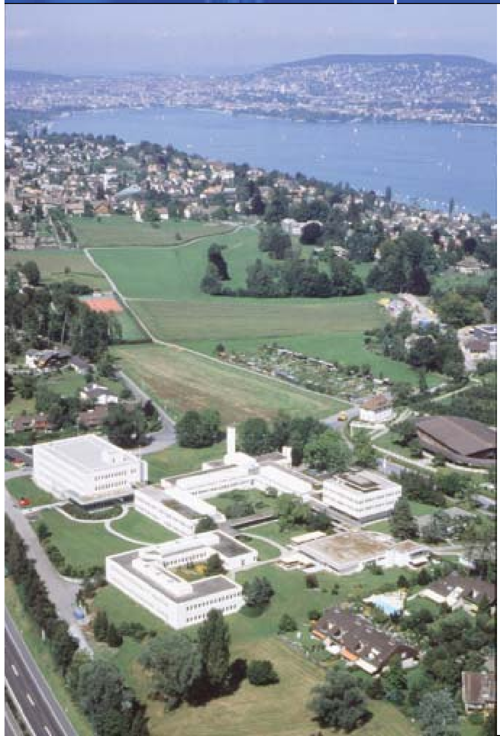
Zurich Research Laboratory



Network Anomaly Detection Based on Behavioral Traffic Pattern Recognition

Andreas Kind
Paul Hurley
Jeroen Massar
Xenofontas Dimitropoulos

Network Anomalies

- Unusual and significant changes in network traffic characteristics
  - Data volume (octets, packets)
  - Flows (number, duration, size, service type)
  - Communication matrix (src/dst IP, src/dst ports)
  - Packets (size, flags)

- Caused by…
  - “Season”
  - Organizational change (eg, new application, new user group, new business process)
  - Flash crowd
  - Vulnerability scan
  - Outage, fault, misconfiguration (eg, port scanning AFS, DNS used by IDS)
  - DoS attack, self-propagating attack (virus, worm)
  - Research on networks

