Traffic Classification in the Fog

Scott E. Coull

February 23, 2006

Overview

- What is traffic classification?
- Communities of Interest for classification
- BLINC
- Profiling Internet Backbone Traffic
- What is missing here?

Traffic Classification

- Determine application-level behavior from packet-level information

- Why bother?
  - Traffic shaping/QoS
  - Security policy creation
  - Detect new/abusive applications

Levels of Classification

- Payload classification – In the clear
  - Becomes a type of text classification
  - Not so interesting, or realistic

- Transport-layer Classification – In the fog
  - Typical 4-tuple (Src. IP, Dst. IP, Src. Port, Dst.Port)
  - Sufficient condition for proving application-layer behavior?

Levels of Classification

- In the Dark Classification
  - Tunneling, NAT, proxying
  - Fully encrypted packets
  - What is left for us?
    - Packet size, inter-arrival times, direction

Communities of Interest

- "…a collection of entities that share a common goal or environment." [Aiello et. al. 2005]

- Uses -
  - Finding groups of malicious users in IRC [Camptepe et. al. 2004]
  - Groups of similar web pages [Google's PageRank]
  - Defining security policy?

Enterprise Security: A Community of Interest Based Approach

Aiello et. al. – NDSS ‘06

- Motivation – Move enterprise protection from perimeter to hosts
  - Perimeter defenses weakening
- Claims:
  - Hosts provide best place to stop malicious behavior
  - Past connection history indicates future connections

Communities of Interest for Enterprise Security

- General Approach:
  1. Gather network data and ‘clean’ it
  2. Create a profile for each host from past behavior
  3. Create security policy to ‘throttle’ connections based on profiles

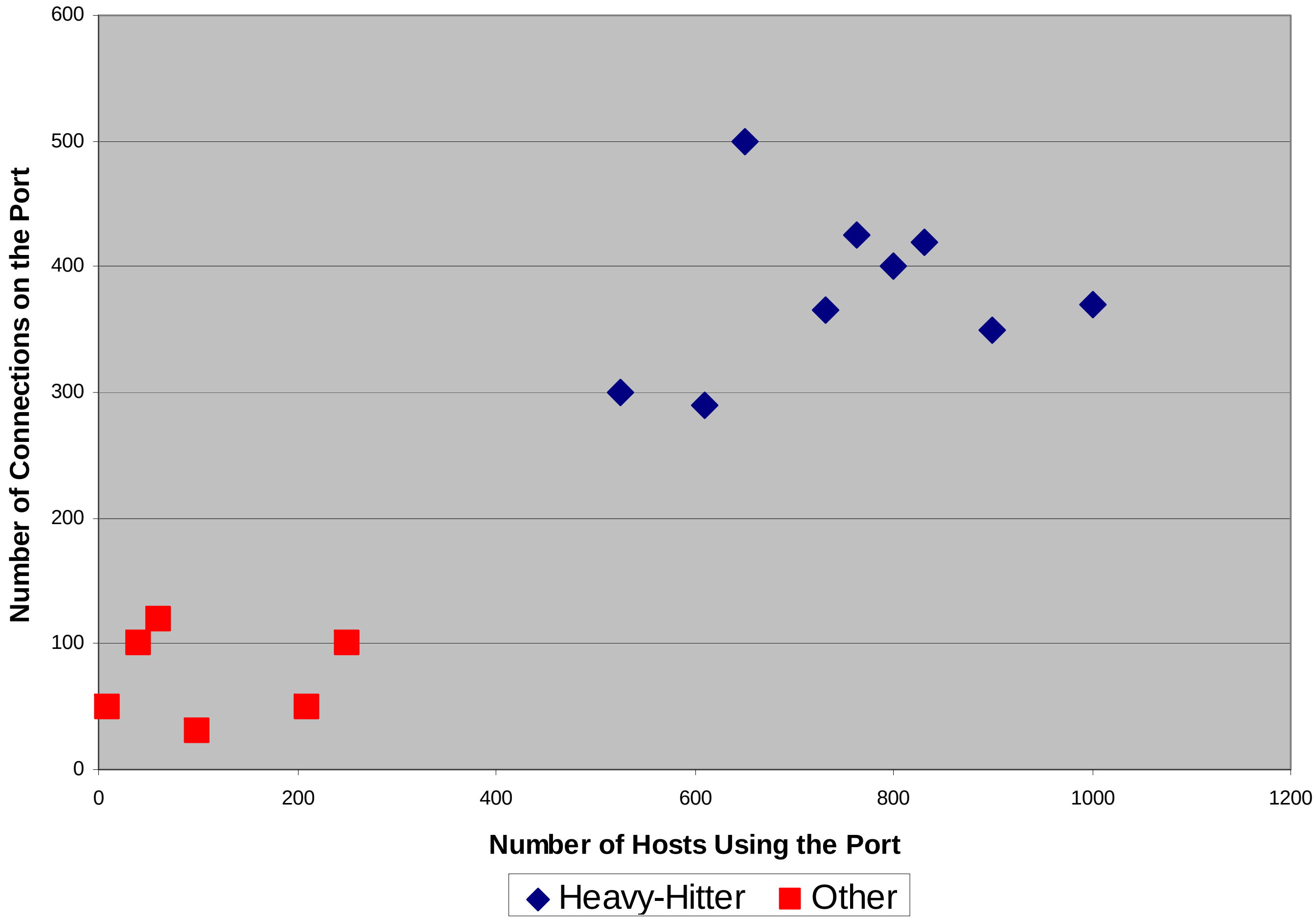
Communication Profiles

- Protocol, Client IP, Server Port, Server IP
  - Very specific communication between a host and server
  - Ex: (TCP, 123.45.67.8, 80, 123.45.67.89)
- Protocol, Client IP, Server IP
  - General communication profile between a host and server
  - Ex: (TCP, 123.45.67.8, 123.45.67.89)

Communication Profiles

- Protocol, Server IP
  - Global profile of server communication
  - Ex: (TCP, 123.45.67.89)
- Extended COI
  - k-means clustering
  - Specialized profile of most used communication channels
  - Global, server-specific, ephemeral, unclassified ports

Extended COI – An Example



Throttling Disciplines

- *n-r-Strict*
  - Very strictly enforce profile behavior with strong punishment
  - No outside profile interaction
  - Block all traffic if > n out of profile interactions in r time
- *n-r-Relaxed*
  - Allow some relaxation of profile behavior, but keep punishment
  - n outside profile interactions allowed in time r
  - Block all traffic if > n out of profile interactions in r time
- *n-r-Open*
  - Allow some relaxation of profile, but minimize punishment
  - n outside profile interactions allowed in time r
  - Block out of profile traffic if > n out of profile interactions in r time

Experimental Methodology

- Test profiles and ‘throttling’ against worm
- Not-so-realistic worm
  - Assume all hosts with worm’s target port in profile are susceptible
  - Fixed probability of infection during each time period
    - No connection with susceptible population distribution or scanning method
  - No exact description of worm scanning
    - ‘Scanning’ based on infection probability

Results and Observations

Infection Probability

Out of Profile Attempts

TD Policy

Profile Types

Port	Policy	$s(\%)$	n	PSP	PCSP	PCSPP	Intelligent
135/tcp	strict	1	10	0.768%	0.741%	1.852%	1.852%
135/tcp	strict	5	10	0.872%	0.741%	1.852%	1.852%
135/tcp	strict	5	100	14.044%	0.785%	1.852%	1.852%
135/tcp	strict	10	100	31.048%	0.818%	1.852%	1.852%
135/tcp	strict	25	1000	33.421%	10.126%	1.852%	1.852%
135/tcp	strict	100	1000	33.421%	12.617%	1.852%	1.852%
135/tcp	relaxed	1	10	0.842%	0.793%	2.143%	2.109%
135/tcp	relaxed	5	10	1.383%	1.495%	3.841%	3.738%
135/tcp	relaxed	5	100	98.938%	98.996%	99.280%	99.331%
135/tcp	relaxed	10	100	99.997%	99.995%	100.000%	100.000%
135/tcp	relaxed	25	1000	100.000%	100.000%	100.000%	100.000%
135/tcp	relaxed	100	1000	100.000%	100.000%	100.000%	100.000%
135/tcp	open	1	10	92.060%	61.871%	1.989%	1.972%
135/tcp	open	5	10	95.734%	50.209%	16.907%	10.065%
135/tcp	open	5	10	98.621%	98.886%	99.949%	99.074%
135/tcp	open	10	100	100.000%	100.000%	99.983%	100.000%
135/tcp	open	25	1000	100.000%	100.000%	100.000%	100.000%
135/tcp	open	100	1000	100.000%	100.000%	100.000%	100.000%

How can we subvert this?

- Topological worms
  - Spread using topology information derived from infected machine
  - Local connection behavior appears normal
  - Weaver et. al.

