heavy but light on TCAM **vs** **TCAM heavy approach**

[1]**UNSW MUD Database:** MUD Profiles

Solution: Header partitioning

Partitioned MUD ACL ruleset

Approach: Reduce these header fields

Forward Traffic

Rule	sMAC	dMAC	typEth	Proto	sPort	dPort	srcIP	dstIP
1	94:16:3e:3b:41:cf	*	0x0800	6	*	8899	*	52.23.33.88
3	94:16:3e:3b:41:cf	*	0x0800	17	*	53	*	*
5	94:16:3e:3b:41:cf	*	0x0800	6	*	9207	*	*

Backward Traffic

Rule	sMAC	dMAC	typEth	Proto	sPort	dPort	srcIP	dstIP
2	*	94:16:3e:3b:41:cf	0x0800	6	8899	*	52.23.33.88	*
4	*	94:16:3e:3b:41:cf	0x0800	17	53	*	*	*
6	*	94:16:3e:3b:41:cf	0x0800	6	9207	*	*	*

Forward Traffic

Rule	sMAC	typEth	Proto	dPort	dstIP
1	94:16:3e:3b:41:cf	0x0800	6	8899	52.23.33.88
3	94:16:3e:3b:41:cf	0x0800	17	53	*
5	94:16:3e:3b:41:cf	0x0800	6	9207	*

Backward Traffic

Rule	dMAC	typEth	Proto	sPort	srcIP
2	94:16:3e:3b:41:cf	0x0800	6	8899	52.23.33.88
4	94:16:3e:3b:41:cf	0x0800	17	53	*
6	94:16:3e:3b:41:cf	0x0800	6	9207	*

Solution: Wildcard optimization

- We summarize the patterns observed in the partitioned MUD ACL rules of all 28 IoT devices[1].
- Only a few cases need to be unfurled

Forward Traffic (FT)

sMAC	typEth	proto	dPort	dstIP	Action
E	E	17	53	*	No
E	E	17	67	*	No
E	E	1	*	*	No
E	E	1	*	E	No
E	*✓	2	*	E	Yes
E	E	6	*✓	*	Yes
E	E	17	*✓	*	Yes
E	E	17	E	*✓	Yes

Backward Traffic (BT)

dMAC	typEth	proto	sPort	srcIP	Action
E	E	17	53	*	No
E	E	17	67	*	No
E	E	1	*	*	No
E	E	1	*	E	No
E	E	6	*✓	*	Yes
E	E	17	*✓	*	Yes

[1]**UNSW MUD Database:** MUD Profiles

Technical challenges

Challenges	Our approach
1. Emulate '*' behavior using purely SRAM-based operations • A decision tree approach requires '*' values to be unfurled.	Partition the MUD ACLs to reduce impact of '∗' values on the size of the ruleset
2. MUD rules of same device types are repeated. • Same IoT device types have similar MUD profiles	Identify and encode rules of an IoT device type only once in the decision tree
3. Map decision tree to the SRAM match action pipeline in the switch data plane	Introduce state values to drive packets through switch pipeline stages

Challenge 2: Decision tree representation

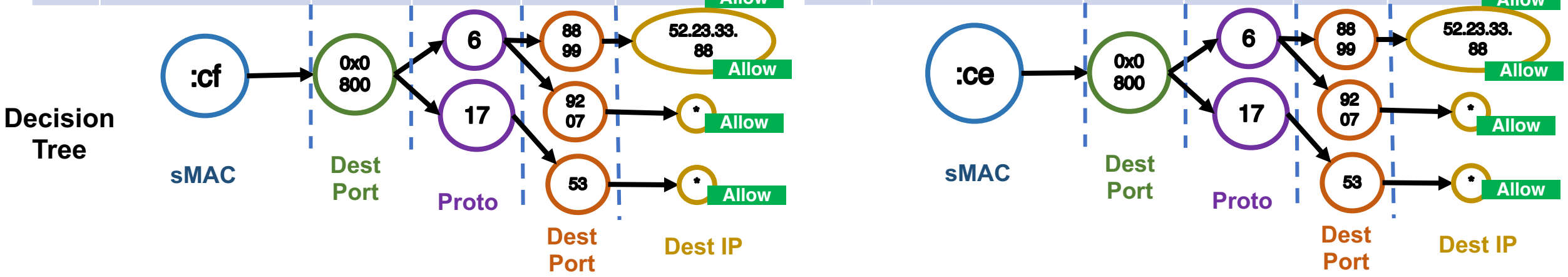
Observation: MUD-based ACLs have repeating values

