SCIENCE DMZ: INTRODUCTION, CHALLENGES, AND OPPORTUNITIES

Jorge Crichigno
College of Engineering and Computing
University of South Carolina

Presentation at John Hopcroft Center for Computer Science
Shanghai Jiao Tong University (SJTU)
May 20, 2019

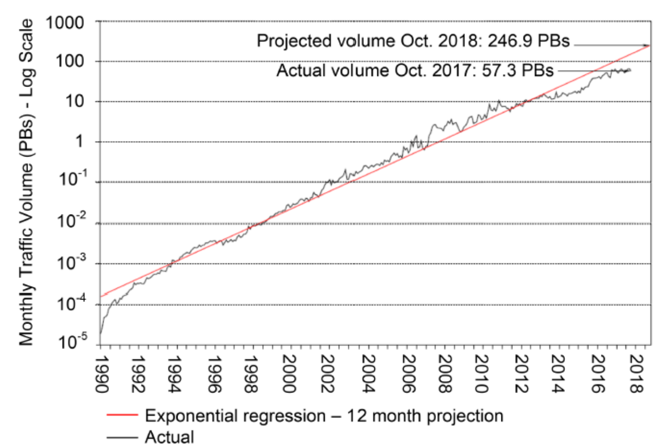
Agenda

- Motivation for a high-speed ‘science’ network architecture
- The Science DMZ
- Research opportunities
  - Enabling pacing using P4 switches (work in progress)
  - Entropy-based intrusion detection system (IEEE ICC 2019)

Motivation for a High-Speed Science Architecture

- Science and engineering applications are now generating data at an unprecedented rate
- From large facilities to portable devices, instruments can produce hundreds of terabytes in short periods of time
- Data must be typically transferred across high-throughput high-latency Wide Area Networks (WANs)

Applications

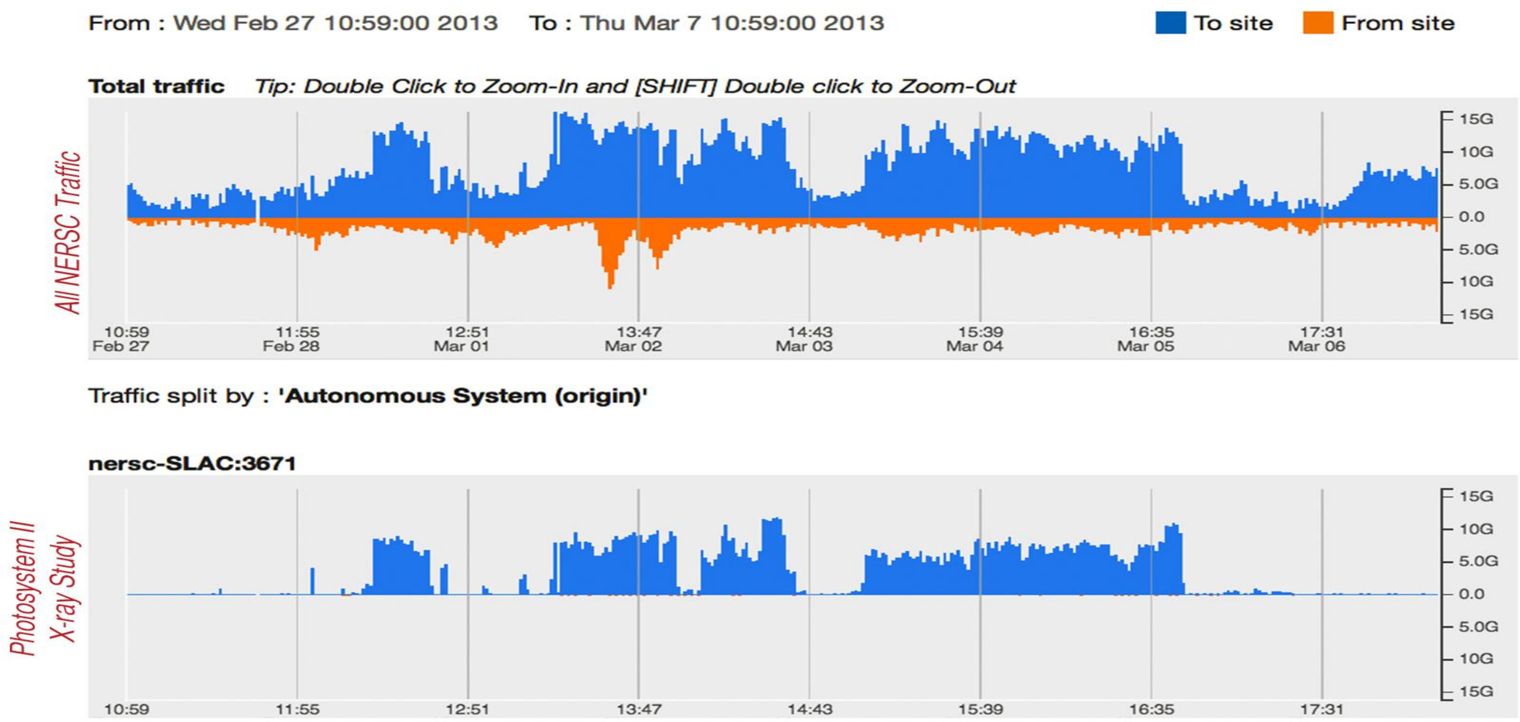


ESnet traffic

The Energy Science Network (ESnet) is the backbone connecting U.S. national laboratories and research centers

Motivation for a High-Speed Science Architecture

- A biology experiment using the U.S. National Energy Research Scientific Computing Center (NERSC) resources



Motivation for a High-Speed Science Architecture

- A biology experiment using the U.S. National Energy Research Scientific Computing Center (NERSC) resources

SnapChat Data produced per day worldwide by millions of people

= 38 TB

One Biology experiment by a team of nine scientists:

= 114 TB

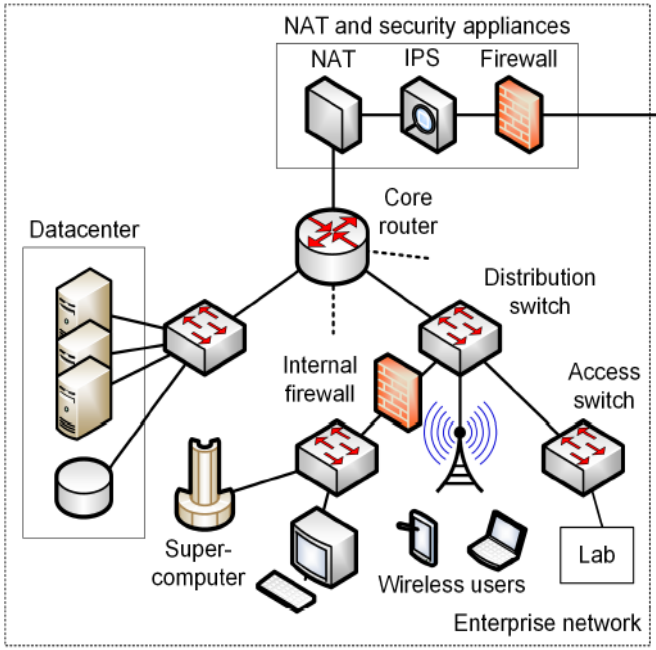
(Photosystem II X-Ray Study)

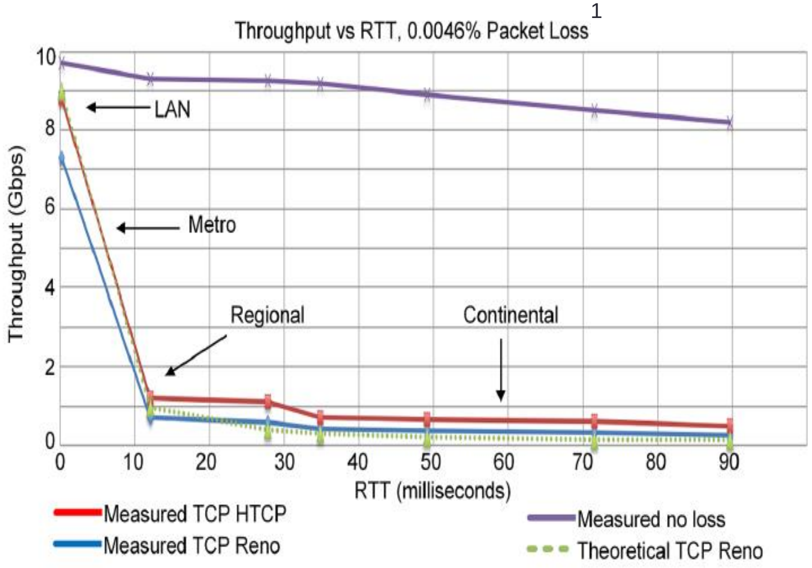
http://www.nature.com/articles/ncomms5371

Motivation for a High-Speed Science Architecture

Enterprise network limitations:

- Security appliances (IPS, firewalls, etc.) are CPU-intensive
- Inability of small-buffer routers/switches to absorb traffic bursts
- At best, transfers of big data may last days or even weeks



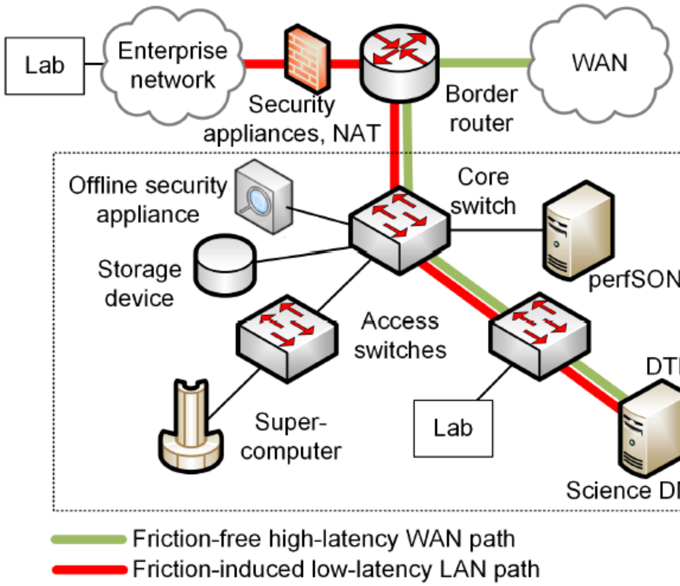


Two devices exchanging data on a 10 Gbps network
Packet loss rate is 1/22,000, or 0.0046%

[1]E. Dart, L. Rotman, B. Tierney, M. Hester, J. Zurawski, "The science dmz: a network design pattern for data-intensive science," *International Conference on High Performance Computing, Networking, Storage and Analysis*, Nov. 2013.

