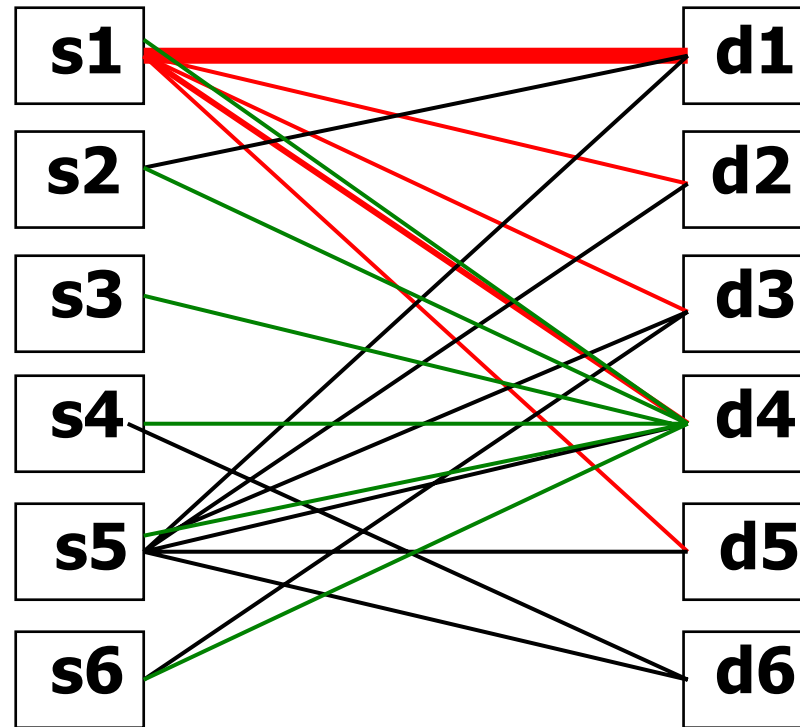


Profiling Internet Backbone Traffic: Behavior Models, Applications and Implementation

Supratik Bhattacharyya
Sprint ATL

Collaborators:
Kuai Xu, Zhi-Li Zhang
University of Minnesota

Our focus: Communication patterns



- Who is talking to whom? How often?
- How are they using ports?
- One-to-many? Few-to-few? Many-to-many? ...
- Are there fundamental shifts in a host's communication pattern over time? Why?

Examining Communication Patterns

- Questions
 - Are there distinct patterns?
 - Do they correspond to application types/host activities?
 - How can we characterize these patterns?
 - How can we automatically discover them?
- Intuition: many informal observations
 - a port scanner sending single pkts to numerous hosts on target port(s)
 - A web-server using port 80 to talk to several clients
 - Few P2P nodes talking frequently among themselves

Problem Setting

- Challenges for a backbone network
 - vast masses of traffic data
 - large number of end-hosts
 - diverse applications

- Our starting point
 - One-way traffic data from a single link
 - Assume availability of packet headers only
 - Make no assumptions about normal (or anomalous) behavior

Talk Outline

- Profiling Methodology
- Observations
- Implementation

Talk Outline

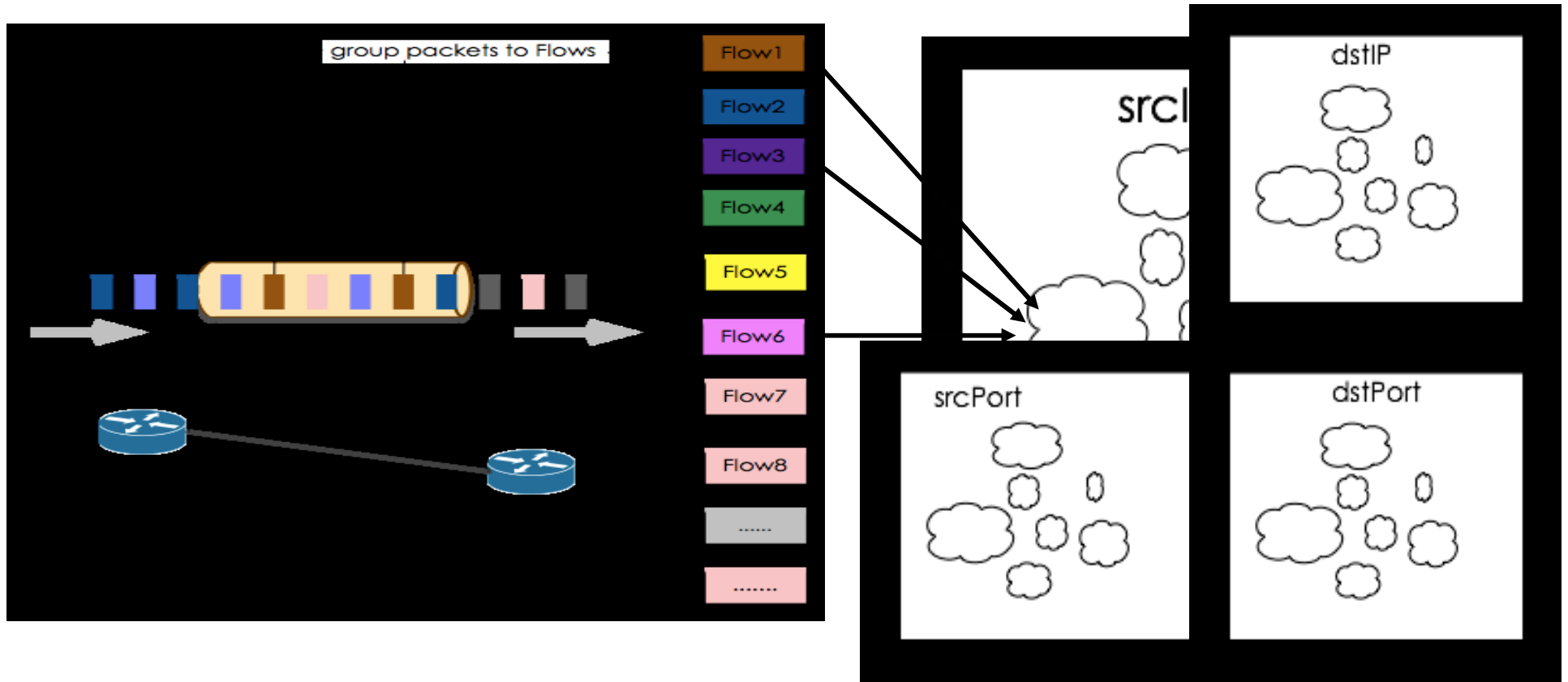
- Profiling Methodology
- Observations
- Implementation