IoT Device 1

Rule	sMAC	typEth	Proto	dPort	dstIP
1	94:16:3e:3b:41:cf	0x0800	6	8899	52.23.33.88
3	94:16:3e:3b:41:cf	0x0800	17	53	*
5	94:16:3e:3b:41:cf	0x0800	6	9207	*

IoT Device 2

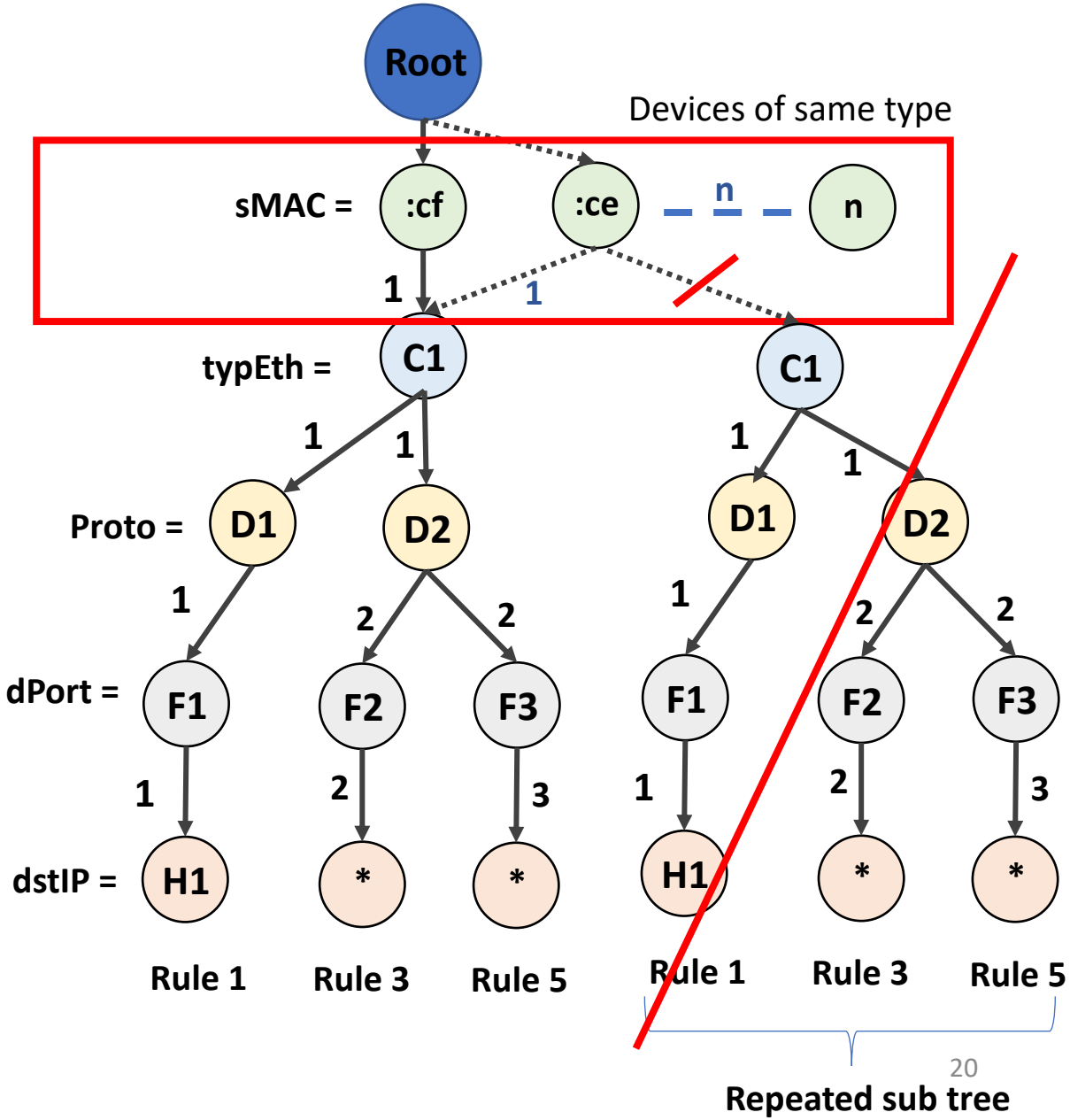
Rule	sMAC	typEth	Proto	dPort	dstIP
1	94:16:3e:3b:41:ce	0x0800	6	8899	52.23.33.88
3	94:16:3e:3b:41:ce	0x0800	17	53	*
5	94:16:3e:3b:41:ce	0x0800	6	9207	*