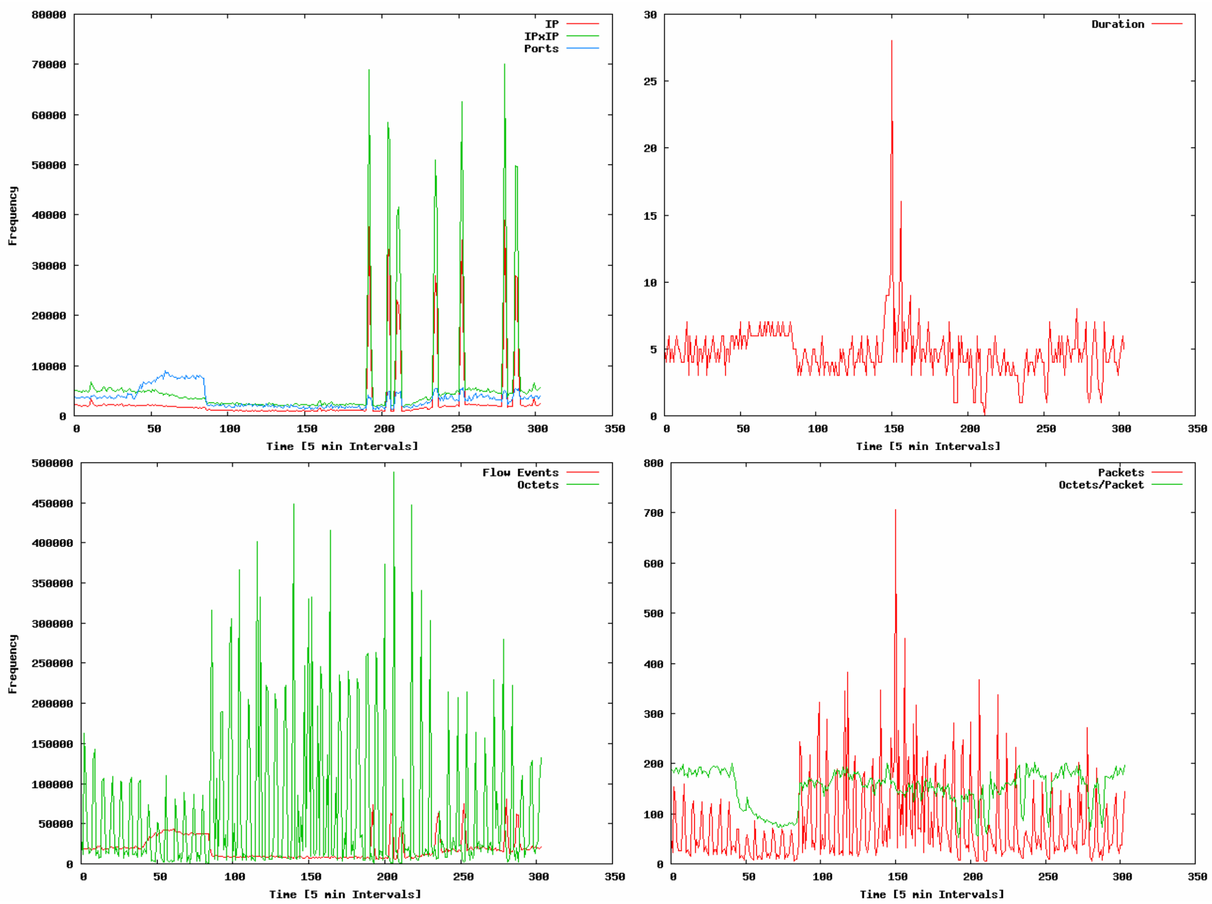
Network Anomalies

IP
IPxIP
Ports

Flow duration

Flows
Octets

Packets
Octets/packets



Detection Requirements

- Scalable for data centers
- No additional equipment (eg, splitters, taps, meter appliances)
- No traffic insertion (eg, active probing)
- No agents, no credentials
- No access to traffic payload
- No increase in monitoring traffic
- Real-time operation
- Low hardware costs
- No explicit configuration of thresholds and confidence intervals
- Applicable to highly varying workloads

  … which is a bit of a contradiction

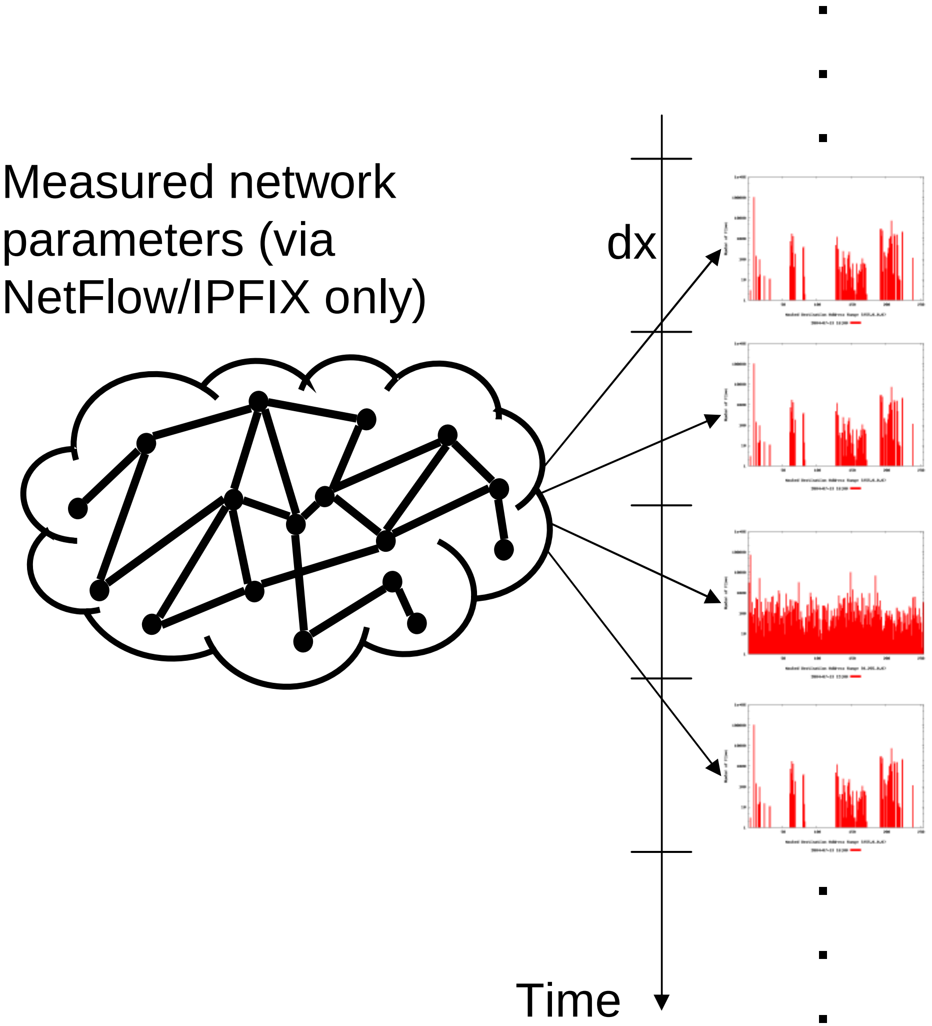
- No automatic prevention, no prediction, but deployment in combination with flow-based network profiling system
  – Which are the end-to-end flows causing the anomaly?