    A Taxonomy of Computer Worms, WORM ‘03
- Non-uniform scanning worms
- Traffic tunneling

Blind Classification (BLINC)
Karagiannis et. al. – SIGCOMM ‘05

- Motivation - payloads can be encrypted, forcing classification to be done ‘in the dark’
  - Use remaining information in flow records
- Claim:
  - Transport-layer info indicates service behavior

‘In the Dark’

- No access to payloads
- No assumption of well-known port numbers

- Only information found in flow records can be used
  - Source and Destination IP addresses
  - Packet and byte counts
  - Timestamps
  - TCP flags

Robust 'In the Dark' Definition

- No information that would not be visible over an encrypted link

- Sun et. al.

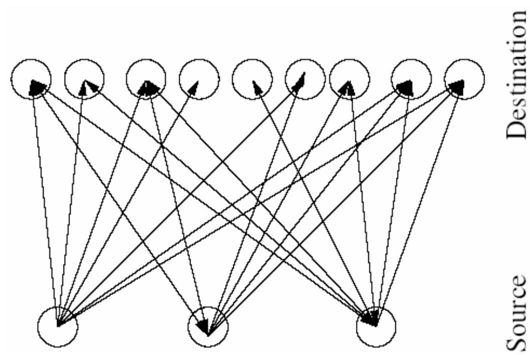
  Statistical Identification of Encrypted Web Browsing Traffic, Oakland '02
  - Examine size and number of objects per page
  - Use similarity metric between observed encrypted page requests and 'signatures'
  - Identify roughly 80% of web pages with near 1% false positive rate

Improvements over COI

- “Multi-level traffic classification”
  - Capture historical ‘social’ interaction among hosts
  - Capture source and destination port usage
- Novel ‘graphlet’ structure

Social Interaction

- Claim: Bipartite cliques indicate underlying protocol type
  - “Perfect” cliques indicate worm traffic



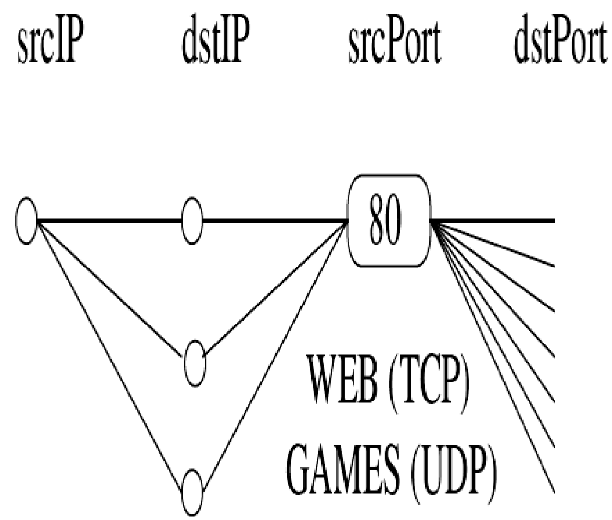
  - Partial overlap indicates p2p, games, web, etc.
  - Partial overlap in same “IP neighborhood” indicates server farm

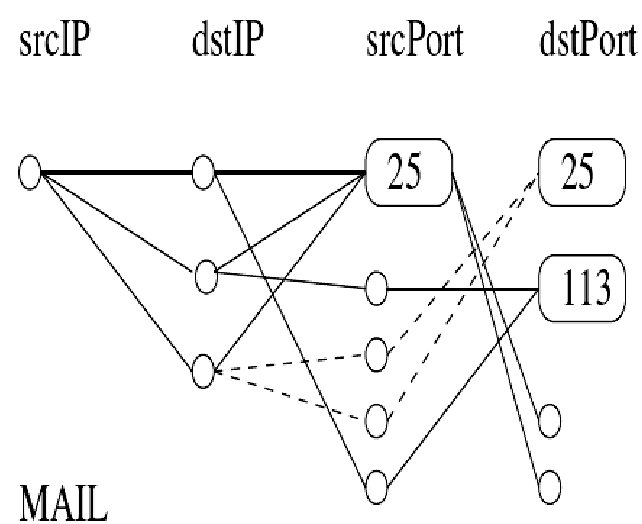
Functional Interaction

- Claim: Source ports indicate host behavior
  - Client behavior indicated by many source ports
  - Server behavior indicated by a single source port
  - Collaborative behavior not easily defined
  - Some protocols don’t follow this model
    - Multi-modal behavior

Graphlets

- Application level – Combine functional and social level into a 'graphlet'
  - Example:





Heuristics

- Claim: Application layer behavior is differentiated by several heuristics
  - Transport layer protocol
  - Cardinality of destination IPs vs. Ports
  - Average packet size per flow
  - Community
  - Recursive detection

Thresholds

- Several thresholds to tune classification specificity
  - Minimum number of destination IPs before classification
  - Relative cardinality of destination IPs vs. Ports
  - Distinct packet sizes
  - Payload vs. nonpayload flows

Experimental Methodology

- Compare BLINC to payload classification
  - Compare completeness and accuracy
  - Ad hoc payload classification method
  - Non-payload data is never classified
    - ICMP, scans, etc…

Experimental Methodology

- Payload classification
  - Manually derive ‘signature’ payloads from observed flows, documentation, or RFCs
  - Classify flows based on ‘signature’ and create (IP, Port) mapping table to associate pair with application
  - Use this pair to classify packets with no ‘signature’ in the payload
  - Remove remaining ‘unknown’ mappings

- Similar to classification performed by: Zhang, Y. Z., and Paxson, V.

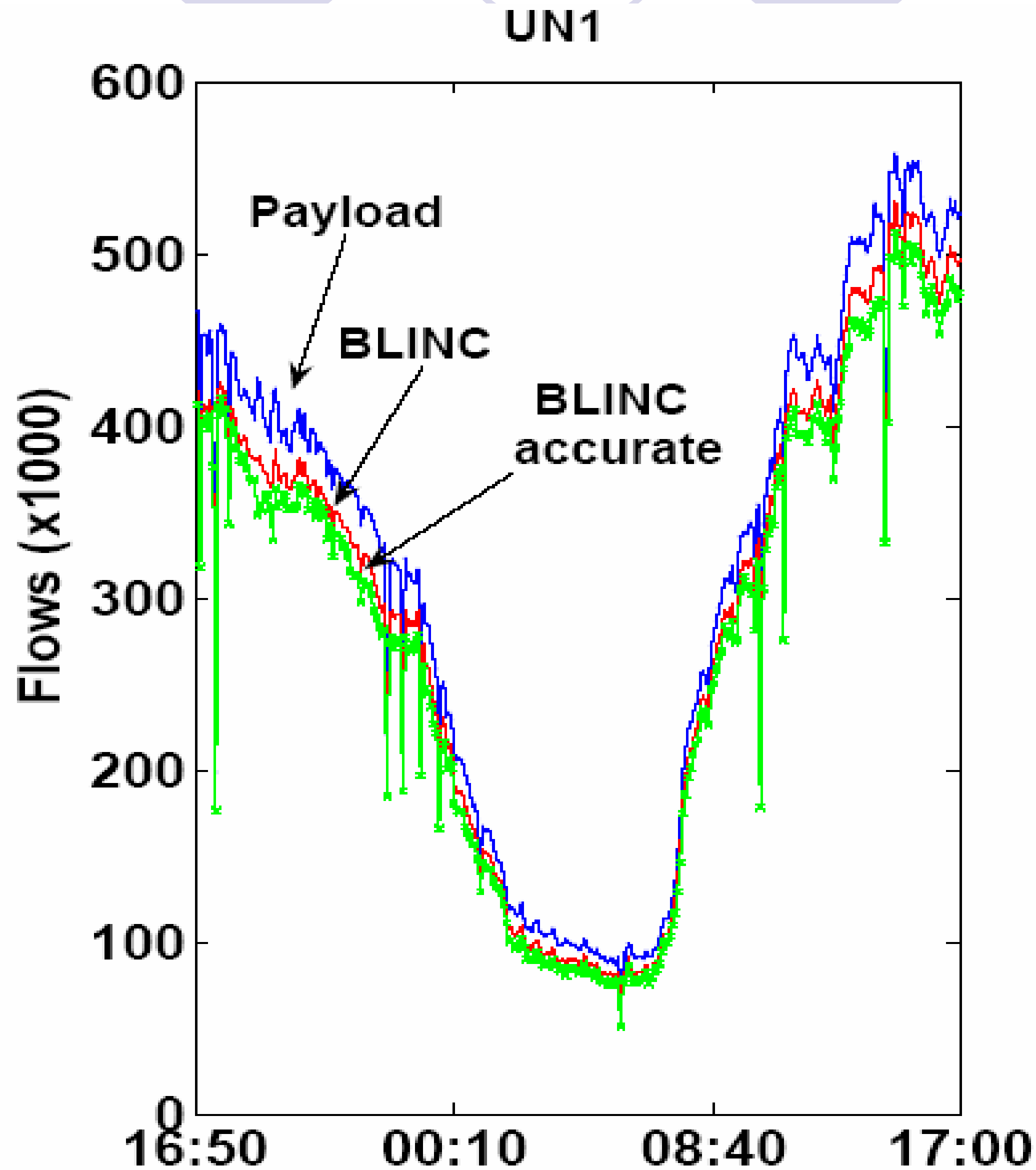
  Detecting Backdoors, USENIX Sec. ‘00

Evaluation

- The Data
  - Collected from Genome Lab and University
  - Collected several months apart to ensure variety
  - Important questions are ignored
    - How long was the data collected for?
    - Which parts, if any, were used to create the ‘graphlets’?
    - How were accuracy and completeness measured?

