ACM SIGCOMM 2005

Profiling Internet Backbone Traffic: Behavior Models and Applications

Kuai Xu, Zhi-Li Zhang, and Supratik Bhattacharyya

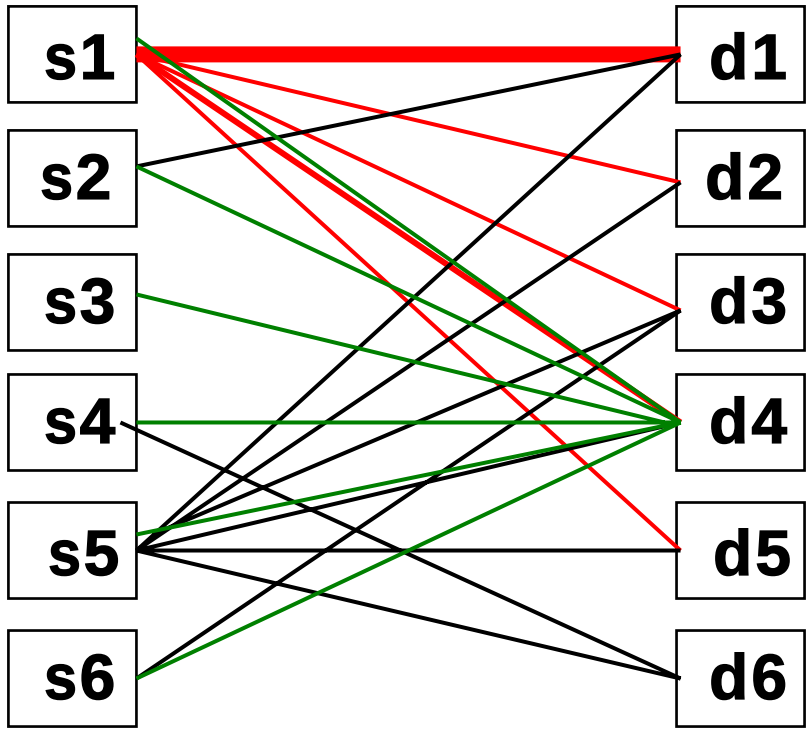
University of Minnesota Sprint ATL

August 24, 2005

Why profile traffic?

- Changes in Internet traffic dynamics
  - increase in unwanted traffic
  - emergence of disruptive applications
  - new services on traditional ports
  - traditional service on non-standard ports
- Existing tools
  - rely on ports for identifying or classifying traffic
  - report volume-based heavy hitters
  - look for specific or known patterns
- **Need better techniques to discover behavior patterns**
  - help network operators secure and manage networks

Communication patterns



- Underlying communication patterns of end hosts
  - who are they talking to? how are ports used?
  - how many packets or bytes transferred?
- Can communication patterns reveal interesting behavior?