Related Work

- Signature-based approaches
  - Too slow, payload needed, only know worms/viruses are addressed
- Statistical approaches
  - Typically based on abrupt changes and therefore error-prone with varying workloads in distributed environments
- Rule-based approaches
  - Difficult to train, complex rule-sets too slow
- Service spoofing
  - Traffic destined to unused addresses is a priori suspicious
  - Most effective for worms
- Pattern-based approaches
  - Capture traffic patterns from network characteristics and compare with baseline pattern
  - How to compose and compare traffic patterns in order to address detection requirements?

Desired Detection System

Measured network parameters (via NetFlow/IPFIX only)



Behavioral Traffic Patterns
(ie, histograms of measured network parameters during time period dx)

OK

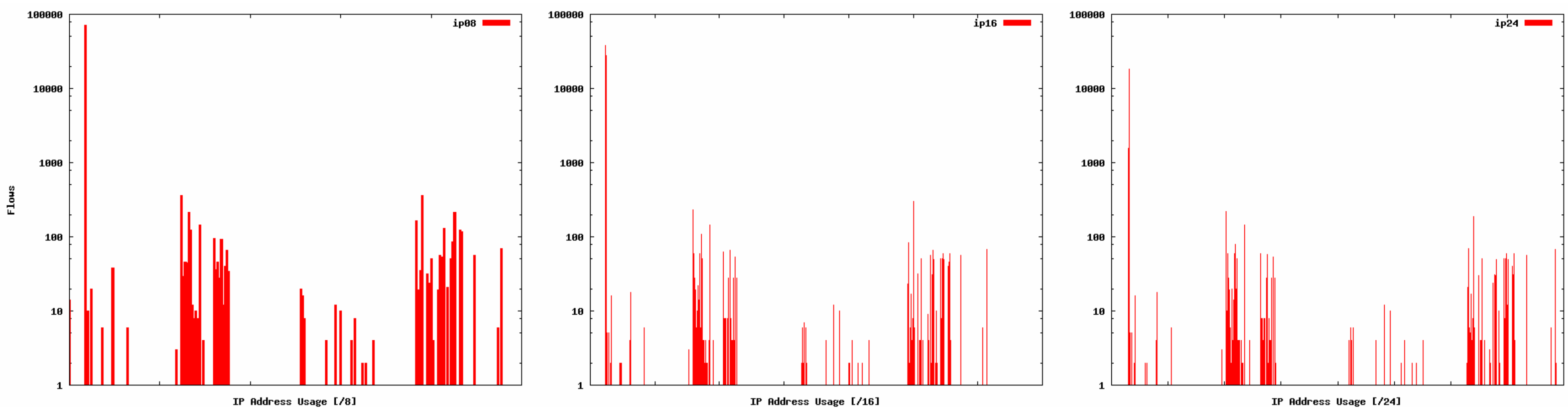
OK

not OK (ie, traffic anomaly)