Roadmap of our methodology

- Data pre-processing
 - aggregate packet streams into 5-tuple flows
 - group flows into clusters
- Extract significant clusters
 - data reduction step using entropy
- Classify cluster behavior based on similarity/dissimilarity of communication patterns
 - characterize using information theory
 - clusters classified into behavior classes
- Interpret behavior classes
 - structural modeling for dominant activities

Data pre-processing

- Aggregate packet streams into 5-tuple flows
- Group flows associated with same end hosts/ports into clusters



Roadmap of our methodology

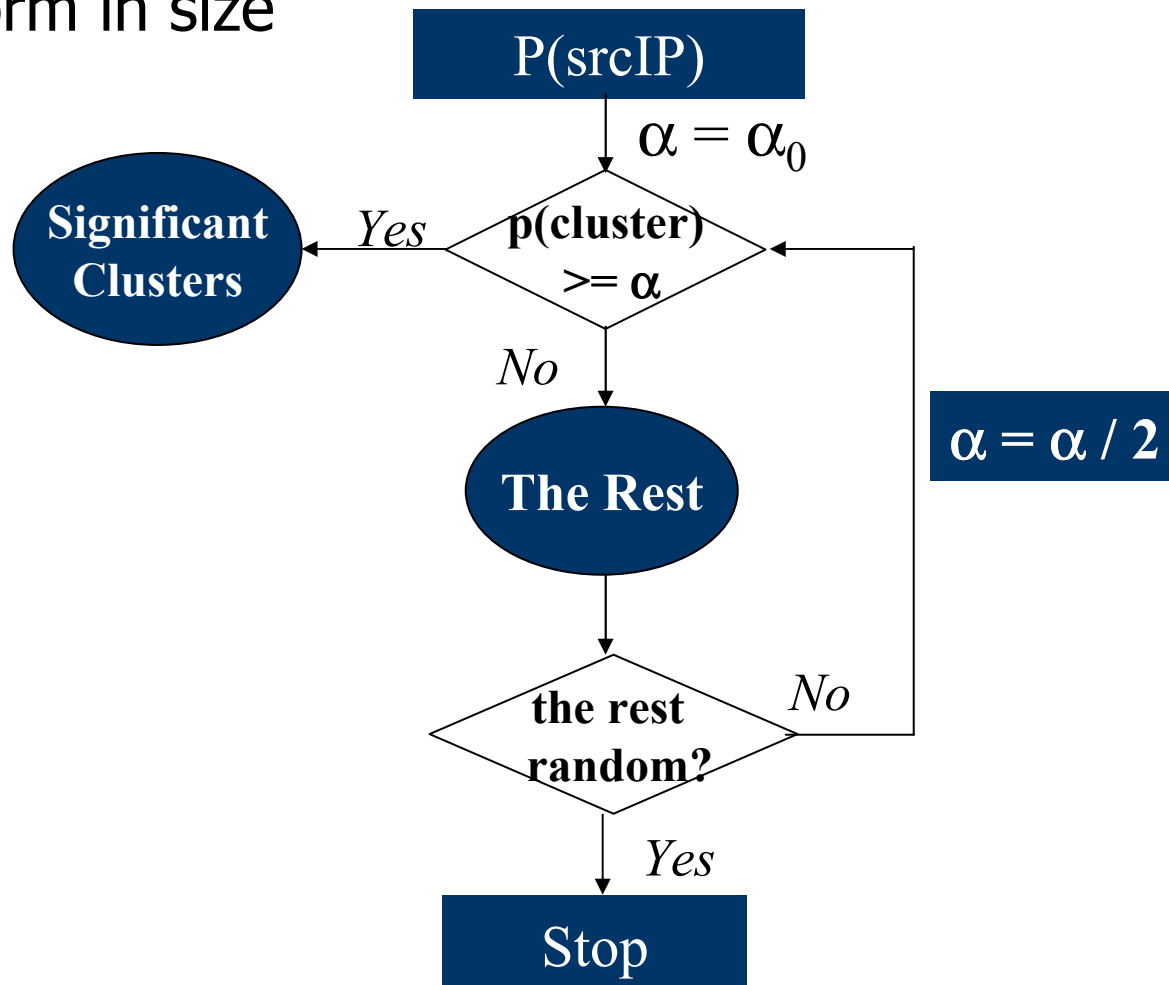
- Data pre-processing
 - aggregate packet streams into 5-tuple flows
 - group flows into clusters
- **Extract significant clusters**
 - **data reduction step using entropy**
- Classify cluster behavior based on similarity/dissimilarity of communication patterns
 - characterize using information theory
 - clusters classified into behavior classes
- Interpret behavior classes
 - structural modeling for dominant activities