Science DMZ

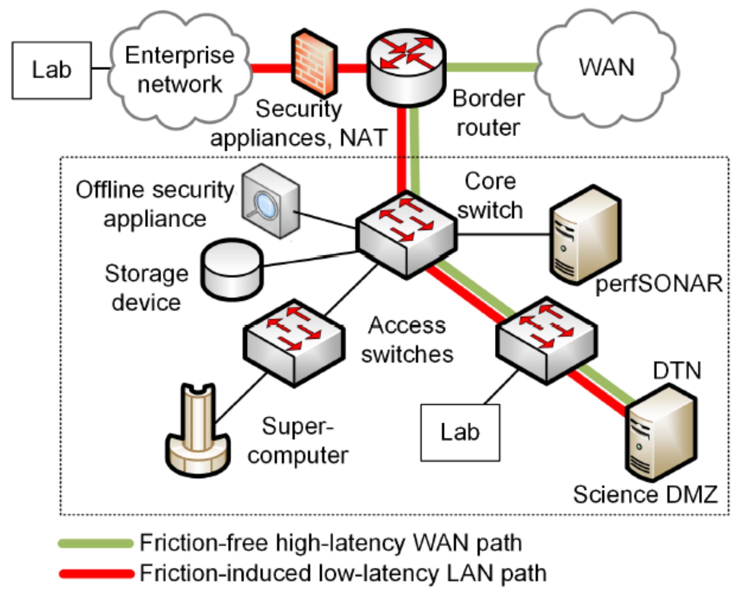
- The Science DMZ is a network designed for big science data[1,2]
- Main elements
  - High throughput, friction free WAN paths (no inline security appliances; routers / switches w/ large buffer size)
  - Data Transfer Nodes (DTNs)
  - End-to-end monitoring = perfSONAR
  - Security = Access-control list + offline appliance/s (IDS)

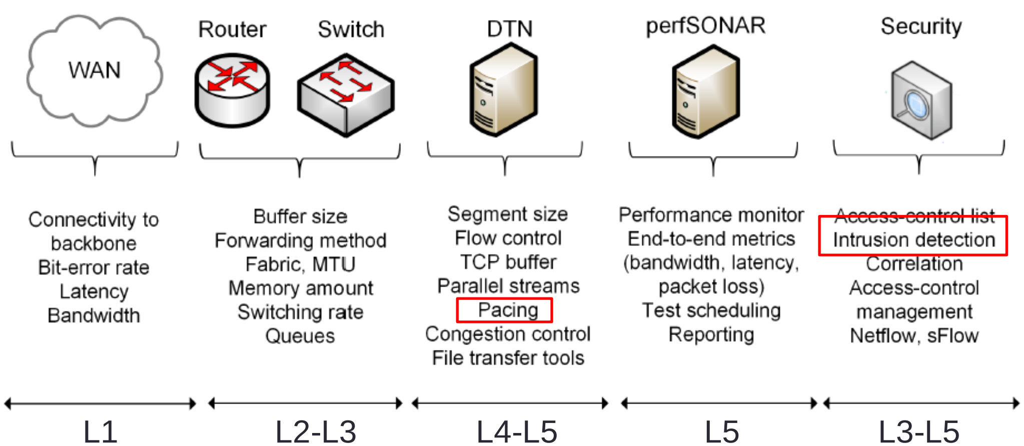


[1]E. Dart, L. Rotman, B. Tierney, M. Hester, J. Zurawski, "The science dmz: a network design pattern for data-intensive science," *International Conference on High Performance Computing, Networking, Storage and Analysis*, Nov. 2013.
[2]J. Crichigno, E. Bou-Harb, N. Ghani, "A comprehensive tutorial on science DMZ," IEEE Communications Surveys and Tutorials, to appear 2nd quarter issue, 2019.

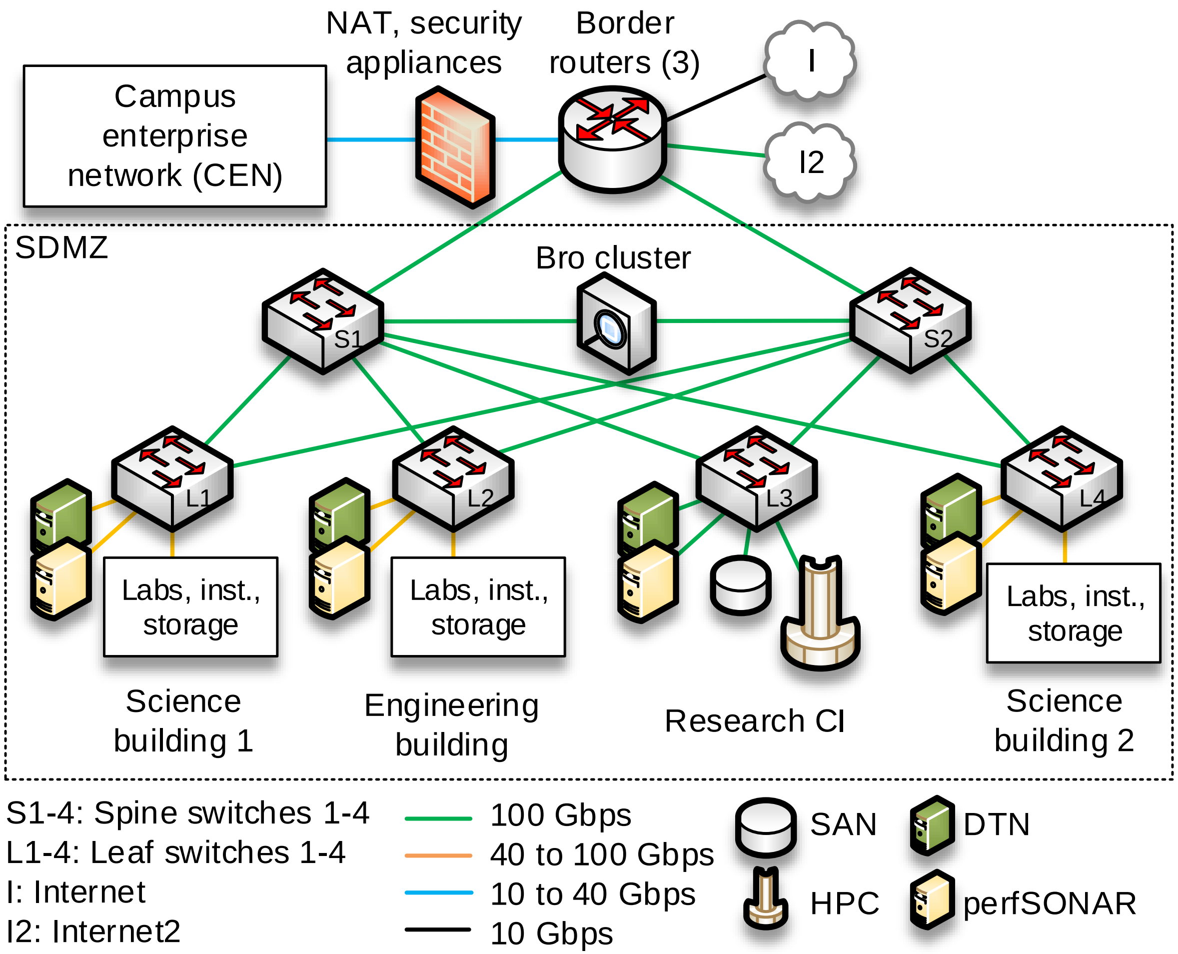
Science DMZ

- The Science DMZ is a network designed for big science data
- Main elements
  - High throughput, friction free WAN paths (no inline security appliances; routers / switches w/ large buffer size)
  - Data Transfer Nodes (DTNs)
  - End-to-end monitoring = perfSONAR
  - Security = Access-control list + offline appliance/s (IDS)