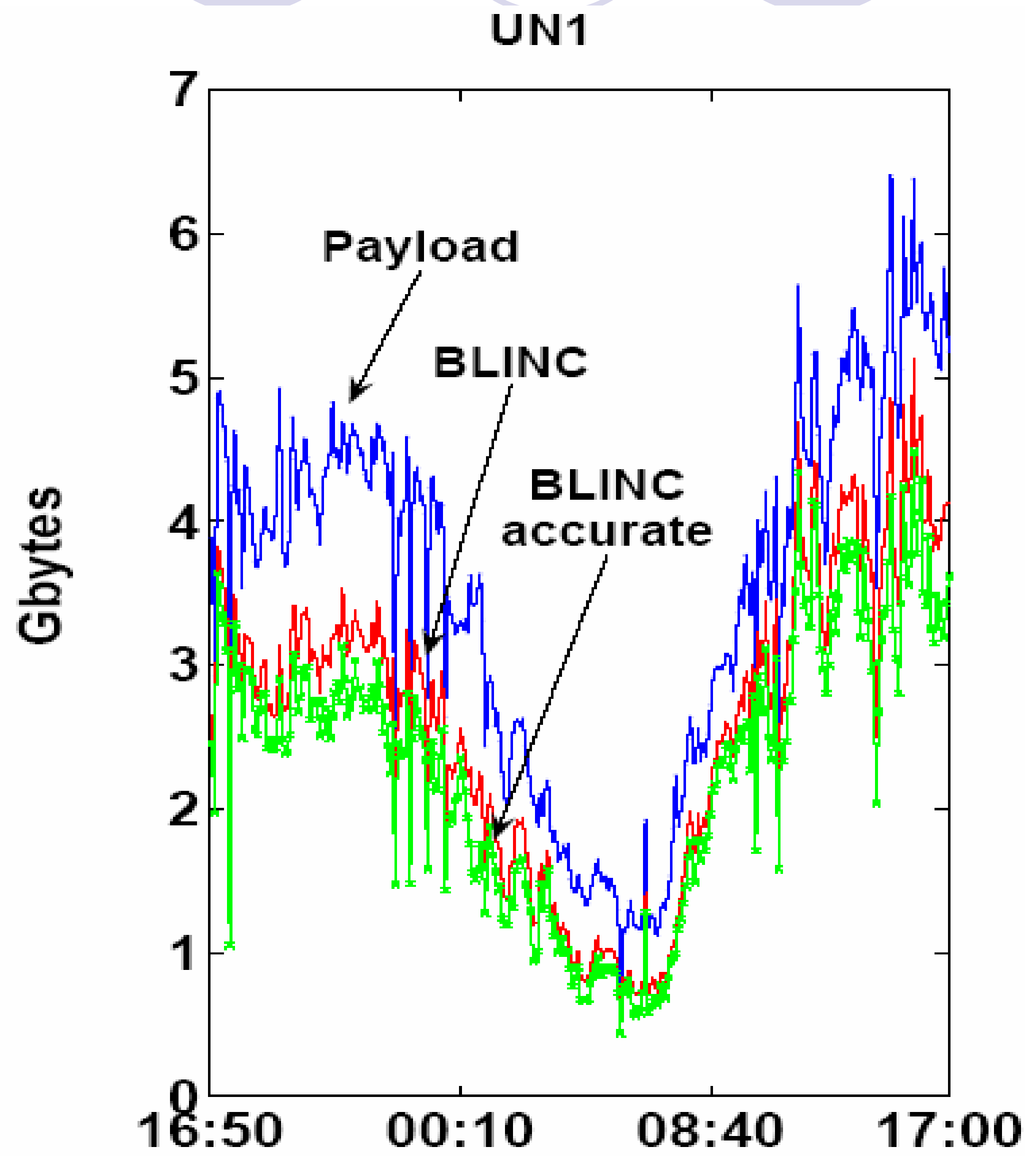
Results – Per Flow

- BLINC classifies almost as many flows as payload classification



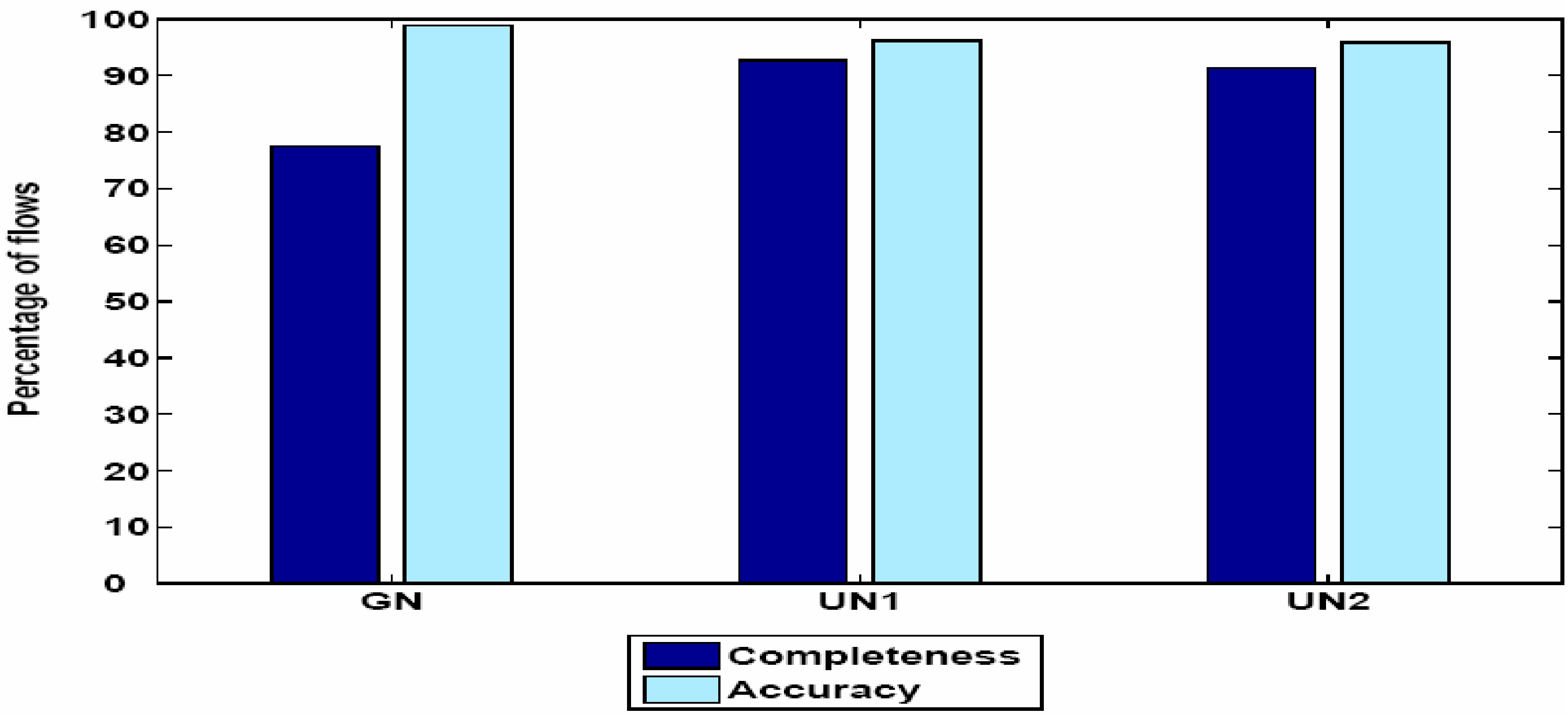
Results – Per GByte

- Significant difference in size of the flows classified by payload versus BLINC



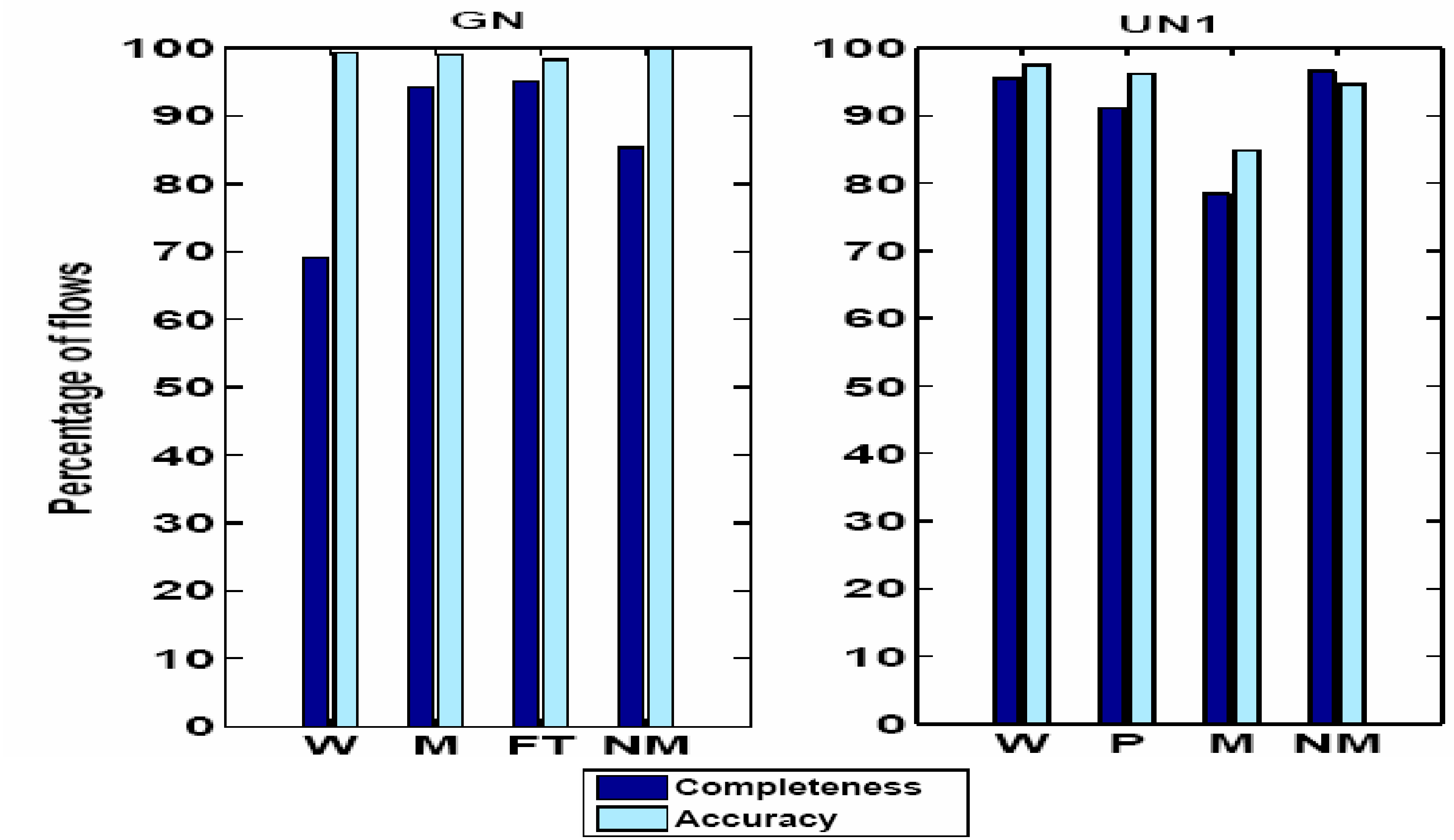
Completeness and Accuracy

- Extremely high accuracy
- Large disparity in completeness for GN



Protocol-Family Results

- Web and Mail classification appear to be highly inconsistent



Recap of BLINC

- Determine social connectivity
- Determine port usage
- Create ‘graphlet’
- Add some additional heuristics
- Test against data that was classified with payload in ad hoc fashion

Unanswered Questions

- How are ‘graphlets’ created?
- What are the effects of their heuristics and how are they used?
- What kind of ‘tunability’ can we achieve from the thresholds?
- Why do they do so well with so little information?

Graphlet Creation

