Deep Packet Inspection as a Service

Yaron Koral†

Joint work with Anat Bremler-Barr‡, Yotam Harchol† and David Hay†

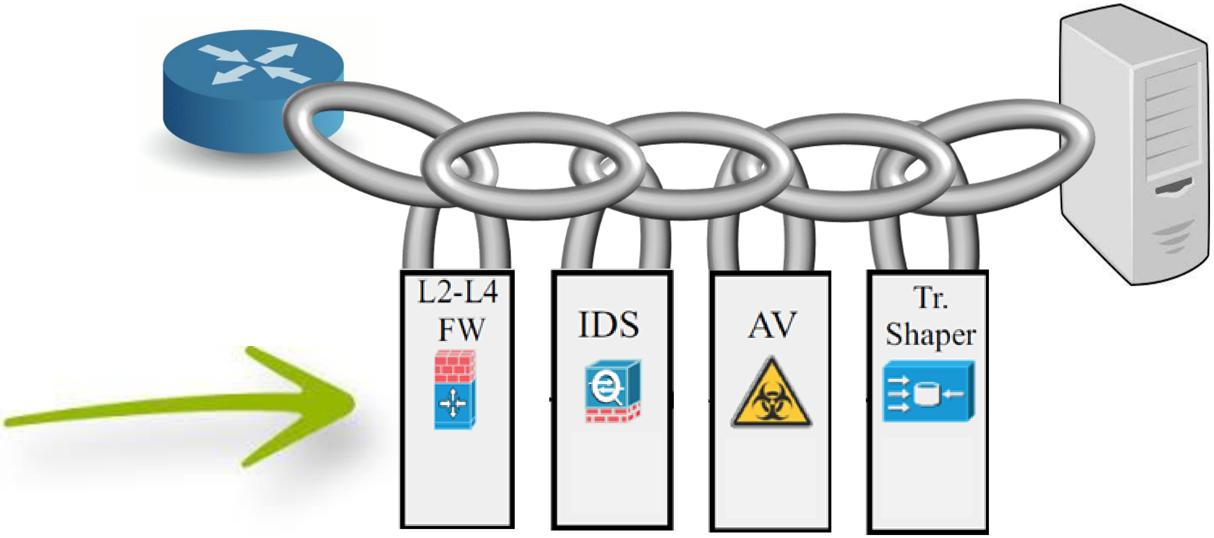
†The Hebrew University, Israel
‡IDC Herzliya, Israel

This work was supported by European Research Council (ERC) Starting Grant no. 259085

NFV and Innovation

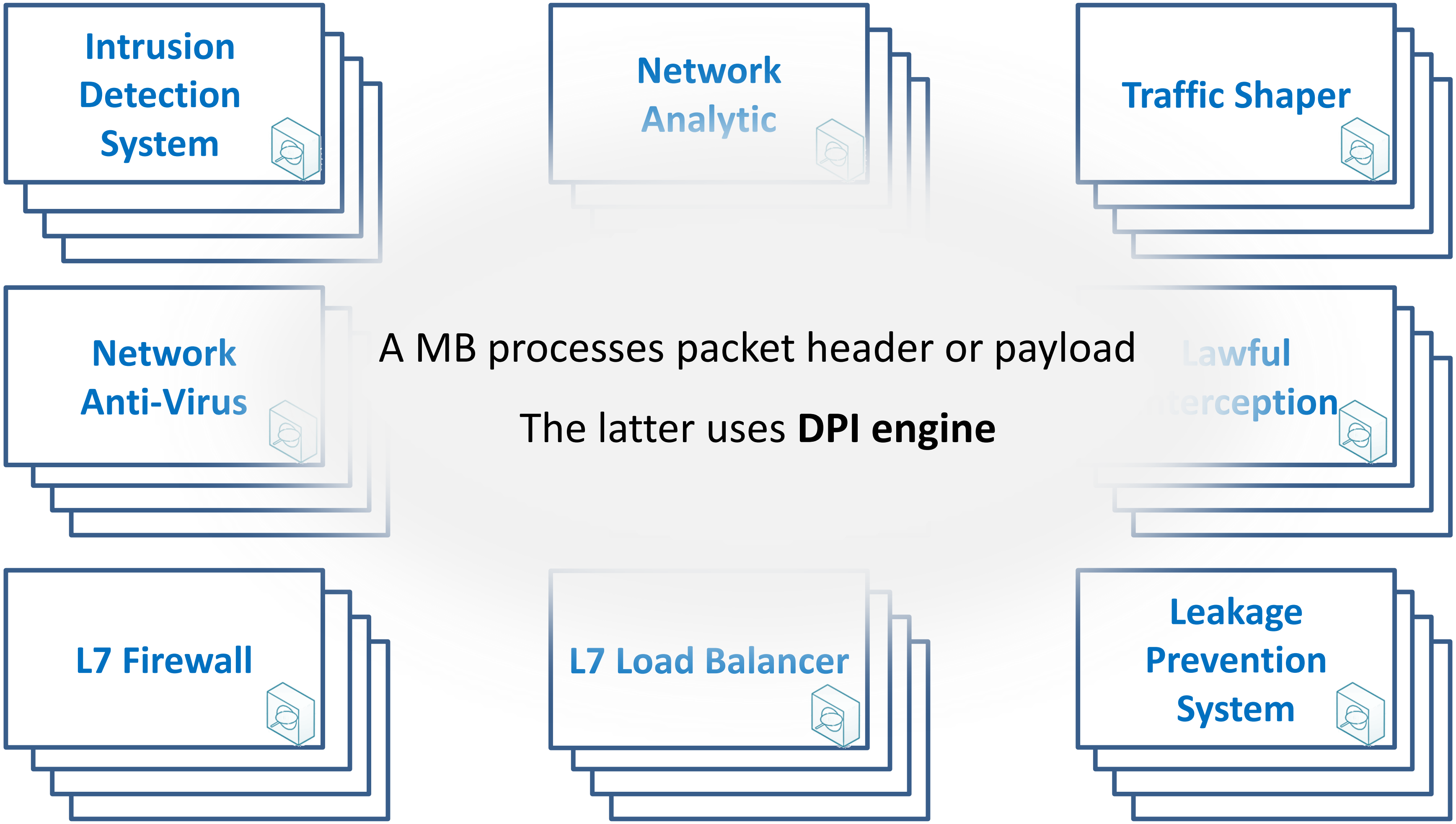
- NFV enables virtualizing network building blocks such as: FW, NAT, IDS, Monitoring, LB, Network AV, WAN Optimizer, etc. …
- Increases network deployment flexibility, and also product introduction times.
- **Opens market for new vendors for network functions (Middleboxes)**
- About 400 SDN-NFV listed companies