Problem settings

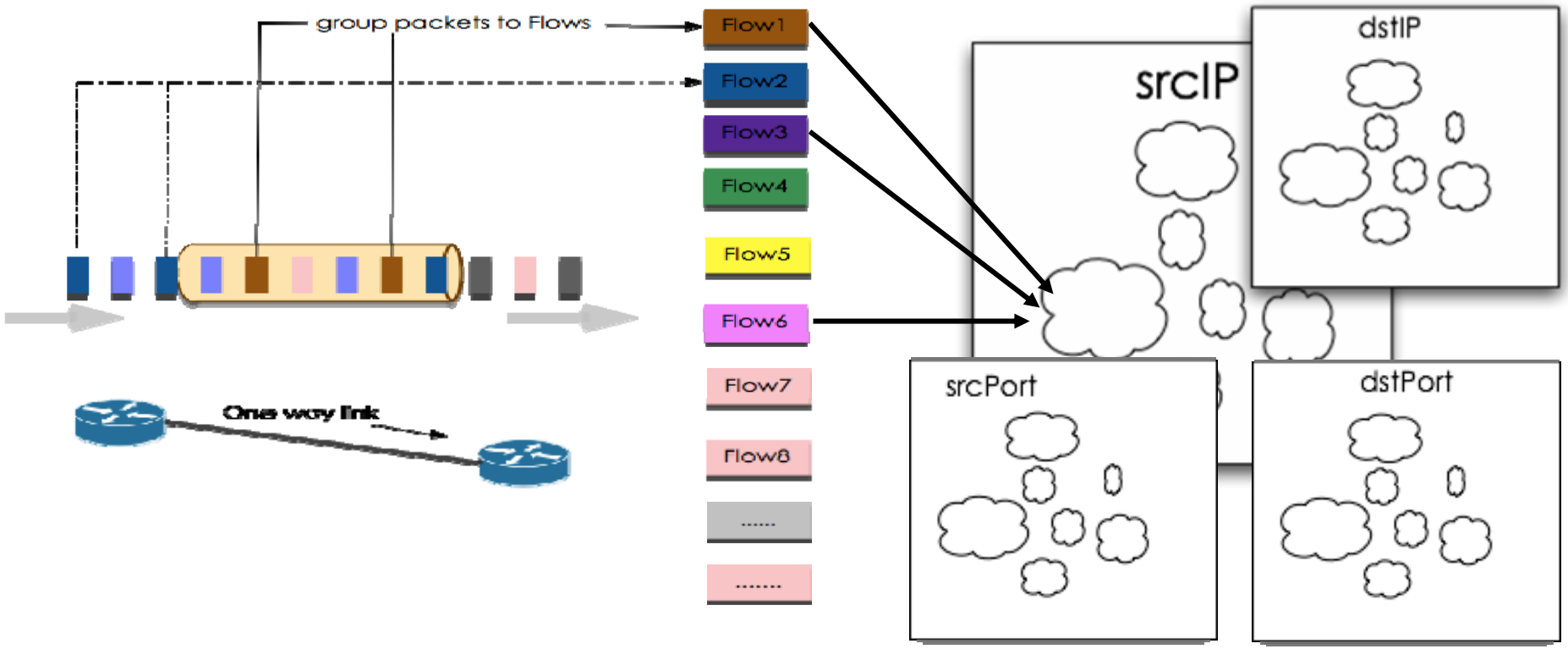
- Problems
  - how to characterize communication patterns?
  - are these patterns meaningful?
  - how to automatically discover such patterns?
- Challenges
  - vast amount of traffic data
  - large number of end hosts
  - diverse applications
- A more specific problem setting
  - use one-way traffic data from single backbone link
  - use only packet header information
  - **no assumption of normal (or anomalous) 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