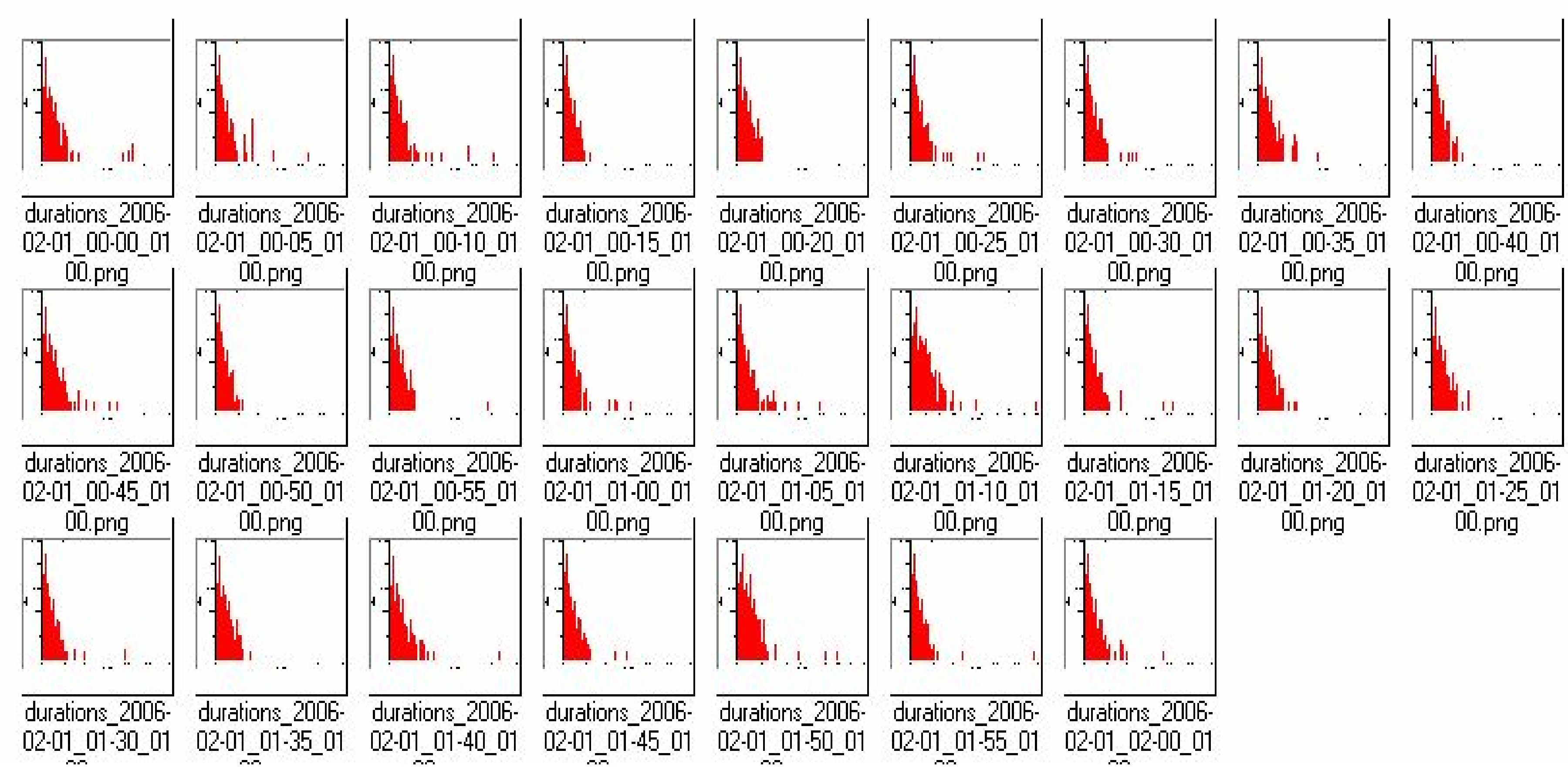
OK

Network Traffic Patterns

- Defined as histograms that display the frequency of flow parameter ranges during observation period
- Examples: IP address range, TCP/UDP port range, flow duration