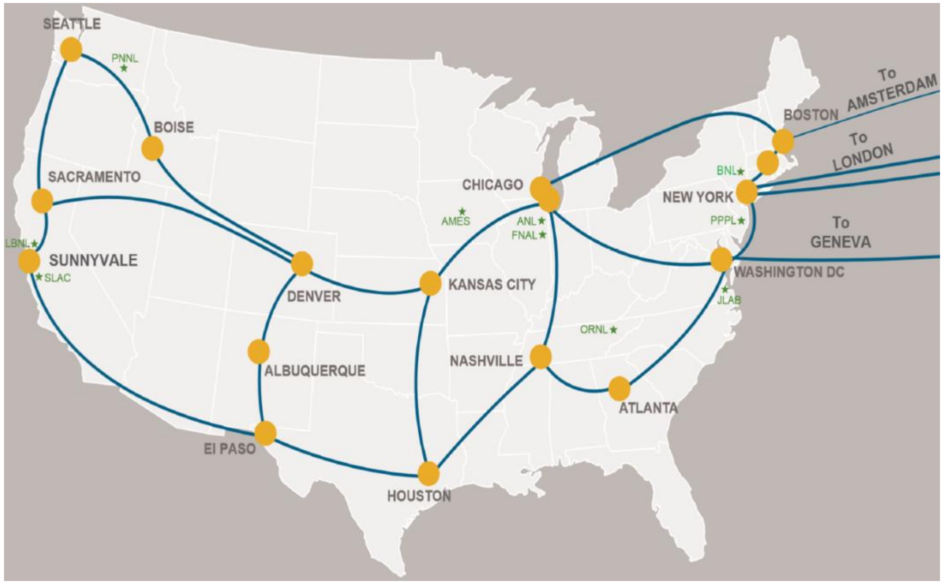
USC's Science DMZ



U.S. Backbones: Internet2 and ESnet

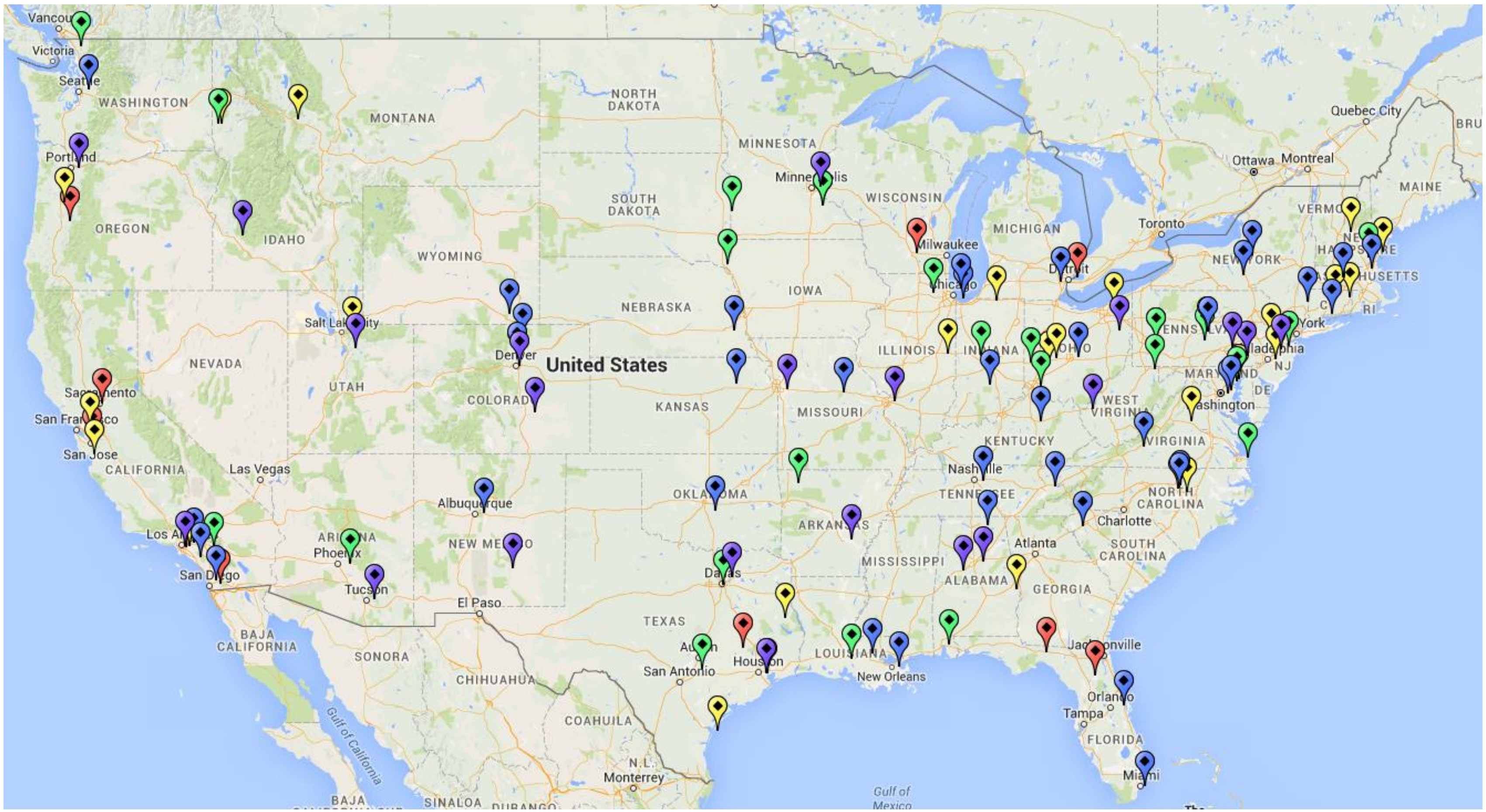
Internet2

ESnet



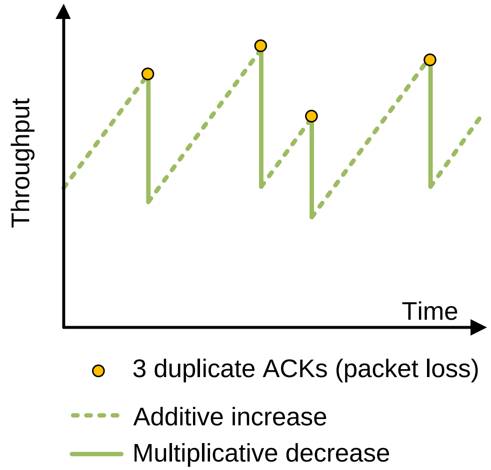
Science DMZs in the U.S.

- Science DMZ deployments, U.S.

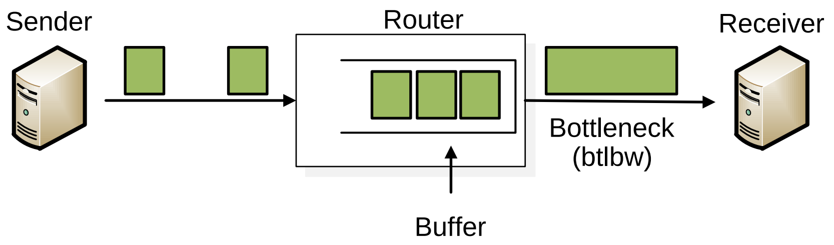


Research Opportunities – Pacing

- Packet loss is expensive in high-throughput high-latency networks



(a) Sawtooth behavior



(b) TCP view of a connection

$$\text{TCP throughput} = \frac{c \cdot MSS}{RTT \cdot \sqrt{p}}$$

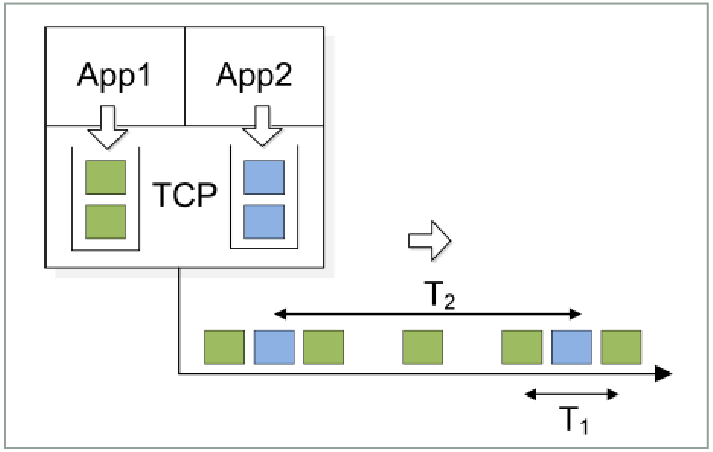
MSS: maximum segment size
RTT: round-trip time
p: loss rate
c: constant

(c) Average throughput

M. Mathis, J. Semke, J. Mahdavi, T. Ott, "The macroscopic behavior of the tcp congestion avoidance algorithm," *ACM Computer Communication Review*, vol. 27, no 3, pp. 67-82, Jul. 1997.

Pacing

- Pacing is a technique by which a transmitter evenly spaces or paces packets at a pre-configured rate
- If the network bottleneck is known, end devices can be set to transfer at a pacing rate rather than ‘discovering’ the rate
- Pacing also helps to mitigate packet bursts

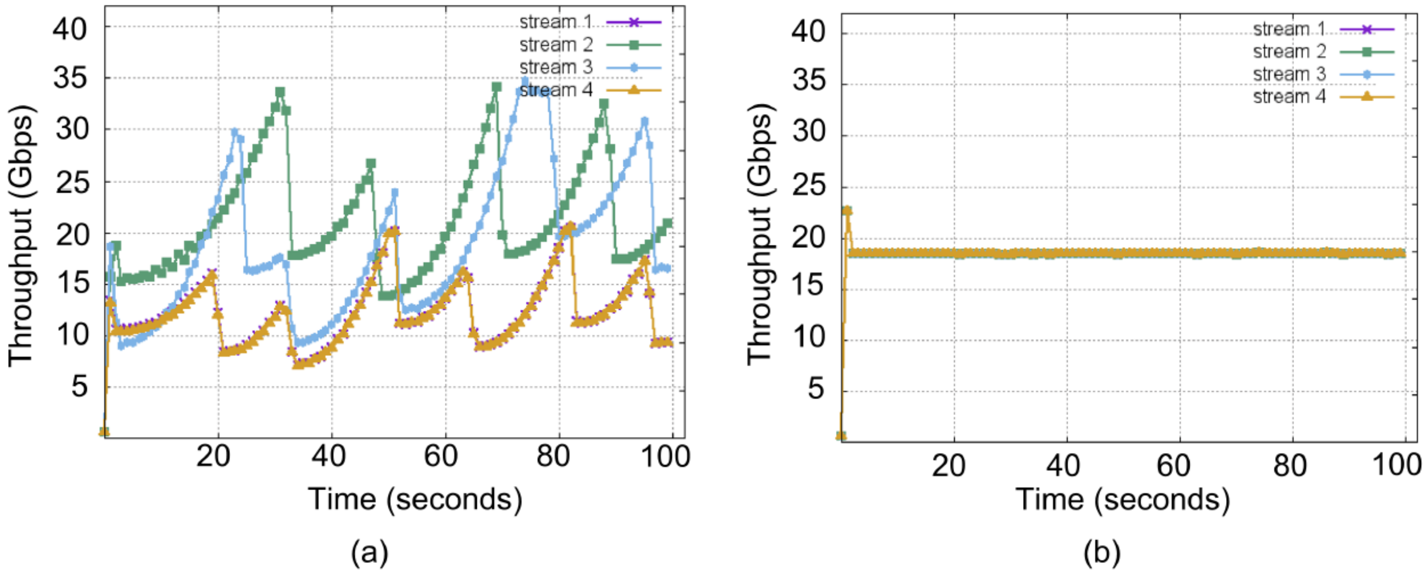


Pacing

Consider tests over ESnet backbone[1]