Data pre-processing

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



Roadmap of our methodology

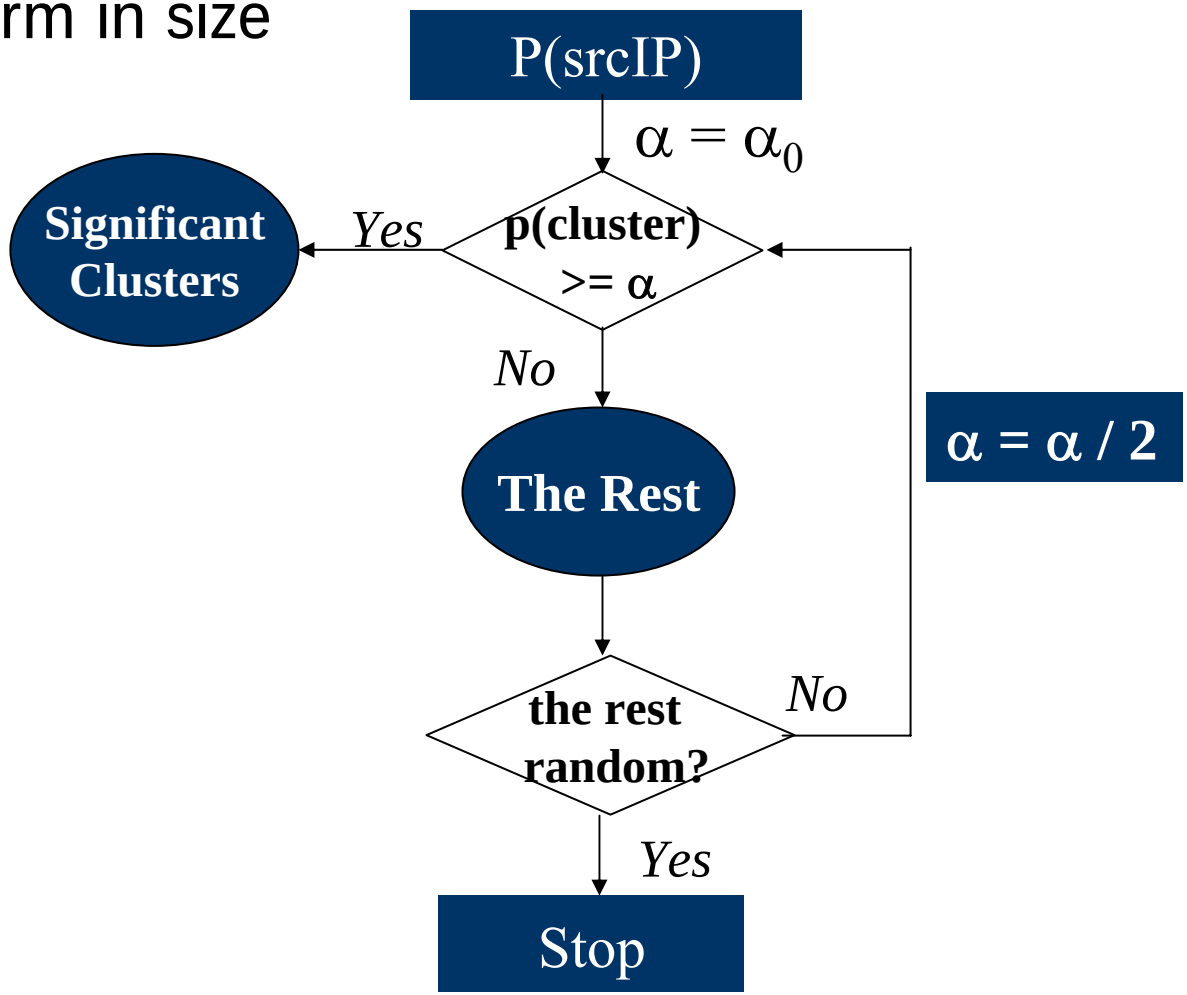
- Data pre-processing
  - aggregate packet streams into 5-tuple flows
  - group flows into clusters
- Extract significant clusters
  - data reduction step using entropy
- Classify cluster behavior based on similarity/dissimilarity of communication patterns
  - characterize using information theory
 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