- *In developing the graphlets, we used all possible means available: public documents, empirical observations, trial and error.*

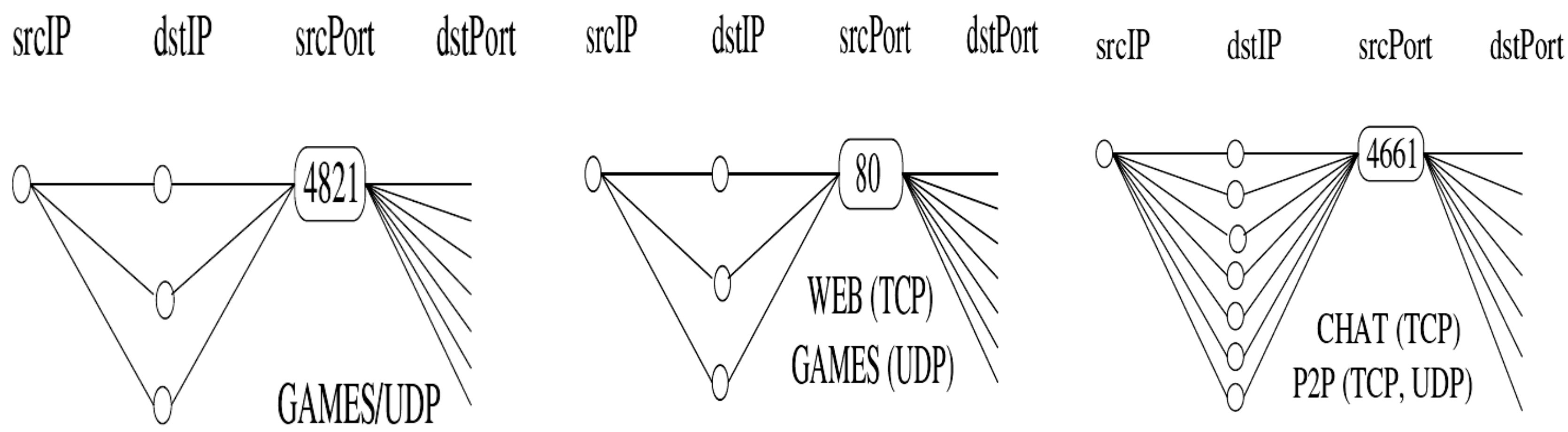
- Is this practical?

Graphlet Creation

- *Note that while some of the graphlets display port numbers, the classification and the formation of graphlets* ***do not associate in any way a specific port number with an application***

- Implication:
  - No one-to-one mapping of port numbers to applications

Graphlet Usage



- Significant similarity in graphlet structure
- Reliance on port numbers for differentiation
- Heuristics and thresholds also play a significant role

Application of Heuristics

- Heuristics recap:
  - Transport protocol, cardinality, packet size, community, recursive detection

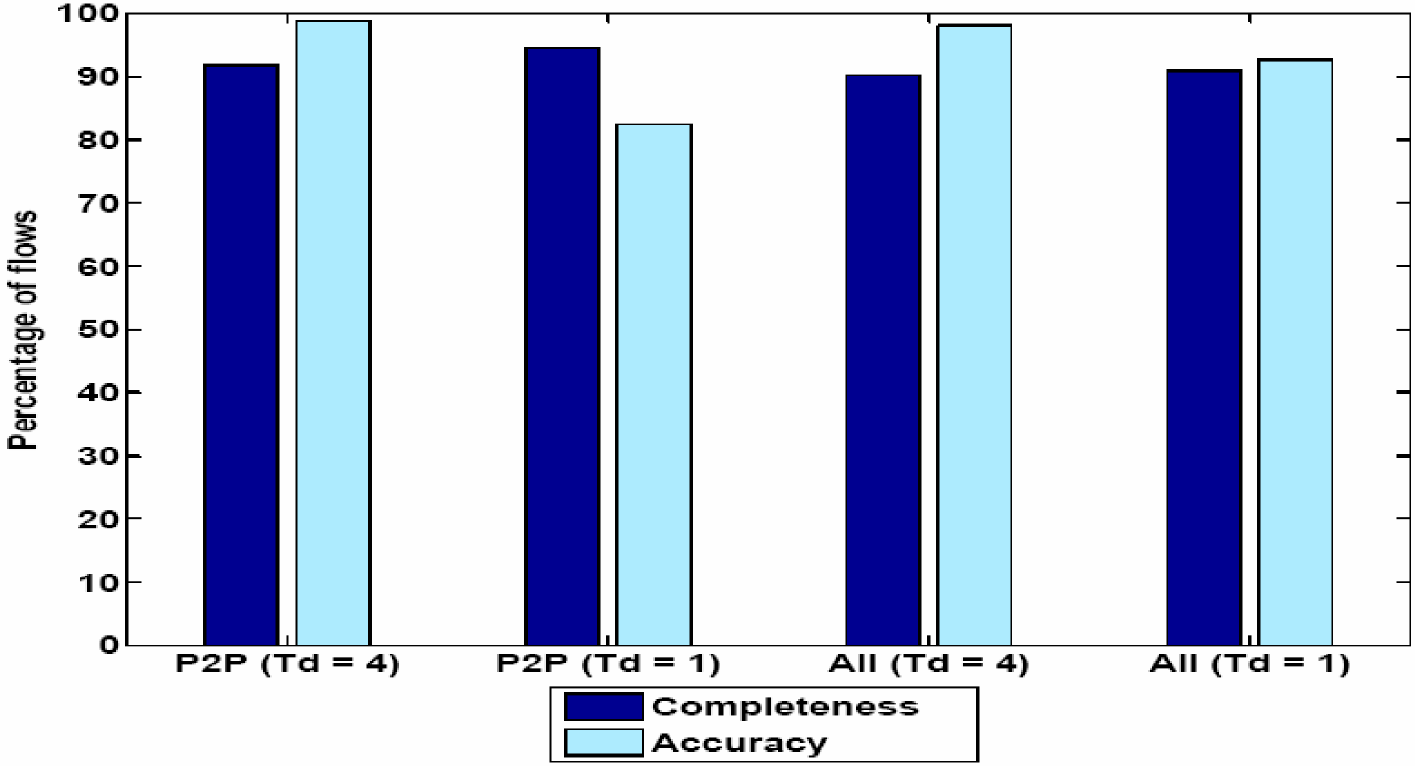
- Transport protocol can be added to the ‘graphlet’
- Cardinality and size in the thresholds
- Recursive detection and community
  - Not discussed in the paper