Network Traffic Patterns

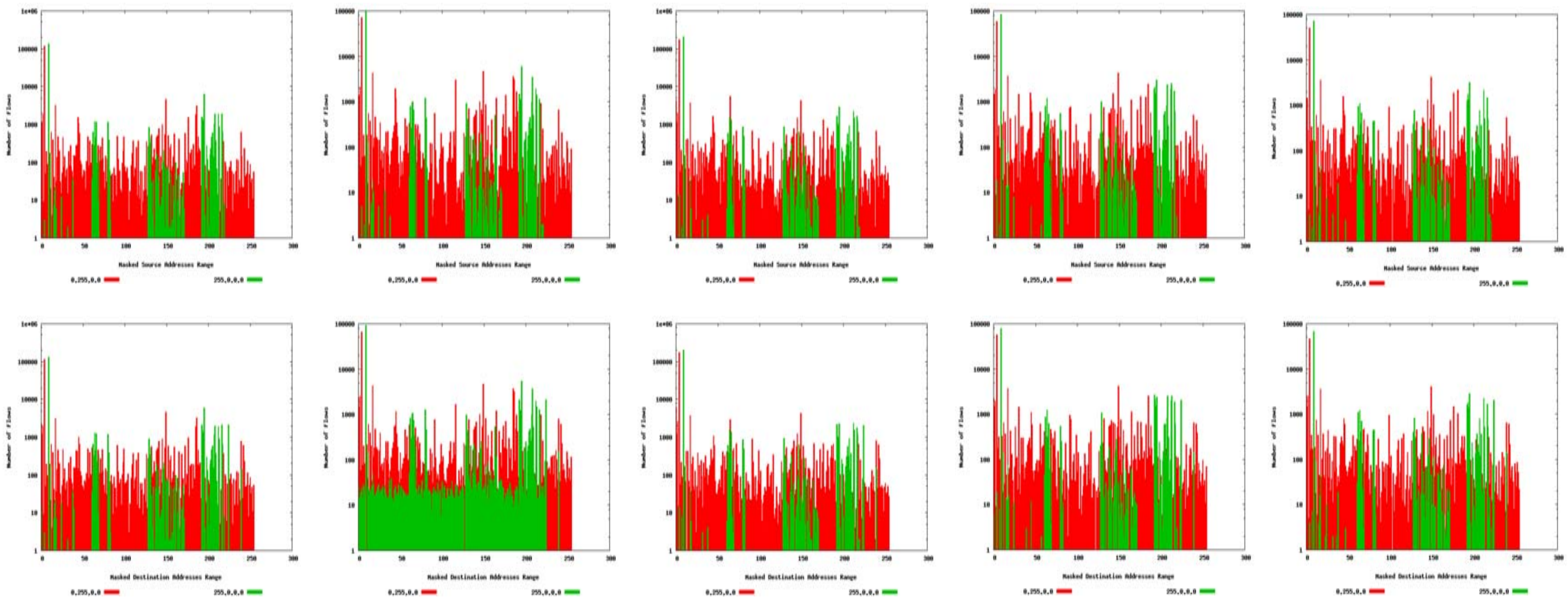


Behavioral Analysis of Virus Activity

Mon Tue Wed Thu Fri

Src Addr

Dst Addr



Host scan

prefix mask 255.0.0.0

prefix mask 0.255.0.0

Distance Between Traffic Patterns

- Defined as the number of changes in the relative order between two patterns

$$ord(w_1[],w_2[],i) = \begin{cases} 0 & \text{if} \quad (\, w_1[i] \geq w_1[mod(i,n)+1] \wedge \\ & \quad\quad w_2[i] \geq w_{2[}mod(i,n)+1] \,) \vee \\ & \quad\quad (\, w_1[i] \leq w_1[mod(i,n)+1] \wedge \\ & \quad\quad w_2[i] \leq w_2[mod(i,n)+1] \,) \\ 1 & \text{otherwise} \end{cases}$$

- Example

Given w_1 = (1,2,3,4), w_2 = (0,7,2,1)