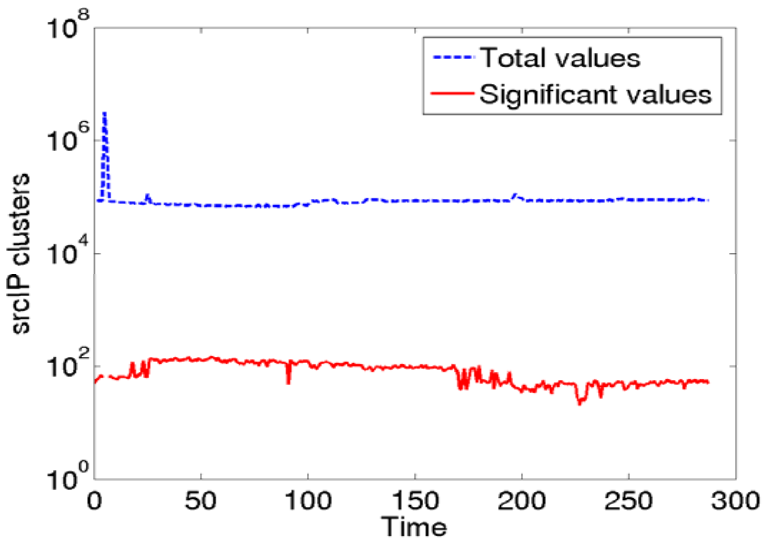
Entropy-based adaptive thresholding

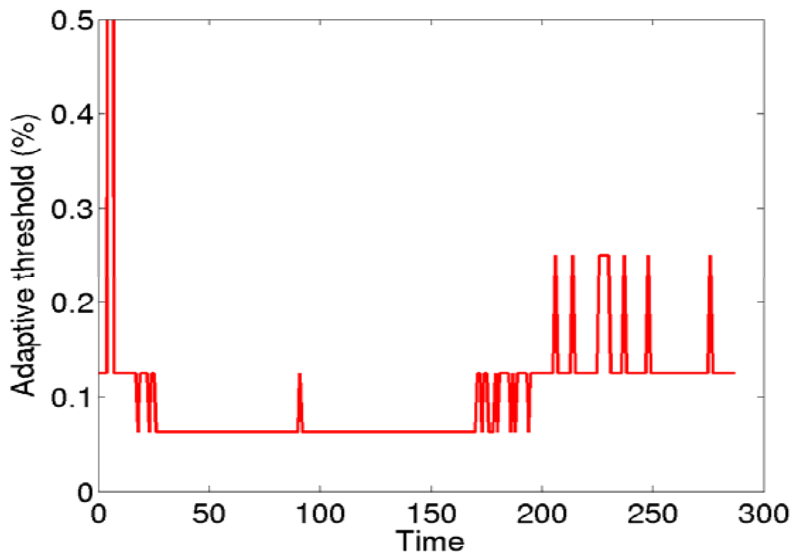
- An iterative process
  - extract significant clusters until the rest look nearly uniform in size



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

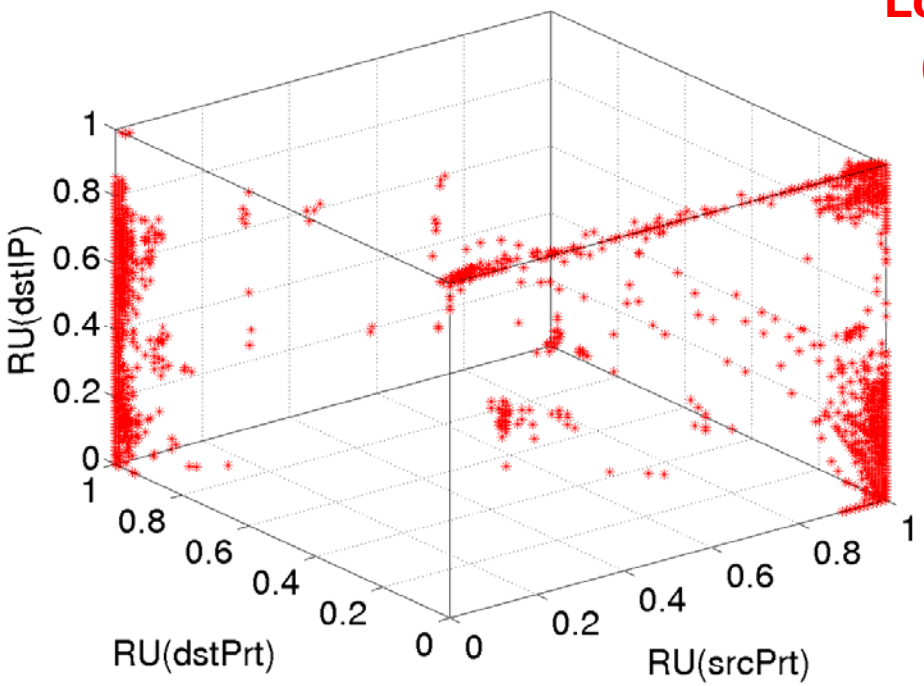
Relative uncertainty

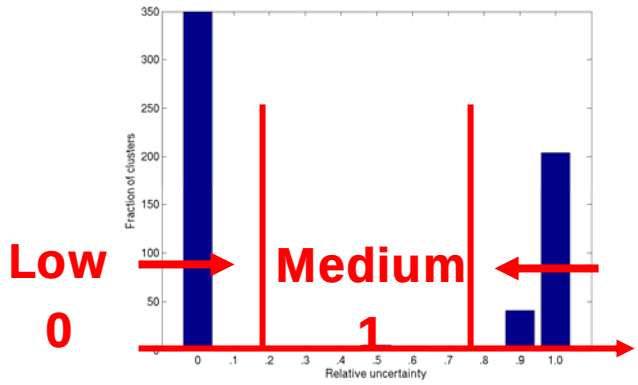
- Entropy: $H(X) = -\Sigma p(x_i) \log p(x_i)$
- Maximum Entropy: $H_{max}(X) = \log [\min(m,N)]$
- Relative Uncertainty of variable X

