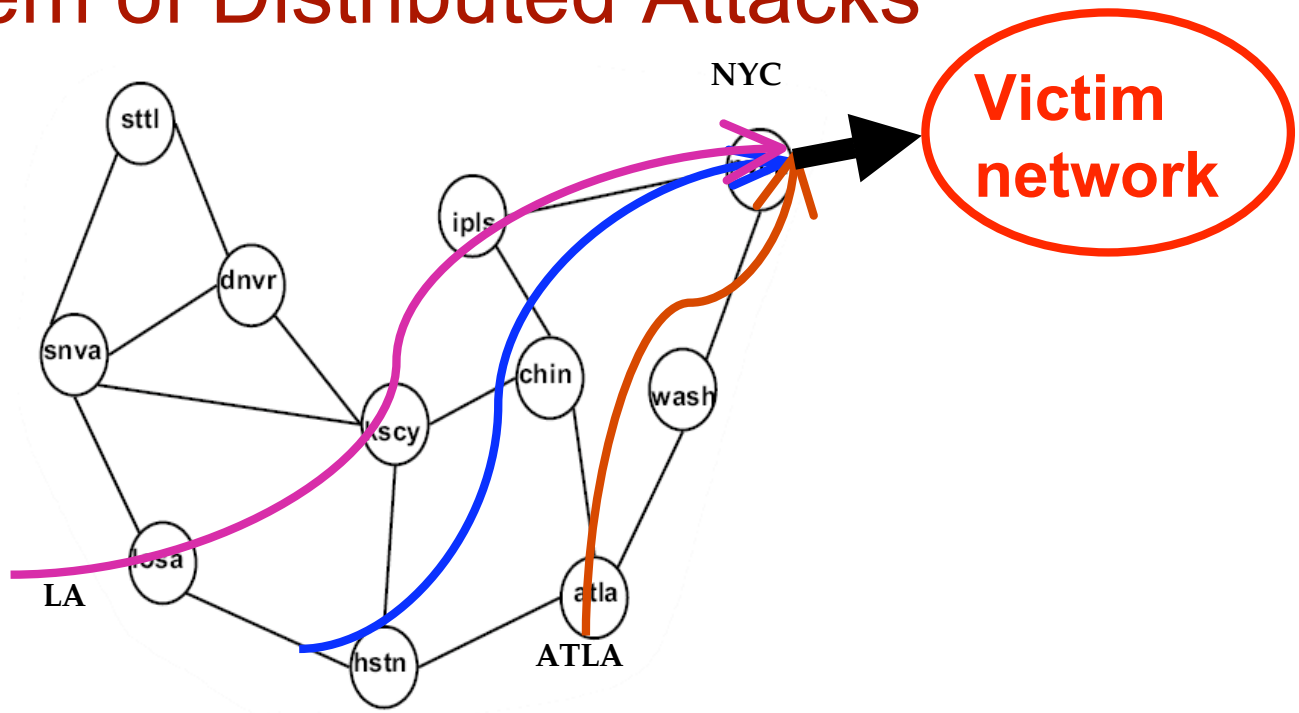
# Detecting Distributed Attacks Using **Network-Wide Flow Data**

Anukool Lakhina
with Mark Crovella and Christophe Diot



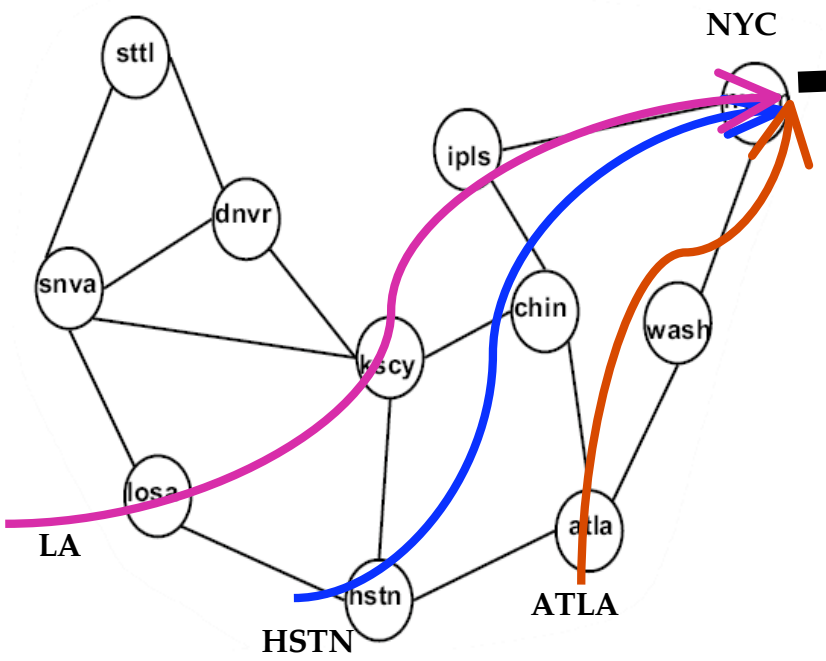
*FloCon, September 21, 2005*

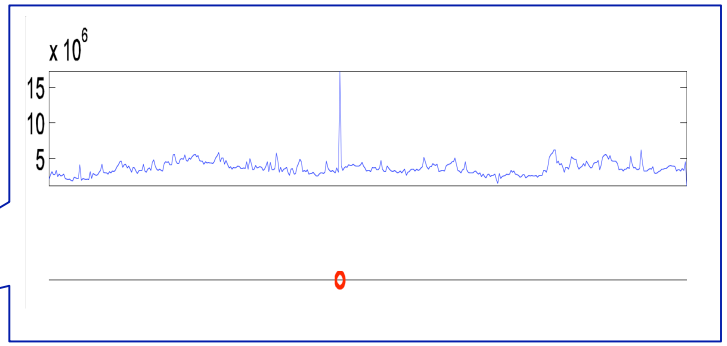
# The Problem of Distributed Attacks




- Continue to become more prevalent [CERT'04]
- Financial incentives for attackers, *e.g.,* extortion
- Increasing in sophistication: worm-compromised hosts and bot-nets are massively distributed