Identify same IoT device types and avoid reencoding their rules

Solution: Repoint the nodes

- Identify IoT device type from the MUD profile.
  - MUD URL in DHCP discover helps identify same IoT device type
- Encode rules of a device type only once.
  - Remove repeated subtree
  - Redirect new device node to existing subtree

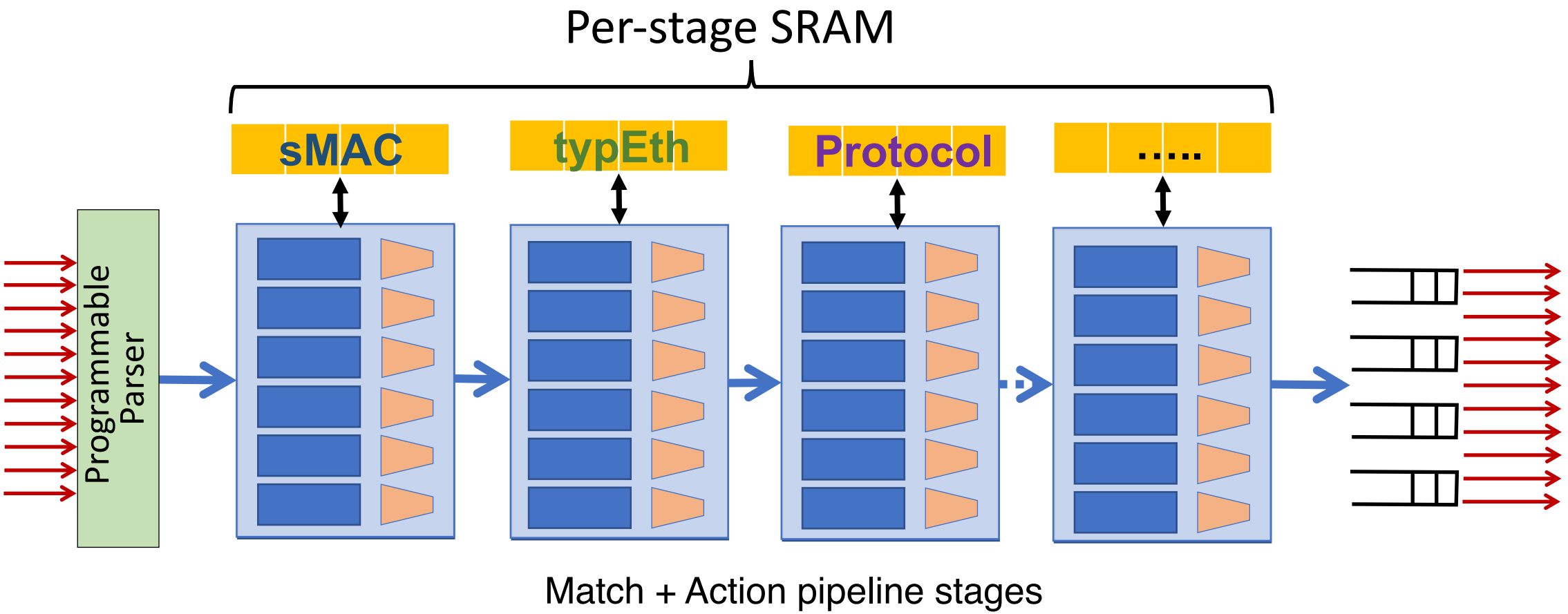


Technical challenges

Challenges	Our approach
1. Emulate '*' behavior using purely SRAM-based operations • A decision tree approach requires '*' values to be unfurled.	Partition the MUD ACLs to reduce impact of '∗' values on the size of the ruleset
2. MUD rules of same device types are repeated. • Same IoT device types have similar MUD profiles	Identify and encode rules of an IoT device type only once in the decision tree
3. Map decision tree to the SRAM match action pipeline in the switch data plane	Introduce state values to drive packets through switch pipeline stages

