

# Deep Packet Inspection of Next Generation Network Devices



Prof. Anat Bremner-Barr  
IDC Herzliya, Israel

[www.deepness-lab.org](http://www.deepness-lab.org)

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

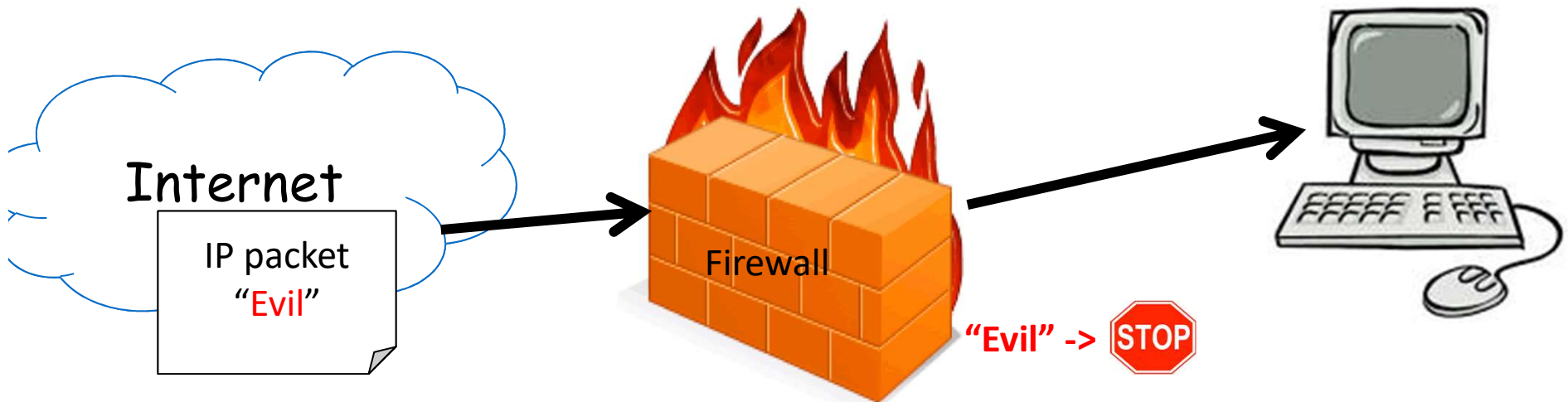
# DEEPNESS Lab

- Deepness - **D**PI **E**ngineering for **E**nhanced **P**erformance of **N**etwork **E**lements and **S**ecurity **S**ystems
- <http://www.deepness-lab.org/> - detailed publication list
- Deepness Lab was founded in November 2010 by Prof. Anat Bremler-Barr and Prof. David Hay
  - Group: more than 20 master, PhD and post doc students
  - More than 40 papers and several patents
  - Currently 1 PhD & 2 Masters



# Deep Packet Inspection (DPI)

- Classify packets according to:
  - Packet payload (data)
  - Against known set of patterns: strings or regular expressions



Snort (cross site inf. disclosure):

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

ClamAV (Cabir A. worm):

```
886f1f10123a001019040010e5f79547e6ad0100bd006f006400750063007
```

Bro (MS Office 2007 xml docs id):

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

- Common task in Network function (Middleboxes)

# DPI-Based Network Functions



**Intrusion  
Detection  
System**



**Network  
Analytic**



**Traffic Shaper**



**Network  
Anti-Virus**



**Copyright  
Enforcement**



**Lawful  
Interception**



**L7 Firewall**



**L7 Load Balancer**



**Leakage  
Prevention  
System**



# DPI Engine – Complicated Challenge

- DPI engine is considered a system bottleneck in many of today's NFs (**30%-80%**)  
[Laboratory simulations over real deployments of Snort and ClamAV]
- A well-studied problem in Computer Science but with no sufficient solutions to current demands.
- Hundreds of academic papers over recent years



# Major Challenges

- **Scalability:**
  - **Rate** - greater than 10 or even 100 Gbps
  - **Memory** - handling thousands of patterns
- **Handling non clear-text traffic**
  - Compressed traffic
- **Security of the DPI itself:**
  - resilient to Distributed Denial of Service (DDoS) attack
- **Opportunities:** DPI in Software Defined Networks(SDN) and Network Function Virtualization(NFV)

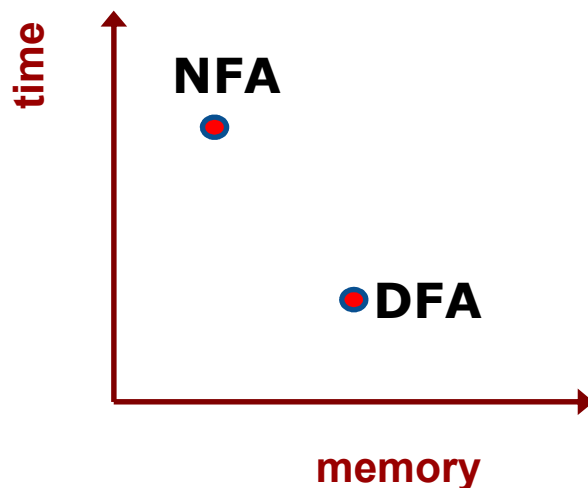
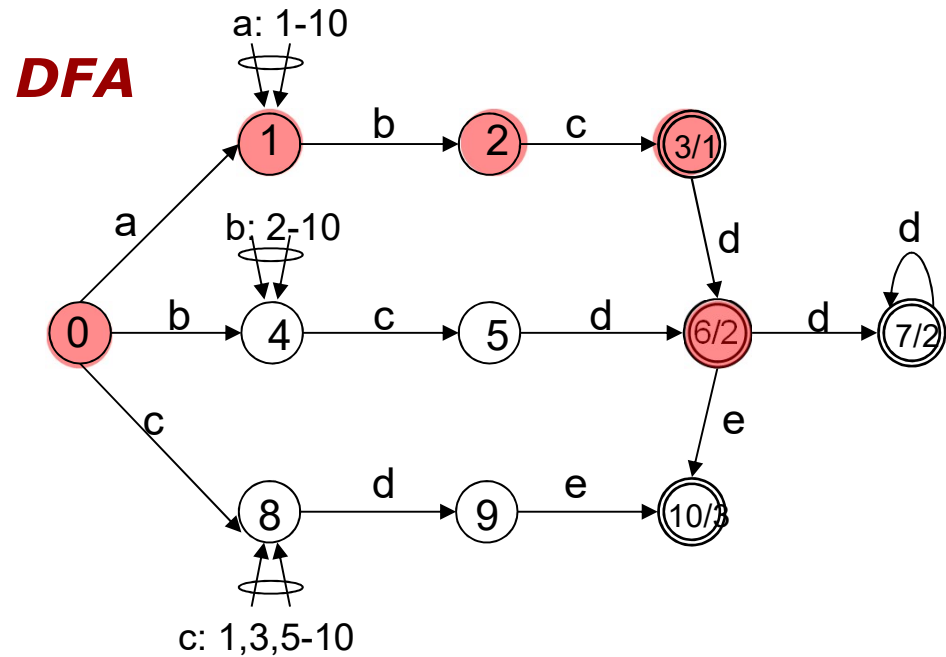
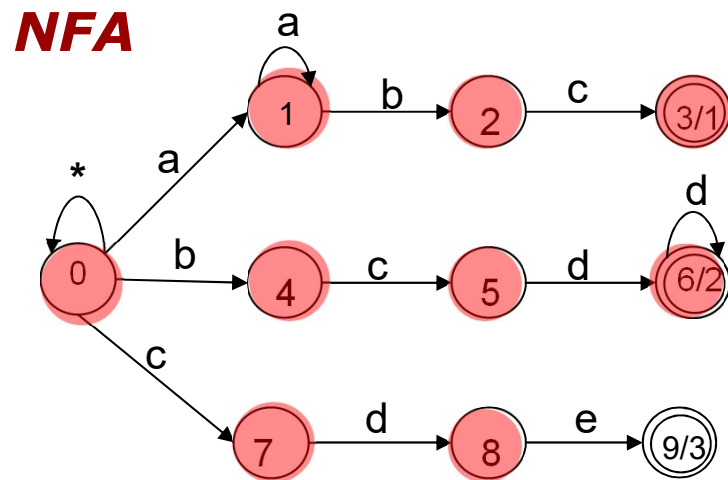
# Challenge #1: Scalability