# Detection at the Edge

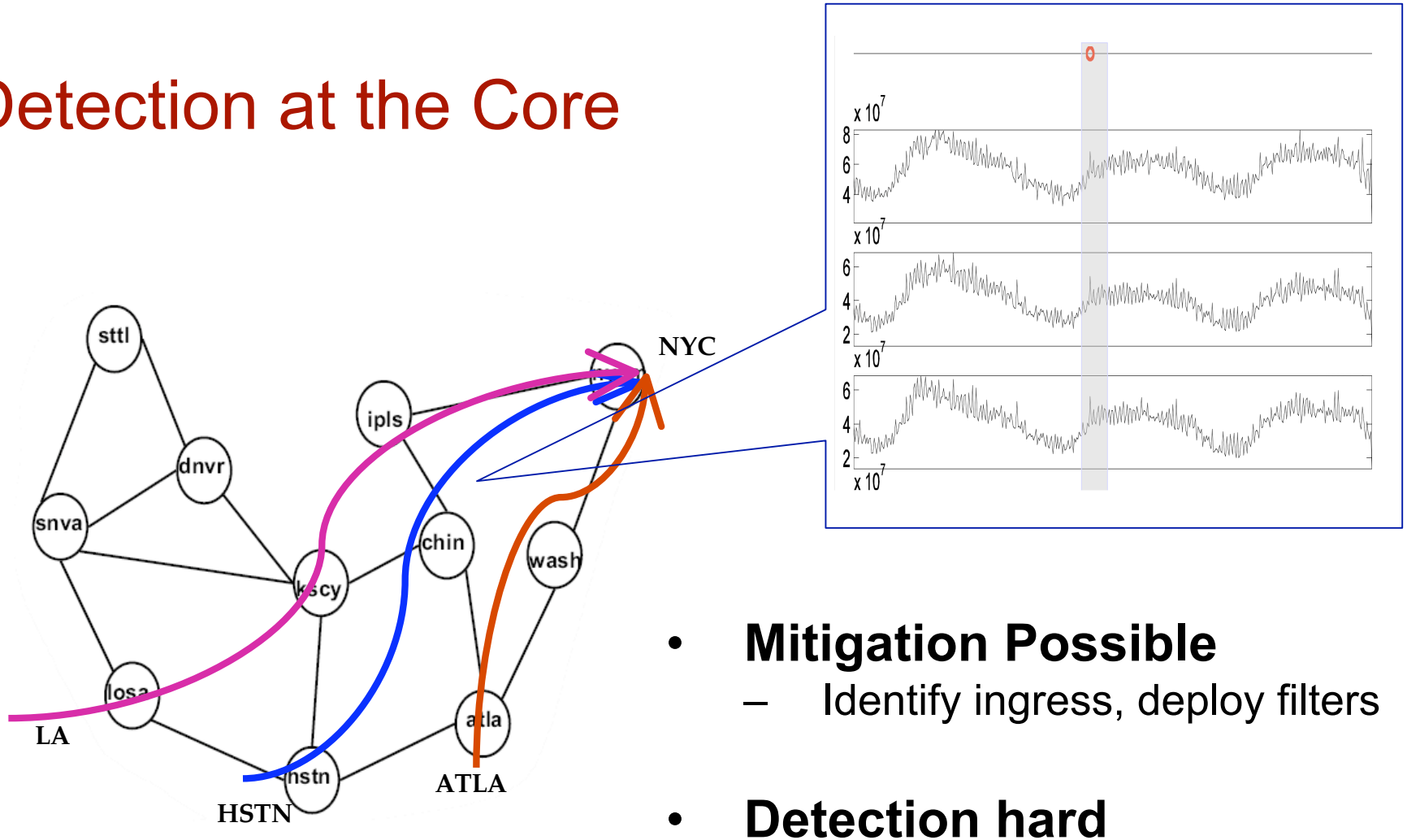



Victim network

- **Detection easy**
  - Anomaly stands out visibly
- **Mitigation hard**
  - Exhausted bandwidth
  - Need upstream provider's cooperation
  - Spoofed sources

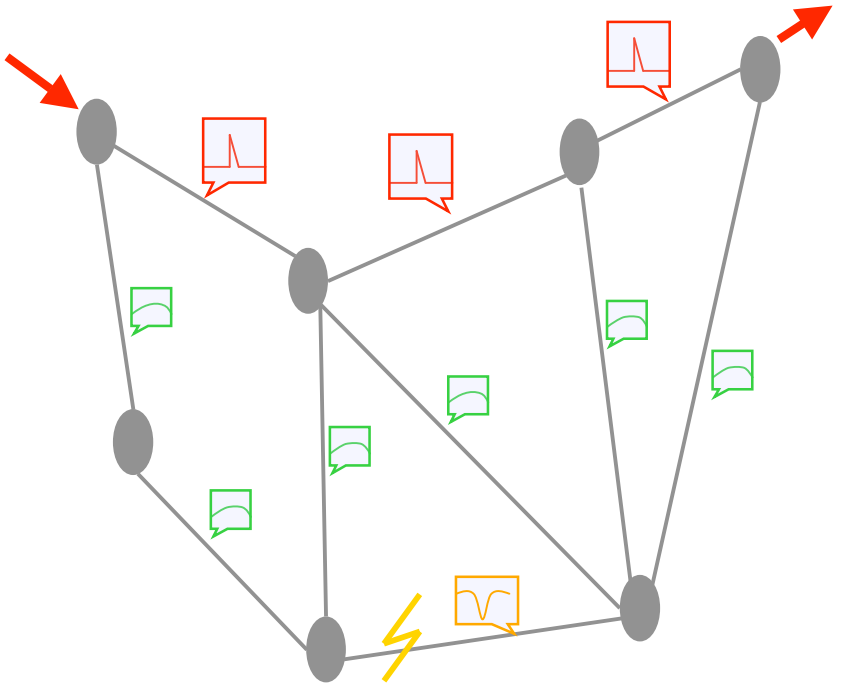
# Detection at the Core




- **Mitigation Possible**
  - Identify ingress, deploy filters
- **Detection hard**
  - Attack does not stand out
  - Present on multiple flows

# A Need for Network-Wide Diagnosis

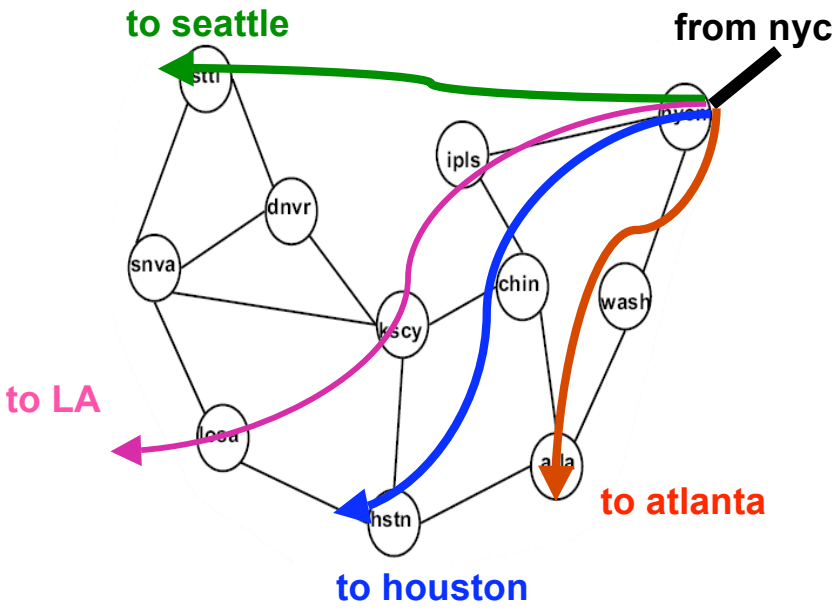
- Effective diagnosis of attacks requires a **whole-network approach**
  - *Simultaneously* inspecting traffic on all links
- Useful in other contexts also:
  - Enterprise networks
  - Worm propagation, insider misuse, operational problems



# Talk Outline

- Methods
  - Measuring Network-Wide Traffic
  - Detecting Network-Wide Anomalies
  - Beyond Volume Detection: *Traffic Features*
  - Automatic Classification of Anomalies
- Applications
  - General detection: scans, worms, flash events, …
  - Detecting Distributed Attacks
- Summary