Challenge 3: Decision tree in data plane

Key Idea: Each decision tree layer is mapped to a stage



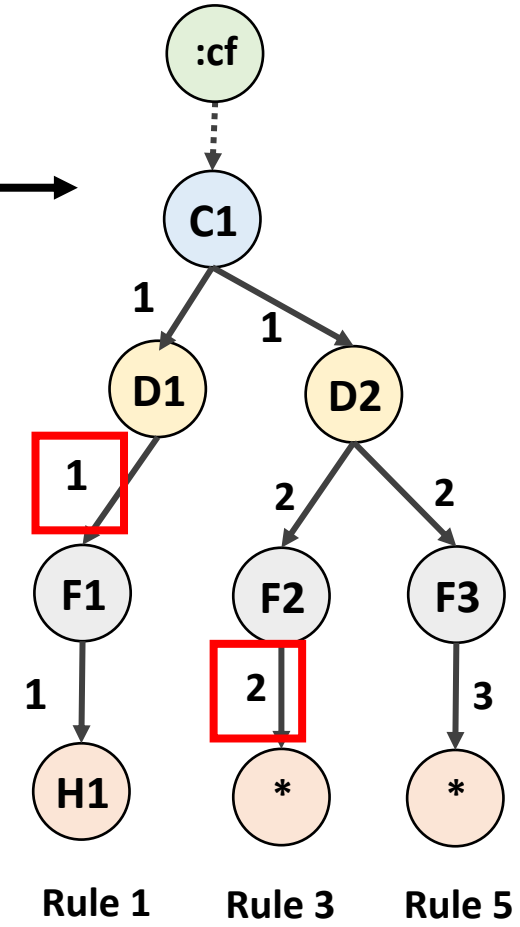
Solution: State values

Key Idea: Drive a packet through the stages using state values.

Forward Traffic

Rule	sMAC	typEth	Proto	dPort	dstIP
1	94:16:3e:3b:41:cf	0x0800	6	8899	52.23.33.88
3	94:16:3e:3b:41:cf	0x0800	17	53	*
5	94:16:3e:3b:41:cf	0x0800	6	9207	*

Decision tree



Match action table

Rule no.	sMAC_ exact		typEth_ exact			typEth_ default		proto_ exact			dPort_ exact			dPort_ default		dstIP_ exact			dstIP_ default	
	val	ns	cs	val	ns	cs	ns	cs	val	ns	cs	val	ns	cs	ns	cs	val	act	cs	act
1	:cf	1	1	C1	1			1	D1	1	1	F1	1			1	H1	f		
3								1	D2	2	2	F2	2						2	f
5											2	F3	3						3	f

Evaluation

How many IoT devices does our approach scale to?

For 4480 devices

- Our approach
  - 114KB of SRAM
- TCAM based approach
  - 2.8 MB of TCAM

Up to 0.7 million devices per switch

Switch latency when scaling IoT devices?

- Min Latency: 364ns
- Max latency: 366ns

Negligible increase in latency when scaling

Switch latency in purely TCAM approach

- Avg Latency: 290ns

Differs by ~75ns

Summary and Future work

- We design a **data plane primitive** to scale MUD enforcement using P4-based programmable switch
- Using a decision tree-based design choice, we can scale well
- Future work:
  - Primitive to support varied network scenarios

Scaling IoT MUD Enforcement using Programmable Data Planes

Harish S A[1], Suvrima Datta[2], Hemanth Kothapalli[1], Praveen Tammana[1], Achmad Basuki[3], Kotaro Kataoka[1], Selvakumar Manickam[4], Venkanna U[2], Yung-Wey Chong[4]

Thank You

9th August 2022

IEEE/IFIP Network Operations and Management Symposium 2023