Four flows on a 100 Gbps network

- "Consistent loss on the network with four streams, no pacing…"
- "Pacing to match bottleneck link works better yet…"
- ESnet approach requires the network operator to statically set the pacing rate, based on the number of big flows



1. https://meetings.internet2.edu/media/medialibrary/2016/10/24/20160927-tierney-improving-performance-40G-100G-data-transfer-nodes.pdf

ENABLING TCP PACING USING PROGRAMMABLE DATA PLANE SWITCHES

E. Kfoury, Jorge Crichigno
College of Engineering and Computing
University of South Carolina

Overview P4 Switches

- P4 is a programming language for switches
- SDN is used to program the control plane
- P4 switches permit operators to program the data plane

  Add proprietary features: invent, *develop custom protocols*
- USC partnered with Barefoot Networks to use Tofino's chip to develop custom protocols

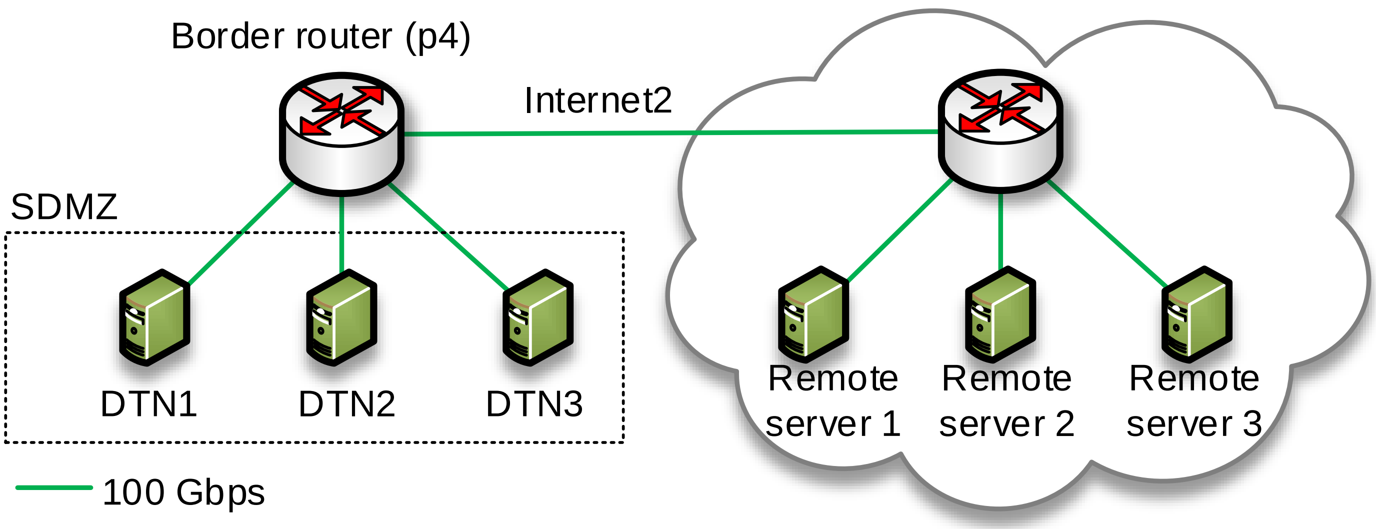
```
/*************************************************************************
*********************** P A R S E R  ***********************************
*************************************************************************/

    state parse_ethernet {
        packet.extract(hdr.ethernet);
        transition select(hdr.ethernet.etherType) {
            TYPE_IPV4: parse_ipv4;
            default: accept;
        }
    }

    state parse_ipv4 {
        packet.extract(hdr.ipv4);
        verify(hdr.ipv4.ihl >= 5, error.IPHeaderTooShort);
        transition select(hdr.ipv4.ihl) {
            5           : accept;
            default     : parse_ipv4_option;
        }
    }
```

P4 code

Barefoot's Tofino (2016)

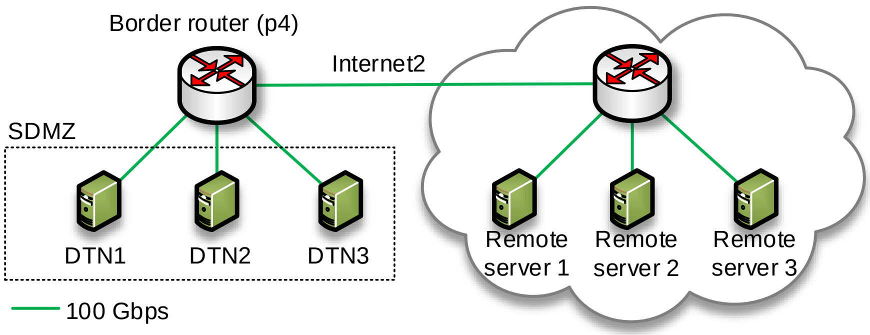
Pacing via P4-Switches

- What if the rate at a sender node is adjusted based on feedback provided by a P4 switch?
- Feedback includes number of large flows and more

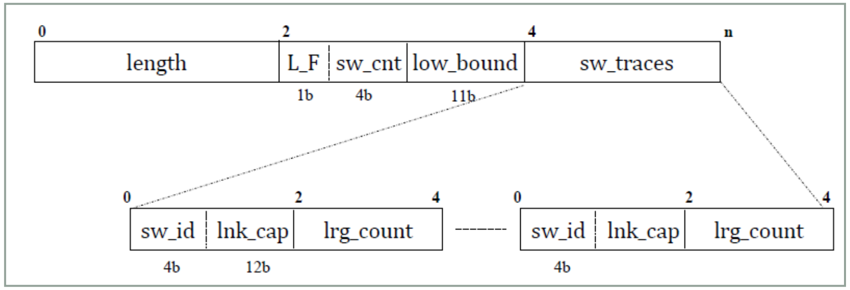


Pacing via P4-Switches

- Switches store network's state (number of large flows)
- To initiate a large flow, a DTN inserts a custom header during the TCP 3-way handshake, using the IP options field
- Switches parse custom header, update number of large flows
- Number of large flows is returned in the SYN-ACK message, and sent to all DTNs. DTNs update their *pacing* rate



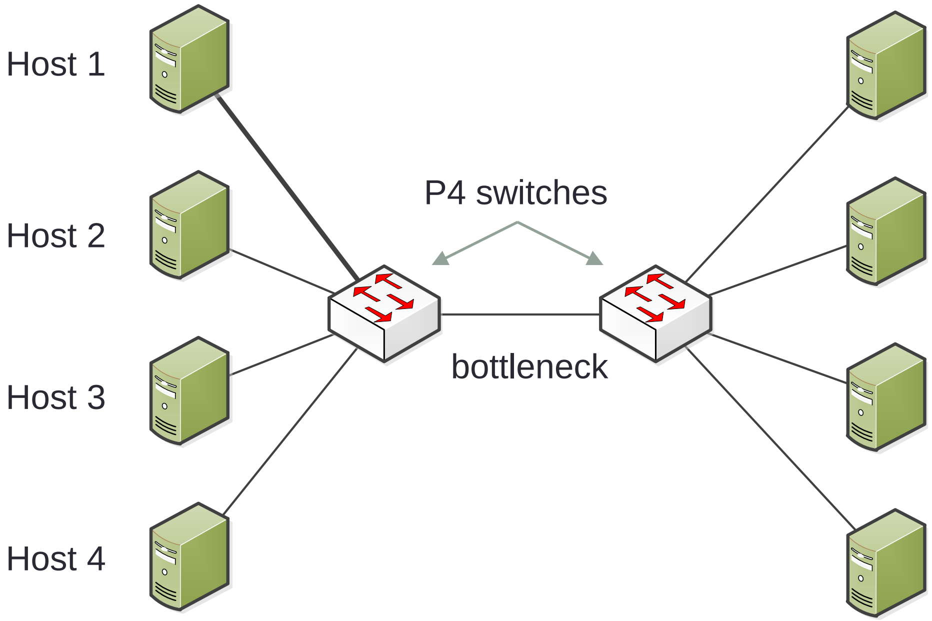
Sample topology



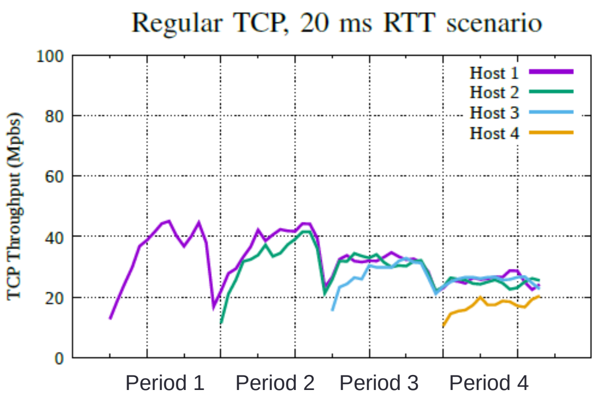
Custom protocol built using IP options field