Middleboxes Policy Chains



SDN allows building policy chain via **traffic steering**

DPI Based Middleboxes



Intrusion Detection System

Network Analytic

Traffic Shaper

Network Anti-Virus

Lawful Interception

L7 Firewall

L7 Load Balancer

Leakage Prevention System

A MB processes packet header or payload

The latter uses **DPI engine**

DPI Pattern Examples

Snort (NIDS/NIPS) – Intrusion Detection

Microsoft XML Core Services cross-site information disclosure attempt

`<\x21DOCTYPE\s+[^>]*SYSTEM[^>]*>.*\x2EparseError`

ClamAV (Anti Virus) – Virus Detection

Cabir.A computer worm signature

`886f1f10123a001019040010e5f79547e6ad0100bd006f0064007500630074004900440054003200200052005300330041005300789c` (binary)

Bro (NIDS) - Application Classification

MS Office 2007 XML documents

`\x50\x4B\x03\x04\x14\x00\x06\x00`

DPI Engine – Complicated Challenge

- Hundreds of academic papers over recent years

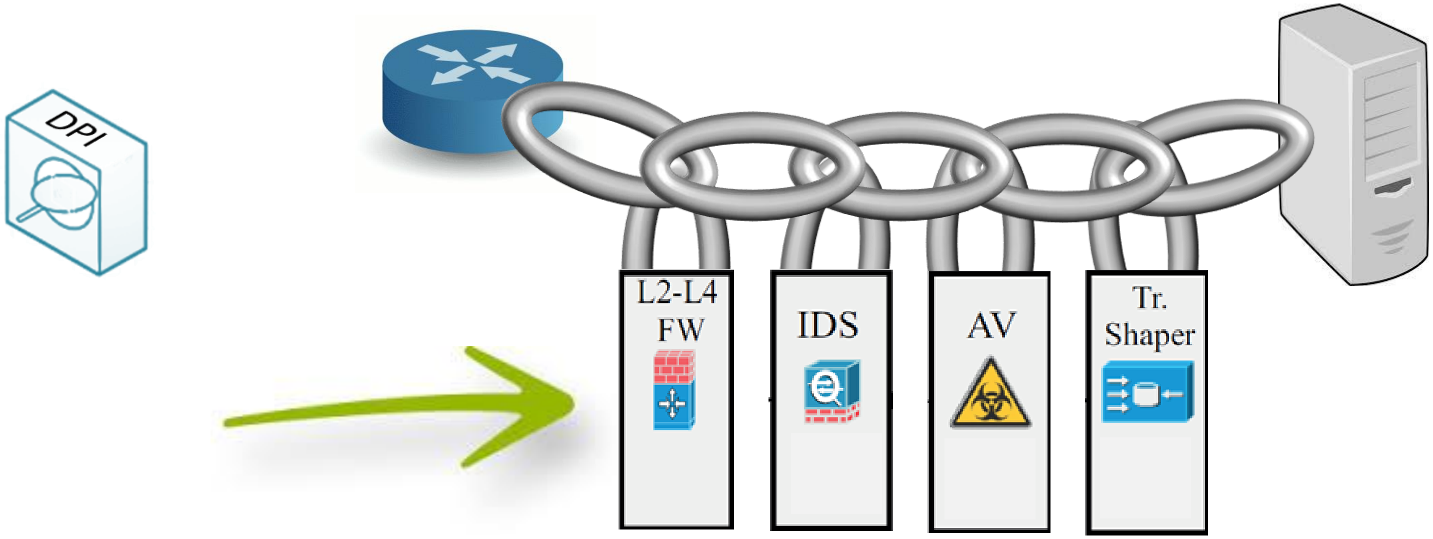
scalability | throughput | latency | low power

resiliency | fast updates | compression

- Pattern set size varies between 10^2-10^5 patterns per MB
- DPI Engine is considered a **system bottleneck** in many of todays MBs (**30%-80%**)

[*Laboratory simulations over real deployments of Snort and ClamAV]

Middleboxes Policy Chains



- Each MB implements its own DPI engine (higher MB costs, reduced features)
- Each packet is scanned multiple times causing waste of computation resources

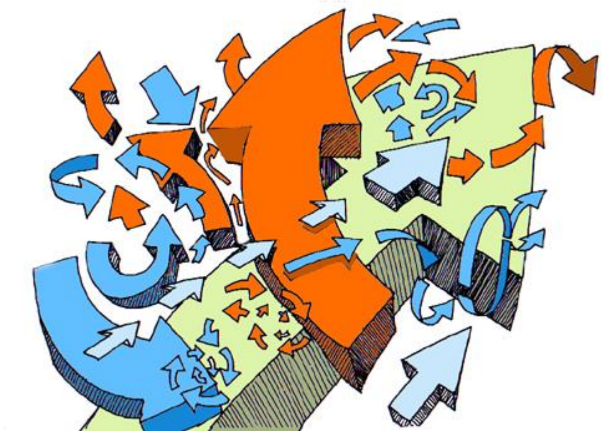
Our Solution: DPI as a Service

Contribution:

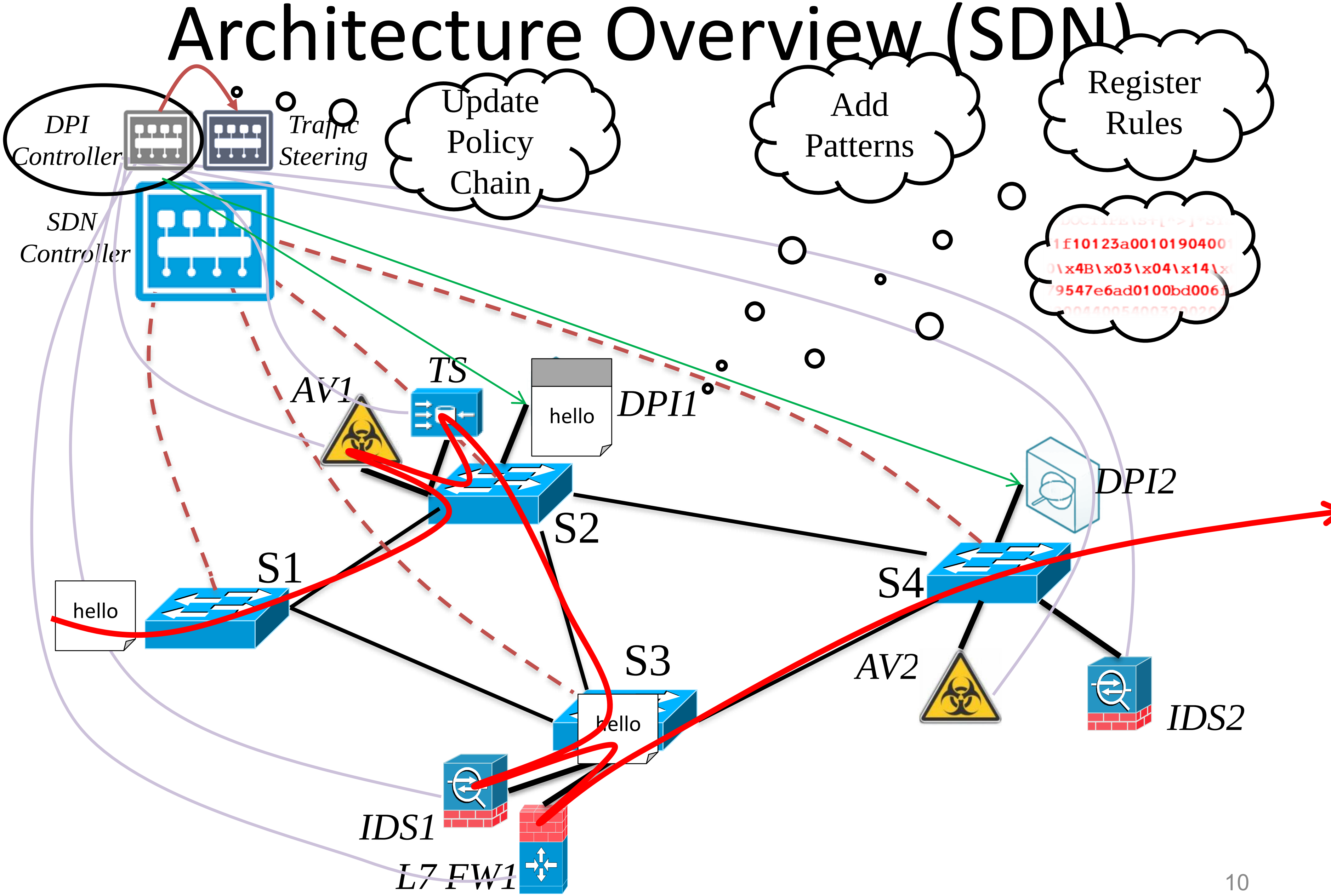
The idea of having
a centralized DPI service
instead of **multiple instances** of it
at each Middlebox

- **Innovation** – Lower entry barriers
- **Rich Functionality** – Invest once for all MB
- **Reduced Costs** – Cheaper MB HW/SW
- **Improved performance** - Scan each packet **once**

ARCHITECTURE

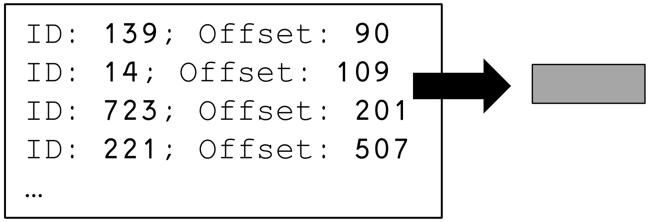


Architecture Overview (SDN)

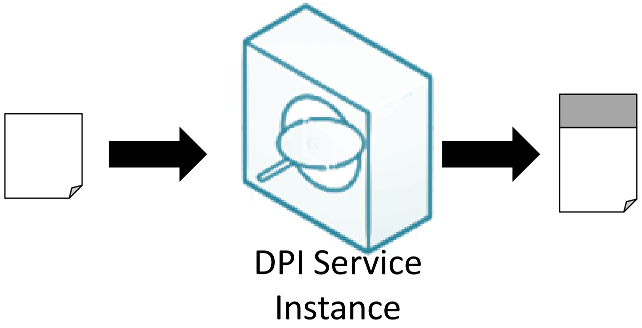


Architecture: Data Plane

- DPI Service Instance Scans incoming packets against an aggregated pattern set
- Each pattern has a unique ID
- Result: <Pattern ID> + <Match Offset>
- Each packet may contain several pattern matches
- All pattern-match results are attached to the packet



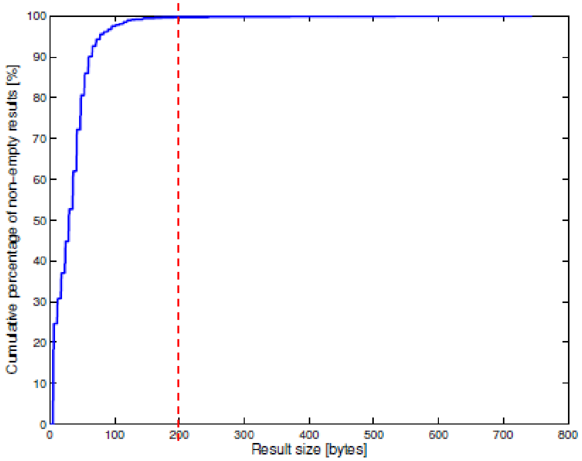
```
ID: 139; Offset: 90
ID: 14; Offset: 109
ID: 723; Offset: 201
ID: 221; Offset: 507
…
```

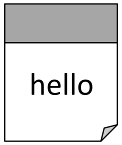


Passing Results

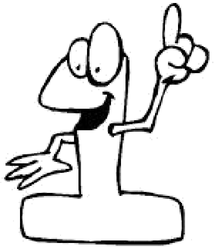
- **Match results header**:

  usually 0B; up to 200B (99%)