# Origin-Destination Traffic Flows



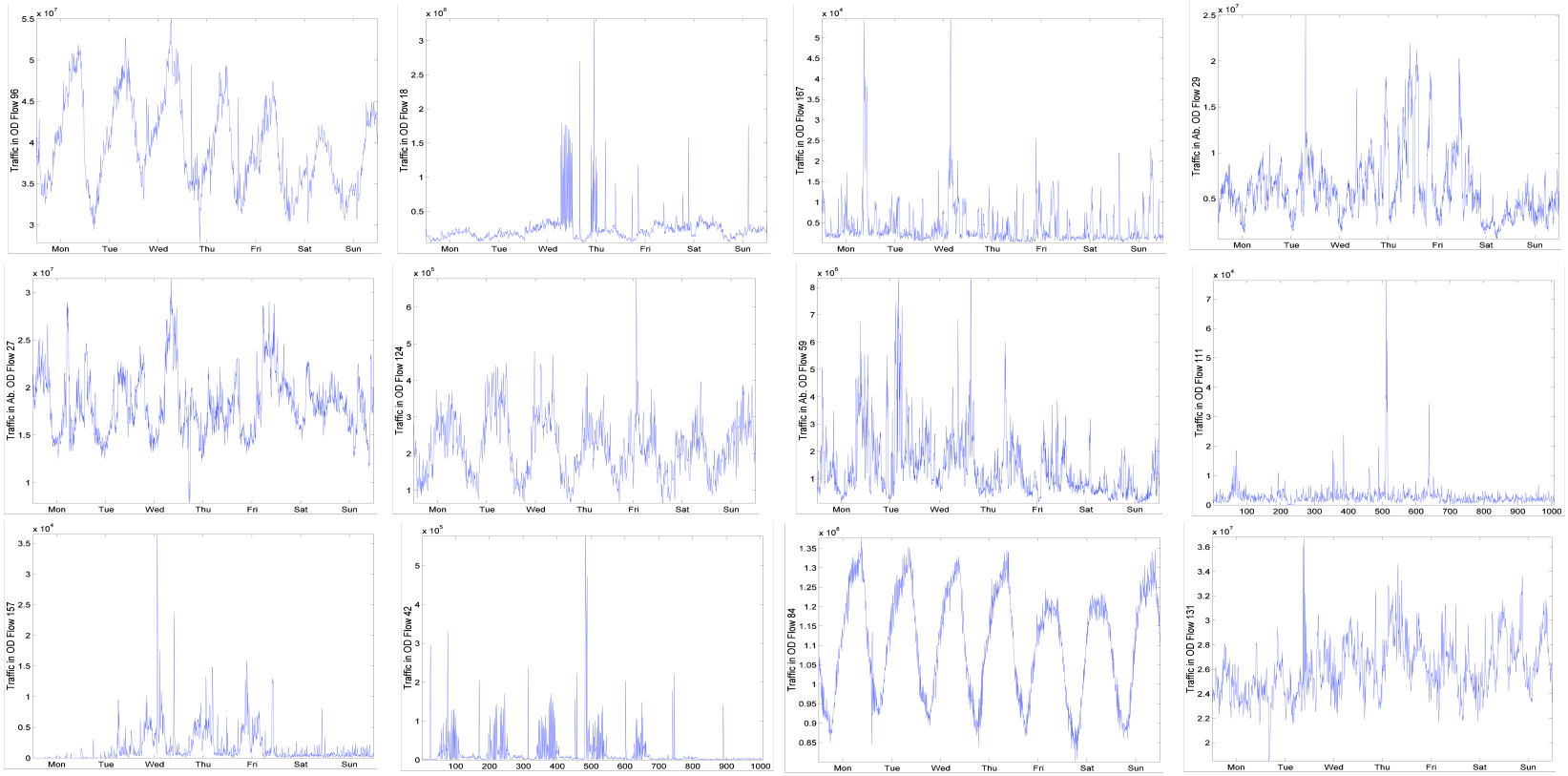

- Traffic entering the network at the *origin* and leaving the network at the *destination* (*i.e.*, the traffic matrix)
- Use routing (IGP, BGP) data to aggregate NetFlow traffic into OD flows
- *Massive* reduction in data collection

# Data Collected

Collect sampled NetFlow data from all routers of:

1. **Abilene Internet 2 backbone research network**
   - 11 PoPs, 121 OD flows, anonymized, 1 out of 100 sampling rate, 5 minute bins
2. **Géant Europe backbone research network**
   - 22 PoPs, 484 OD flows, not anonymized, 1 out of 1000 sampling rate, 10 minute bins
3. **Sprint European backbone commercial network**
   - 13 PoPs, 169 OD flows, not anonymized, aggregated, 1 out of 250 sampling rate, 10 minute bins

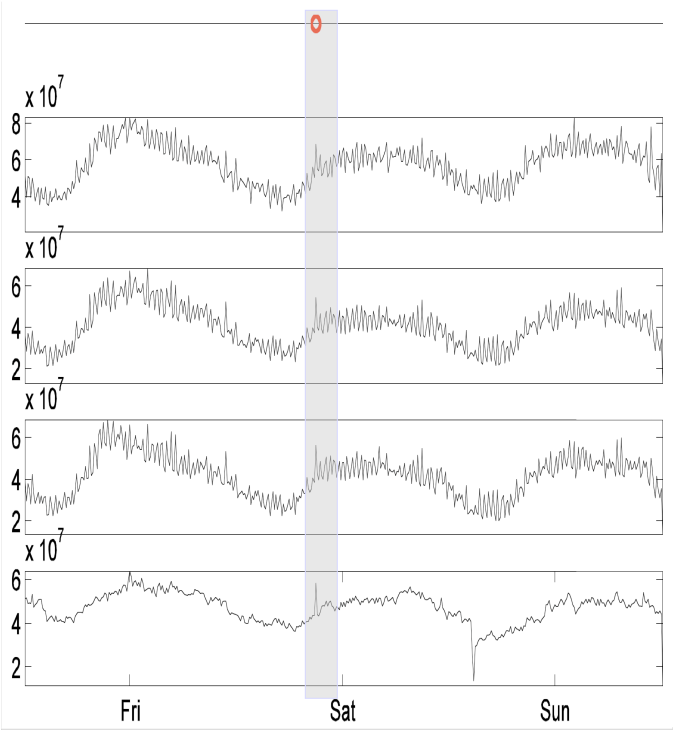
# But, This is Difficult!




How do we extract **anomalies** and **normal behavior** from noisy, **high-dimensional** data in a systematic manner?