Extract significant clusters

- Focus on significant clusters
 - sufficiently large number of flows
 - represent behavior of significant interest
- One definition: using a fixed threshold
 - a cluster is significant if containing at least $x\%$ of flows
 - how to choose x for all links?
- Our definition: adaptive thresholding using entropy
 - a cluster is significant if “standing out” from the rest
 - use entropy to quantify whether the rest looks random

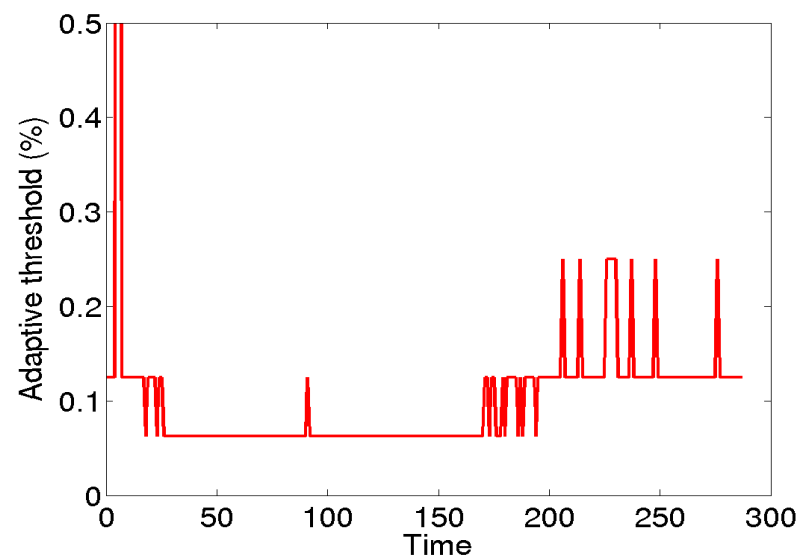
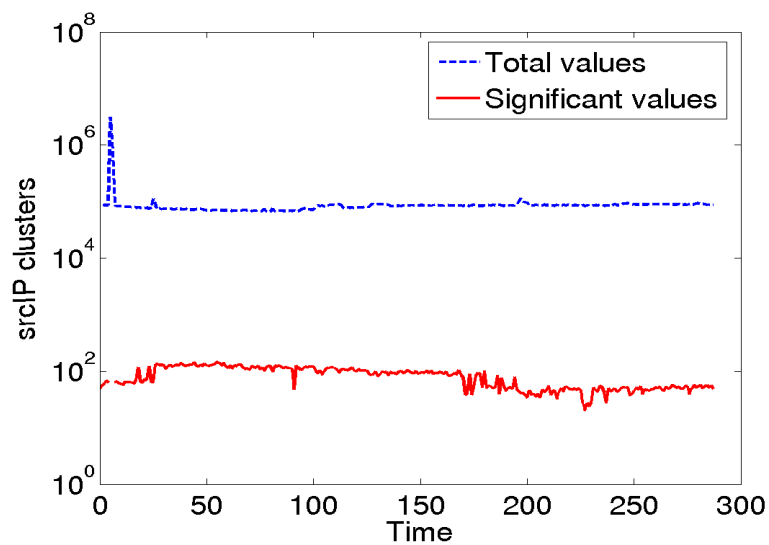
Entropy-based adaptive thresholding

- An iterative process
 - extract significant clusters until the rest look nearly uniform in size



Sample results

- Packet traces
 - OC-48 link during 24 hours
 - extract clusters every 5 minutes



Roadmap of our methodology

- Data pre-processing
 - aggregate packet streams into 5-tuple flows
 - group flows into clusters
- Extract significant clusters
 - data reduction step using entropy
- **Classify cluster behavior based on similarity/dissimilarity of communication patterns**
 - characterize using information theory
 - clusters classified into behavior classes
- Interpret behavior classes
 - structural modeling for dominant activities

Understanding behavior patterns

- Still many significant clusters in each time interval
 - can we characterize their behavior patterns?
 - are there similarities/dissimilarities in behavior?
 - communication patterns provide more insight than volume metrics
- What traffic features should we look at? And how?
 - for each cluster, look at distributions of flows by ports and IP addresses
 - distribution summarized by relative uncertainty
 - each cluster characterized by a point in 3-D space