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

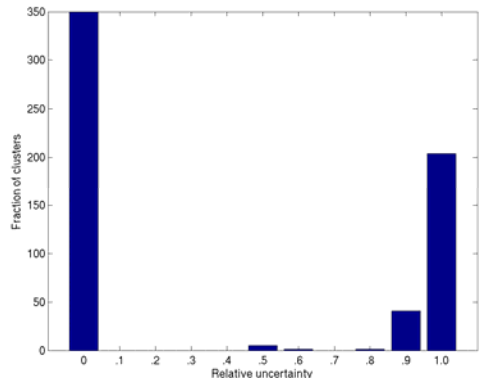
  – RU(X) = 0: X is deterministic
  – RU(X) = 1: X is randomly distributed

Behavior characterization

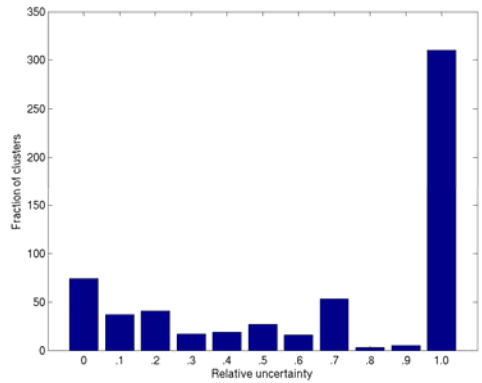




srcPort

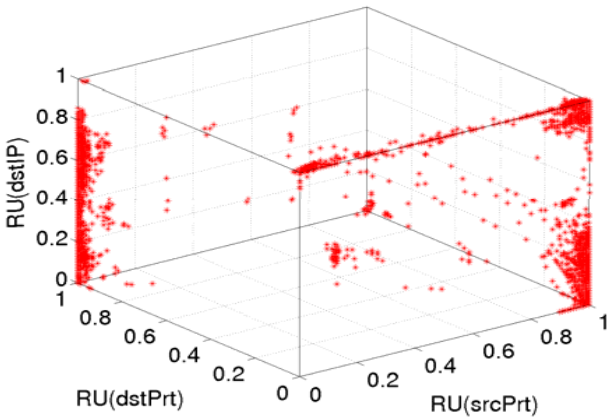


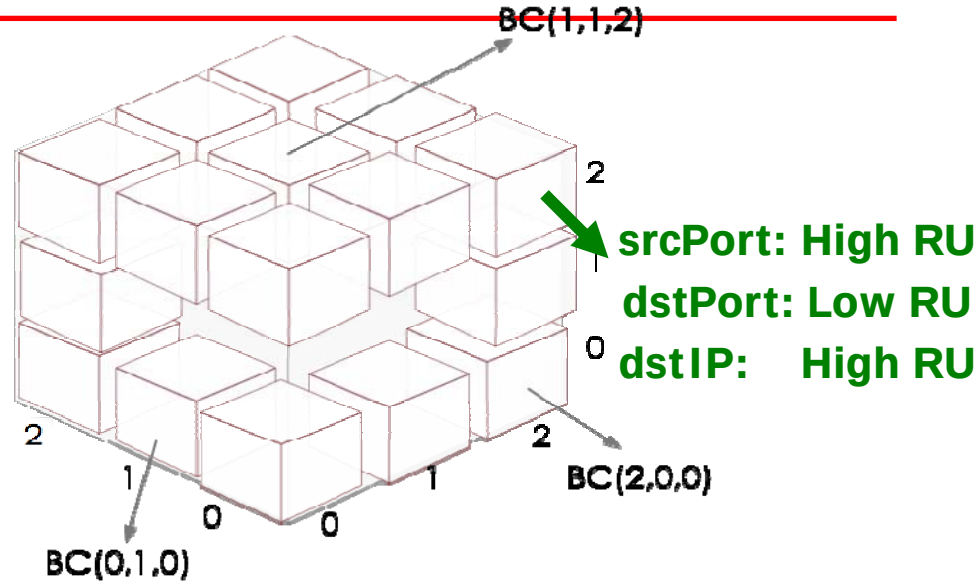
dstPort



dstIP

Behavior classifications