# Turning High Dimensionality into a Strength

- Traditional traffic anomaly diagnosis builds normality in *time*
  - Methods exploit temporal correlation
- Whole-network view is an attempt to examine normality in *space*
  - Make use of spatial correlation
- Useful for anomaly diagnosis:
  - Strong trends exhibited throughout network are likely to be *"normal"*
  - Anomalies break relationships between traffic measures



# The Subspace Method [LCD:SIGCOMM '04]

- An approach to separate normal & anomalous network-wide traffic
- Designate temporal patterns most common to all the OD flows as the **normal subspace**
- Remaining temporal patterns form the **anomalous subspace**
- Then, decompose traffic in all OD flows by *projecting* onto the two subspaces to obtain:

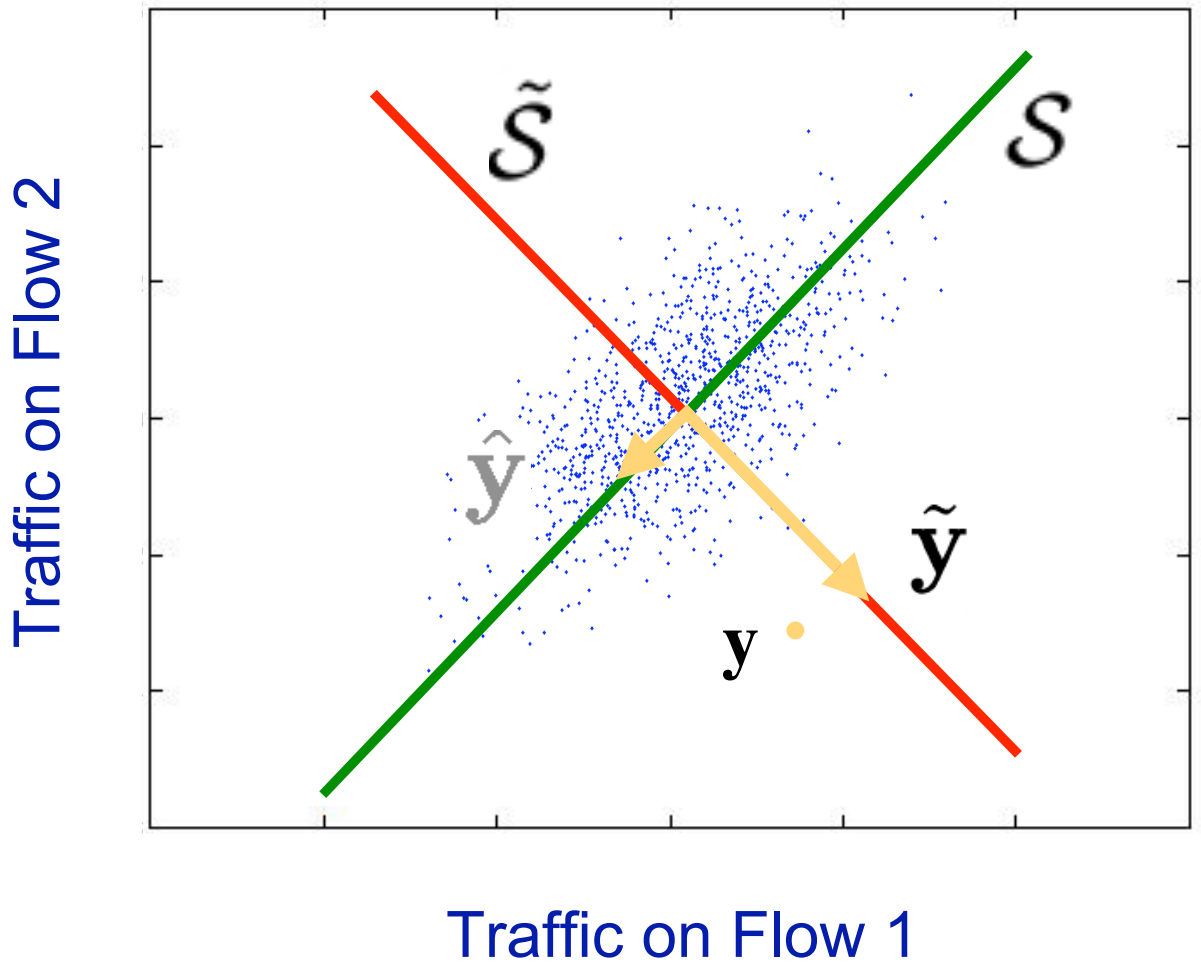
$$\mathbf{y} = \hat{\mathbf{y}} + \tilde{\mathbf{y}}$$