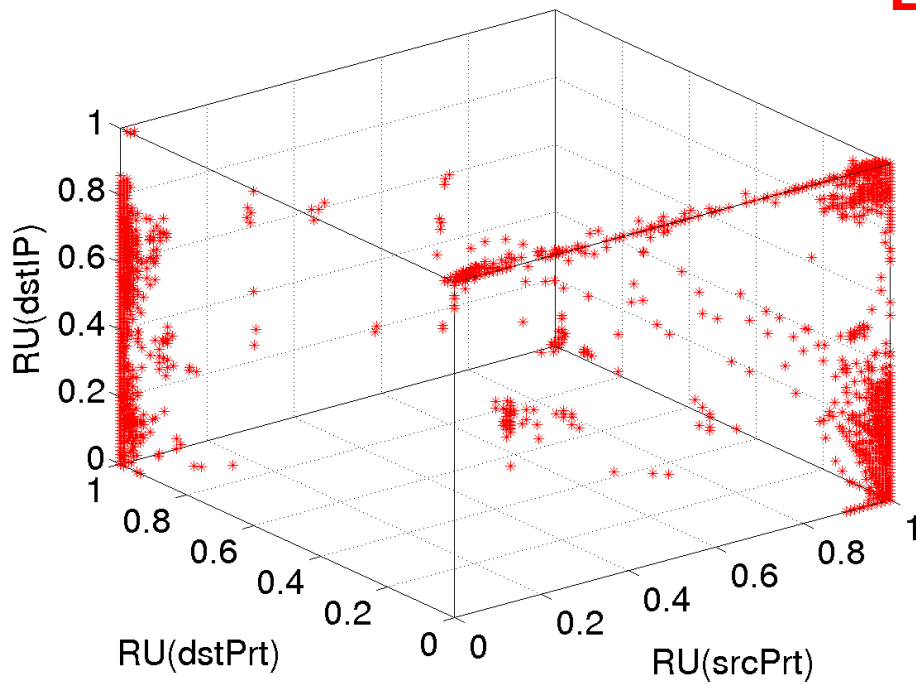
Relative uncertainty

- Entropy: $H(X) = -\sum p(x_i) \log p(x_i)$
- Maximum Entropy: $H_{\max}(X) = \log [\min(m, N)]$
- Relative Uncertainty of variable X

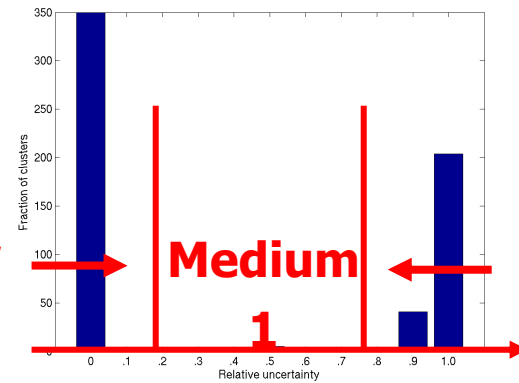
$$RU(X) := H(X) / H_{\max}(X), RU \in [0, 1]$$

- $RU(X) = 0$: X is deterministic distribution
- $RU(X) = 1$: X is randomly distributed

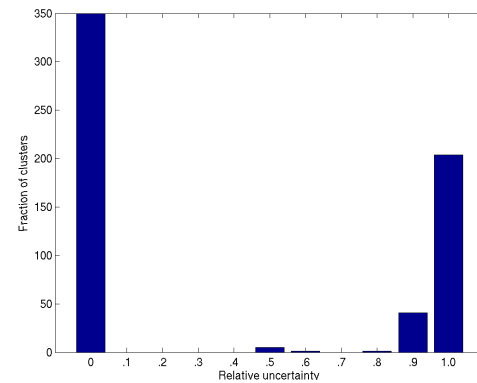
Behavior characterization



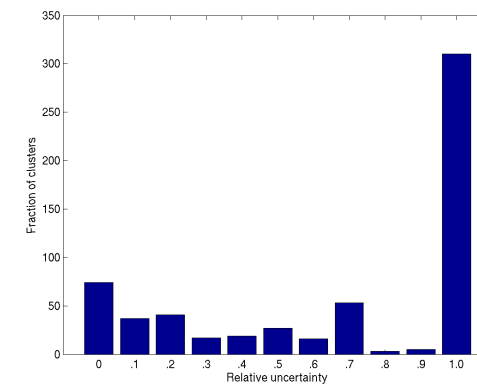
Low
0



srcPort

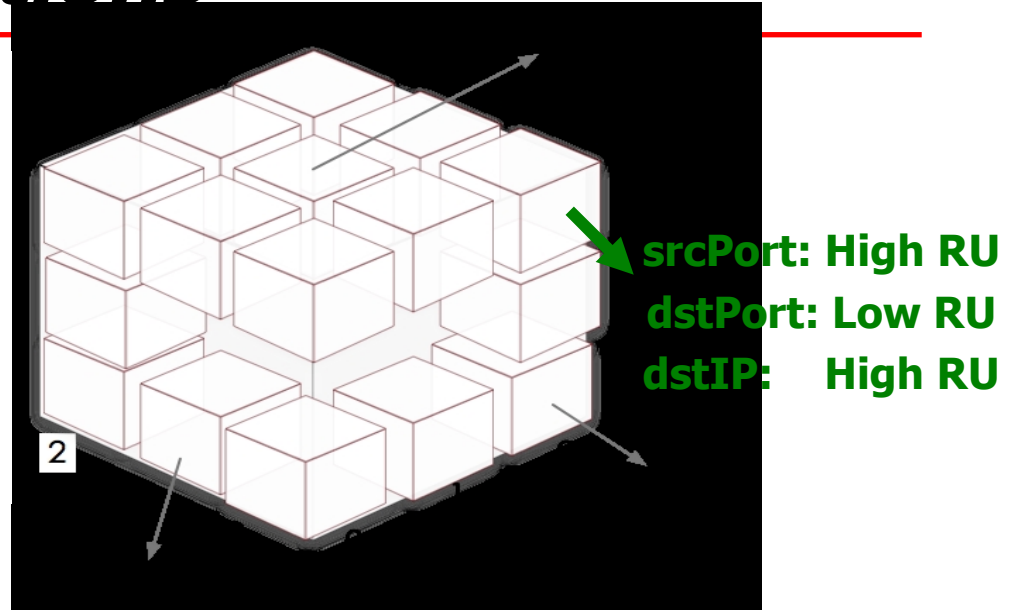
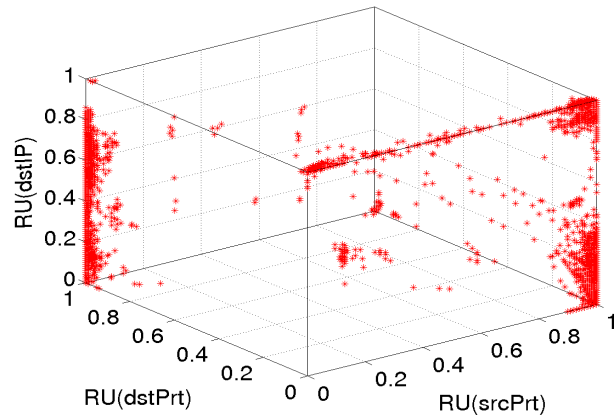