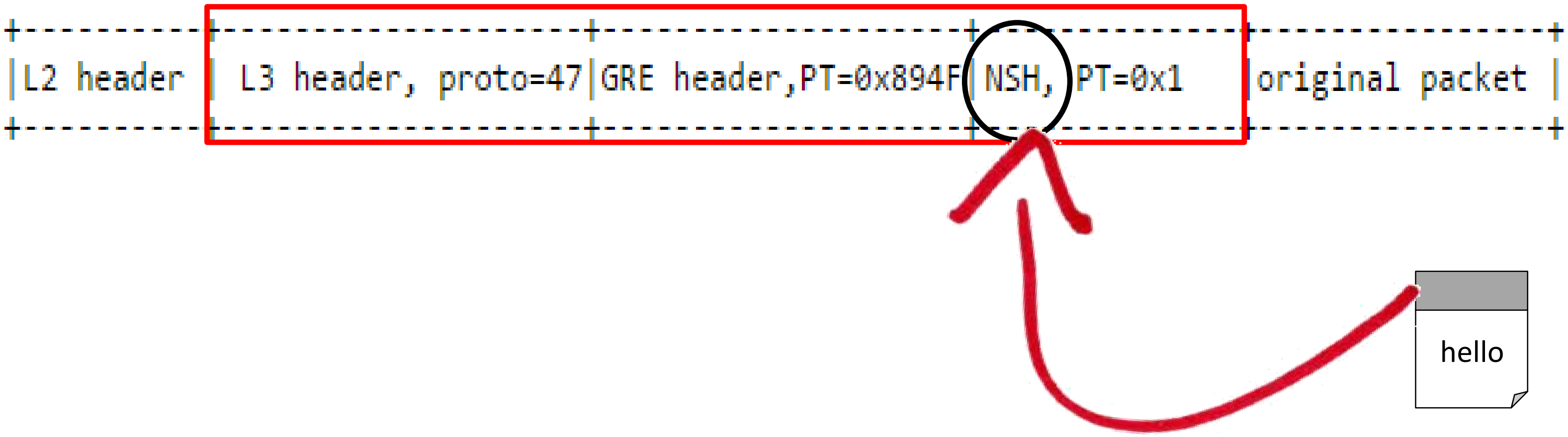


- Using existing tag-fields (i.e. VLAN / MPLS) does not suffice



Passing Results Alternative 1

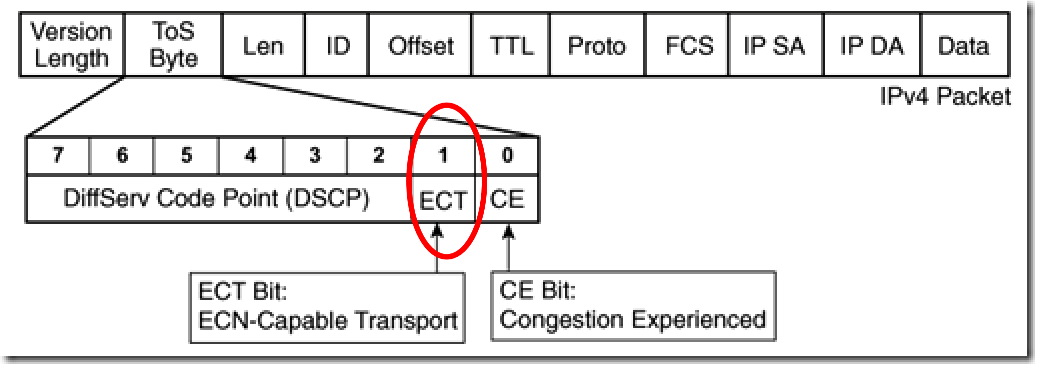
- **Network Service Header (NSH)**
  - Supports a header per network service
  - Not limited in size
  - Resize MTU

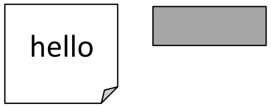


```
+----------+--------------------+--------------------+--------------+----------------+
|L2 header | L3 header, proto=47|GRE header,PT=0x894F|NSH, PT=0x1   |original packet |
+----------+--------------------+--------------------+--------------+----------------+
```

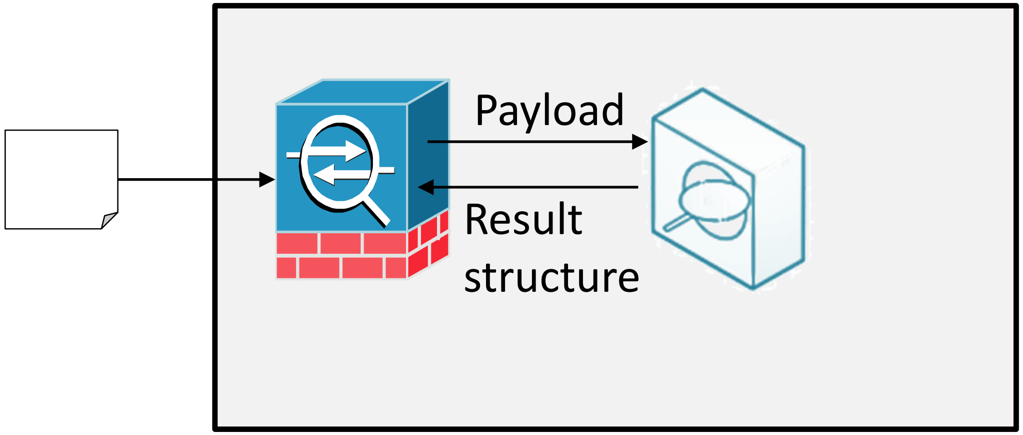
Passing Results Alternative 2

- **Separate result-packet**
  - – Mark original packet upon match (set ECN)
  - – Delay packet until result-packet arrives