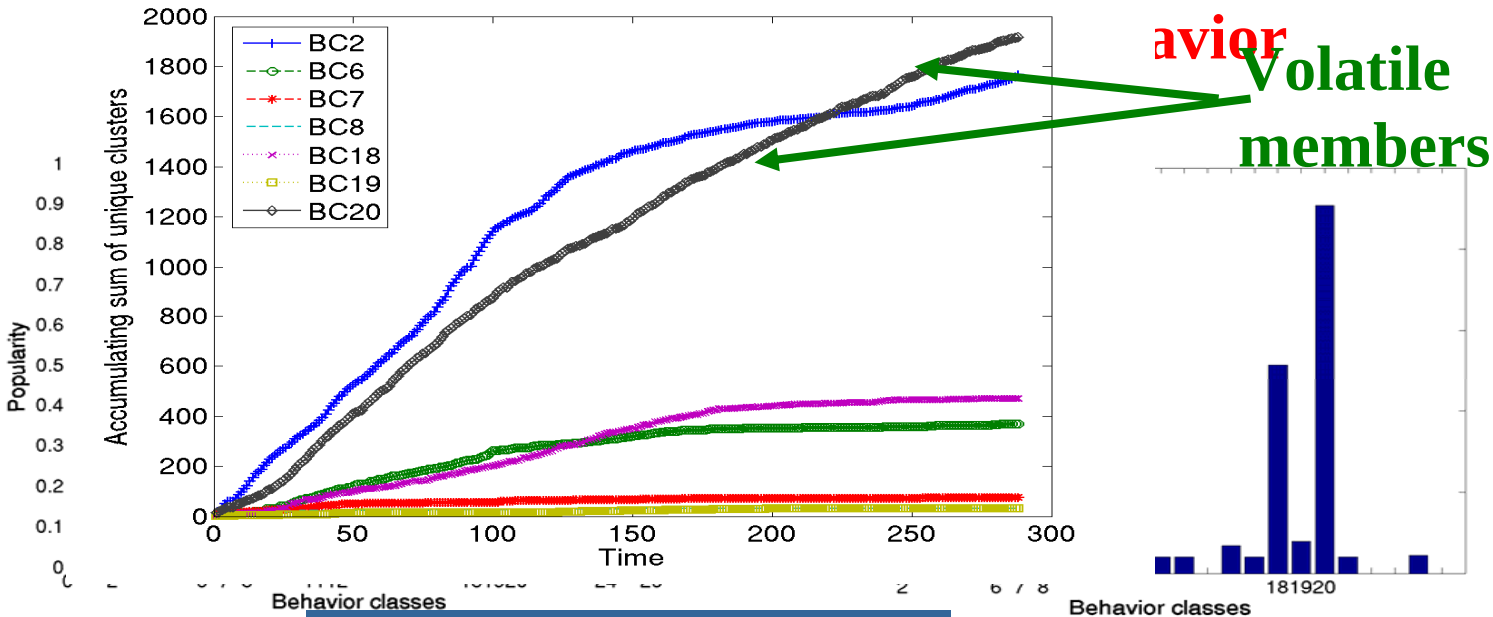




- Behavior classes (BC)
  - summarize three feature distributions into 27 classes
  - [0, 0, 0] ... [2, 2, 2], for convenience BC_0 to BC_{26}
- What is the difference between behavior classes?
  - are there common vs. rare behavior classes?
  - are BCs have many or a few clusters?
  - are memberships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