**Traffic vector of all OD flows at a particular point in time**

**Normal traffic vector**

**Residual traffic vector**

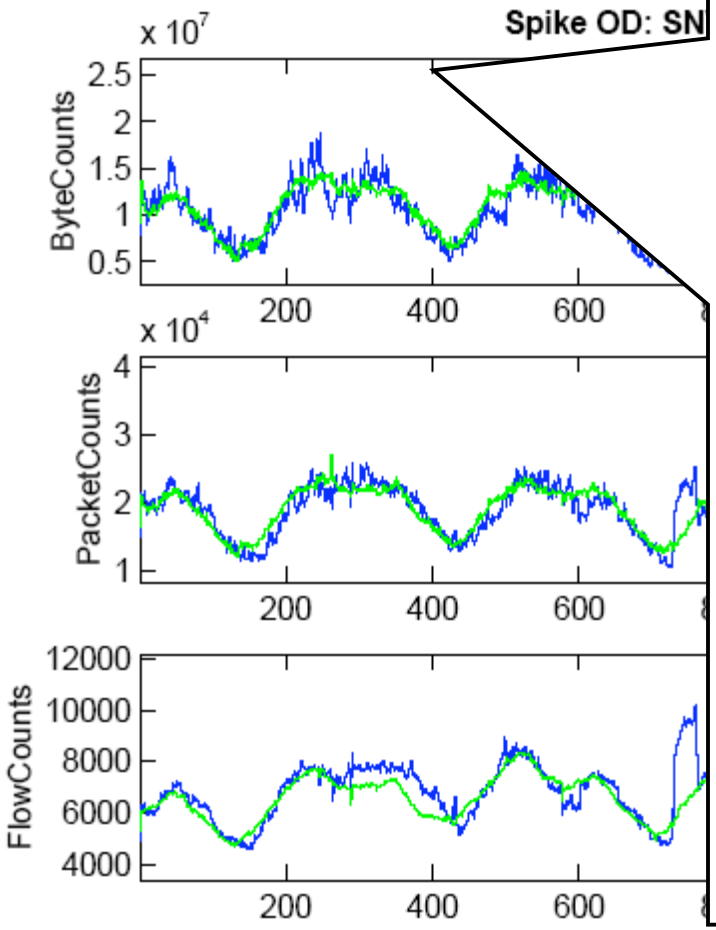
# The Subspace Method, Geometrically

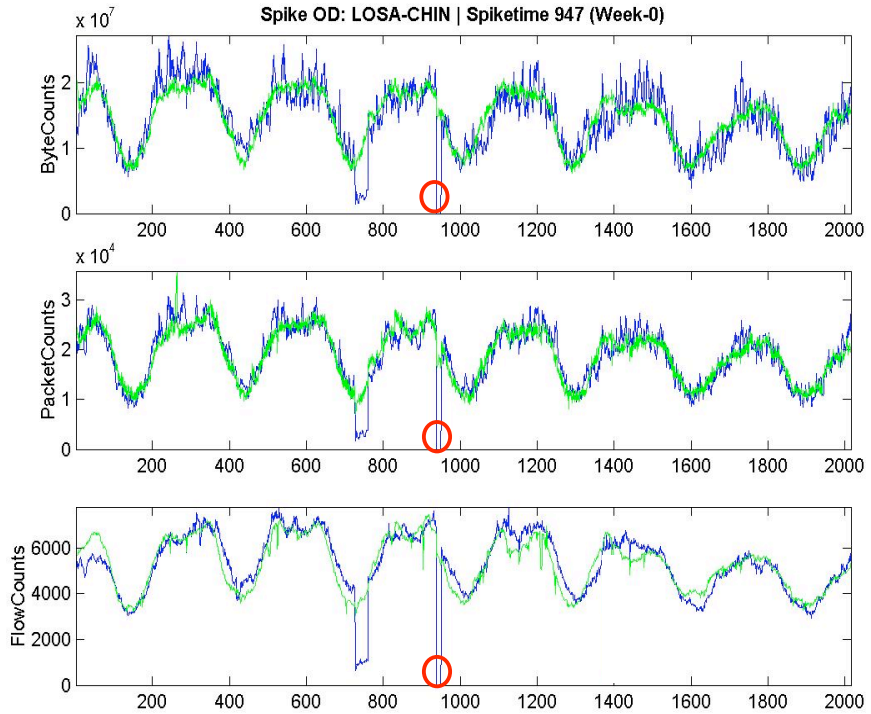



In general, anomalous traffic results in a large size of $\tilde{y}$

For higher dimensions, use Principal Component Analysis
[LPC+:SIGMETRICS '04]

# Example of a Volume Anomaly [LCD:IMC '04]




Multihomed customer CALREN reroutes around outage at LOSA

# Talk Outline

- Methods
  - Measuring Network-Wide Traffic
  - Detecting Network-Wide Anomalies
  - **Beyond Volume Detection: *Traffic Features***
  - Automatic Classification of Anomalies
- Applications
  - General detection: scans, worms, flash, *etc.*
  - Detecting Distributed Attacks
- Summary

# Exploiting Traffic Features

- **Key Idea:**

  Anomalies can be detected and distinguished by inspecting *traffic features*:

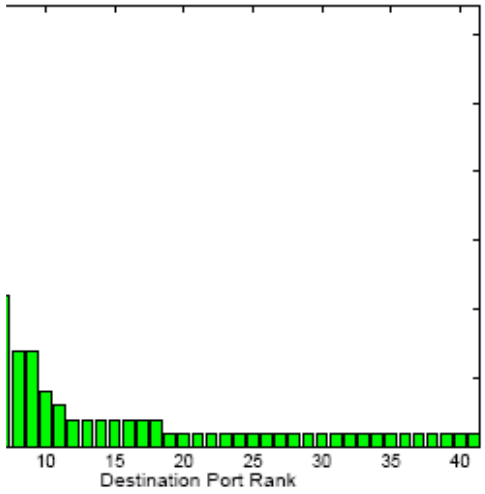
  **`SrcIP, SrcPort, DstIP, DstPort`**

- **Overview of Methodolgy:**

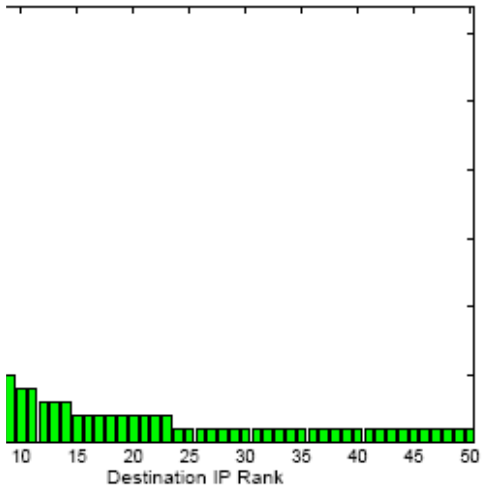
  1. Inspect *distributions* of traffic features
  2. Correlate distributions *network-wide* to detect anomalies
  3. *Cluster* on anomaly features to classify

# Traffic Feature Distributions [LCD:SIGCOMM '05]

**Dispersed Histogram**
High Entropy




**Concentrated Histogram**
Low Entropy



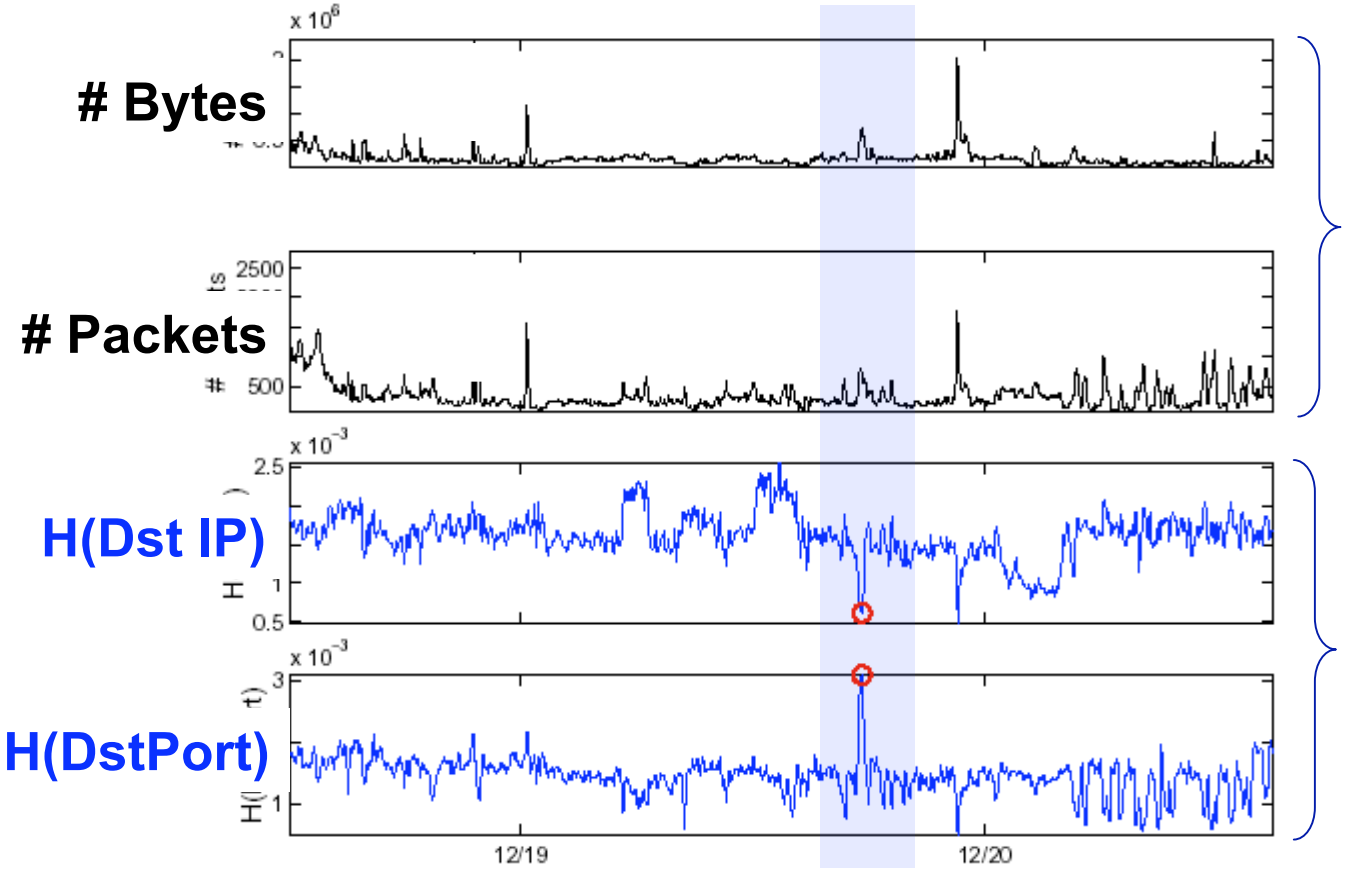

**Typical Traffic**

Summarize using **sample entropy** of histogram *X*:

$$H(X) = -\sum_{i=1}^{N} \left(\frac{n_i}{S}\right) \log_2 \left(\frac{n_i}{S}\right)$$

where symbol *i* occurs $n_i$ times; *S* is total # of observations

# Feature Entropy Timeseries




Port scan dwarfed in volume metrics…

But stands out in feature entropy, which also reveals its structure

# How Do Detected Anomalies Differ?

| Anomaly Label | # Found in Volume | # Additional in Entropy |
|---|---|---|
| Alpha | 84 | 137 |
| DOS | 16 | 11 |
| Flash Crowd | 6 | 3 |
| Port Scan | 0 | 30 |
| Network Scan | 0 | 28 |
| Outage | 4 | 11 |
| Point Multipoint | 0 | 7 |
| Unknown | 19 | 45 |
| False Alarm | 23 | 20 |
| **Total** | **152** | **292** |

3 weeks of Abilene anomalies classified manually

# Talk Outline

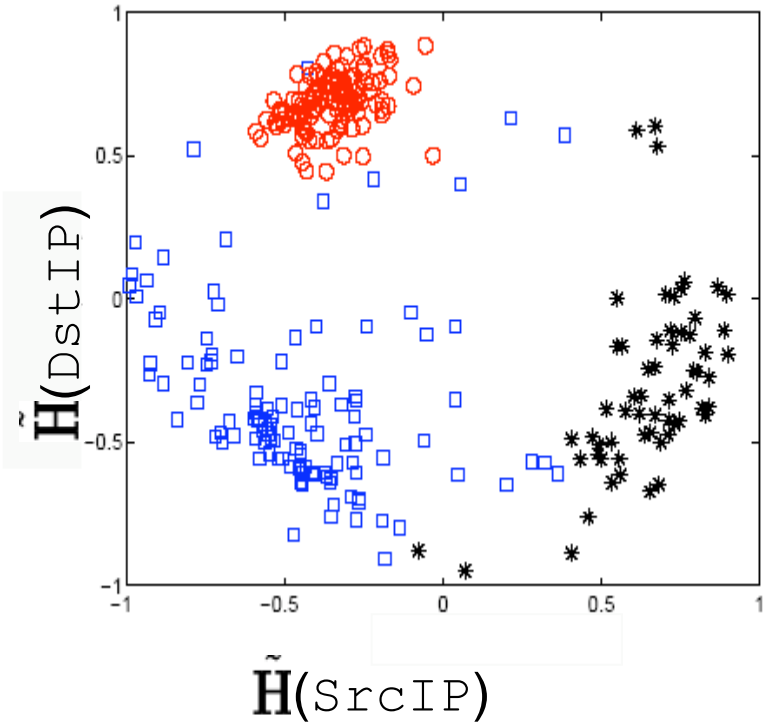
- Methods
  - Measuring Network-Wide Traffic
  - Detecting Network-Wide Anomalies
  - Beyond Volume Detection: *Traffic Features*
  - **Automatic Classification of Anomalies**
- Applications
  - General detection: scans, worms, flash events, …
  - Detecting Distributed Attacks
- Summary

# Classifying Anomalies by Clustering

- Enables unsupervised classification
- Each anomaly is a point in 4-D space:
  [ $\tilde{H}$(SrcIP), $\tilde{H}$(SrcPort), $\tilde{H}$(DstIP), $\tilde{H}$(DstPort) ]
- Questions:
  - Do anomalies form clusters in this space?
  - Are the clusters meaningful?
    - Internally consistent, externally distinct
  - What can we learn from the clusters?