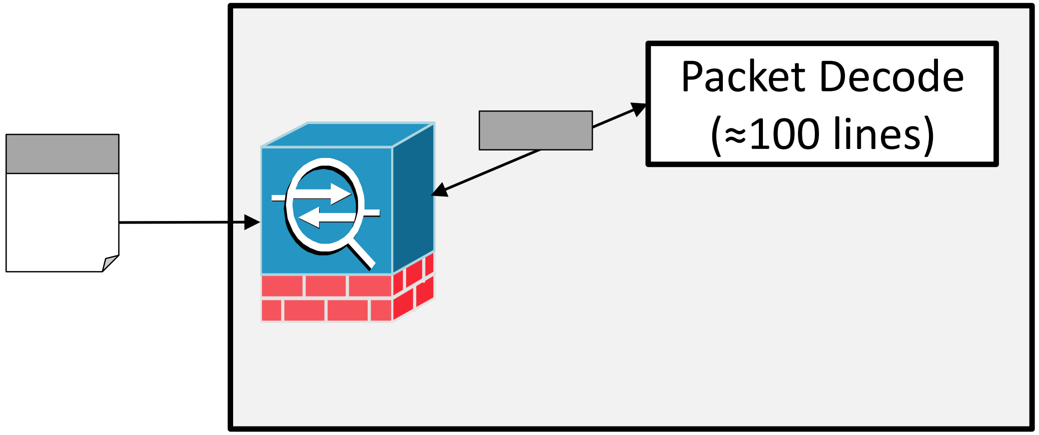
MiddleBox Support



Payload

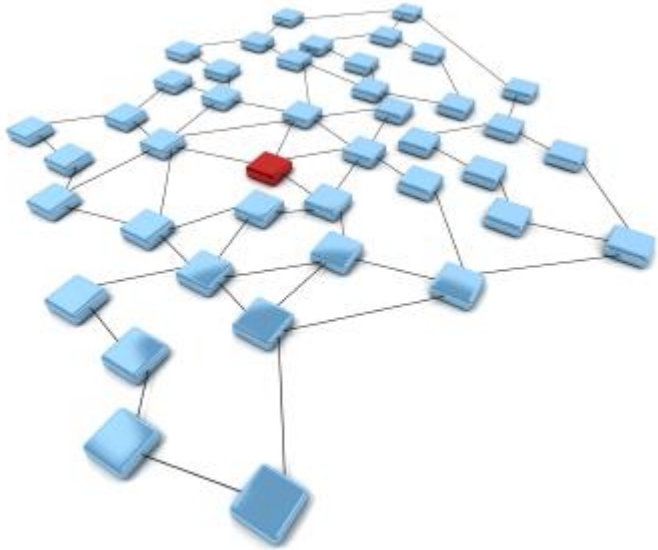
Result structure

MB with internal DPI engine



Packet Decode (≈100 lines)

MB with external DPI service



QUESTION: ARE THE DPI ALGORITHMS SCALABLE?

Are DPI Algorithms Scalable?

- Short Answer: **YES!**

- What are the DPI Algorithms?

String Matching ⊂ Regex Matching

```
886f1f10123a001019040010e5f79547e6ad0100bd006
f006400750063007400490044005400320020005200530
0033004100530078 9c
```

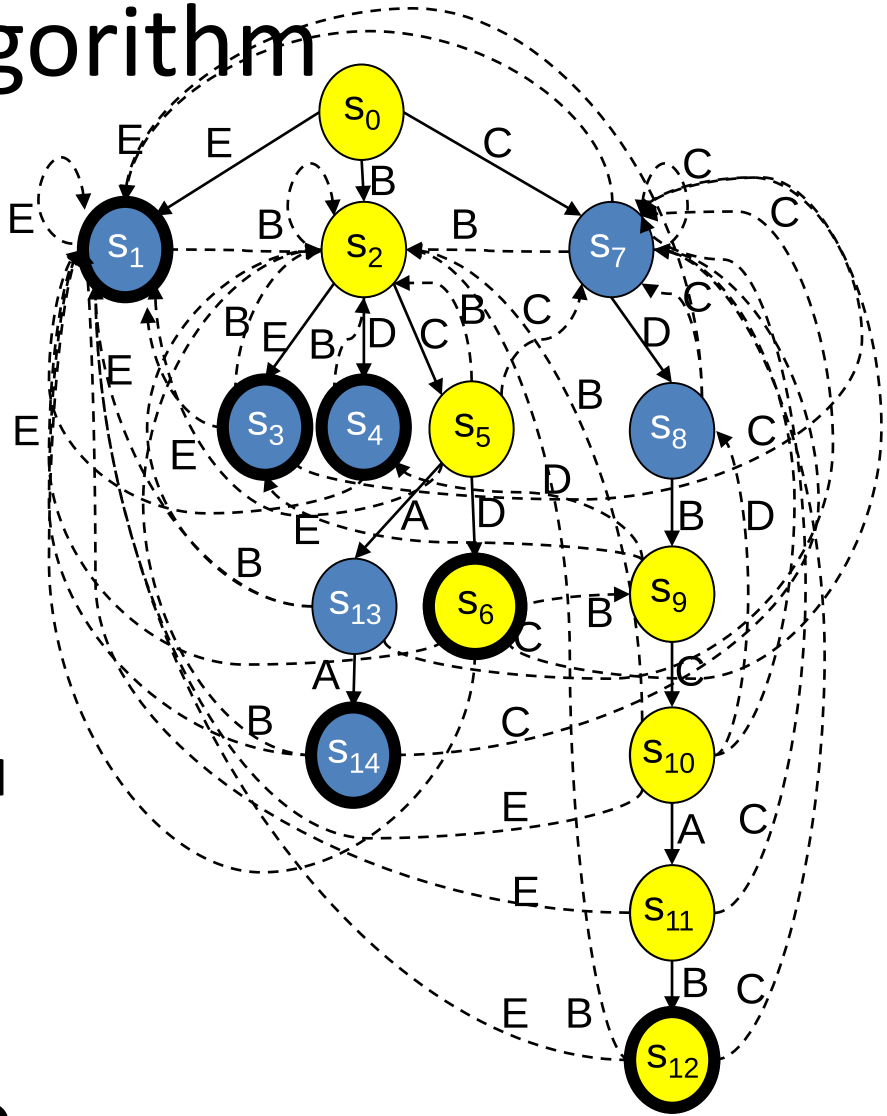
```
<\x21DOCTYPE\s+[^>]*SYSTEM[^>]*>.*\x2EparseError
```