dstPort



dstIP

Behavior classifications

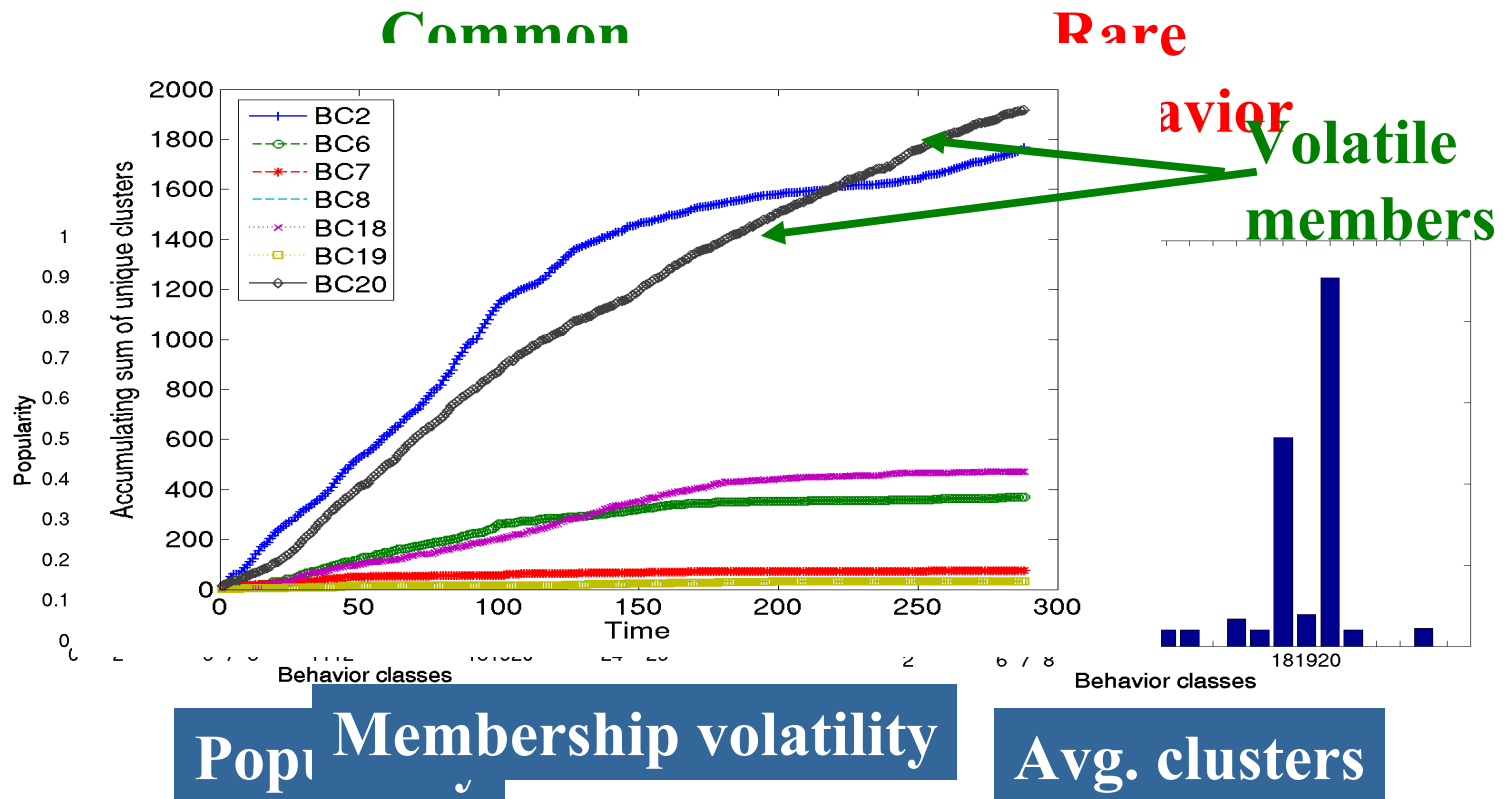


- Behavior classes (BC)
 - summarize three feature distributions into 27 classes
 - $[0, 0, 0] \dots [2, 2, 2]$, for convenience BC_0 to BC_{26}
- What is the difference between behavior classes?
 - are there common vs. rare behavior classes?
 - do BCs have many or a few clusters?
 - are memberships in BCs stable?

Temporal Properties

Metrics

- Popularity: how many time slots do we see a BC in?
- Avg. size: how many clusters in each BC per time slot?
- Membership volatility : does a BC contain the same clusters over time?



Summary of behavior classifications

- Behavior classes classify clusters based on communication patterns
- Behavior classes have distinct temporal properties
- Clusters have stable behavior over time