Anat Bremler-Barr, Yaron Koral, David Hay, "CompactDFA: Scalable Pattern Matching using Longest Prefix Match Solutions",  
in IEEE/ACM Transactions on Networking, 2013

# Regular expression – classical solutions:

## Deterministic vs. Non-Deterministic Finite Automaton

RegEx: (1)  $.^*a^+bc$  (2)  $.^*bcd^+$  (3)  $.^*cde$



Text:  $d a b c d$

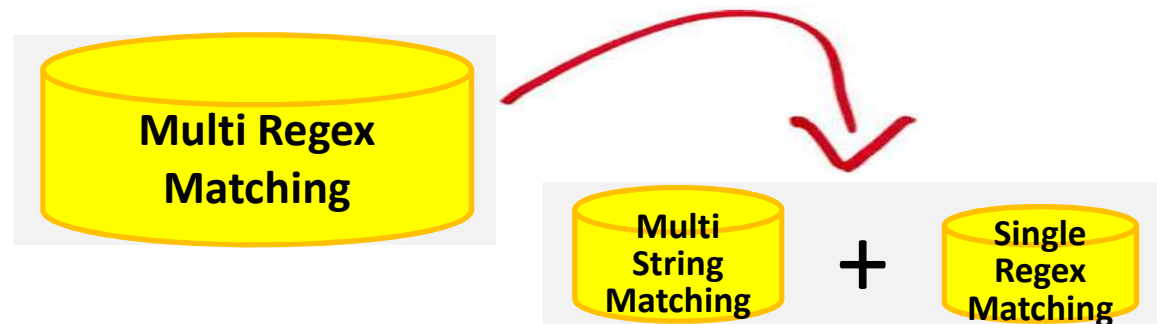
In practical cases **single DFA infeasible!**



# Feasible Solutions to Regular expressions

- **Common approach (e.g. Snort) implement two-phase approach:**
  1. String matching over all strings that appeared in the combined set of regular expressions
  2. Running a single regular expression DFA

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



# The challenge in string-matching

	Current	Sym	Next State
1	0000( $s_0$ )	A	0000 ( $s_0$ )
2	0000( $s_0$ )	B	0110( $s_6$ )
3	0000( $s_0$ )	C	1100( $s_{12}$ )
4	0000( $s_0$ )	D	0000( $s_0$ )
5	0000( $s_0$ )	E	0001( $s_1$ )
6	0000( $s_0$ )	F	0000( $s_0$ )
7	0001( $s_1$ )	A	0000( $s_0$ )
8	0001( $s_1$ )	B	0010( $s_2$ )
9	0001( $s_1$ )	C	0000( $s_0$ )
10	0001( $s_1$ )	D	0000( $s_0$ )
11	0001( $s_1$ )	E	0000( $s_0$ )
12	0001( $s_1$ )	F	0000( $s_0$ )
13	0010( $s_2$ )	A	0000( $s_0$ )
14	0010( $s_2$ )	B	0100( $s_4$ )
15	0010( $s_2$ )	C	0011( $s_3$ )
16	0010( $s_2$ )	D	0000( $s_0$ )
84	1101( $s_{13}$ )	F	0000 ( $s_0$ )

- Common algorithm Aho-Corasick
- Common implementation full table DFA :  $|\text{States}| \times |\text{Alphabet}|$
- Cannot be fully in fast SRAM:
  - Snort: 73MB
  - ClamAV: 1.5GB

	Current	Sym	Next State
1	0000( $s_0$ )	A	0000 ( $s_0$ )
2	0000( $s_0$ )	B	0110( $s_6$ )
3	0000( $s_0$ )	C	1100( $s_{12}$ )
4	0000( $s_0$ )	D	0000( $s_0$ )
5	0000( $s_0$ )	E	0001( $s_1$ )
6	0000( $s_0$ )	F	0000( $s_0$ )
7	0001( $s_1$ )	A	0000( $s_0$ )
8	0001( $s_1$ )	B	0010( $s_2$ )
9	0001( $s_1$ )	C	0000( $s_0$ )
10	0001( $s_1$ )	D	0000( $s_0$ )
11	0001( $s_1$ )	E	0000( $s_0$ )
12	0001( $s_1$ )	F	0000( $s_0$ )
13	0010( $s_2$ )	A	0000( $s_0$ )
14	0010( $s_2$ )	B	0100( $s_4$ )
15	0010( $s_2$ )	C	0011( $s_3$ )
16	0010( $s_2$ )	D	0000( $s_0$ )
⋮			
84	1101( $s_{13}$ )	F	0000 ( $s_0$ )



	current	sym	next state
1	00000	C	01001 ( $s_5$ )
2	00101	C	01010 ( $s_3$ )
3	00101	B	00000 ( $s_4$ )
4	10001	B	00101 ( $s_2$ )
5	010**	D	11001 ( $s_{11}$ )
6	000**	A	11000 ( $s_9$ )
7	01***	F	11010 ( $s_{13}$ )
8	00***	C	01001 ( $s_{10}$ )
9	00***	B	00001 ( $s_8$ )
10	00***	A	10010 ( $s_7$ )
11	*****	E	10001 ( $s_1$ )
12	*****	C	01100 ( $s_{12}$ )
13	*****	B	00100 ( $s_6$ )
14	*****	*	10000 ( $s_0$ )