String Matching: Aho-Corasick Algorithm

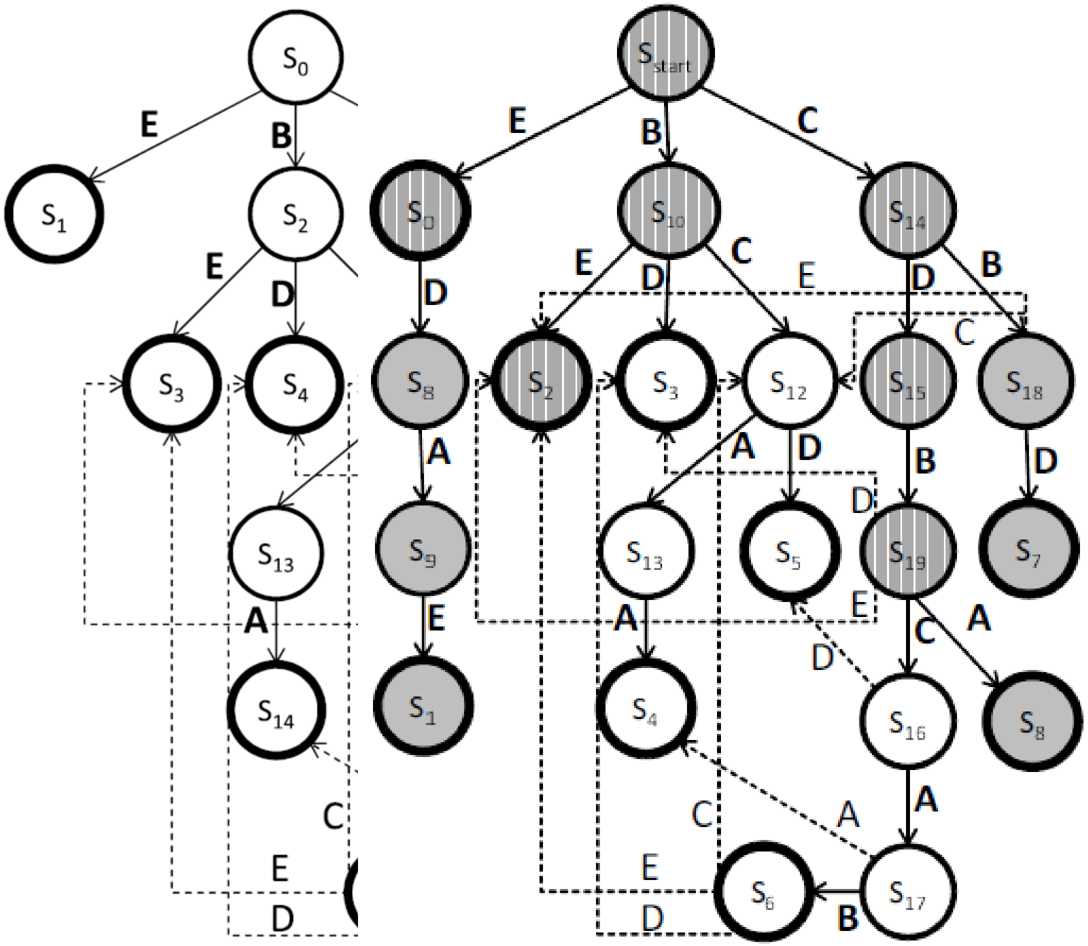
- Build a Deterministic Finite Automaton (basic full-table variant)
- Each input byte requires single lookup **regardless the number of patterns!!**
- $ Cost Function:

  1 Mem. access per **input byte**
- More patterns may results in a **marginal** performance reduction (cache)
- Example:
  {E, BE, BD, BCD, CDBCAB, BCAA}

BCDBCAB



Pattern Set Aggregation



Pattern Set 1

Pattern Set 2

match table

0	$\langle 0, E\rangle$
1	$\langle 0, E\rangle$, $\langle 1, EDAE\rangle$
2	$\langle 0, E\rangle$, $\langle 0, BE\rangle$, $\langle 1, BE\rangle$
3	$\langle 0, BD\rangle$
4	$\langle 0, BEAA\rangle$
5	$\langle 0, BCD\rangle$
6	$\langle 0, CDBCAB\rangle$
7	$\langle 0, BD\rangle$, $\langle 1, CBD\rangle$
8	$\langle 1, CDBA\rangle$

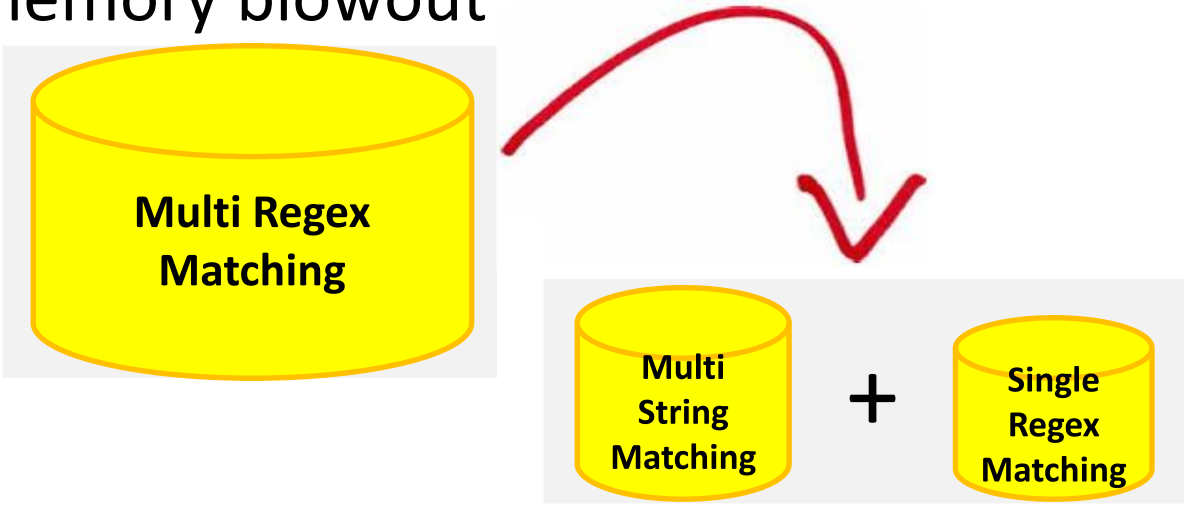
○ Pattern set 1

● Pattern set 2

▥ Both sets

Regular Expressions Matching

- Repetition operators (e.g. Kleen star) may cause memory blowout

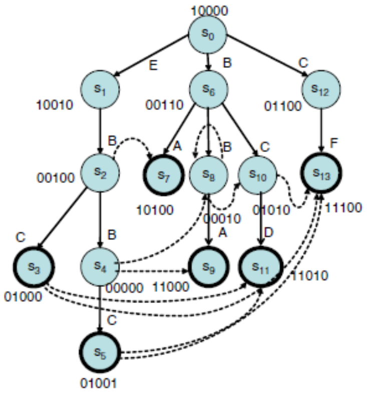


- Common approach:
  string matching w/global DFA >> single regex DFA

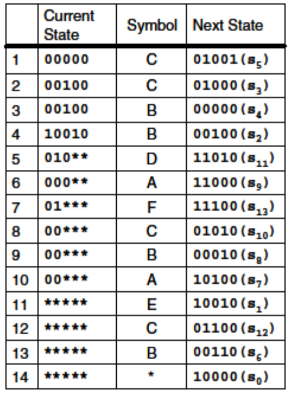
```
<\x21DOCTYPE\s+[^>]*SYSTEM[^>]*>.*\x2EparseError

<\x21DOCTYPE         SYSTEM         \x2EparseError
```