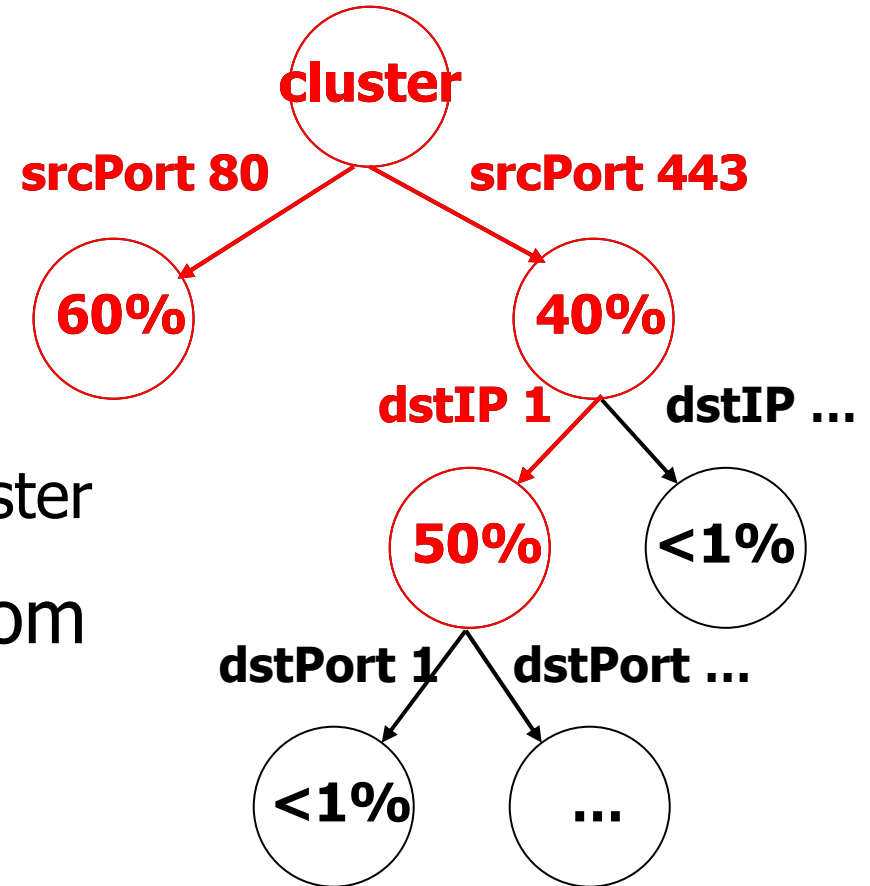
How can we interpret observed behavior?

Roadmap of our methodology

- Data pre-processing
 - aggregate packet streams into 5-tuple flows
 - group flows into clusters
- Extract significant clusters
 - data reduction step using entropy
- Classify cluster behavior based on similarity/dissimilarity of communication patterns
 - characterize using information theory
 - clusters classified into behavior classes
- Interpret behavior classes
 - structural modeling for dominant activities

Structural modeling

- Each cluster has hundreds or thousands of flows.
 - an exhaustive approach is not practical
 - need a compact summary
- Dominant state analysis
 - dominant activities in each cluster
- An example: a web server from srcIP perspective
 - $RU_{srcPort} \leq RU_{dstPort} \leq RU_{dstIP}$
 - feature dependency: srcPort, dstIP, dstPort



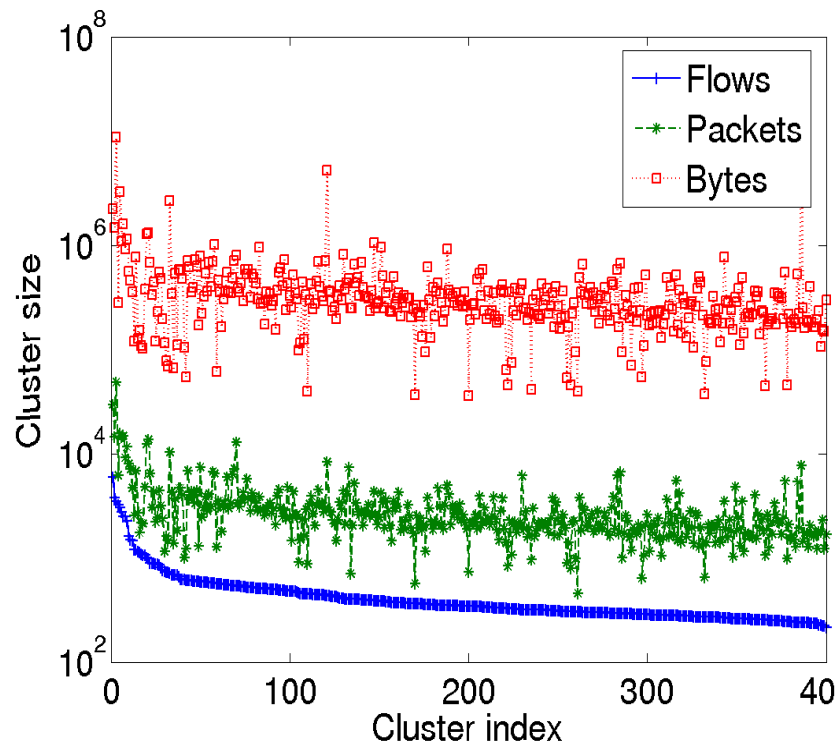
Dominant state analysis : Example

BCs	Structural models	Comments
BC ₂	srcPort(.)->dstPort(.)->dstIP(*) srcPort(1025)->dstPort(137)->dstIP(*) srcPort(1081)->dstPort(137)->dstIP(*) srcPort(1153)->dstPort(1434)->dstIP(*) srcPort(220)->dstPort(6129)->dstIP(*)	scan activities