**DFA → CompactDFA**

Snort: 73MB → 0.6MB

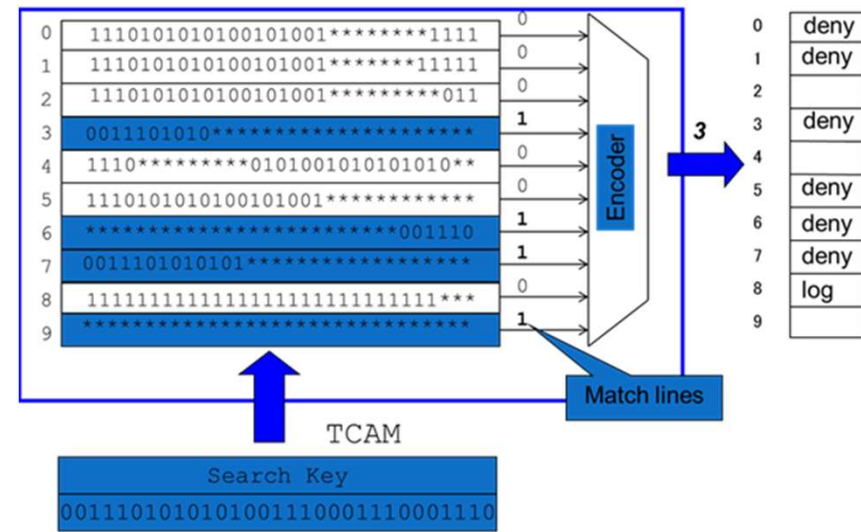
ClamAV: 1.5GB → 26MB

Observation :  
degree of freedom of encoding states name

# CompactDFA

- Reducing the problem of pattern matching to IP lookup
  - Longest prefix match
- Using TCAM to represent a huge DFA in a compact manner
- TCAM – Fully associative ternary memory
  - Common in today routers

	current	sym	next state
1	00000	C	01001 ( $s_5$ )
2	00101	C	01010 ( $s_3$ )
3	00101	B	00000 ( $s_4$ )
4	10001	B	00101 ( $s_2$ )
5	010**	D	11001 ( $s_{11}$ )
6	000**	A	11000 ( $s_9$ )
7	01***	F	11010 ( $s_{13}$ )
8	00***	C	01001 ( $s_{10}$ )
9	00***	B	00001 ( $s_8$ )
10	00***	A	10010 ( $s_7$ )
11	*****	E	10001 ( $s_1$ )
12	*****	C	01100 ( $s_{12}$ )
13	*****	B	00100 ( $s_6$ )
14	*****	*	10000 ( $s_0$ )



## Challenge #2: Compressed Traffic

- A. Bremler-Barr, Y. Koral " Accelerating Multi-patterns Matching on Compressed HTTP Traffic ", in IEEE/ACM Transaction on Networking 2011
- Yehuda Afek, Anat Bremler-Barr, Yaron Koral, "Efficient Processing of Multi-Connection Compressed Web Traffic", in Computer Communication 2012
- Michela Becchi, Anat Bremler-Barr, David Hay, Omer Kochba, Yaron Koral, "Accelerating Regular Expression Matching Over Compressed HTTP". In IEEE INFOCOM, April 2015

# Motivation: Compressed HTTP

- 76% of all the sites compress traffic.



- Goal: reduce Bandwidth !
- Data compression is done by adding references (pointers) to repeated data: GZIP (+Huffman)

## Yahoo Decompressed file:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en-US"><head><meta http-equiv=
"Content-Type" content="text/html; charset=UTF-8">
<script type="text/javascript">
var now=new Date,t1=t2=t3=t4=t5=t6=t7=t8=t9=t10=t11=t12=0,cc="
ylp=";t1=now.getTime();
</script>
```

## Yahoo LZ77 form:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD(26,6)4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<(20,4) lang="en-US(20,5)head(7,3)meta (73,4)-equiv=
"Content-Type" c(14,6)="text(92,5); charset=UTF-8(75,4)
script t(50,3){41,6}java(22,6){32,3}
var now=new Date,t1=t2=t3=t4=t5=t6=t7=t8=t9=t10(4,3){32,3}12=0,cc="
ylp(7,3);{54,3}{70,3}.getTime();
</(100,6)>
```

- Current security tools do not deal with compressed traffic due to the high challenges in **time** and **space**

# Compressed Traffic : Time Challenge

- **Need to decompress prior to pattern matching**
  - HTTP compression is an adaptive compression
    - The same string will be encoded differently depending on its location in the text
- General belief:

**Decompression + pattern matching  
>> pattern matching**

# Our solution: Accelerating DPI

- Compression is done by compressing repeated sequences of bytes, so store information about the pattern matching results
- No need to fully perform *again* pattern matching on repeated sequences which were already scanned
- x 2-3 time reduction

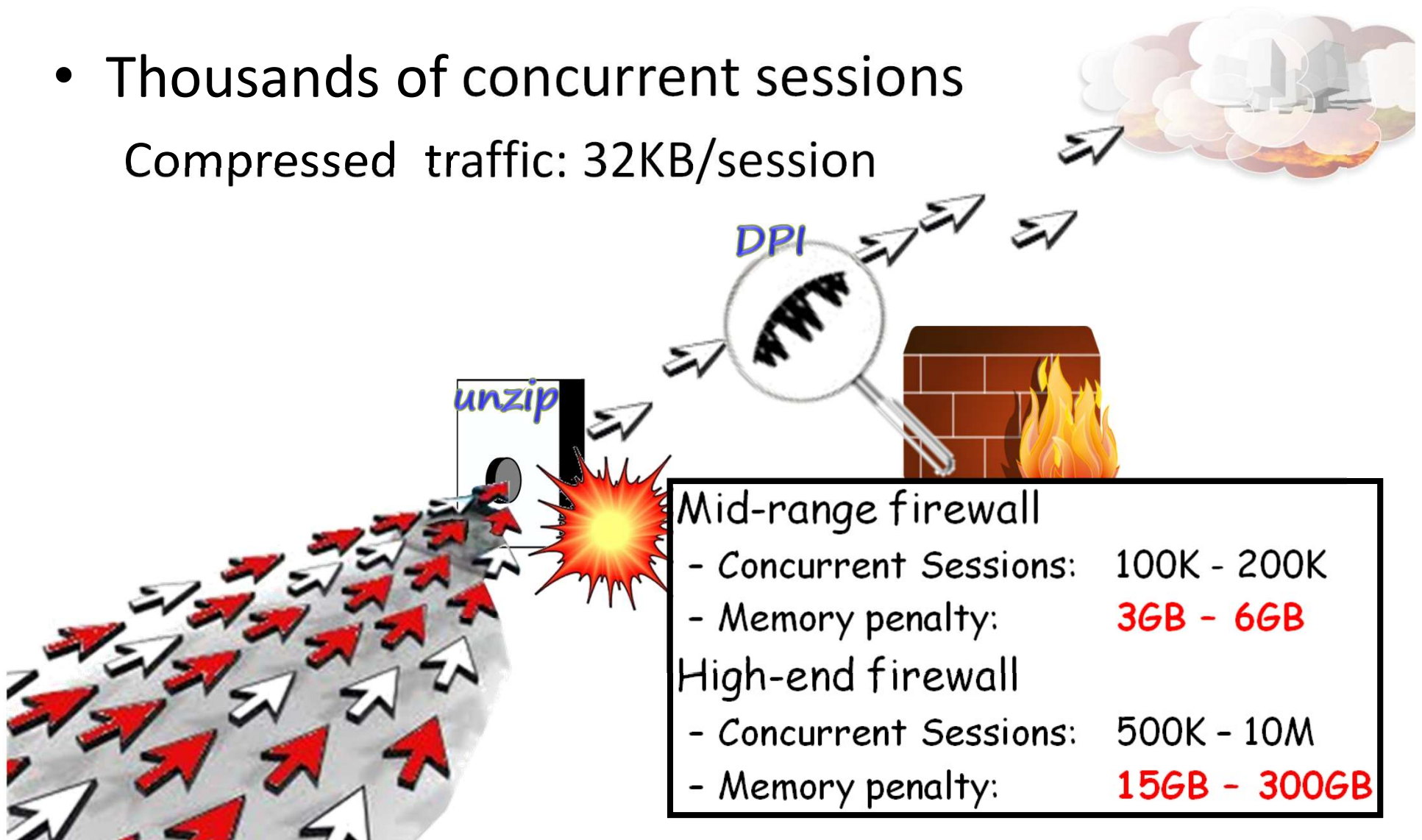
**Decompression + pattern matching  
<< pattern matching**