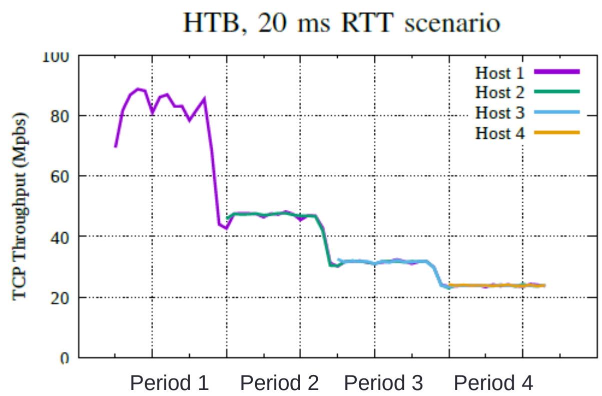
Emulation Results

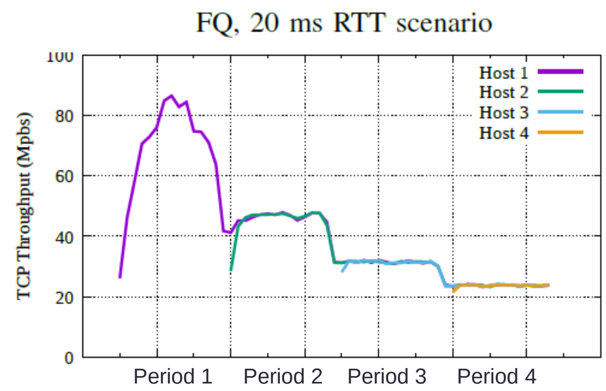
- The custom protocol was implemented in Mininet
- The P4 switch is the BMv2 from P4.org
- Four hosts (DTNs) generating flows; 100 Mbps, 20ms RTT
- Hosts adjusted their pacing rate using two pacing disciplines
  - Fair Queue (FQ)
  - Hierarchical Token Bucket (HTB)



Emulation Results







Throughput

	Regular TCP					**HTB**					**FQ**				
Period	$\sum T_i$	T_1	T_2	T_3	T_4	$\sum T_i$	T_1	T_2	T_3	T_4	$\sum T_i$	T_1	T_2	T_3	T_4
P_1 (01-15 sec)	**33.62**	33.62	N/A	N/A	N/A	**81.25**	81.25	N/A	N/A	N/A	**66.59**	66.59	N/A	N/A	N/A
P_2 (16-30 sec)	**67.27**	36.06	31.21	N/A	N/A	**93.1**	46.40	46.70	N/A	N/A	**89.91**	45.85	44.06	N/A	N/A
P_3 (31-45 sec)	**88.83**	31.27	30.61	26.95	N/A	**94.42**	31.40	31.37	31.65	N/A	**93.72**	31.40	31.36	30.96	N/A
P_4 (46-60 sec)	**91.86**	25.32	24.63	25.32	16.59	**95.12**	23.78	23.75	23.73	23.86	**94.52**	23.71	23.71	23.67	23.43

Coefficient of variation and Jain's fairness

	Regular TCP					**HTB**					**FQ**				
Period	**F**	CV_1	CV_2	CV_3	CV_4	**F**	CV_1	CV_2	CV_3	CV_4	**F**	CV_1	CV_2	CV_3	CV_4
P_1 (01-15 sec)	1.00	32.32	N/A	N/A	N/A	1.0000	8.188	N/A	N/A	N/A	1.0000	28.427	N/A	N/A	N/A
P_2 (16-30 sec)	.994	22.63	30.08	N/A	N/A	.99998	3.773	2.998	N/A	N/A	.99960	4.351	14.142	N/A	N/A
P_3 (31-45 sec)	.994	9.349	10.90	19.69	N/A	.99998	2.065	2.081	1.985	N/A	.99960	1.618	1.317	3.879	N/A
P_4 (46-60 sec)	.974	7.806	5.260	6.447	17.27	.99999	1.168	1.138	.755	.684	.99997	1.022	1.020	.996	3.336

Work in progress

- Implement proposed protocol using a real P4 switched network
- Support for more complex topologies
- Extend the sharing bandwidth scheme for scenarios where an uneven allocation is desirable (priorities)
- Use proposed protocol in the production Science DMZ at USC

A FLOW-BASED ENTROPY CHARACTERIZATION OF A NATED NETWORK AND ITS APPLICATION ON INTRUSION DETECTION

Jorge Crichigno
College of Engineering and Computing
University of South Carolina

IEEE International Conference on Communications (ICC)
Shanghai, China
May 22, 2019

Agenda

- Motivation flow-based intrusion detection systems (IDSs)
- Overview of campus NATed networks
- Entropy of flow tuples
- Characterization of a campus enterprise network
- Conclusion

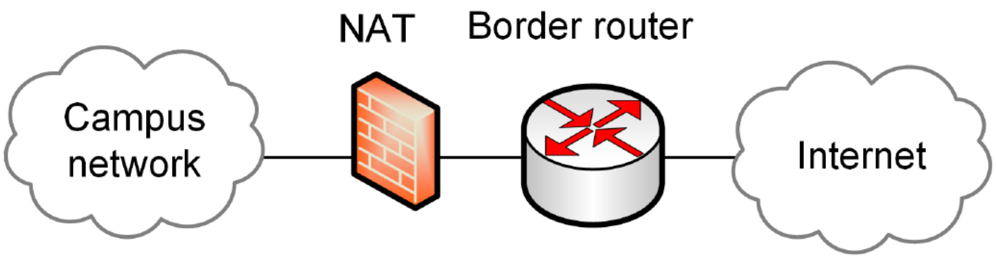
Motivation

- Offline scalable security appliances are required in high-speed networks such as Science DMZs
- There are two approaches to characterize traffic:
  - Flow-based: information collected from header fields
  - Payload-based: information collected from payload (deep inspection)
- The amount of processing of payload-based approaches may become excessive at very high rates[1, 2]

1. R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, A. Pras, “Flow monitoring explained: from packet capture to data analysis with netFlow and ipfix,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, 2014.
2. A. Gonzalez, J. Leigh, S. Peisert, B. Tierney, A. Lee, J. Schopf, “Monitoring big data transfers over international research network connections,” in *Proceedings of the IEEE International Congress on Big Data,*, Jun. 2017.

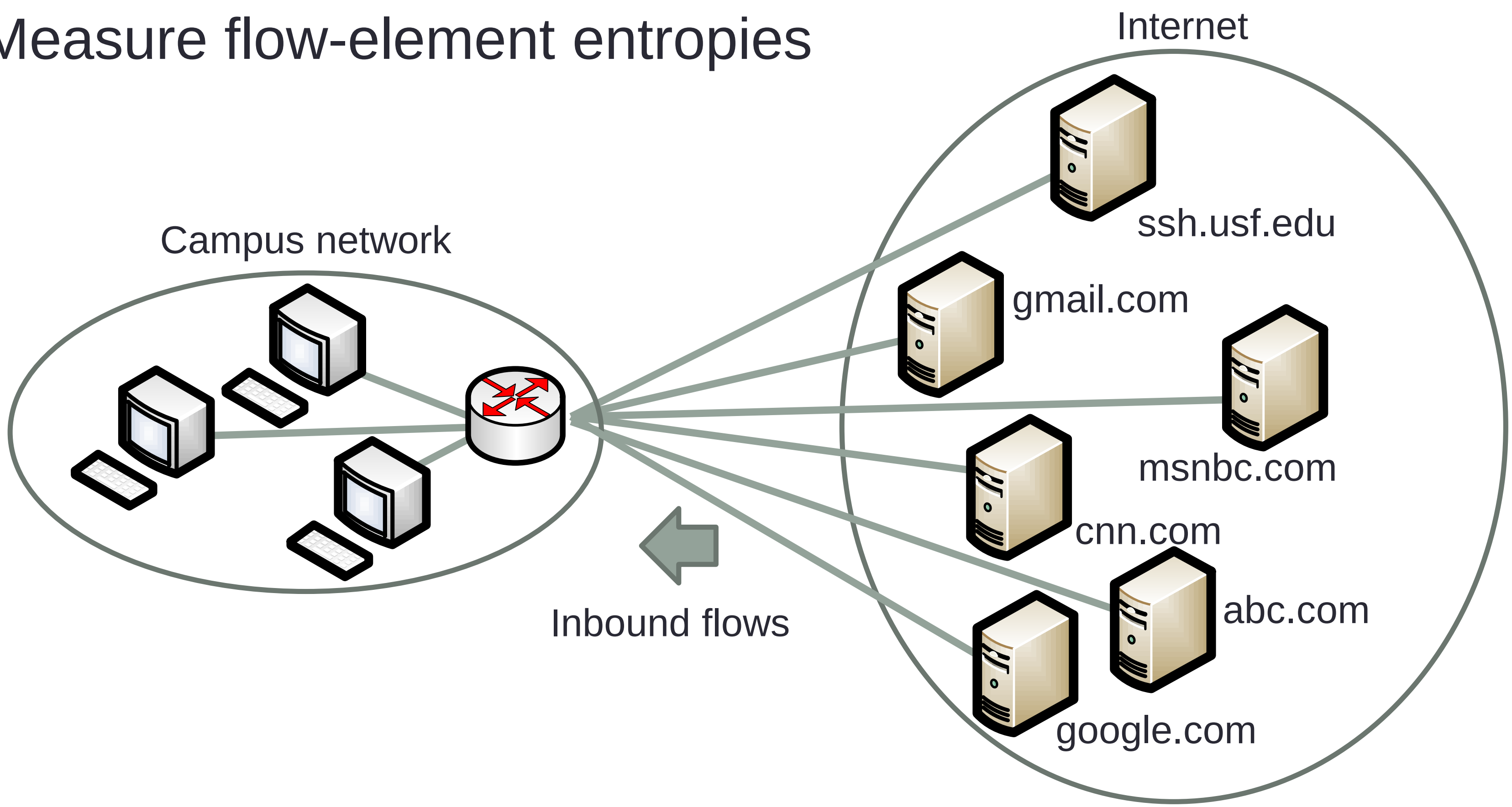
Motivation

- Most networks use Network Address Translation (NAT)
- Although NAT has been used since early 2000s, traffic behind NAT has not been characterized
- One approach for flow characterization is to measure the *randomness* or *uncertainty* of elements of a flow
- E.g., entropy of IP addresses, ports, and combinations
- Goal: characterize normal traffic behavior (entropy) by using flow information