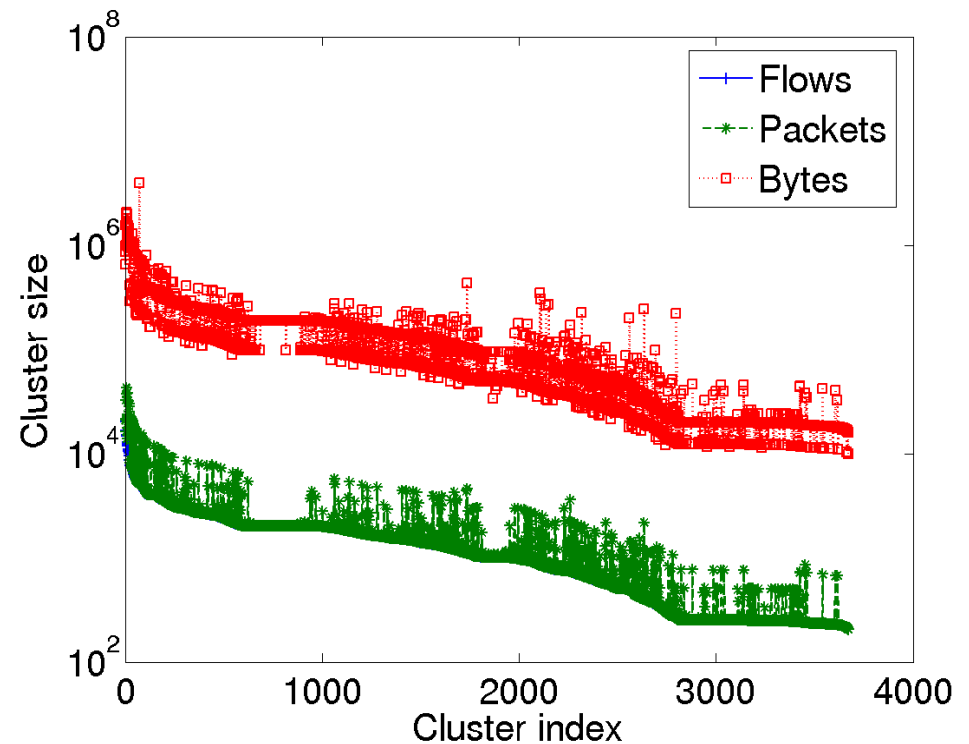
- Observations
 - clusters within the same BCs have similar structural models
 - But they could have different dominant states (or activities)

Additional flow features

- Flow, packet and byte counts
 - average counts of packets and bytes per flow



srcIPs in $BC_{\{6,7,8\}}$



srcIPs in $BC_{\{2,20\}}$

Talk Outline

- Profiling Methodology
- **Observations**
- Implementation

Canonical behavior profiles

Profile	Interpretation	BC	Cluster freq.	Flow feature
Server	server talking to a large number of clients, e.g., web, DNS, FTP	srcIP BC{6,7,8} dstIP BC{18,19}	frequently occurring	diverse packet sizes and byte counts
Heavy hitter	host talking a lot to one or several IP addresses (typically servers), e.g., NAT, web proxies, etc.	srcIP BC{18,19} dstIP BC{6,7}	frequently occurring	diverse packet sizes and byte counts
Exploit	host attempting to spread malicious exploits	srcIP BC{2,20}	highly volatile	single packet of fixed size

Applications (1): finding unknowns

- Discover “servers” running on high ports
 - we found servers on ports 56192, 56193, 60638
- Zoom in on potentially interesting behavior
 - srcIP with “heavy hitter” profile frequently talking to dstIP on port 7070 (RealAudio)
 - appears to be a proxy inside large corporation
- Discover unknown exploit activity
 - we found a srcIP with “exploit profile” doing single pkt scans on UDP port 12827

Applications(2): security

- Distinguish between exploit behaviors
 - srcIP BC2[0,0,2] => single src port
 - srcIP BC20[2,0,2] => random src ports
- Rare BCs contain exploits/attacks
 - DOS attack in dstPrt BC15[1,2,0]
 - dstPrt clusters (6667, 113,8083) – 94% of flows to single dstIP from random srcIPs
 - Note: may not be visible in srcIP/dstIP dimensions
 - Low intensity mixed scans in srcIP BC10[1,0,1]
 - srcIP 200.164.36.21 appears once, sends lot of ICMP probes towards small set of hosts + port (139, 445) scans

Applications(3): unusual behavior

- Unusual profiles for popular service ports
 - Found srcIPs scanning ports 25, 53, 80
 - Classified with exploits in BC2, BC20, not with servers or heavy hitters!
- Clusters changing BCs
 - 95% of clusters never do!
 - Found Yahoo web-server changing from srcIP BC8[0,2,2] to BC6[0,2,0] and back
 - transitioned from talking to a lot of clients to a single client
 - All flows (87%) to this single client have same packet and byte count