- We also dealt with regular expression [Infocom 2015]
- We also dealt with SDCH [Infocom 2012]



# Compressed Traffic : Space Challenge

- Thousands of concurrent sessions  
Compressed traffic: 32KB/session



# Our solution: Space Reduction

- Observation: The 32KB needed for decompression are not used most of the time
- Key idea: therefore the 32KB can be kept in compressed form most of the time
  - Some light version on the compressed form of the traffic
- x5 space reduction
- Overall: improve space by 80% and Time by 40%

# The Other Side of the Coin: Acceleration by Identifying Repetitions in Uncompressed Traffic

There are repetitions in uncompressed HTTP traffic

- Entire files (e.g., images)
- Parts of the files (e.g., HTML tags, javascripts)

→ We keep scanning again and again the same thing (and get the same scanning results..)

1. Identify frequently repeated data
2. Perform DPI on the data **once** and remember the results
3. When encountering a repetition, recover the state without re-scanning

Anat Bremler-Barr, Shimrit Tzur David, Yotam Harchol, and David Hay: “Leveraging Traffic Repetitions for High-Speed Deep Packet Inspection”. Infocom, 2015

## Challenge #3: Securing the DPI

- Yehuda Afek, Anat Bremler-Barr, Yotam Harchol, David Hay, Yaron Koral, "Making DPI Engines Resilient to Algorithmic Complexity Attacks". In IEEE/ACM Transactions on Networking, Volume 24, Issue 6, 2016

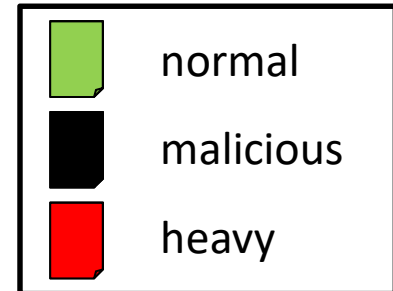
# Complexity DoS Attack Over NIDS

- Regular operation
- 2 Steps attack:

Attacker



Internet



1. Kill IPS/FW






2. Launch original attack  
(e.g., steal credit cards)

# Complexity DDoS Attack Over IDS

- Easy to craft – very hard to process packets
- 2 Steps attack:



Internet

	normal
	malicious
	heavy

1. Kill IPS/FW



2. Sneak into the network

# Attack on Security Elements

The screenshot shows the eWEEK.com website with a navigation bar at the top. The main headline is "IT Security & Network Security News & Reviews". The featured article is titled "Sony Data Breach Was Camouflaged by Anonymous DDoS Attack" by Fahmida Y. Rashid, dated 2011-05-05. The article text describes how a distributed denial of service (DDoS) attack against Sony's PlayStation Network and Qriocity music service masked network intrusions and compromised 101 million user accounts. It mentions that Sony didn't notice the security breaches because it was distracted by the DDoS attacks. Several Sony divisions had been hit by a large-scale coordinated denial-of-service attack in early April from the Anonymous hacker collective protesting the company's lawsuit against George Hotz, a PlayStation 3 hacker. Sony wrote in a letter to the United States House of Representatives on May 4 that the combined effect of the DDoS attack and the sophisticated methods used by the cyber-thieves made it difficult for Sony's administrators to detect the data breach, according to the letter. Sony disclosed on April 26 that thieves had stolen account information of up to 77 million users on the PlayStation Network and Qriocity. A week later, the company admitted on May 2 that the Sony Online Entertainment gaming service had also been breached, affecting an additional 24.6 million users. About 101 million user accounts have been compromised to date. The stolen data included names, addresses, email addresses, dates of birth. Some credit card information may have been stolen, but Sony claimed the numbers were securely saved as a cryptographic hash.

On the left side of the page, there is a sidebar with the heading "Your News. Anytime. Anywhere." and an image of a smartphone displaying the eWEEK mobile app. Below this, it says "Get the latest Enterprise news on any device with our new mobile apps" and a "Learn More" button. At the bottom left, there is an "AdChoices" link and a "Password Encrypted Email" link with the text "exchange email via encrypted SMTP service".

On the right side, there is a "Today's Featured Video" section titled "Accelerating Big Data Analysis with In Memory Technology" with a "Watch Now" button. Below this is a "Newest Videos" section with a grid of video thumbnails. At the bottom right, there is a "Suggested Related Content" section.

Combined Attack:  
DDoS on Security Element  
exposed the network –  
theft of customers'  
information



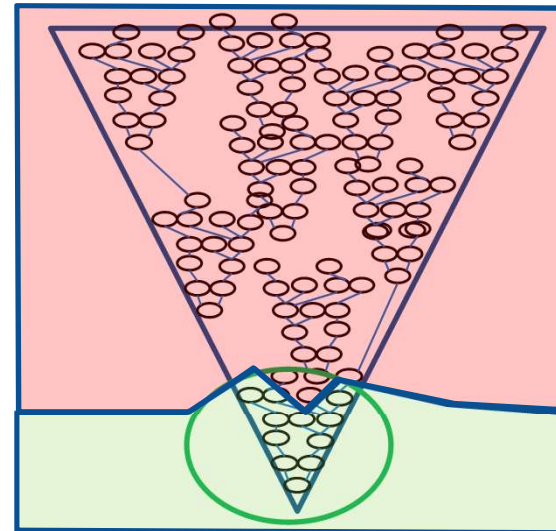
# Deep Packet Inspection: the environment

High Capacity  
Slow Memory



Locality-based  
Low Capacity  
Fast Memory

Cache  
Memory



In reality, in *security* network function ,*most memory accesses* are done to the cache.