Temporal Properties

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



Popu Membership volatility

Avg. clusters

Summary of behavior classifications

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

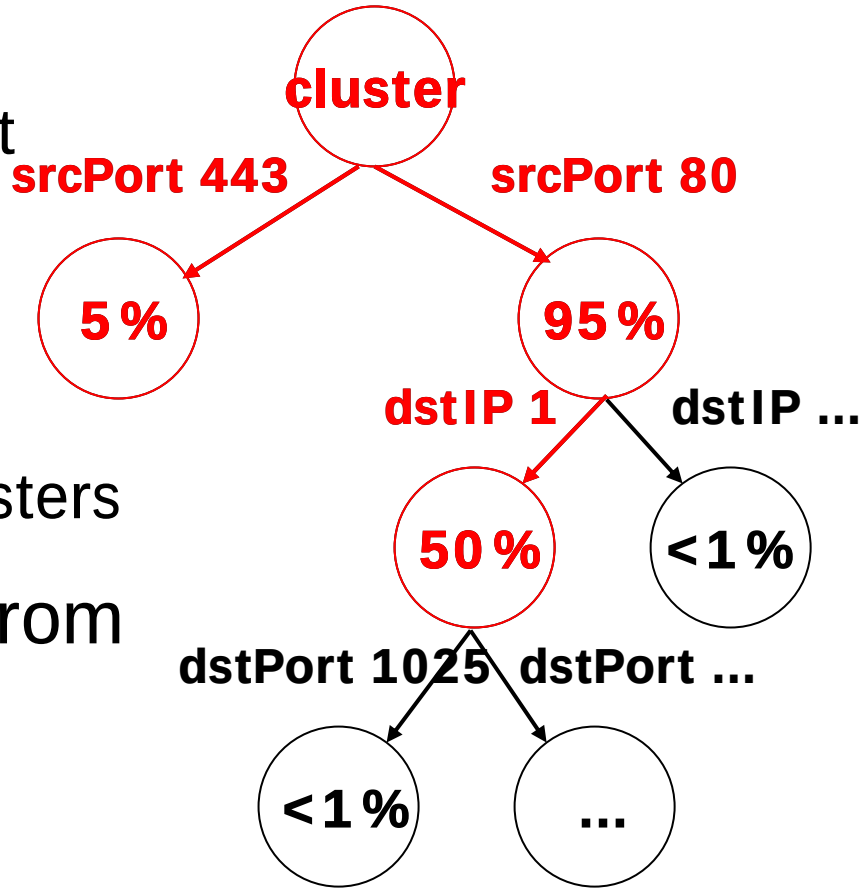
How can we interpret observed behavior?

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 of the clusters
- An example: a web server from srcIP perspective
  - $RU_{srcPort} \leq RU_{dstIP} \leq RU_{dstPort}$
  - feature dependency: srcPort, dstIP, dstPort