DPI is Scalable, not Trivial…



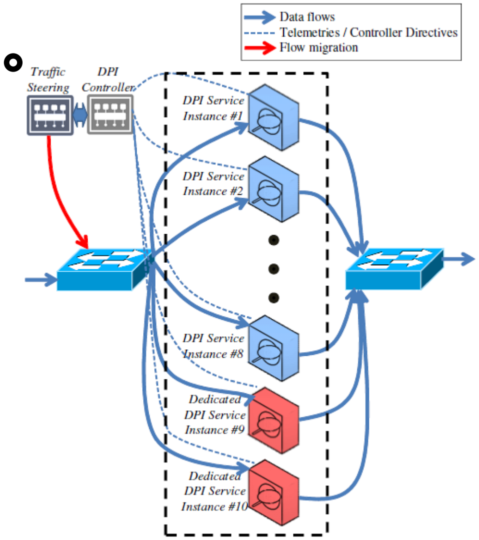
CompactDFA, ToN 2014



	Current State	Symbol	Next State
1	00000	C	01001(s_5)
2	00100	C	01000(s_3)
3	00100	B	00000(s_4)
4	10010	B	00100(s_2)
5	010**	D	11010(s_{11})
6	000**	A	11000(s_9)
7	01***	F	11100(s_{13})
8	00***	C	01010(s_{10})
9	00***	B	00010(s_8)
10	00***	A	10100(s_7)
11	*****	E	10010(s_1)
12	*****	C	01100(s_{12})
13	*****	B	00110(s_6)
14	*****	*	10000(s_0)

Encode DFA in TCAM

Resilient Multi-Core DPI



MCA^2, ANCS 2012

gzip

ACCH, ToN 2012

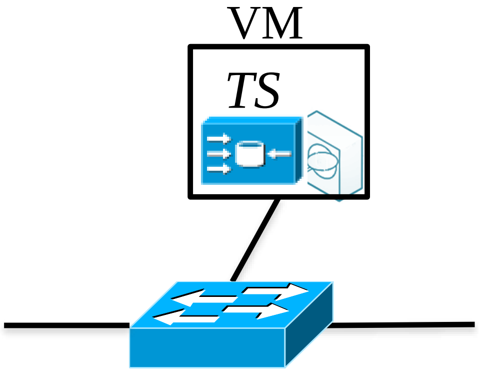
DPI over Compressed Traffic

DPI AS A SERVICE & DIFFERENT MIDDLEBOXES LAYOUTS

Related Network-Functions Layouts

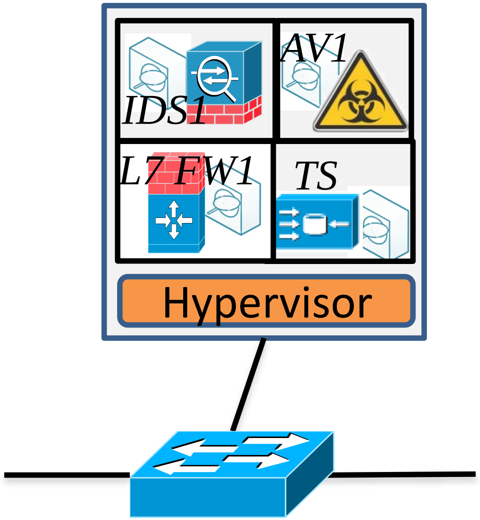
- SDN + NFV

  ETSI. Network functions virtualization
  Gember et al., HotNets 2012
  Rajagopalan et al., NSDI 2013



- MB Consolidation

  Comb, NSDI 2012
  xOMB, ANCS 2012
  Crossbeam, 2012
  Kekely et al., Infocom 2014



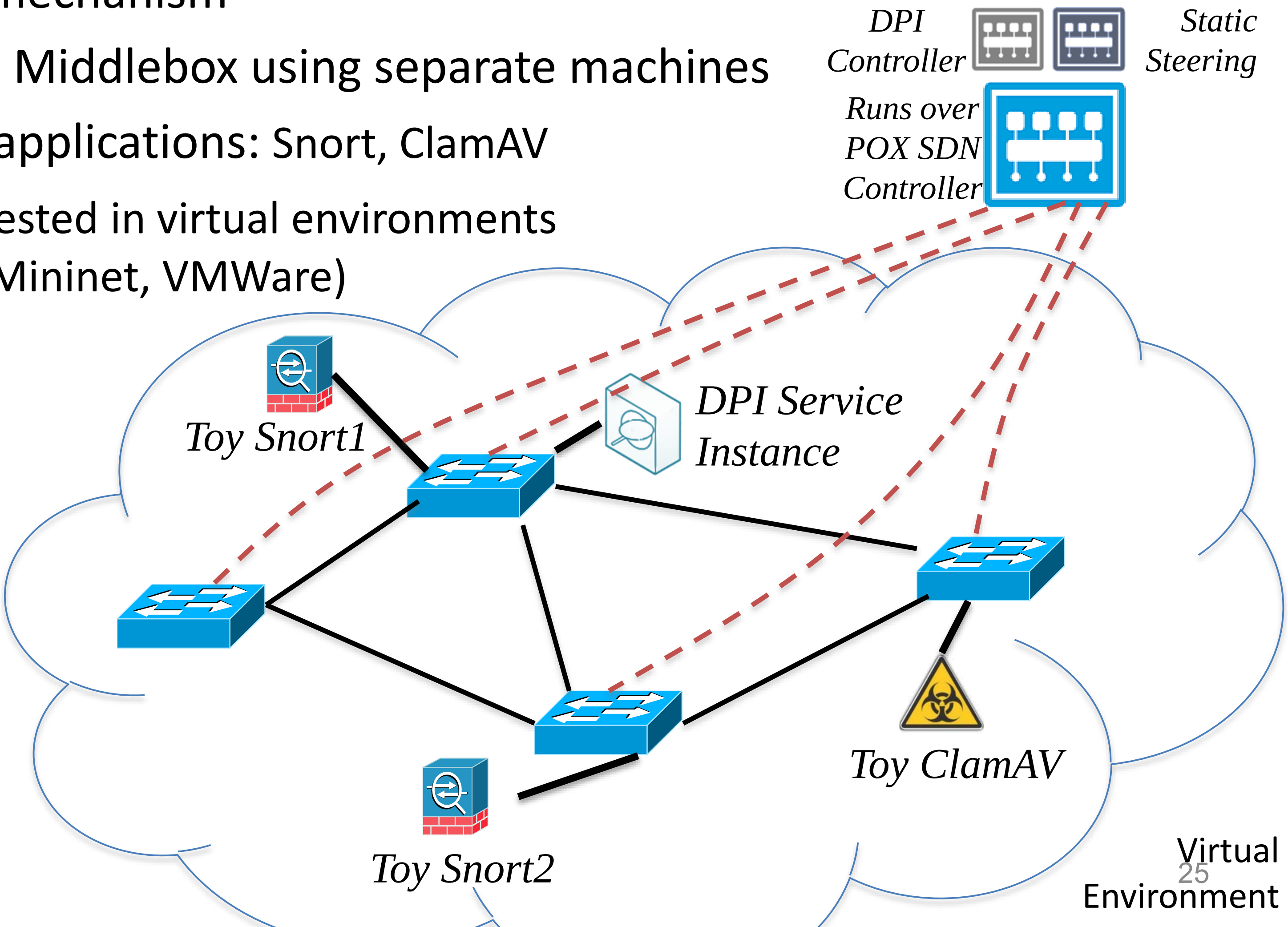
- Outsource MB (out-of-network)