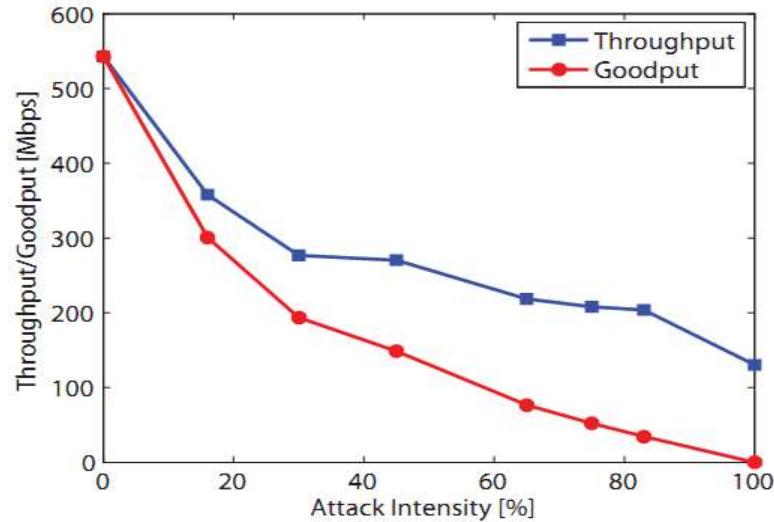
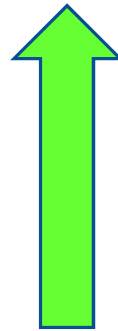
**BUT**

One can *attack* the implementation by reducing its locality, getting it out of cache - and making it *much slower!*





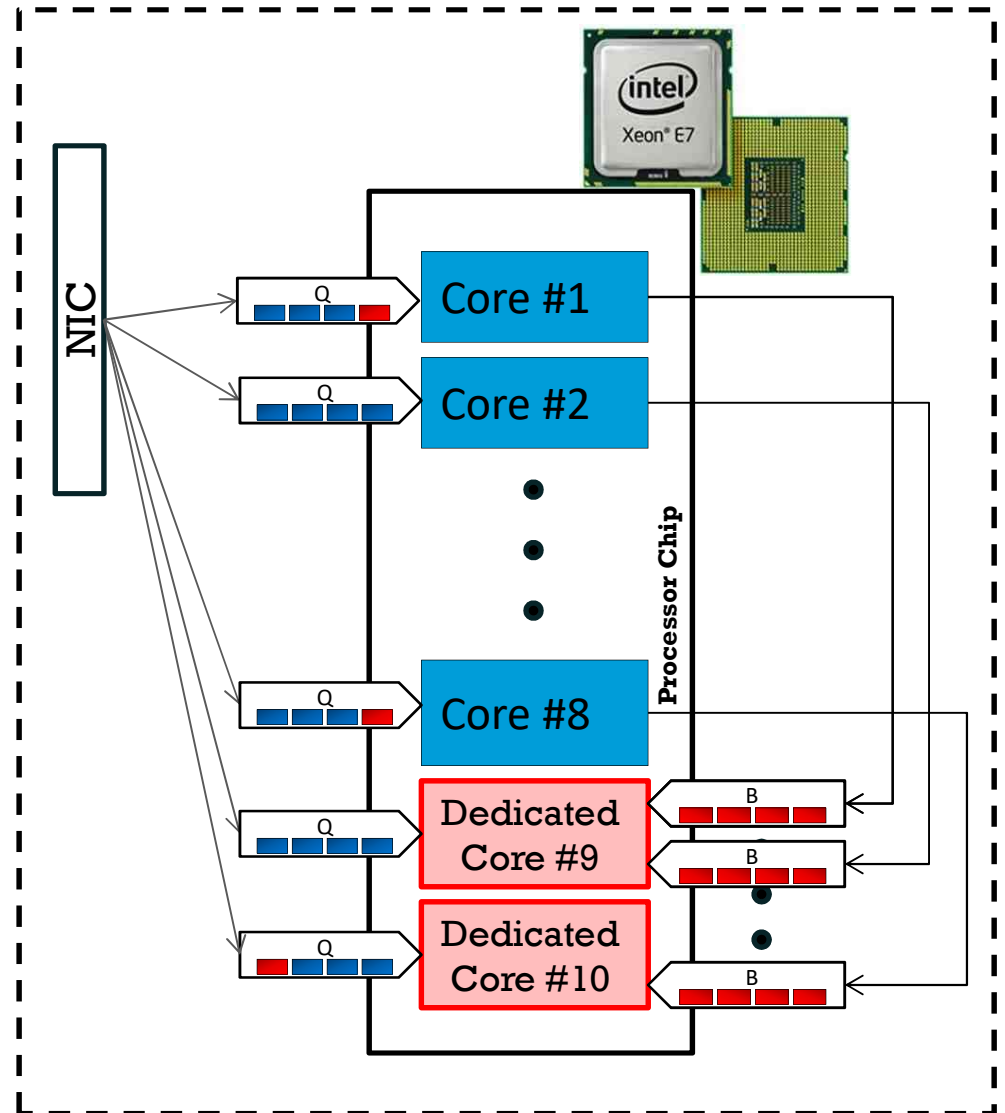
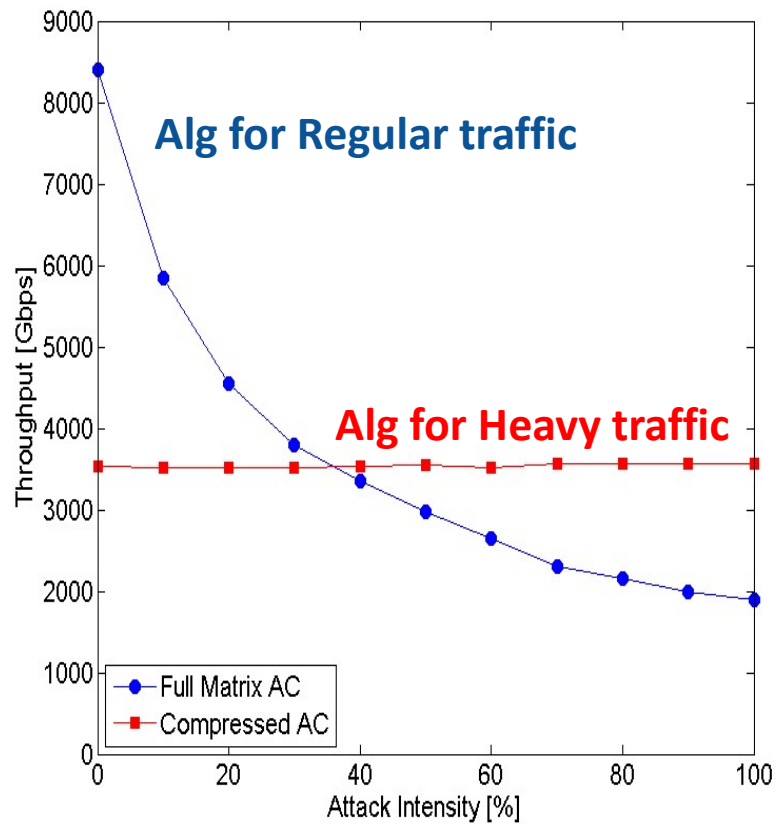
# Attack on Snort