# Clustering Known Anomalies (2-D view)

**Known Labels**



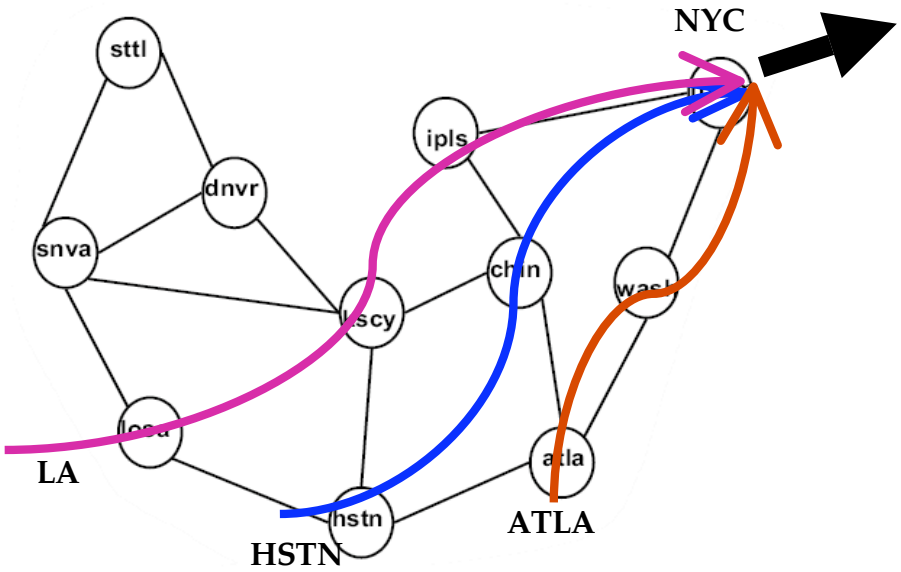

Legend

**Code Red Scanning**

**Single source DOS attack**

**Multi source DOS attack**

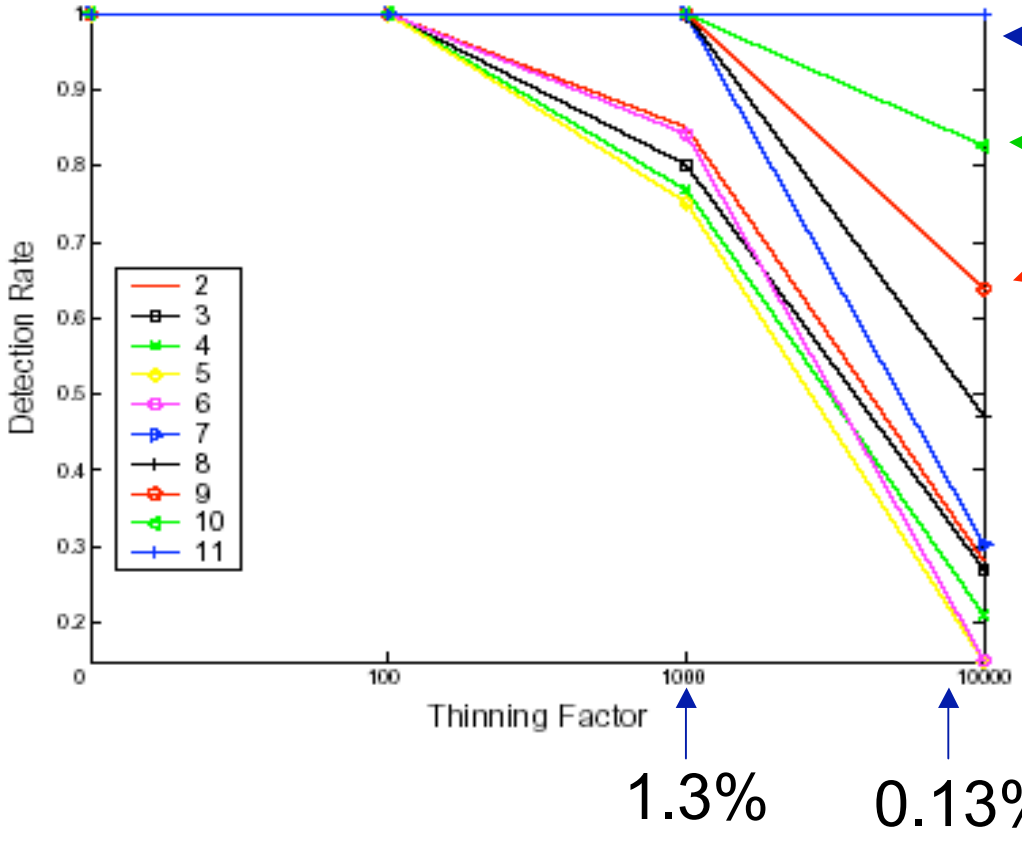
# Back to Distributed Attacks…




**Evaluation Methodology**

1. Superimpose known DDOS attack trace in OD flows
2. Split attack traffic into varying number of OD flows
3. Test sensitivity at varying anomaly intensities, by thinning trace
4. Results are average over an exhaustive sequence of experiments

# Distributed Attacks: Detection Results




11 OD flows

10 OD flows

9 OD flows

**The more distributed the attack, the easier it is to detect**

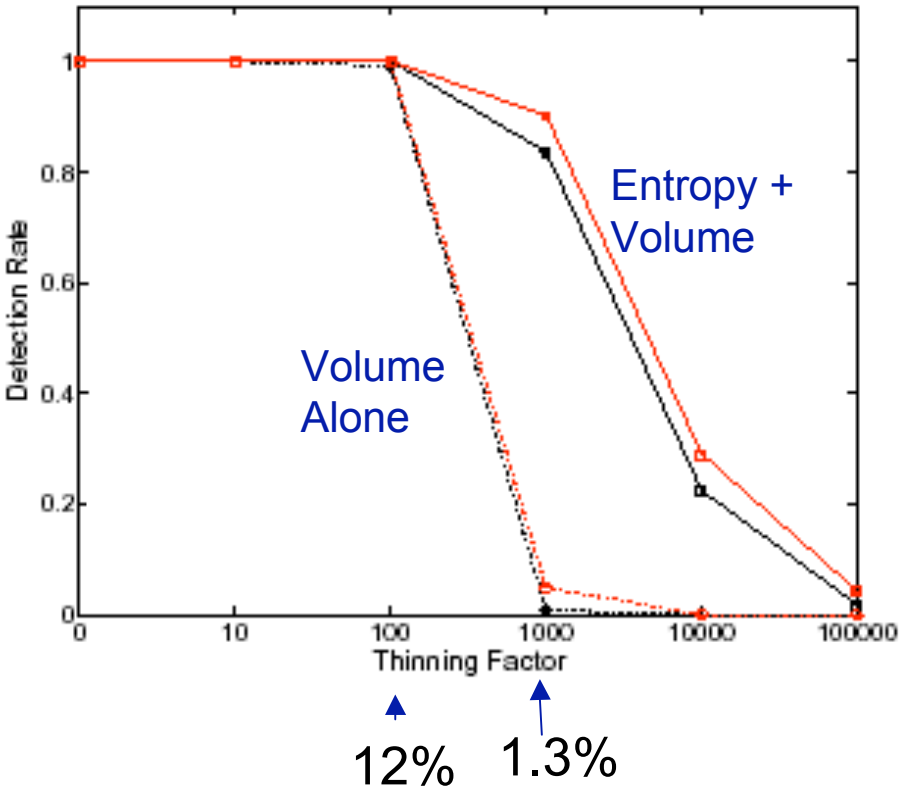
# Summary

- Network-Wide Detection:
  - Broad range of anomalies with low false alarms
  - Feature entropy significantly augment volume metrics
  - Highly sensitive: Detection rates of 90% possible, even when anomaly is 1% of background traffic
- Anomaly Classification:
  - Clusters are meaningful, and reveal new anomalies
  - In papers: more discussion of clusters and Géant
- Whole-network analysis and traffic feature distributions are promising for general anomaly diagnosis

# Backup Slides

# Detection Rate by Injecting Real Anomalies

Multi-Source DOS
[Hussain et al, 03]

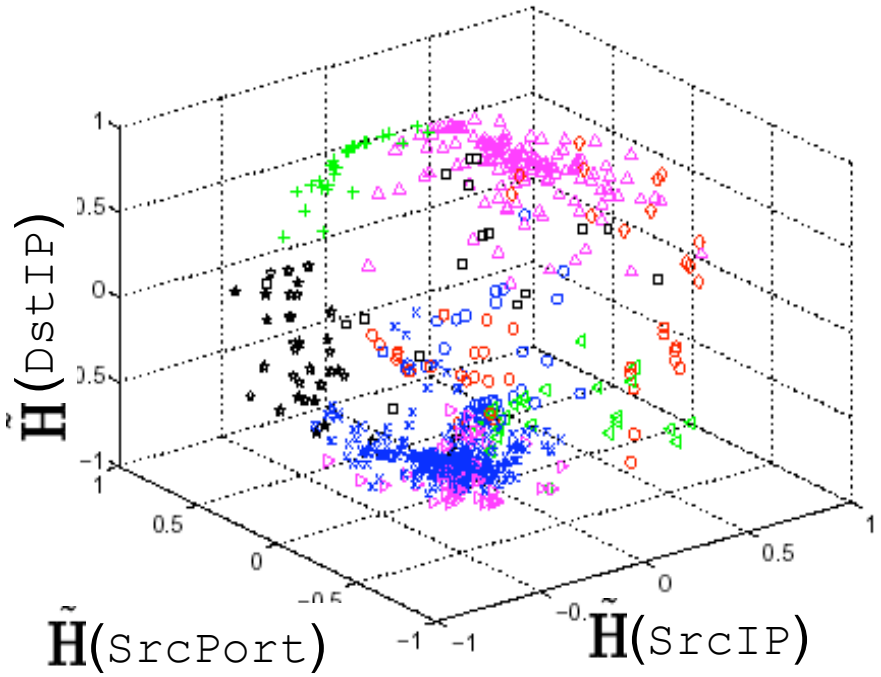



Detection rate vs. Anomaly intensity
(intensity % compared to average flow bytes)