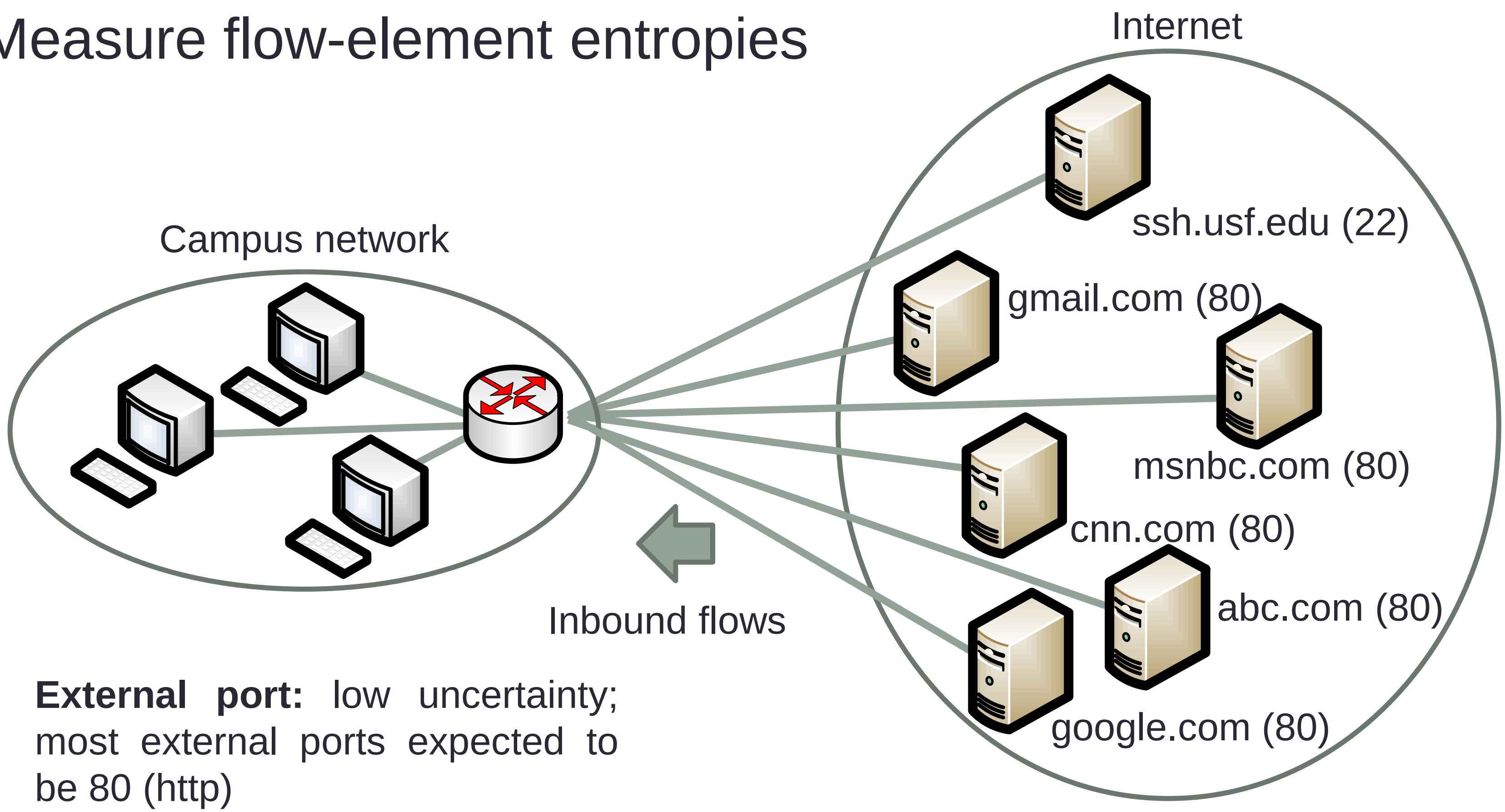
Methodology

- A flow is uniquely identified by the external IP, campus IP, external port, campus port, protocol
- Measure flow-element entropies



Methodology

- A flow is uniquely identified by the external IP, campus IP, external port, campus port, protocol
- Measure flow-element entropies

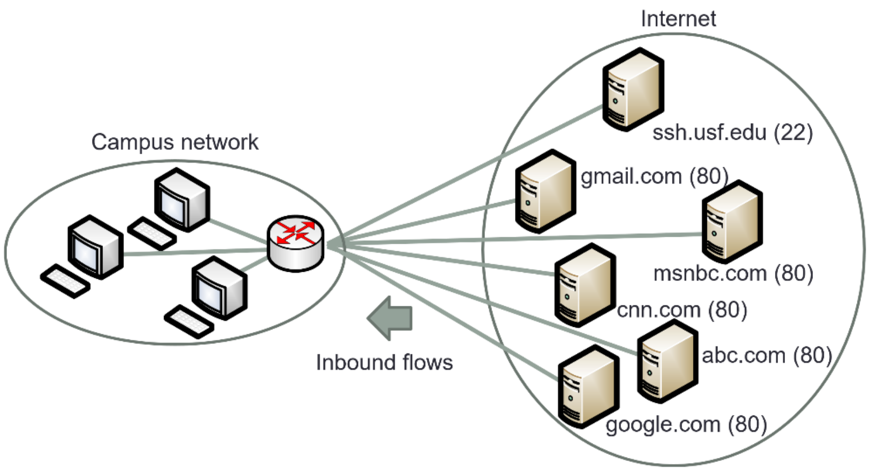


External port: low uncertainty; most external ports expected to be 80 (http)

Methodology

- Entropy provides a measure of randomness or uncertainty
- For a variable X, entropy of X = $\sum_{x \in X} p_x \log_2 \left(\frac{1}{p_x} \right)$
- For the previous port example, let *X* be the variable indicating the external port

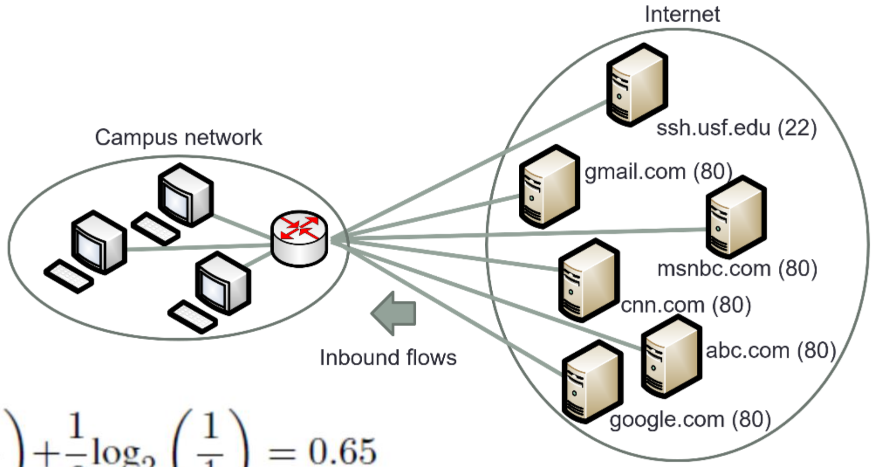
$$X = \begin{cases} 80 \text{ with probability } p_1 = \frac{5}{6} \\ 22 \text{ with probability } p_2 = \frac{1}{6} \end{cases}$$



Methodology

- Entropy provides a measure of randomness or uncertainty
- For a variable X, entropy of X = $\sum_{x \in X} p_x \log_2 \left(\frac{1}{p_x} \right)$
- For the previous port example, let *X* be the variable indicating the external port

$$X = \begin{cases} 80 \text{ with probability } p_1 = \frac{5}{6} \\ 22 \text{ with probability } p_2 = \frac{1}{6} \end{cases}$$

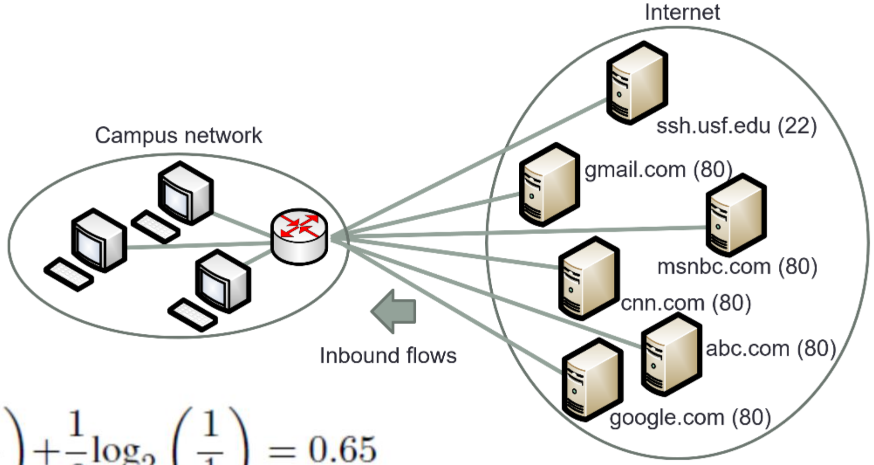


$$\text{Entropy External Port} = \sum_{i=1}^{2} p_i \log_2 \left(\frac{1}{p_i} \right) = \frac{5}{6} \log_2 \left(\frac{1}{\frac{5}{6}} \right) + \frac{1}{6} \log_2 \left(\frac{1}{\frac{1}{6}} \right) = 0.65$$

Methodology

- Entropy provides a measure of randomness or uncertainty
- For a variable X, entropy of X = $\sum_{x \in X} p_x \log_2 \left(\frac{1}{p_x}\right)$
- For the previous port example, let *X* be the variable indicating the external port

$$X = \begin{cases} 80 \text{ with probability } p_1 = \frac{5}{6} \\ 22 \text{ with probability } p_2 = \frac{1}{6} \end{cases}$$