Application of Thresholds

- Threshold recap:
  - Distinct destinations, relative cardinality, distinct packet sizes, payload vs. non-payload packets
- Only distinct destination is ever discussed
  - Are two settings really enough to generalize the behavior?

System Tunability

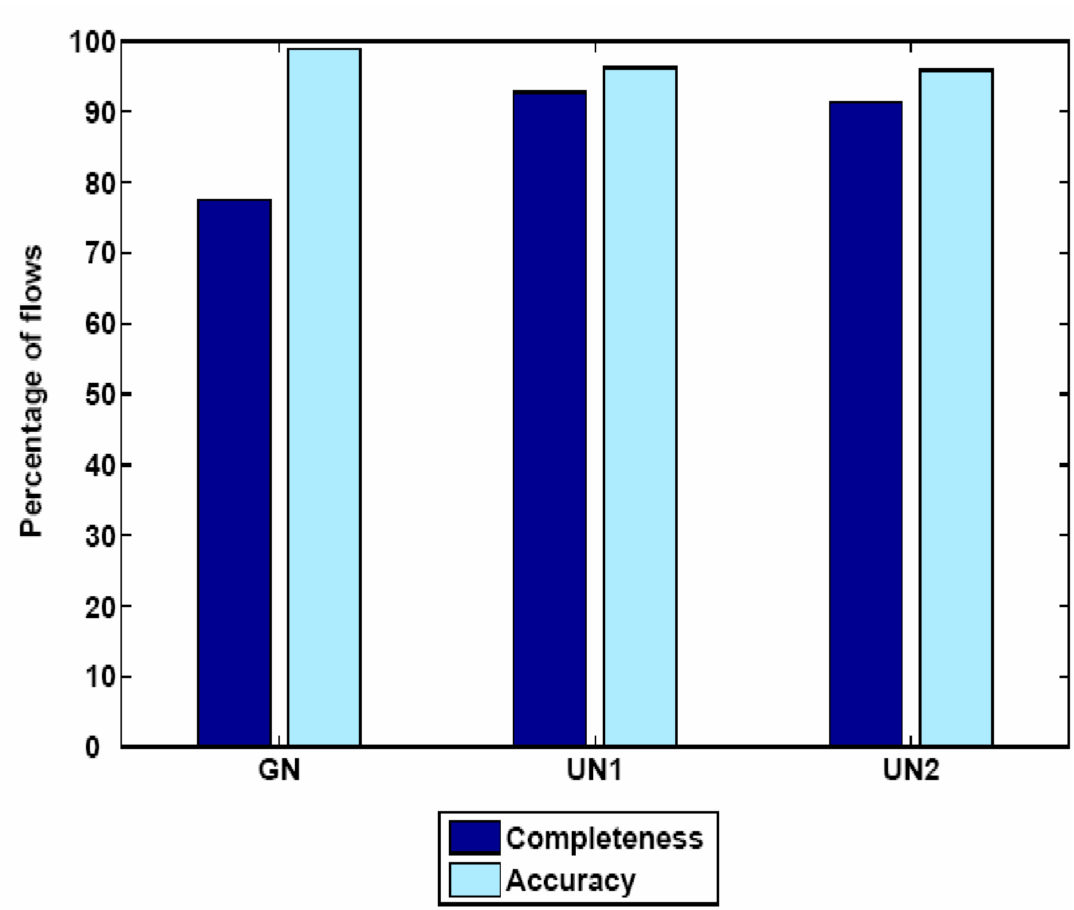
- <u>Claim</u>: Increasing the number of distinct IPs required will **increase** accuracy and **decrease** completeness



Why do they do so well?

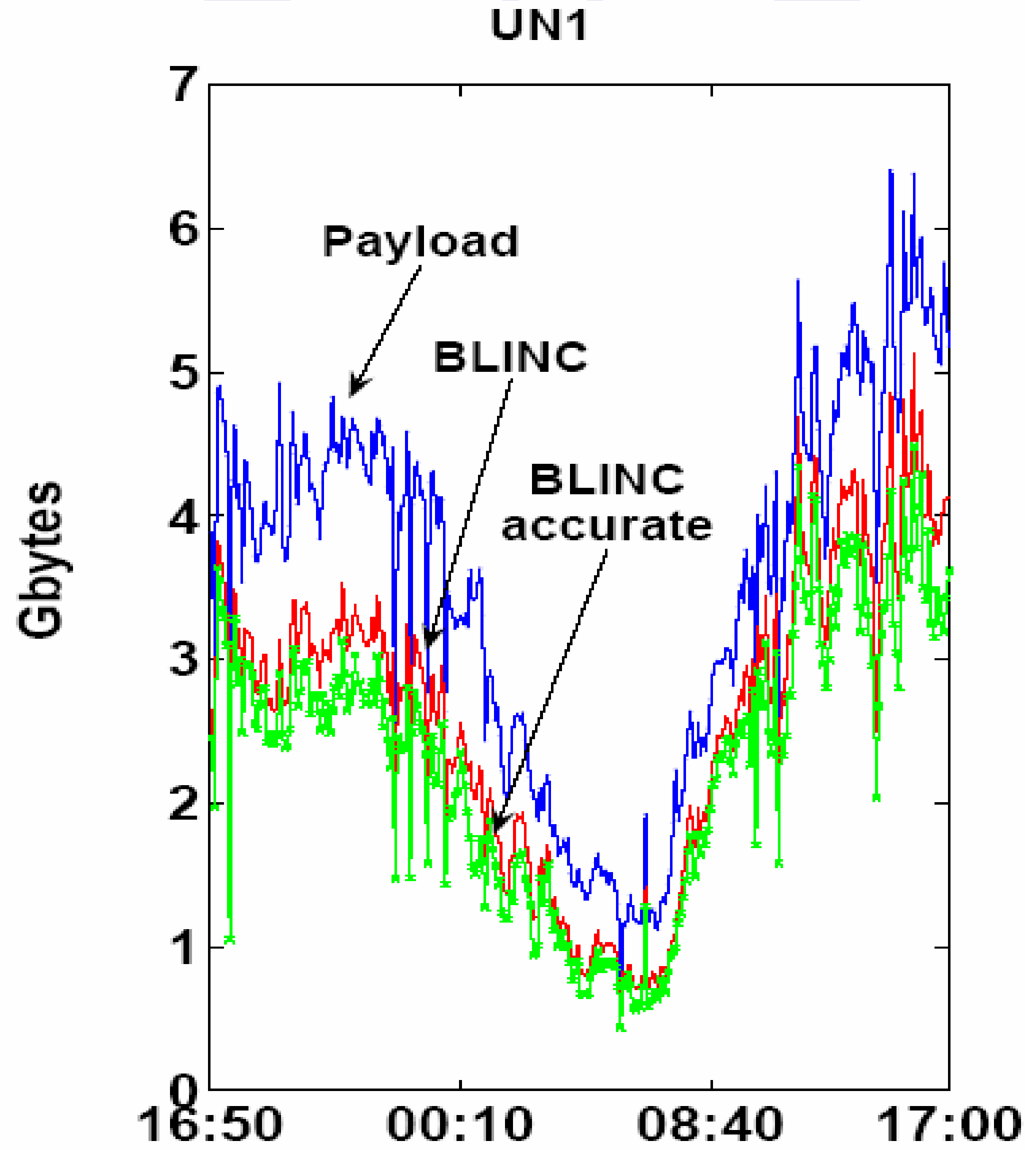
- Top applications:
  - Web
  - P2P
  - Non-payload
- 77.6% of flows at GN
- 82.2% at UN1
- 74.2% at UN2