$ord(w_1,w_2,1) = 0$

$ord(w_1,w_2,2) = 1$

$ord(w_1,w_2,3) = 1$

$ord(w_1,w_2,4) = 0$

Distance Between Traffic Patterns

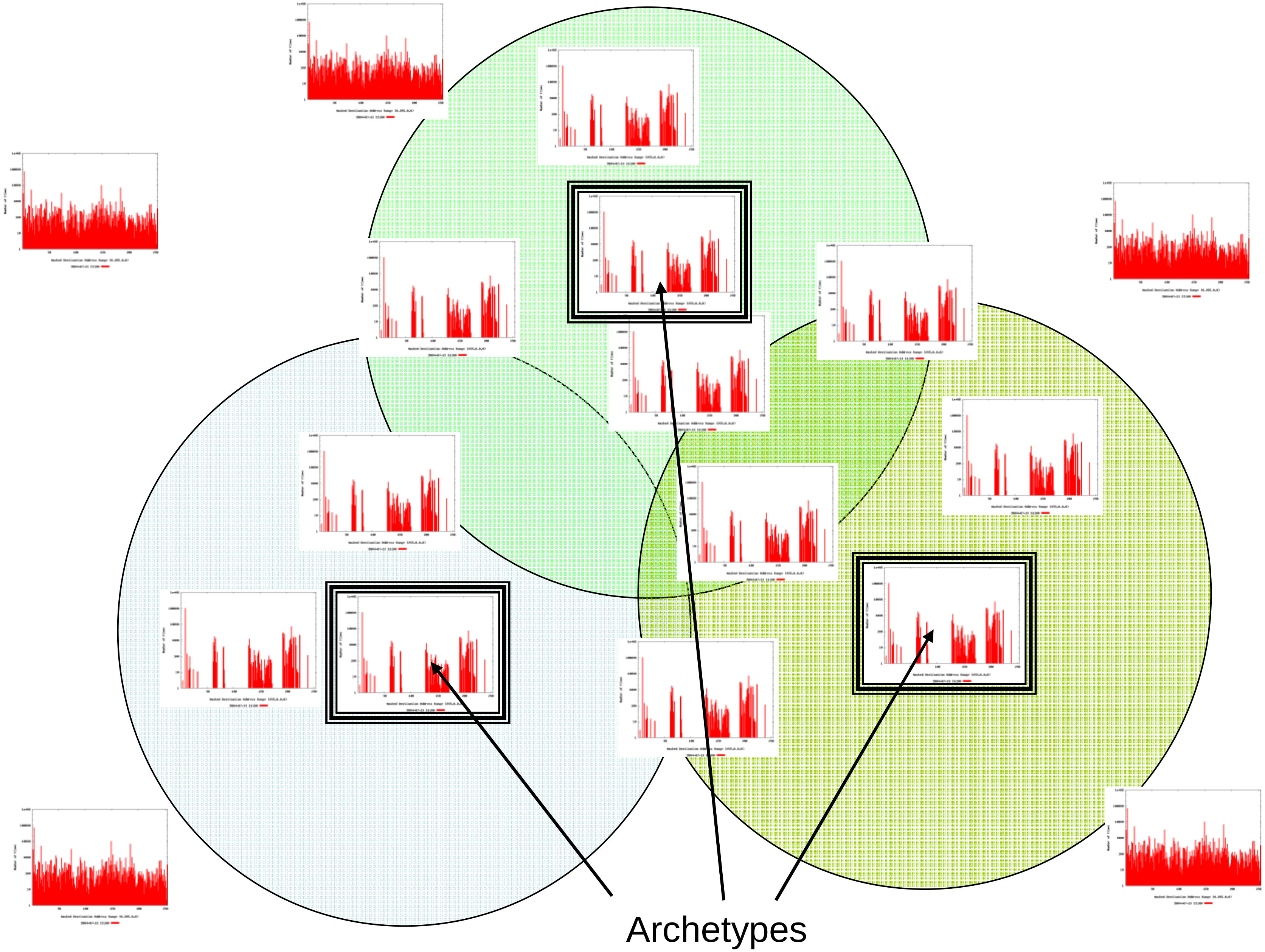
- Distance function

  $d(w_1, w_2) = 1/n \sum ord(w_1, w_2, i)$ for $0 < i \leq n$

- Example

  Given $w_1 = (1,2,3,4)$, $w_2 = (0,7,2,1)$

  $d(w_1, w_2) = 1/4 * 2 = 0.5$


Archetypes

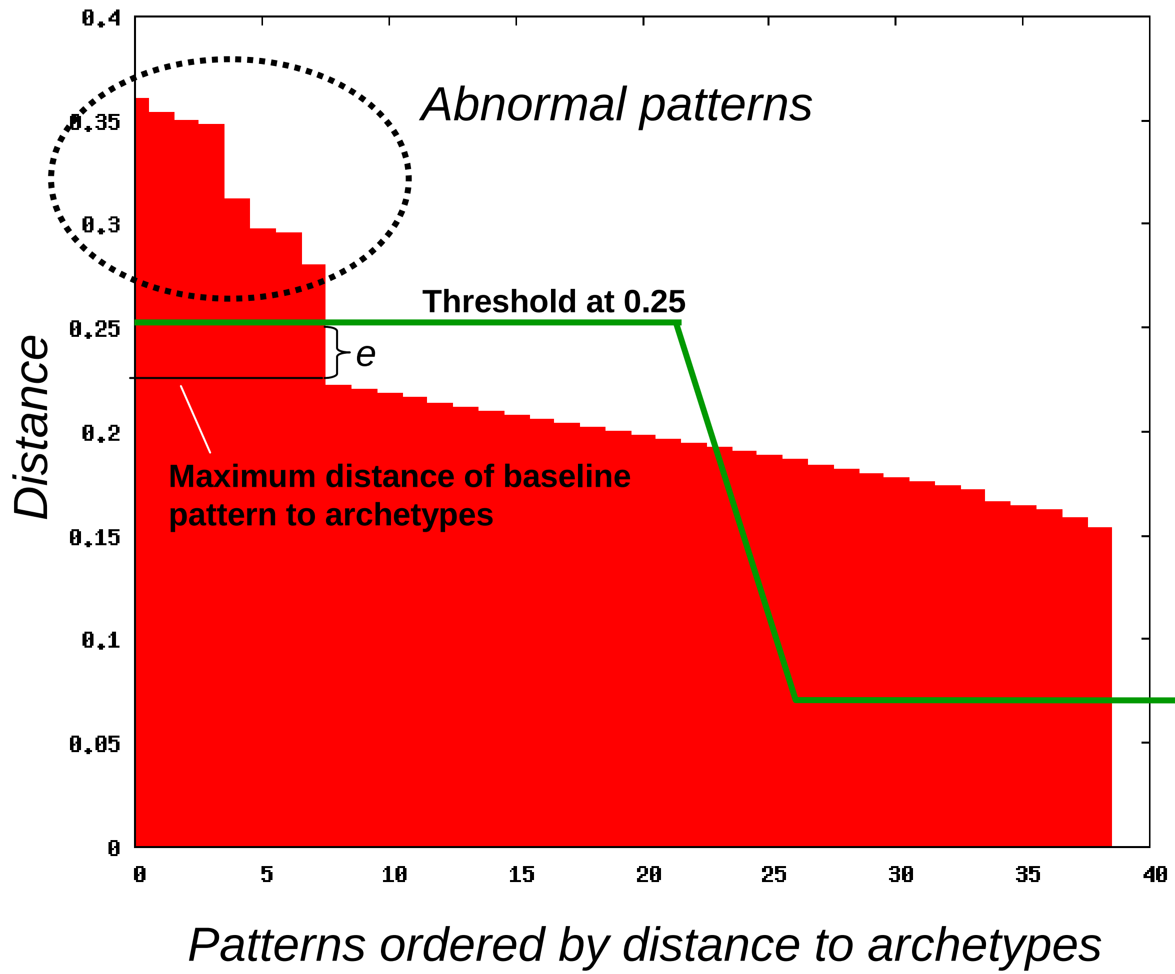
Clustering Traffic Patterns

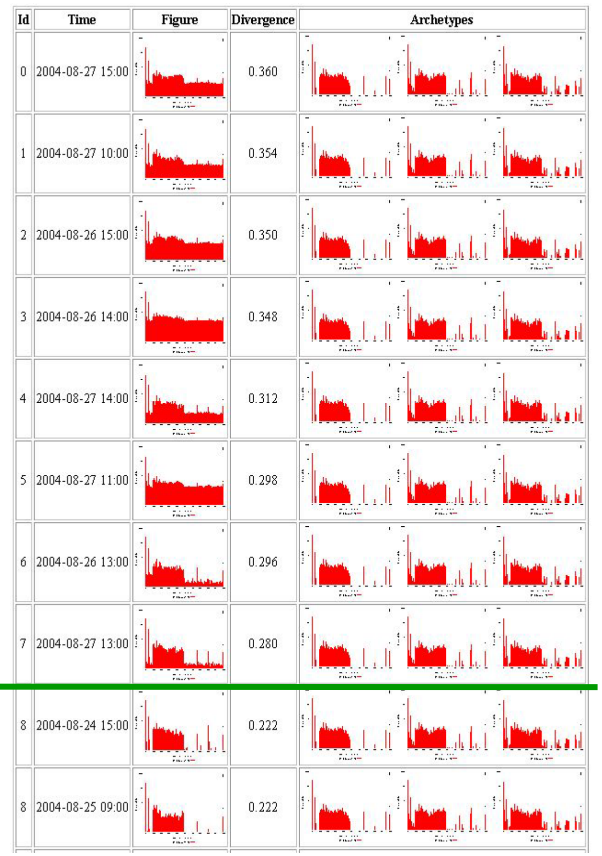
- Tree Clustering
  - Joining patterns into successively larger clusters using distance function
  - Results in hierarchical tree
  - But: How to determine mean (most likely “dummy”) pattern for which variability in distances to other cluster members is the smallest?
- *k*-Means Clustering
  - Given fixed member of *k* clusters
  - Assign patterns to clusters so that overall variability in distances to other cluster members is minimized