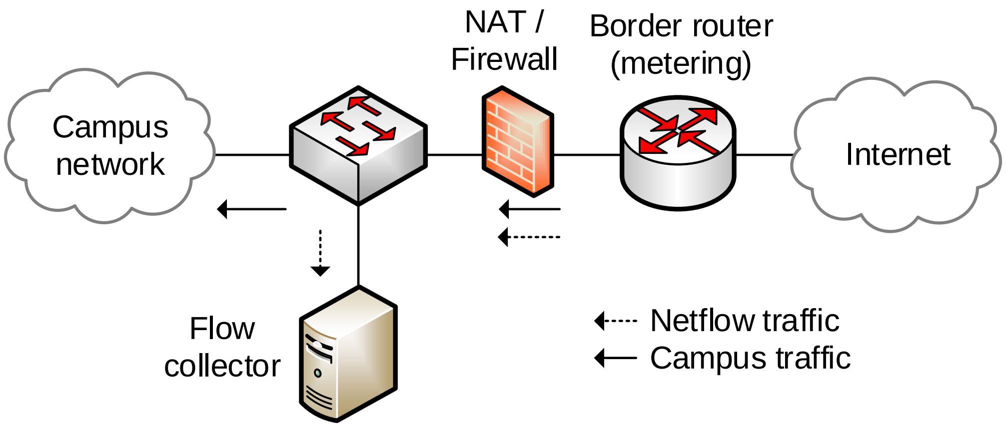


$$\text{Entropy External Port} = \sum_{i=1}^{2} p_i \log_2 \left(\frac{1}{p_i}\right) = \frac{5}{6}\log_2\left(\frac{1}{\frac{5}{6}}\right) + \frac{1}{6}\log_2\left(\frac{1}{\frac{1}{6}}\right) = 0.65$$

- 0 entropy ~ no uncertainty (e.g., all external ports are 80)
- 1 entropy ~ random -> high uncertainty

Methodology

- Campus network with 15 buildings
- Inbound traffic is used as a reference (external IP address is in the Internet, campus IP address is on campus)
- The collector organizes flow data in five-minute time slots
- Traffic data observed during a week is representative of the campus traffic



Methodology

- The entropy of a random variable *X* is:

$$H(X) = \sum_{i=1}^{N} p(x_i)\log_2\left(\frac{1}{p(x_i)}\right),$$

where $x_1, x_2, \ldots x_N$ is the range of values for *X*, and $p(x_i)$ is the probability that *X* takes the value x_i

- For each external (campus) IP address (port) x_i, the probability $p(x_i)$ is calculated as

$$p(x_i) = \frac{\text{Flows with } x_i \text{ as external (campus) IP addr. (port)}}{\text{Total number of flows}}$$

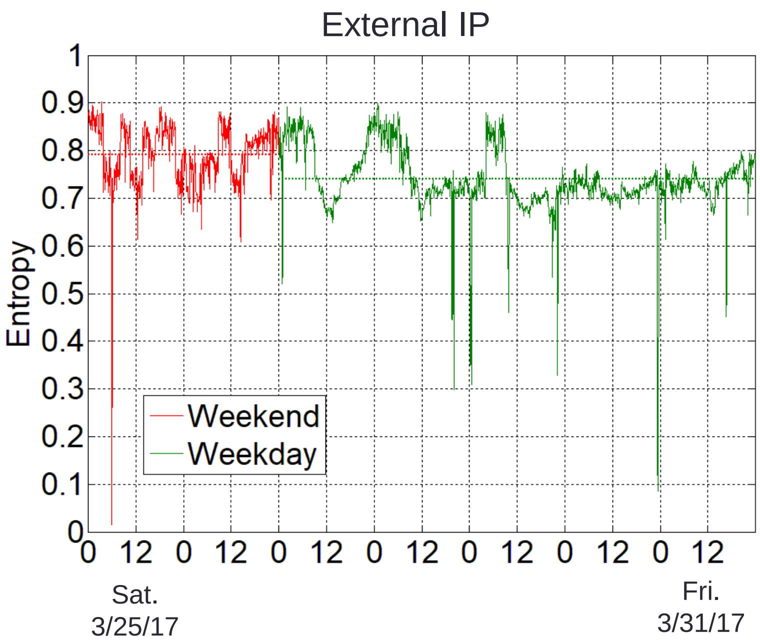
- Entropies are normalized to that of the uniform distribution

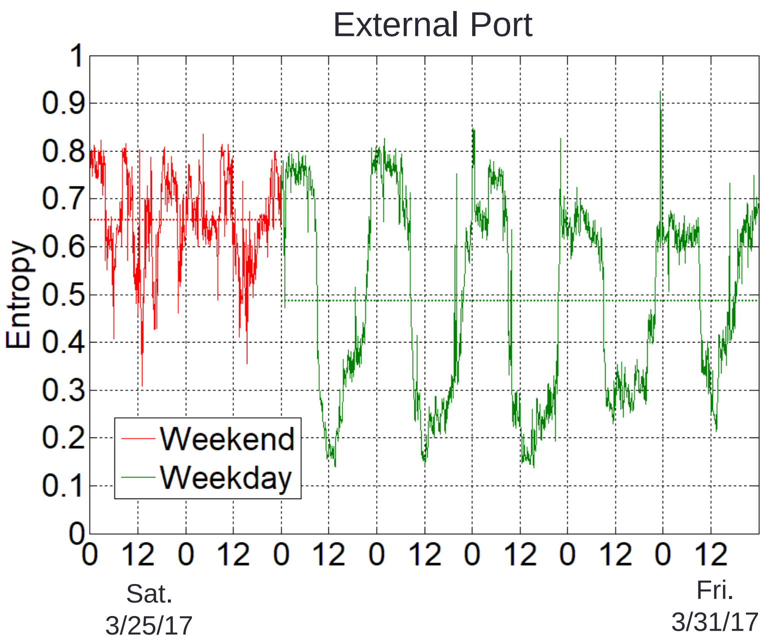
Methodology

- This paper also considers the entropy of the 3-tuple {external IP, campus IP, campus port}
- For a given 3-tuple x_i, the corresponding probability is calculated as

$$p(x_i) = \frac{\text{Flows with } x_i \text{ as 3-tuple}}{\text{Total number of flows}}$$

Results





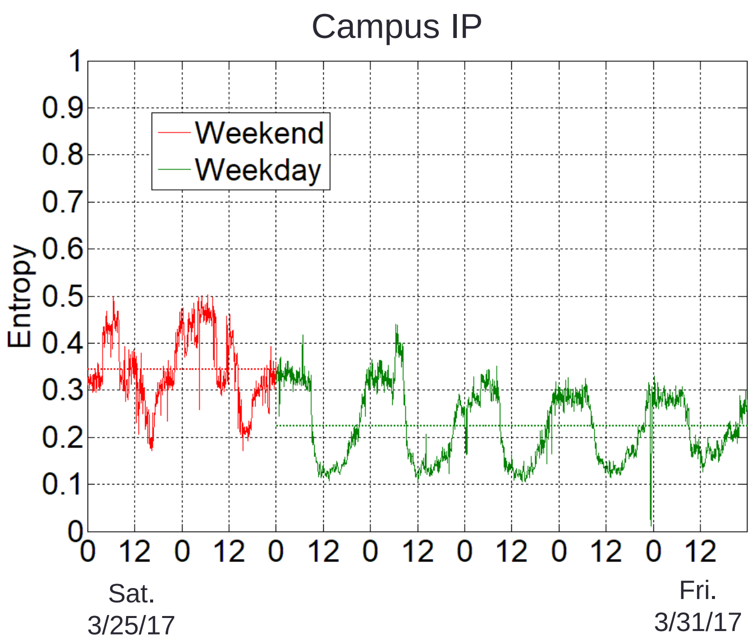
External IP

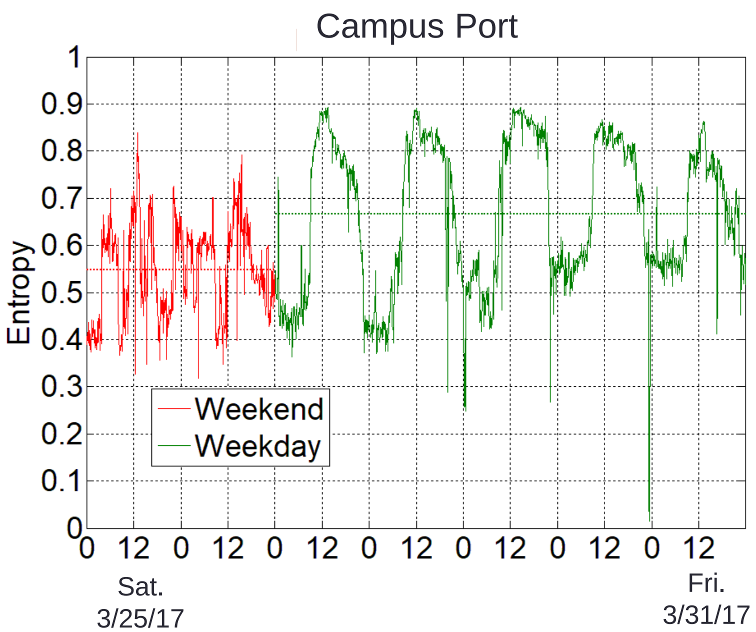
- In general, high entropy, ‘many’ external IP addresses
- External IPs dispersed in the Internet
- Abnormal low entropy points
- Entropy near zero (no uncertainty of the external IP address), or ‘very low’ level (few external IP addresses dominate the distribution)

External port

- Higher entropy during the night, weekends
- Low entropy during the day, noon
- Large volume of http flows when students are on campus (less uncertainty/entropy on external port)
- Abnormal high entropy points
- Entropy widely varies over ‘hours’ but not over very short time periods