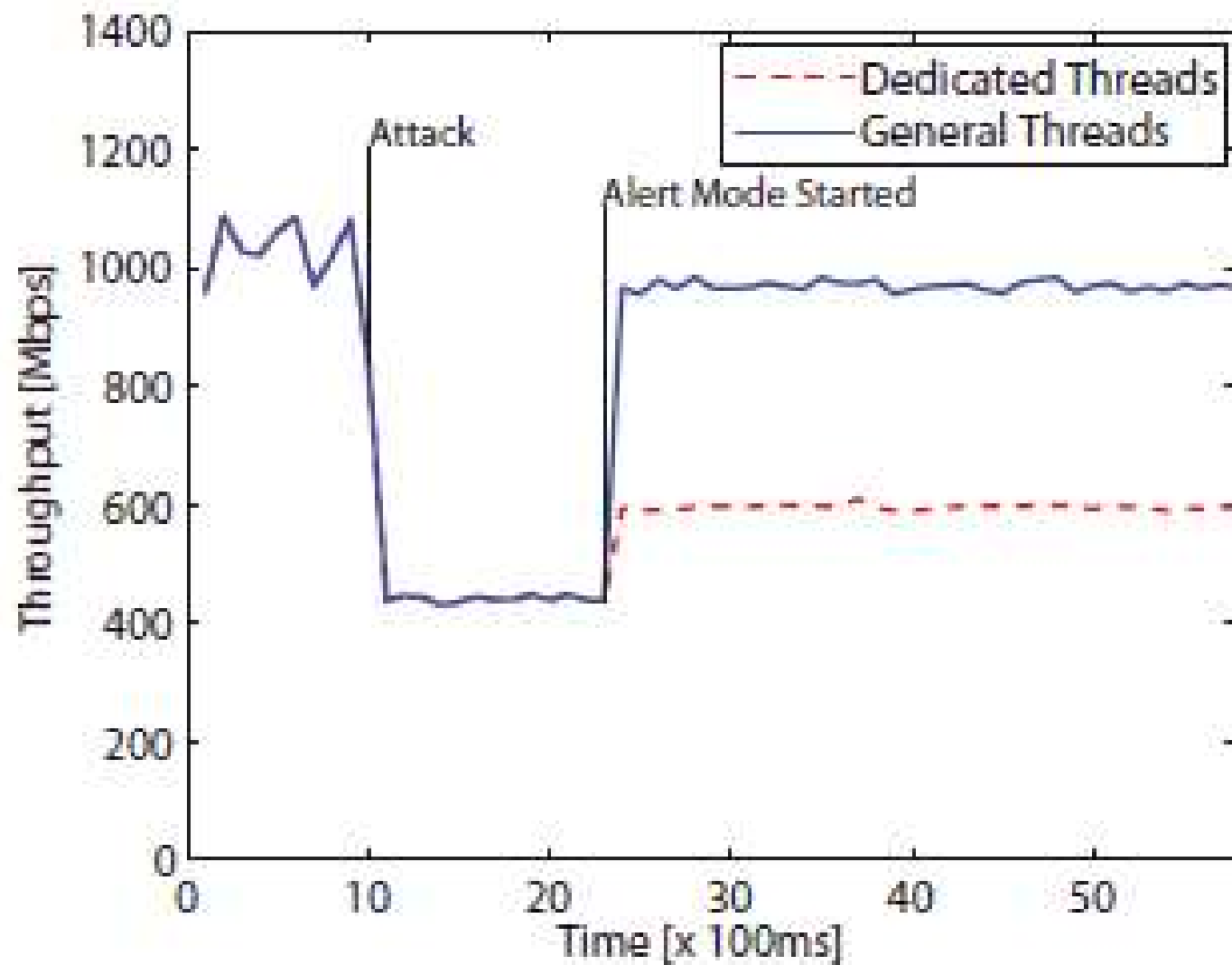
Heavy packets rate

We present a **multi-core (multi- VMs)** system architecture, which is **robust** against **complexity DDoS attacks**

# Solution Outline



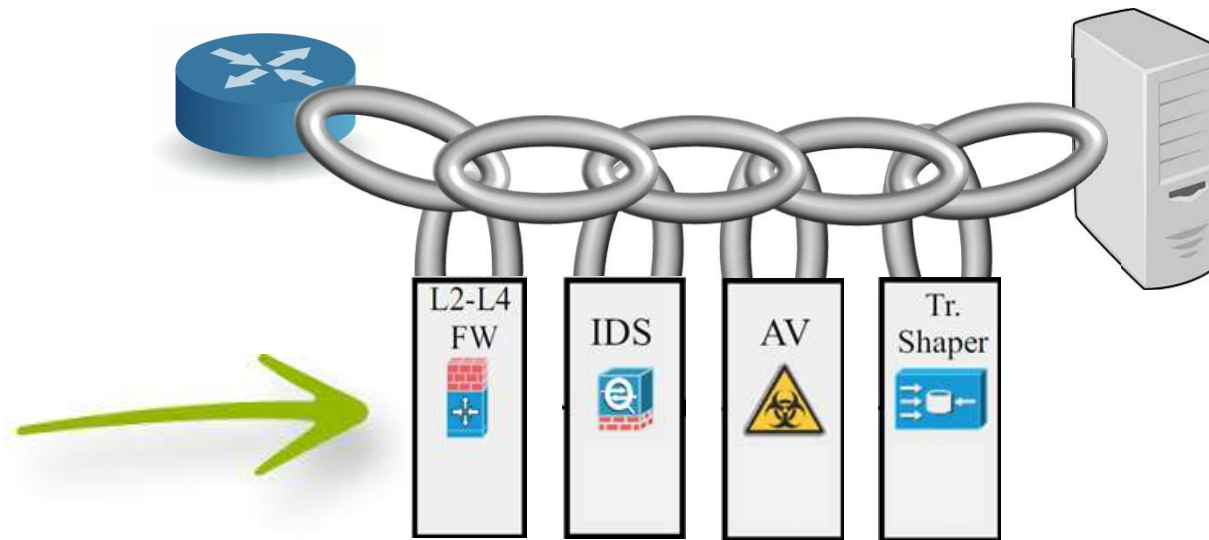
# System Throughput Over Time



# DPI in SDN & NFV

- Anat Bremler-Barr, Yotam Harchol, David Hay, Yaron Koral, "Deep Packet Inspection as a Service". in ACM CoNEXT, 2014
- Anat Bremler-Barr, Yotam Harchol, David Hay, "OpenBox: A Software-Defined Framework for Developing, Deploying, and Managing Network Functions", in SIGCOMM, 2016