- BLINC only classifies approximately 75-80% of GN flows



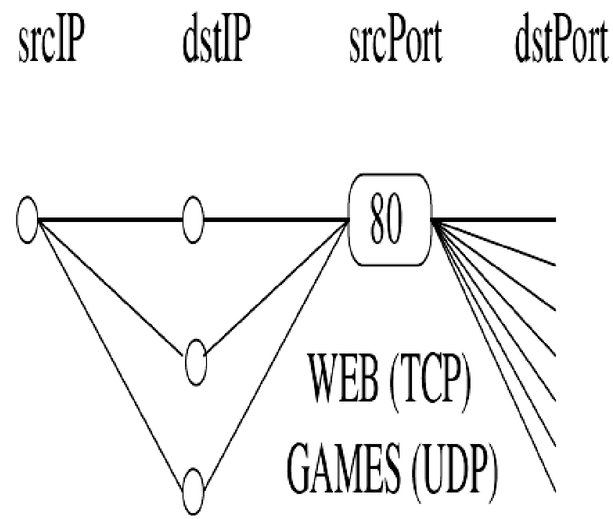
Why do they do so well?

- Non-payload flows are never classified by the payload classifier
- Large proportion of non-payload flows explains size difference



Subverting BLINC

- Mimicry attack
  - Replicate connectivity
  - Replicate port number
  - Replicate destination port behavior
  - Be aware of thresholds
- Traffic tunneling
- NAT devices



Profiling Internet Backbone Traffic

Xu et. al. – SIGCOMM ‘05

- Motivation – Profile backbone traffic to automatically find significant behavior
  - Interpret behavior to identify classes of traffic
  - Allow for easy summary to network ops

Information Theory Refresher

- Entropy
  - Measure of uncertainty in empirical data

$$H(X) := -\sum_{x_i \in X} p(x_i)\log(p(x_i))$$

- Relative Uncertainty
  - Measures uniformity of empirical data regardless of sample (m) or support size (N_x)

$$RU(X) := \frac{H(X)}{\log(\min\{N_x, m\})}$$

Information Theory Refresher

- Conditional Relative Uncertainty
  - RU conditioned on a specific set
  - The sample size (m) equals the cardinality of the set (A)

$$RU(X \mid A) := \frac{H(X)}{\log(|A|)}$$

  - Values near 1 indicate uniform distribution of values in set A

Connection to Classification

- Utilize the standard 4-tuple
  - (Src. IP, Dst. IP, Src. Port, Dst. Port)
  - Each dimension (e.g. Src. IP) in the tuple is analyzed individually to determine significant values
  - Set of all observed values in the dimension is the set A
    - e.g. A is the set of all source IPs seen in the data

Entropy-based Cluster Extraction

- Gather the most significant values from each dimension of the 4-tuple based on Conditional Relative Uncertainty
  - We will call these the ‘fixed’ dimensions from here on

Algorithm 1 Entropy-based Significant Cluster Extraction

Parameters: $\alpha := \alpha_0$; $\beta := 0.9$; $S := \emptyset$;
Initialization: $S := \emptyset$; $R := A$; $k := 0$;
compute prob. dist. $\mathcal{P}_R$ and its RU $\theta := RU(\mathcal{P}_R)$;
while $\theta \leq \beta$ **do**
  $\alpha = \alpha \times 2^{-k}$; $k++$;
  for each $a_i \in R$ **do**
    if $\mathcal{P}_A(a_i) \geq \alpha$ **then**
      $S := S \cup \{a_i\}$; $R := R - \{a_i\}$;
    end if
  end for
  compute (cond.) prob. dist. $\mathcal{P}_R$ and $\theta := RU(\mathcal{P}_R)$;
end while

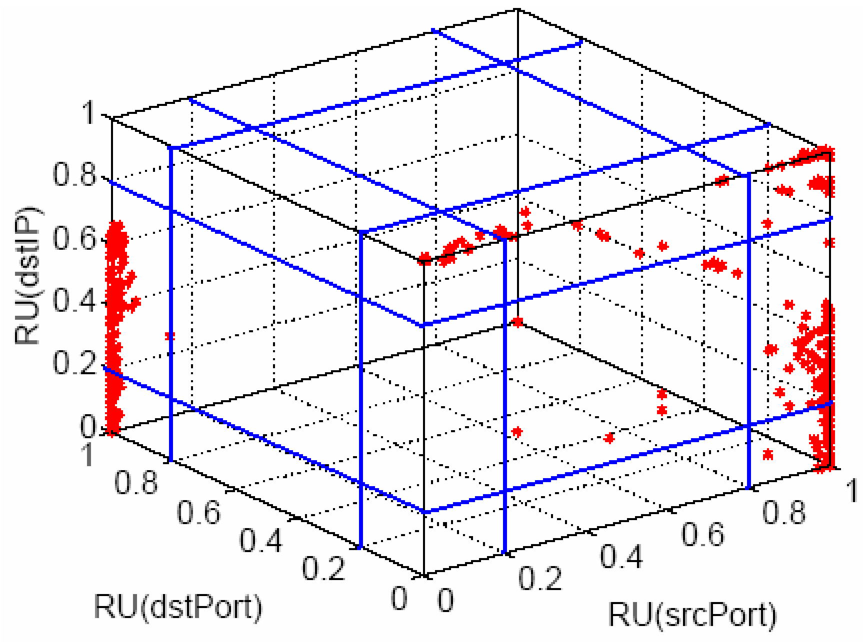
Entropy-based Cluster Extraction

- For each fixed dimension of the tuple
  - Partition the remaining 3-tuple dimensions based on RU
  - e.g. With fixed dimension of Src. IP, partition the Dst. IP, Src. Port, and Dst. Port dimensions individually

$$L(ru) = \begin{cases} 0(low), & \text{if } 0 \leq ru \leq \epsilon, \\ 1(medium), & \text{if } \epsilon < ru < 1 - \epsilon, \\ 2(high), & \text{if } 1 - \epsilon \leq ru \leq 1, \end{cases}$$

Behavioral Classes

- 27 classes based on the RU category of each of the dimensions in the remaining 3-tuple
  - e.g. With fixed dimension Src. IP, [0,2,2] indicates stable Src. Ports, but highly variable Dst. IPs and Ports



Dominant State Analysis

- Specific instantiations of the behavioral class that occur often

- Step 1:
  - For each 3-tuple within the class, order the dimensions by their RU

Dominant State Analysis

- Step 2:
  - Compute marginal probability of the lowest RU dimension and select all values greater than the threshold, δ
  - e.g. Src. Port is lowest RU dimension and
    $a \in SrcPort$

$$\boxed{p(a) := \sum_{b \in DstIP} \sum_{c \in DstPort} p(a,b,c) \geq \delta}$$

Dominant State Analysis

- Step 3:
  - Compute conditional marginal probability for each of the values of the next lowest dimension
  - e.g. Given a particular Src. Port value, calculate the probability of the Dst. IP values

$$\boxed{p(b_j \mid a_i) := \frac{\sum_{c \in DstPort} p(a_i, b_j, c)}{p(a_i)} \geq \delta}$$

Dominant State Analysis

- Step 4:
  - Compute conditional marginal probability for each of the values of the highest RU dimension
  - e.g. Given a particular Src. Port and Dst. IP value, calculate the probability of the Dst. Port values

Example Behavioral Classes

BC_6 [0, 2, 0]	srcPrt(·)→dstIP(· · ·)→dstPrt(*)	server replying to a few hosts
	srcPrt(25)→dstIP(· · ·)→dstPrt(*) srcPrt(53)→dstIP(· · ·)→dstPrt(*) srcPrt(80)→dstIP(· · ·)→dstPrt(*) srcPrt(443)→dstIP(· · ·)→dstPrt(*)	25: Email 53: DNS 80: Web 443: https
BC_7 [0, 2, 1]	srcPrt(·)→dstIP(· · ·)→ dstPrt(*)	server replying to many hosts
	srcPrt(25)→dstIP→dstPrt(*) srcPrt(80)→dstIP→dstPrt(*)	25: Email 80: Web
BC_8 [0, 2, 2]	srcPrt(.)→(dstPrt(*),dstIP(*))	server replying to large # of hosts
	srcPrt(80)→(dstPrt(*),dstIP(*))	80: Web

- **Variability in the Dst. IP dimension allows for classification of server load**