  Gibb et al., HotSDN 2012
  Sherry et al., SIGCOMM 2012

EXPERIMENTAL RESULTS

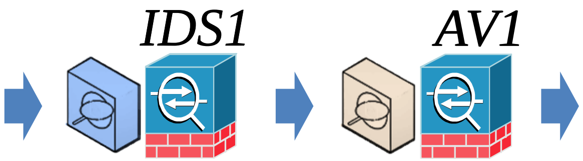
Experimental Environment

- POX SDN Controller (OpenFlow 1.0)
- Static steering mechanism
- DPI Service and Middlebox using separate machines
- Toy middlebox applications: Snort, ClamAV
- Functionality: Tested in virtual environments (Mininet, VMWare)
- Performance: no Mininet (overhead)



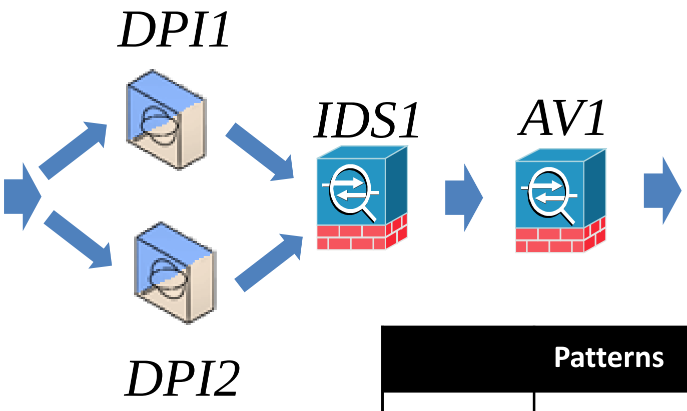
Performance Results

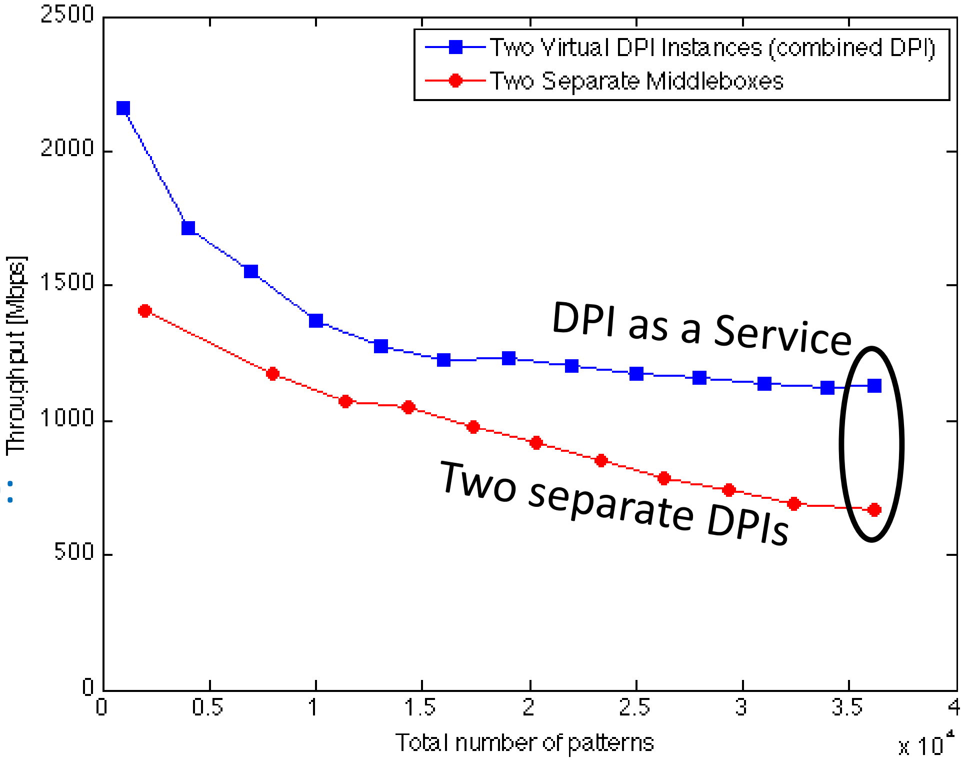
Policy Chain with Two DPIs :



Each using separate machines

Combined DPI instances (DPI as a Service):





	Patterns	Space	Throughput	Latency	Overall Throughput	Overall Latency
Snort	4356	71.18MB	807.7Mbps	9.69us/p	**668.4Mbps**	**21.5us/p**
ClamAV	31827	1.87GB	668.4Mbps	11.91us/p		
DPI1	36183	1.94GB	563.3Mbps	13.82us/p	**1126Mbps**	**13.8us/p**
DPI2	36183	1.94GB	563.3Mbps	13.82us/p		

Future Work

- Potential tasks to be “outsourced” as a service:
  - Payload Processing (Decryption/Decompression)
    - Retrieve raw data
  - Session reconstruction (Connection Tracking)
    - For session processing rather than packet processing
  - Header/protocol analysis
    - For protocol aware network functions
- Use the DPI to extend OpenFlow based switches
  - Use the tags created by the DPI service to drive policies in conventional switches.

Conclusions

- DPI is a common service used by today’s MB
- Thanks to its scalability it may be easily exported as a stand-alone network service
- DPI as a Service provides:
  - Innovation (Lower entry barriers)
  - Network scalability
  - Lower costs (Cheaper MB Hardware)

Thank You!!