# Network Function Service Chains



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

# Our Solution: DPI as a Service

## Contribution:

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

## Benefits:

- **Innovation** – Lower entry barriers
- **Reduced costs** – Cheaper NF HW/SW
- **Improved performance** - Scan each packet **once**  
Beneficial - time requirement is sub linear with #patterns
- **Rich DPI functionality** – Invest once for all NF

# Solution Outline

- Architecture aspects of DPI as a service

- DPI Instance

- One or multiple DPI instances



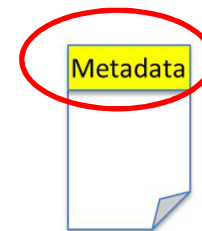
- DPI controller

- Received the patterns sets from all the NFs
    - Divide the patterns to different sets of DPI instances

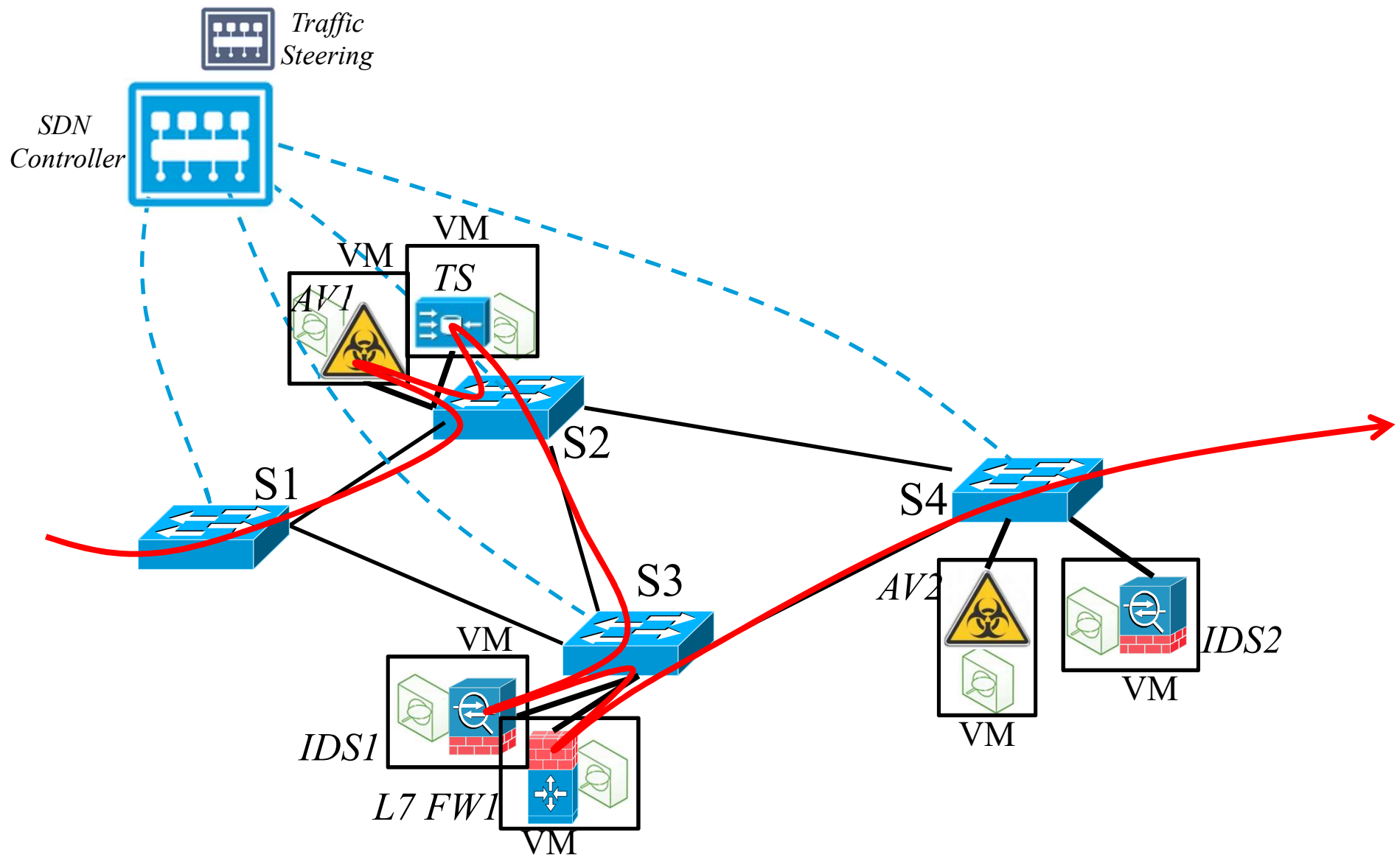


- Mechanism for passing results from the DPI to the NFs:

- Network Service Header (NSH)