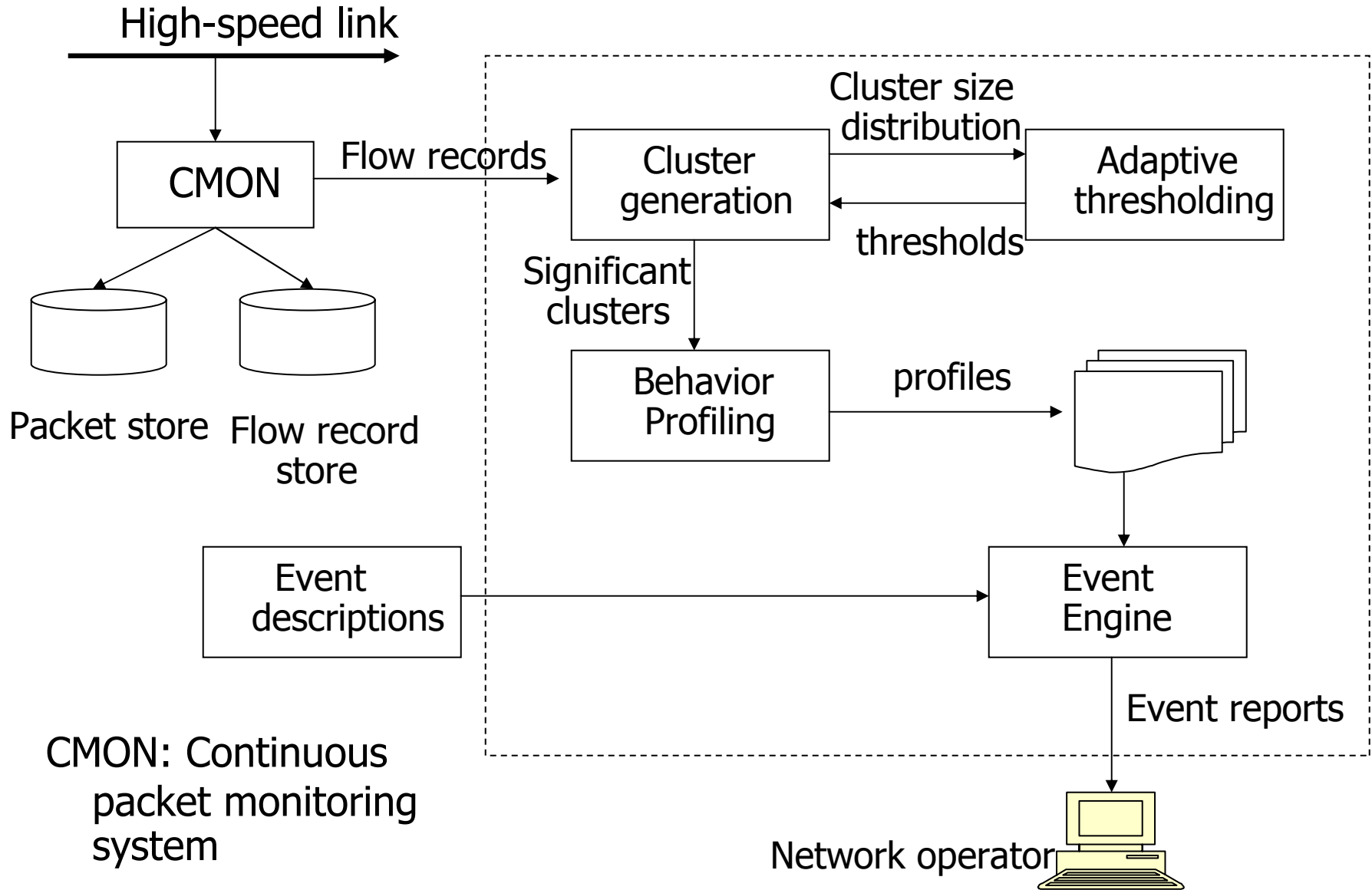
Talk Outline

- Profiling Methodology
- Observations
- **Implementation**
 - Architecture
 - Initial findings
 - Identifying top sources of unwanted traffic

Deployment in a transit backbone

- Data collection systems at strategic locations
 - link-level taps to collect packets, build flow records
 - Netflow records from router interfaces
- Behavior profiling systems use flow records
 - Co-located with taps or centralized (e.g., per PoP)
 - Profiles traffic per link/interface
 - Will correlate multiple links/interfaces in the future
- An iterative approach
 - profiling narrows down events of interest
 - Deep packet inspection or flow-level analysis of these events

Architecture



Implementation Issues

- Can we profile in real time?
 - Initial result: flows for 5 min time windows can be usually profiled in less than 1 minute
- Profiling time increases linearly with flow table size
 - How do we handle DOS attacks?
- Reading a flow table all at once is expensive
 - the input of flow records has to be spread over time
- Need a concise language to describe events of interest
- Event reports have to be of manageable size

Identifying top sources of unwanted traffic

- Goal: Reduce the number of sources reported
- Approach
 - Identify all sources with “exploit” profile
 - Base rule: report each exploit source + port(s) it attacks
 - Use additional rules to narrow down most severe exploit sources
- Severity of exploit traffic
 - frequency: # of time slots in which an exploit source is significant
 - persistency: # of consecutive slots (frequency > 1)
 - intensity: (average) # of targets touched per minute

Heuristics for reporting sources

Rule	Heuristic
Base rule	report every source with an exploit profile
Rule 1	report exploit sources from the top x <i>origin ASes</i>
Rule 2	report source with an exploit profile for one of the <i>top k popular ports</i>
Rule 3	Report sources with an exploit profile for at least n <i>consecutive periods</i>
Rule 4	Report source that touches at least m <i>targets per minute</i>

Sample result

Rule	Cost	Flow reduction	Packet reduction	Byte reduction	Wastage (%)
Base rule	3756	76.8%	71.1%	67.2%	1310 (34.8%)
Rule 1 (top 10 ASes)	1942	22.7%	19.5%	17.9%	1071 (55.1%)
Rule 2 (top 5 ports)	3471	67.1%	56.3%	52.1%	1216 (35.0%)
Rule 3 (2 consecutive time slots)	1586	48.4%	43.5%	37.9%	505 (31.8%)
Rule 4 (300 targets per minute)	1789	64.7%	57.2%	48.8%	302 (16.9%)

Cost – number of sources reported

Wastage – number of times a reported source is never seen again

Summary

- Developed a systematic methodology to automatically discover and interpret communication patterns
- Used information-theoretic approach to build behavior models of end hosts and applications
- Applied dominant state analysis to explain traffic behavior
- Discovered canonical profiles as well as rare and deviant behaviors

Ongoing and Future work

- Correlate behavior profiles across multiple links
- Validate behavior profiles using additional features, e.g., packet payload
- Implement traffic profiling framework and deploy in conjunction with continuous packet monitoring system.

Thank you!

Contact info:

supratik@sprintlabs.com

<http://www.sprintlabs.com/~supratik>