Contributions

- Information theoretic application of ‘thresholds’ discussed in BLINC
- Discover significant traffic patterns without manual intervention

Contributions

- Multiple ‘views’ on the patterns
  - Fix the source port dimension
    - Uncertainty in source IP can indicate global ports
  - Fix the destination IP dimension
    - Uncertainty in source IP and port indicate the ‘activity’ of the client

Contributions

- Insight based on behavioral change
  - If a server moves from BC8 to BC6, it could indicate DoS
  - Appearance in certain behavioral classes indicate worm infection

BC_6 [0, 2, 0]	srcPrt(·)→dstIP(· · ·)→dstPrt(*)	server replying to a few hosts
	srcPrt(25)→dstIP(· · ·)→dstPrt(*) srcPrt(53)→dstIP(· · ·)→dstPrt(*) srcPrt(80)→dstIP(· · ·)→dstPrt(*) srcPrt(443)→dstIP(· · ·)→dstPrt(*)	25: Email 53: DNS 80: Web 443: https
BC_7 [0, 2, 1]	srcPrt(·)→dstIP(· · ·)→ dstPrt(*)	server replying to many hosts
	srcPrt(25)→dstIP→dstPrt(*) srcPrt(80)→dstIP→dstPrt(*)	25: Email 80: Web
BC_8 [0, 2, 2]	srcPrt(.)→(dstPrt(*),dstIP(*))	server replying to large # of hosts
	srcPrt(80)→(dstPrt(*),dstIP(*))	80: Web

Contributions

- Canonical clusters
  - Servers have low uncertainty in source port
  - Scan/exploits have low uncertainty in dest. port
  - Heavy hitters have low uncertainty in the dest. port

What is missing from these schemes?

- Transport-layer is easy to fool
  - Most characteristics are under user control
- Transport-layer characteristics are not a sufficient condition for proving the presence of a particular service/protocol

What is missing from these schemes?

- Attacks become difficult when additional information is added
  - COI – General profile of communication behavior
  - BLINC – Application-specific profile of communication behavior
  - Profiling Backbone Traffic – Robust profiles of significant behavior
  - Flow-specific profiles based on underlying protocol artifacts

Challenges

- Single encrypted tunnel (IPSec)
  - Multiple hosts
  - Multiple protocols
  - What protocols are running in the tunnel?
  - How many connections in the tunnel?
- Single transport-layer profile no matter what protocols are running, or how many hosts are present

Open Questions

- Can classification occur in the tunnel?
- Does the tunnel assumption make it easier for attackers to fool the classification?
- Can we stop the mimicry attack completely?

References

- Aiello, W., Kalmanek, C., McDaniel, P., Sen, S., Spatscheck, O., and Van der Merwe, J. *Analysis of Communities of Interest in Data Networks.* In Proceedings of 6th Annual Workshop on Passive and Active Network Monitoring, Boston, MA. March 31 – April 1, 2005. pp. 83-97.

- Campete, S. A., Krishnamoorthy, M., and Yener, B. *A Tool for Internet Chatroom Surveillance.* In Proceedings of the 2nd Symposium on Intelligence and Security Informatics. June 2004. pp. 252-265.

- McDaniel, P., Sen, S., Spatscheck, O., Van der Merwe, J., Aiello, W., and Kalmanek, C. *Enterprise Security: A Community of Interest Based Approach*. In Proceedings of the 13th Annual Network and Distributed System Security Conference. February 2006.

- Karagiannis, T., Papagiannaki, K., and Faloutsos, M. *BLINC: Multilevel Traffic Classification in the Dark.* In Proceedings of 2005 ACM SIGCOMM. August, 2005.

References

- Sun, Q., Simon, D. R., Yi-Min, W., Russell, W., Padmanabhan, V. N., and Qiu, L. *Statistical Identification of Encrypted Web Browsing Traffic*. In Proceedings of the 2002 IEEE Symposium on Security and Privacy, Oakland, CA. May, 2002.

- Weaver, N., Paxson, V., Staniford, S., and Cunningham, R. *A Taxonomy of Computer Worms*. In Proceedings of the 2003 ACM Workshop on Rapid Malcode, Washington, DC. October, 2003. pp. 11-18.

- Xu, K. Zhang, Z., and Bhattacharyya, S. *Profiling Internet Backbone Traffic: Behavior Models and Applications*. In Proceedings of 2005 ACM SIGCOMM. August, 2005.

- Zhang, Y., and Paxson, V. *Detecting Backdoors*. In Proceedings of the 9th Annual USENIX Security Symposium, Denver, CO. August 2000.

Traffic Classification: Reloaded

Scott E. Coull

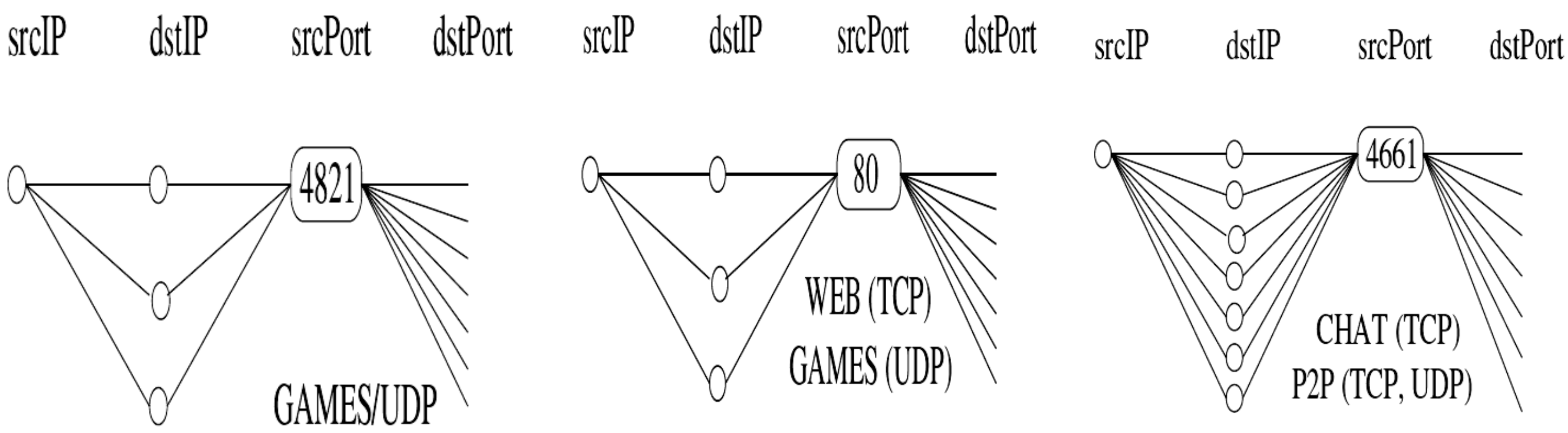
February 24, 2006

Graphlet Creation

- *Note that while some of the graphlets display port numbers, the classification and the formation of graphlets* ***do not associate in any way a specific port number with an application***

- Implication:
  - No one-to-one mapping of port numbers to applications

Graphlet Usage



- Significant similarity in graphlet structure
- Reliance on port numbers for differentiation
- Heuristics and thresholds also play a significant role