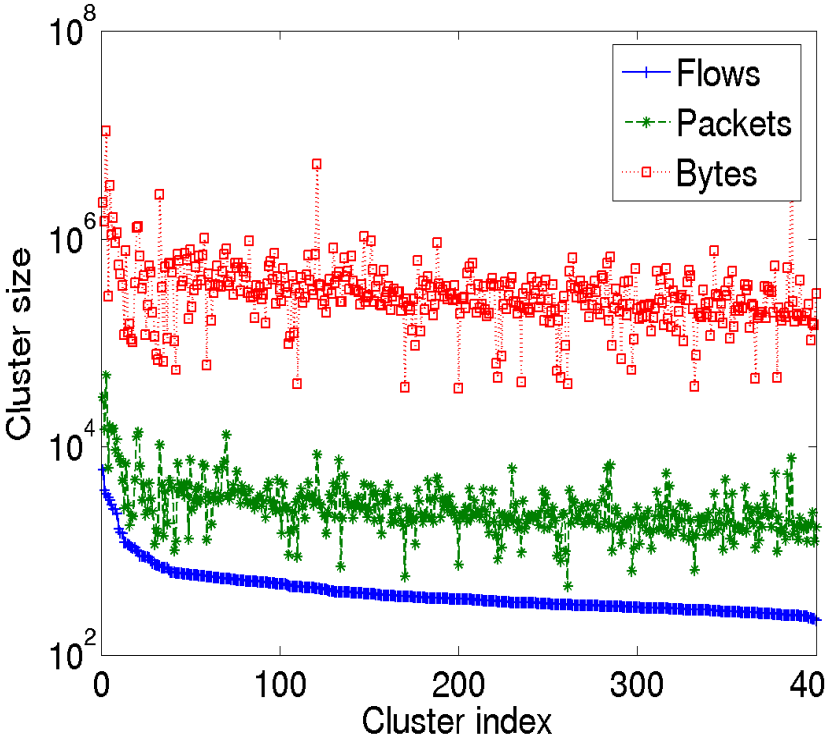
Dominant state analysis

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

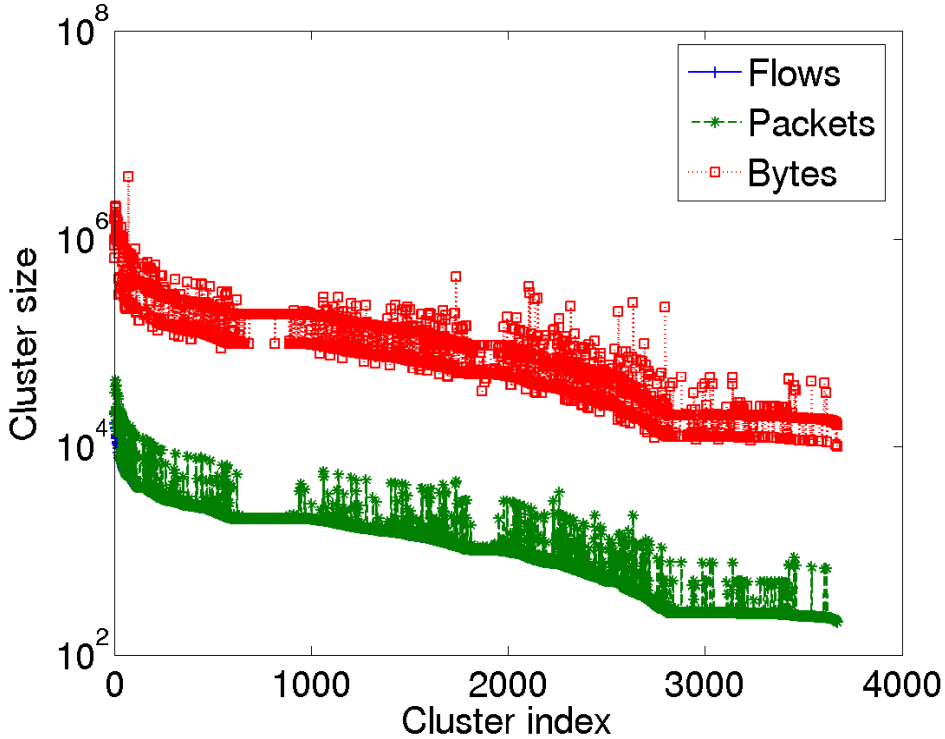
- Observations
  - clusters within the same BCs have similar structural models
  - they could have different dominant states (or activities)

Additional flow features

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



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



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

Canonical behavior profiles

Profile	Interpretation	BC	Freq.	Flow feature
Server/ service	servers talk to a large number of clients	srcIP BC{6,7,8} dstIP BC{18,19}	frequently occurring	diverse packets and bytes
Heavy hitter	hosts talk to many or several IP addresses (typically servers)	srcIP BC{18,19} dstIP BC{6,7}	frequently occurring	diverse packets and bytes
Scan/ exploit	hosts attempt to spread malicious exploits	srcIP BC{2,20}	highly volatile	single packet, same bytes

Case Studies

- Identify interesting events using typical profiles
  - server profiles on high ports, e.g., 60638
  - p2p traffic on alternative ports
  - exploit activities on unknown ports, e.g., an end host probing random dstIPs on dstPort 12827
- Rare behaviors
  - behavior patterns that rare happen are interesting
  - case study: exploit traffic from NAT boxes
- Deviant behaviors
  - clusters change from its usual BCs to a different
  - case study: a web server under DoS attack

Conclusions

- Develop a systematic methodology to automatically discover and interpret communication patterns
- Use information-theoretical techniques to build behavior models of end hosts and applications
- Apply dominant state analysis to explain traffic behavior
- Discover typical behavior profiles as well as rare and deviant behaviors

Future work

- Correlating behavior profiles across multiple links
- Validate behavior profiles using additional features, e.g., packet payload
- Integrate traffic profiling framework with a real-time monitoring system