Results





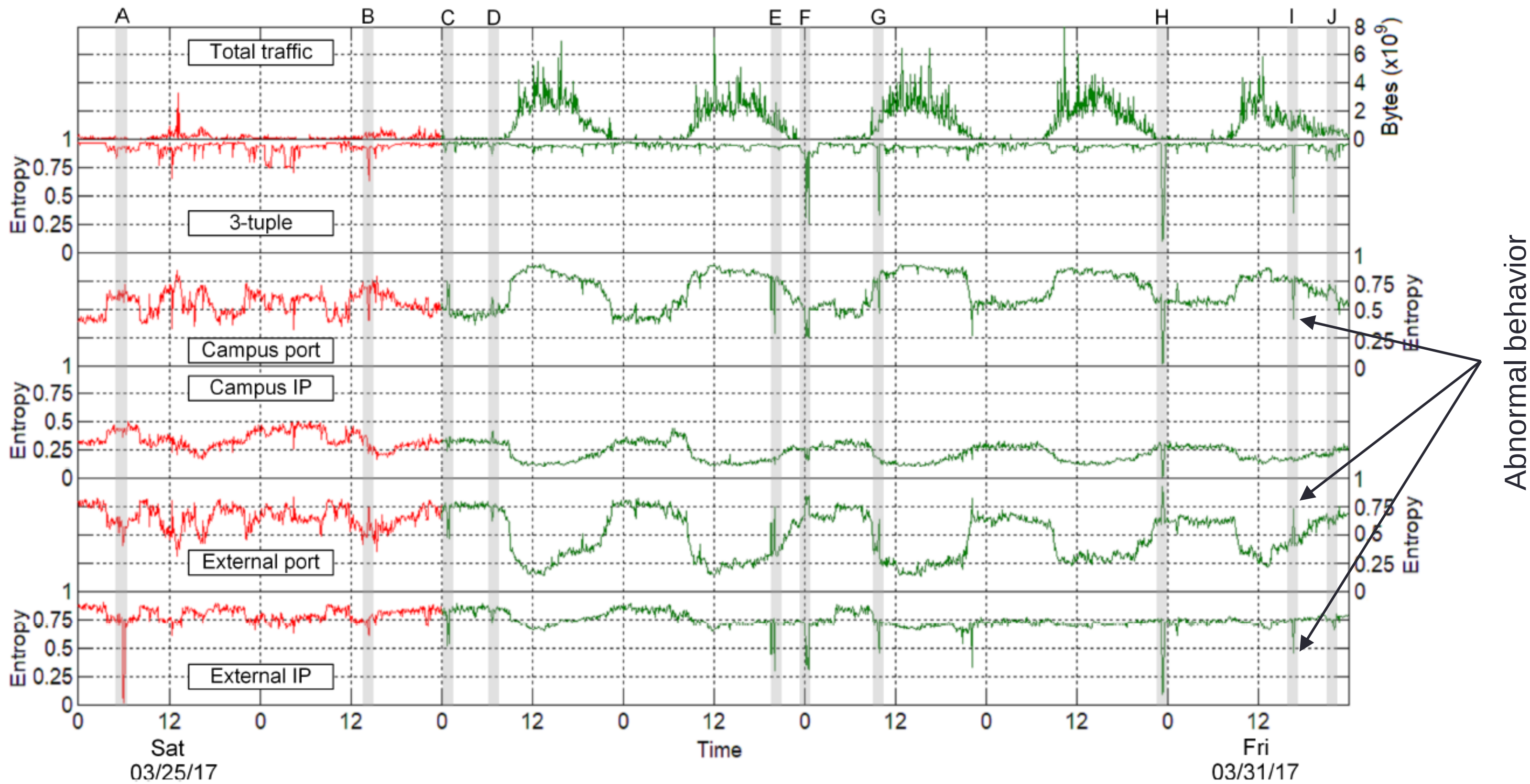
Campus IP

- In general, low entropy, 'few' IP addresses on campus
- Higher entropy on weekends and at night
- Lower entropy when students are on campus
- A handful of public IP addresses used for regular Internet connectivity (NAT operation)
- Entropy varies over 'hours' but not over very short time periods

Campus port

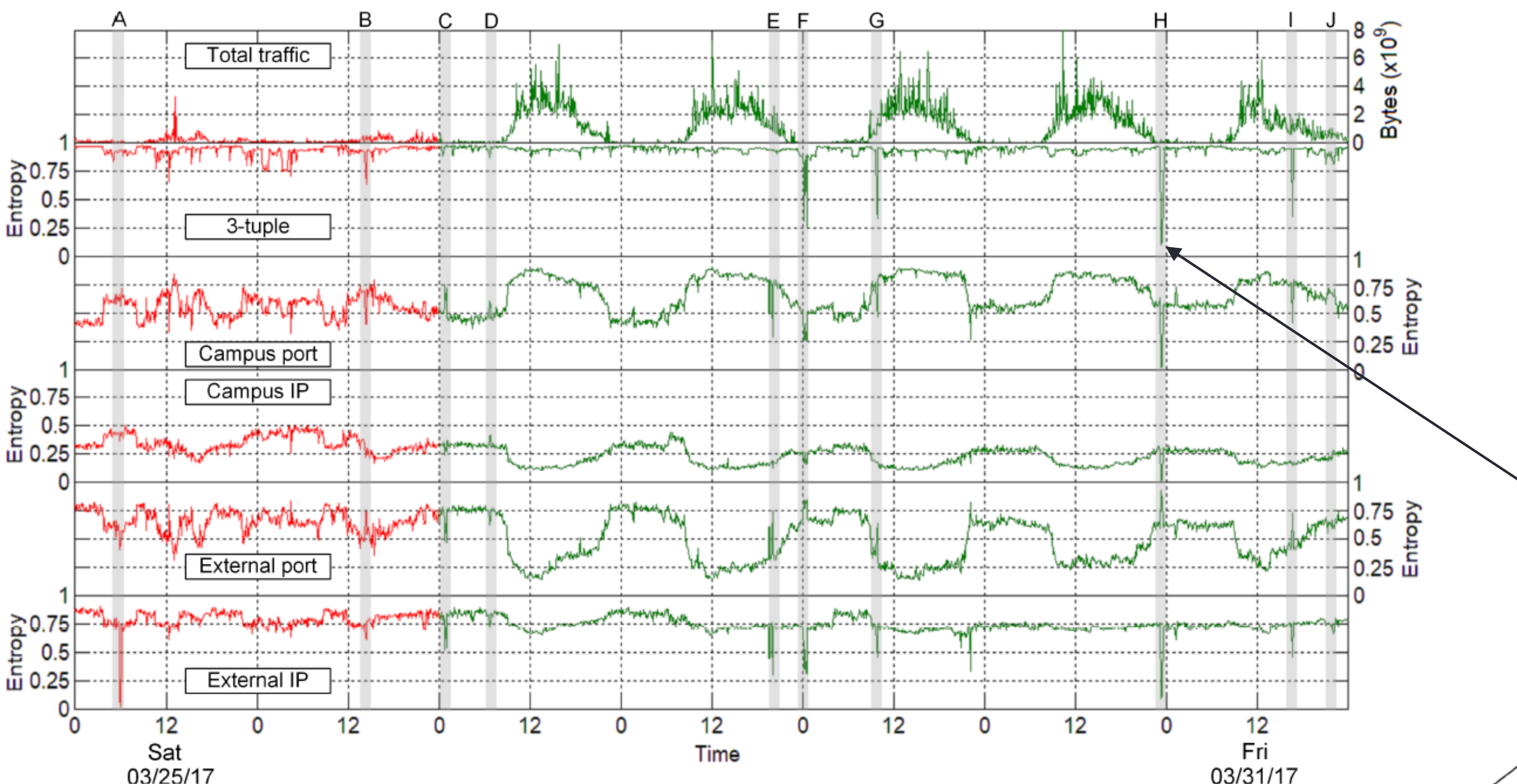
- Lower entropy at night
- High entropy (close to uniform distribution) at noon
- Dynamic ports used by browsers when students connect to the Internet
- Abnormal low entropy points
- Entropy widely varies over 'hours' but not over very short time periods

Results

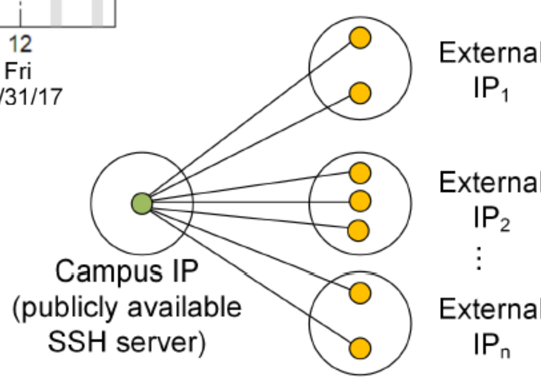


- Anomalies are detected by a single feature or by correlating multiple features
- E.g., event I: low campus port's entropy, high external port's entropy, low external IP's entropy

Results



Distributed brute force attack



Results

- Correlation of entropy time-series

	Campus IP	Campus port	External IP	External port	Total traffic
Weekday					
3-tuple	0.23	0.1	0.6	-0.02	-0.05
Campus IP		-0.85	0.6	0.89	-0.8
Campus port			-0.37	-0.98	0.78
External IP				0.45	-0.36
External port					-0.81
Weekend					
3-tuple	-0.23	-0.12	0.56	0.06	-0.03
Campus IP		0.15	-0.38	0.06	-0.38
Campus port			-0.48	-0.93	0.31
External IP				0.48	-0.05
External port					-0.39

Conclusion

- In a NATed environment, entropies may widely vary. E.g.,
  - External and campus ports vary from below 0.2 to above 0.8 (in a normalized entropy scale 0-1)
  - Campus IP address varies from 0.1 to 0.4
- Despite the wide range of values, building a granular (small time slots) entropy characterization helps to detect anomalies
- Strong correlation exists between entropy time-series, which facilitates the detection of potential attacks
- Future work includes anomaly detection algorithms that exploit the entropy characterization of flow elements