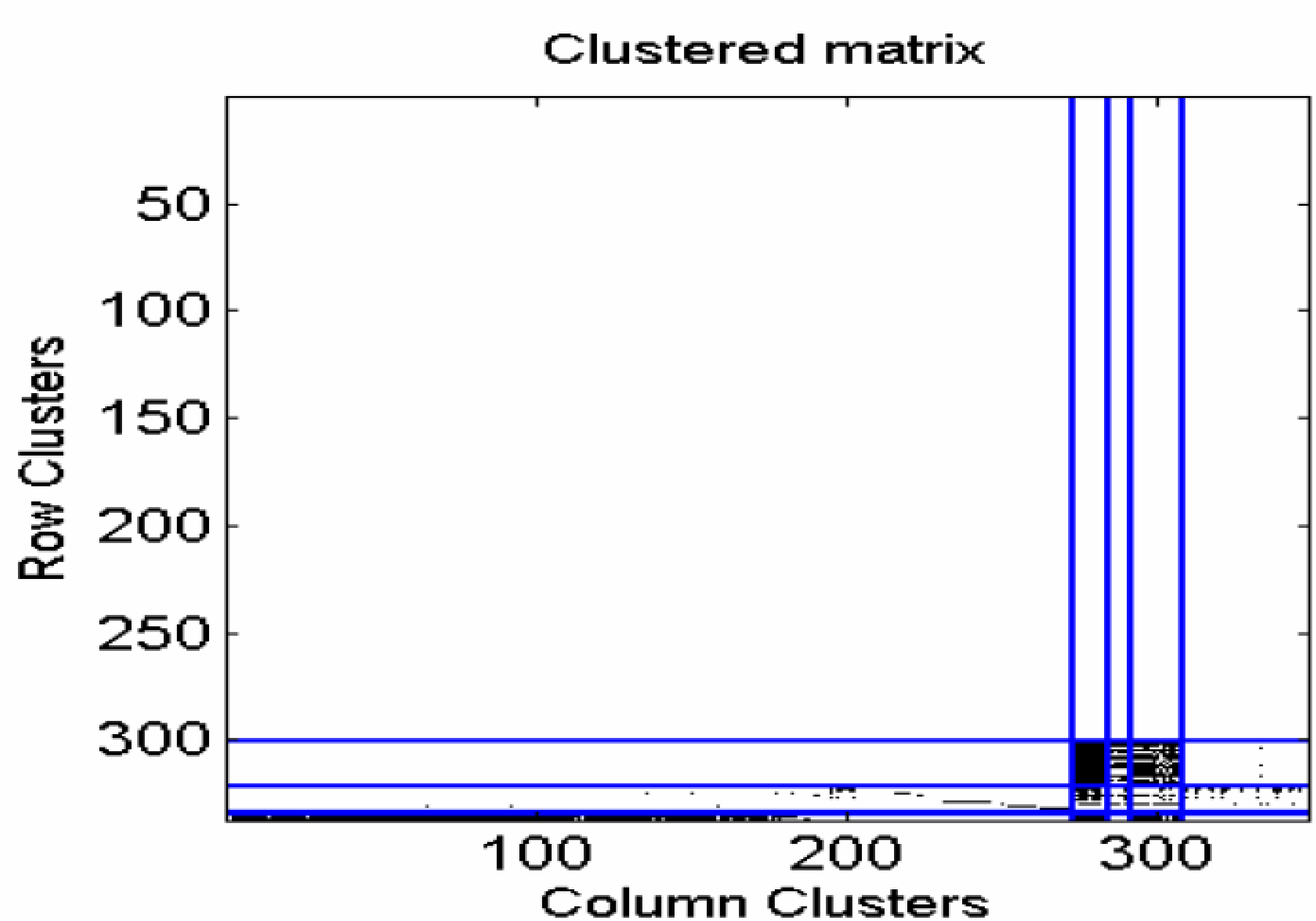
Application of Heuristics

- Heuristics recap:
  - Transport protocol, cardinality, packet size, community, recursive detection

- Transport protocol can be added to the ‘graphlet’
- Cardinality and size in the thresholds
- Recursive detection and community
  - Not discussed in the paper

A Question of 'Cliques'

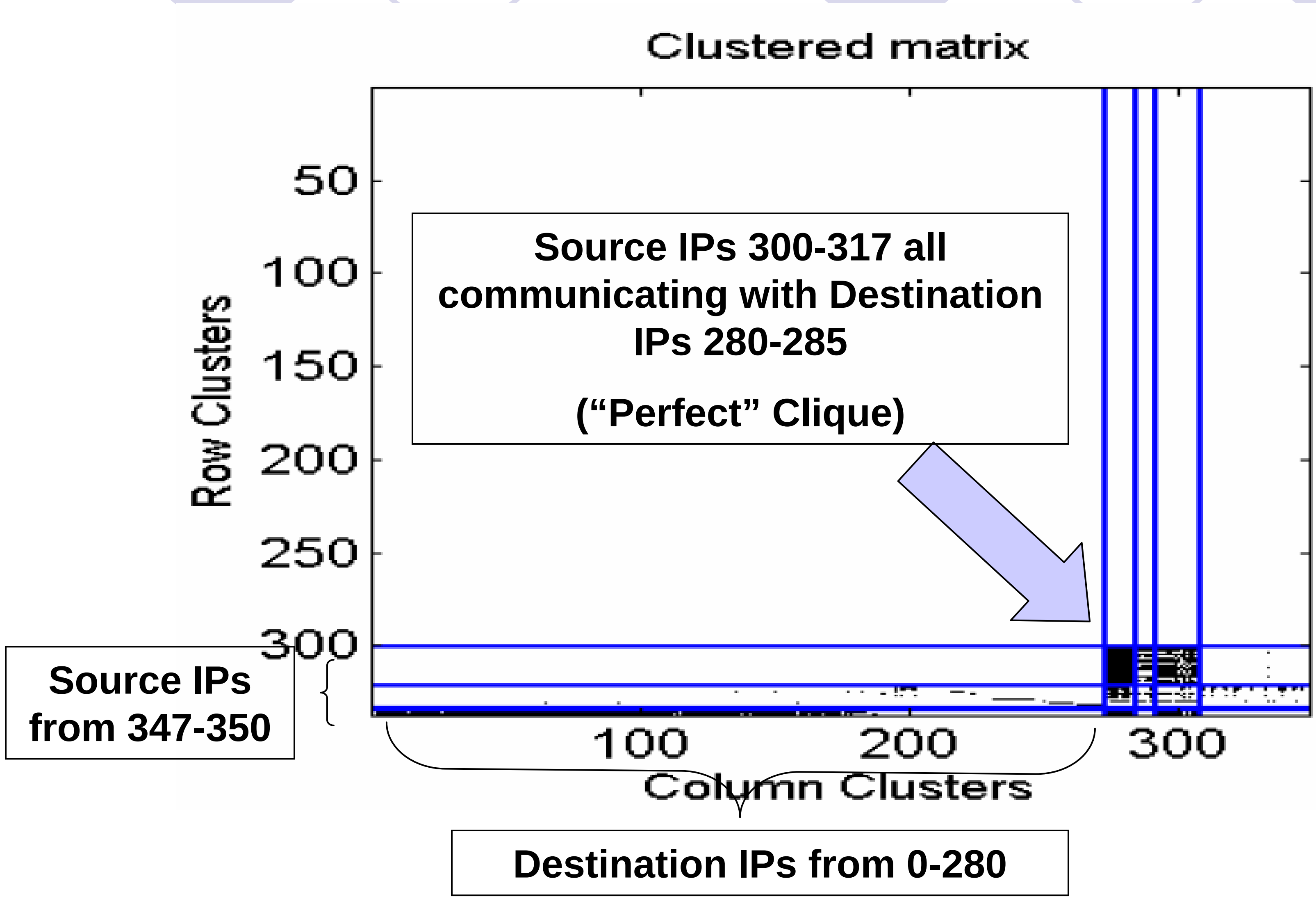
- What is this figure showing us?



A Question of ‘Cliques’

- Column Clusters are indexed destination IPs
- Row Clusters are indexed source IPs
- Binary matrix representing interaction between Column Index and Row Index

A Question of 'Cliques'



Defining Traffic Behavior

- COI
  - Simplistic profiles that blindly capture behavior straight from log data
  - k-means clustering algorithm which uses frequency to determine significant behaviors
- BLINC
  - Manually derived ‘graphlets’ to capture behaviors
- Profiling Internet Backbone Traffic
  - Entropy-based clustering for general behavioral classes
  - Dynamic State Analysis for significant behavior within those classes

Information Theory Refresher

- Entropy
  - Measure of uncertainty in empirical data
- Relative Uncertainty
  - Measures uniformity of empirical data regardless of sample or support size
  - Values near 1 indicate uniform distribution

Entropy-based Clustering

- Find the so-called ‘heavy hitters’ for a dimension of the 4-tuple
  - Example: Find Src. IPs that occur frequently within the set of all Src. IPs seen

Entropy-based Clustering

- While the distribution of values in the set of Src. IPs is skewed there are particular Src. IPs which occur very frequently
  - i.e. while the Relative Uncertainty is low

Entropy-based Clustering

- Take the values from the Src. IP set that occur most frequently
  - i.e. take the Src. IP values which have a probability greater than some threshold

Entropy-based Clustering

- Continue taking the most frequent in the Src. IP set until the remaining Src. IP values are nearly uniformly distributed
  - i.e. continue taking values until the relative uncertainty of the remaining values is near 1

Entropy-based Clustering

- After this iteration is complete, we have a set of tuples that contain ‘heavy hitter’ Src. IPs

Behavioral Classes

- 3 “Free” dimensions for each 4-tuple taken in the Entropy-based Clustering
  - e.g. when we cluster on Src. IP, we have Dst. IP, Dst. Port, and Src. Port “free”
- 27 behavioral classes based on the relative uncertainty of each “free” dimension

Dominant States

- 4-tuples from Entropy-based Clustering lie within these 27 classes
- Probable values of the 3 “free” dimensions within these classes are used as the most significant states
  - i.e. if we see a particular Src. Port occurring often, then this is a dominant state

Wrap Up

- Entropy-based Clustering gets us the most significant tuples based on a particular dimension
  - e.g. we get the tuples that have Src. IPs that have very low entropy
- Behavioral classes denote a specific type of behavior for the dimension that was clustered
- Dominant states denote specific, significant instances of behavior within a class