**Evaluation Methodology**

- Superimpose known anomaly traces into OD flows
- Test sensitivity at varying anomaly intensities, by thinning trace
- Results are average over a sequence of experiments

# 3-D view of Abilene anomaly clusters

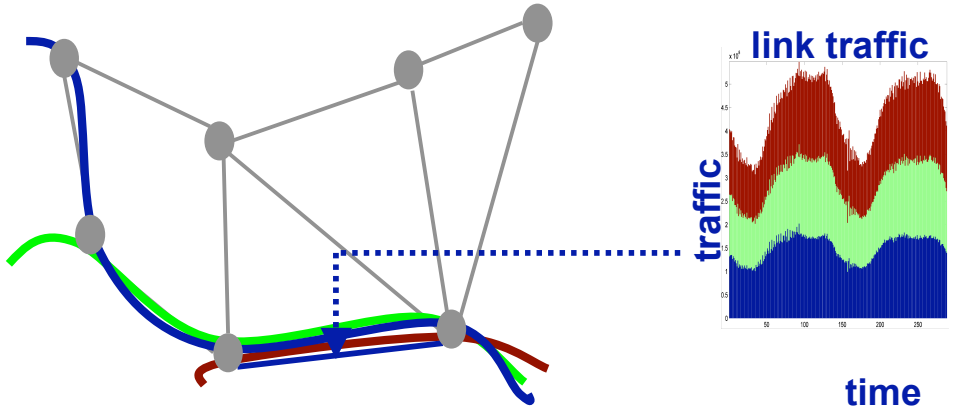



- Used 2 different clustering algorithms
  - Results consistent
- Heuristics identify about 10 clusters in dataset
  - details in paper

# Anomaly Clusters in Abilene data

| ID | # points | Plurality Label | $\tilde{H}$(srcIP) | $\tilde{H}$(srcPort) | $\tilde{H}$(dstIP) | $\tilde{H}$(dstPort) |
|---|---|---|---|---|---|---|
| 1 | 191 | Alpha | – | 0 | – | – |
| 2 | 53 | Network Scan | 0 | + | 0 | 0 |
| 3 | 35 | Port Scan | – | + | – | + |
| 4 | 30 | Port Scan | 0 | – | 0 | + |
| 5 | 24 | Alpha | 0 | 0 | + | 0 |
| 6 | 22 | Outage | 0 | 0 | 0 | + |
| 7 | 22 | Alpha | – | 0 | – | 0 |
| 8 | 8 | Point Multipoint | 0 | 0 | 0 | + |
| 9 | 8 | Flash Crowd | 0 | 0 | 0 | – |
| 10 | 4 | Alpha | 0 | – | 0 | 0 |

# Why Origin-Destination Flows?




- All link traffic arises from the superposition of OD flows
- OD flows capture *distinct* traffic demands; no redundant traffic
- A useful primitive for whole-network analysis

# Subspace Method: Detection

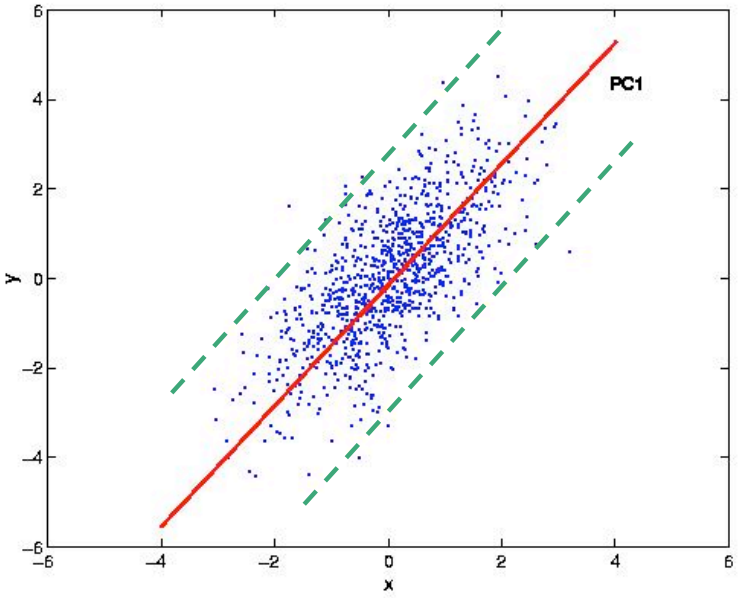
- Error Bounds on Squared Prediction Error:

$$\mathrm{SPE} \equiv \|\tilde{\mathbf{y}}\|^2 = \|\tilde{\mathbf{C}}\mathbf{y}\|^2$$

- Assuming Normal Errors:

$$\mathrm{SPE} \leq \delta_\alpha^2$$

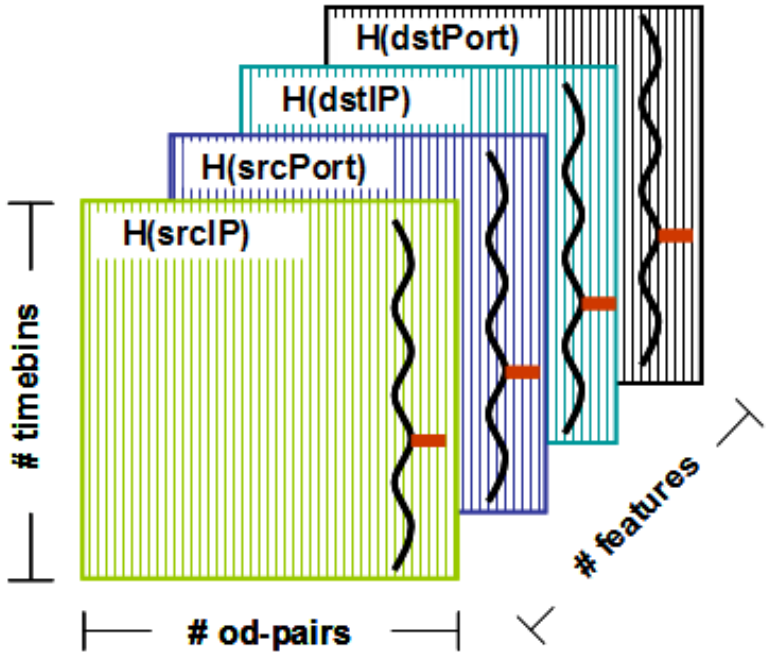
- Result due to [Jackson and Mudholkar, 1979]



# Subspace Method: Identification

- An anomaly results in a displacement of the state vector away from $S$
- The direction of the displacement gives information about the nature of the anomaly
- Intuition: find the OD flow that best describes the direction associated with a detected anomaly
- More precisely, we select the OD flow that accounts for maximum residual traffic

# Network-Wide Traffic Data Collected




**Multivariate, multiway timeseries to analyze**

- Compute entropy on packet histograms for 4 traffic features: **`SrcIP, SrcPort, DstIP, DstPort`**

# Multiway Subspace Method

1. “Unwrap” the multiway matrix into one matrix