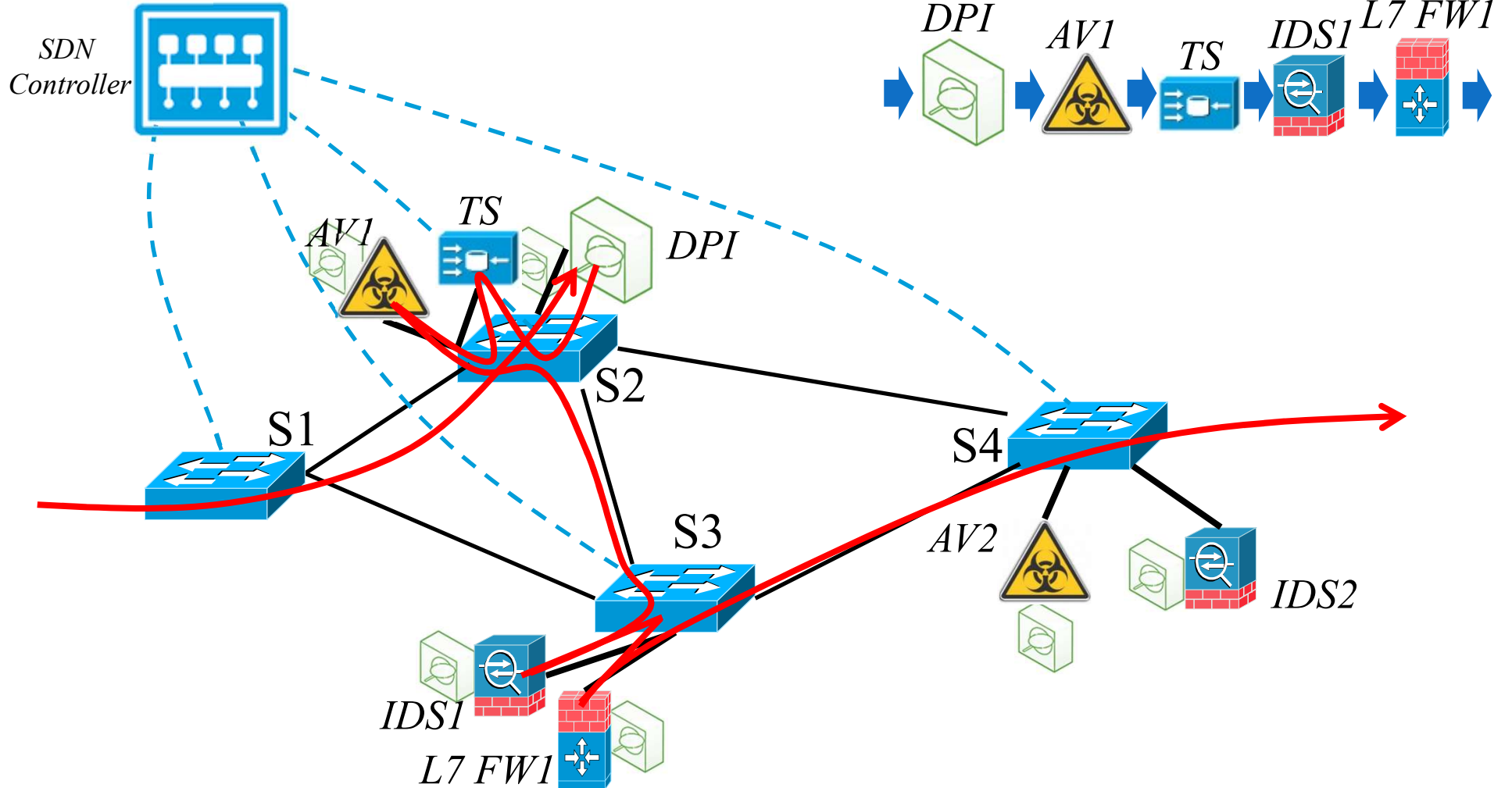
# Service chain of NFs in NFV





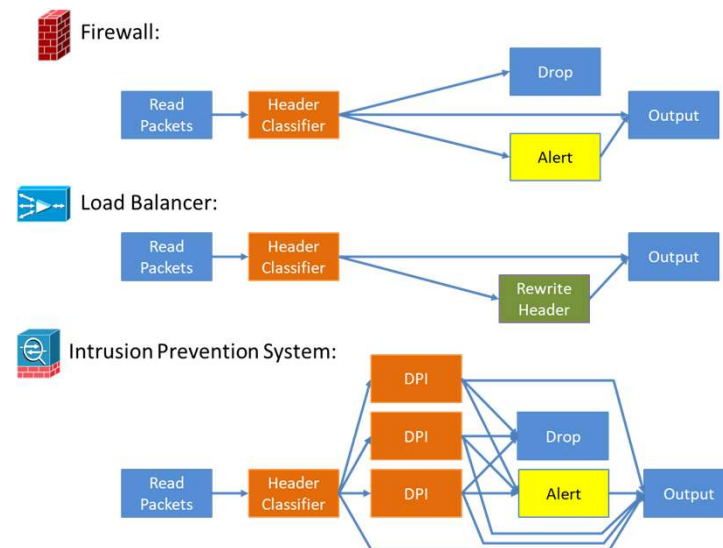
# DPI as a Service

DPI Controller Traffic Steering



# Observation

- Most network functions do very similar processing steps (DPI, Header Classifier...)  
But there is no re-use...
- OpenBox [sigcomm 2016] framework is based on this observation



# OpenBox



[www.openboxproject.org](http://www.openboxproject.org)

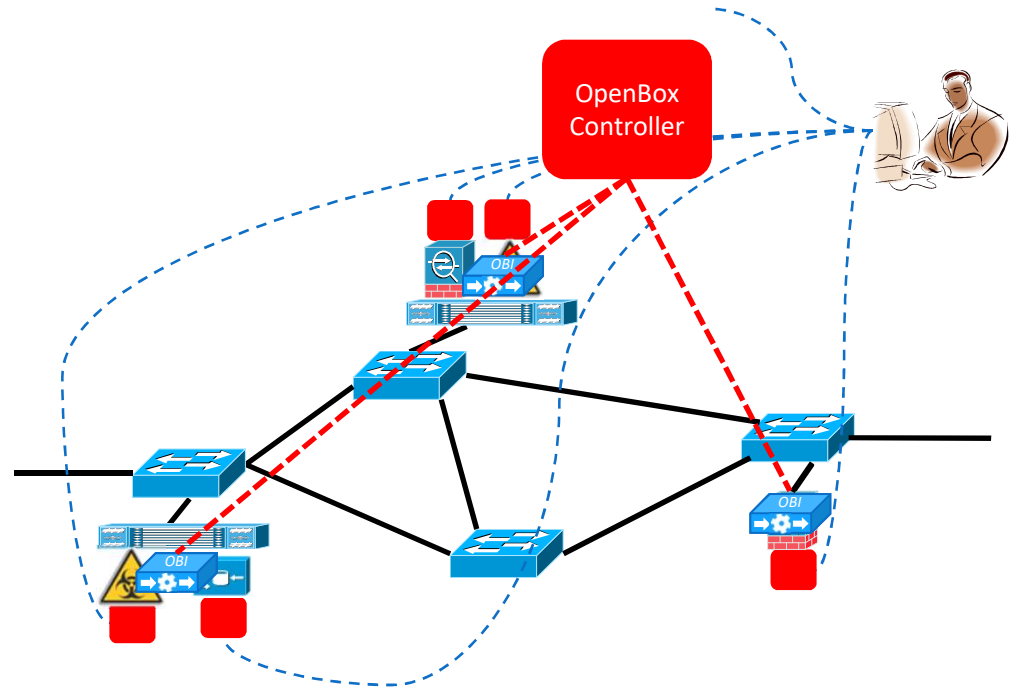


[github.com/OpenBoxProject](https://github.com/OpenBoxProject)

- **OpenBox: A new software-defined framework for network functions**
- Decouples network function control from their data plane
- OpenBox Instances (OBI): Data plane entities (e.g. DPI, packet classification)
- OpenBox Controller : Logically centralized control plane
- NFs are written as OpenBox applications on top of OpenBox Controller using north bound programming API

## Benefits:

- ✓ Easier, unified control
- ✓ Better performance
- ✓ Scalability
- ✓ Flexible deployment
- ✓ Inter-tenant isolation
- ✓ Innovation



# DPI: Conclusion

- Evolving area
- SDN & NFV change the field of Network Function and among other the DPI area

**Thank You!!!**