Traffic Pattern Archetypes

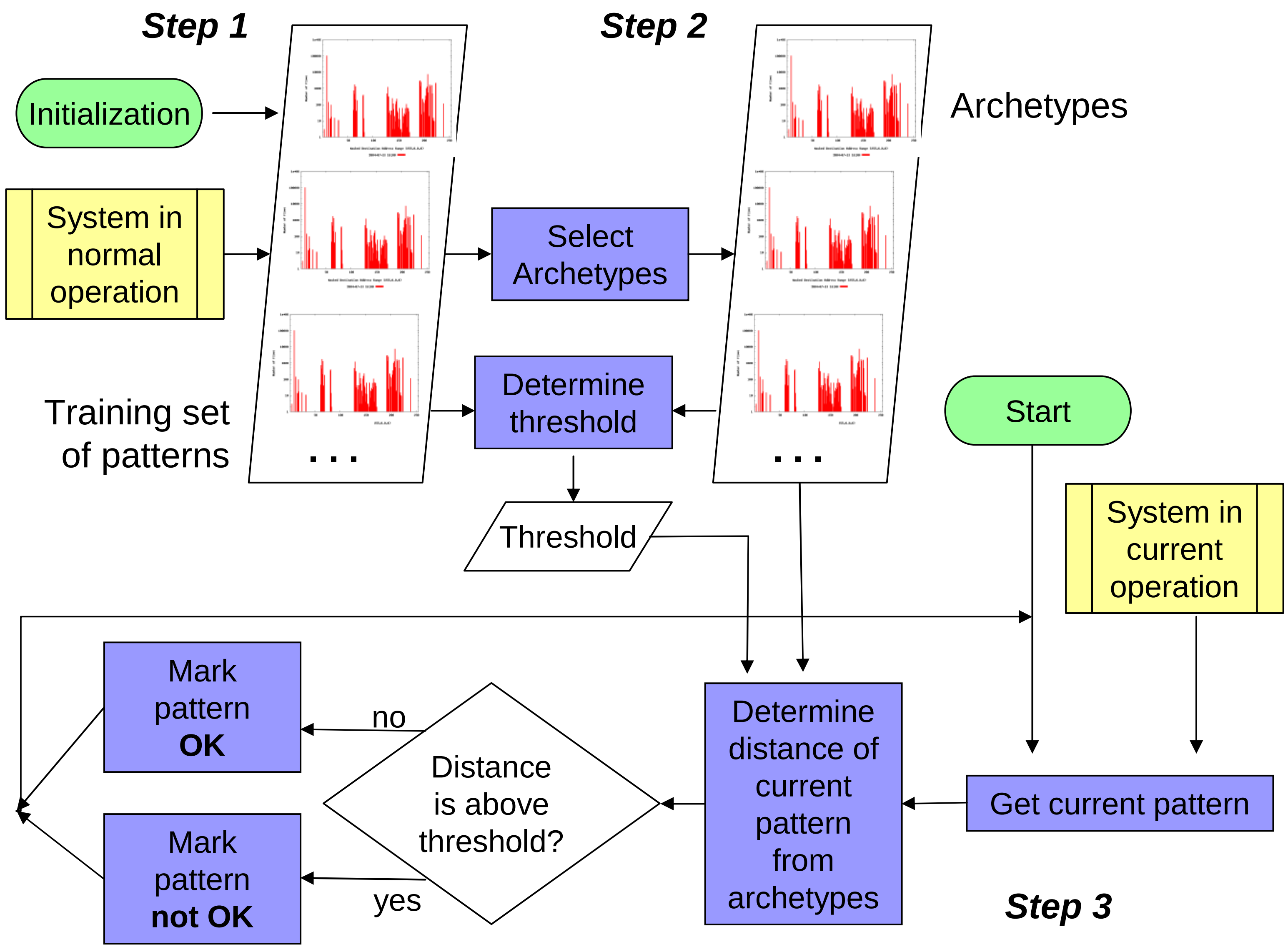
- Traffic pattern archetypes are computed with *k*-means clustering
- Find $w_1, \ldots, w_i \in W$ so that $\sum_i \mathrm{MIN}\ d(w_i, w_k)$ with $w_k \in W \setminus \{w_1, \ldots, w_i\}$ is minimized
  - Find the *i* patterns for which the sum of the minimum distances to all other patterns is minimized
  - We used *i* up to 4

Validation



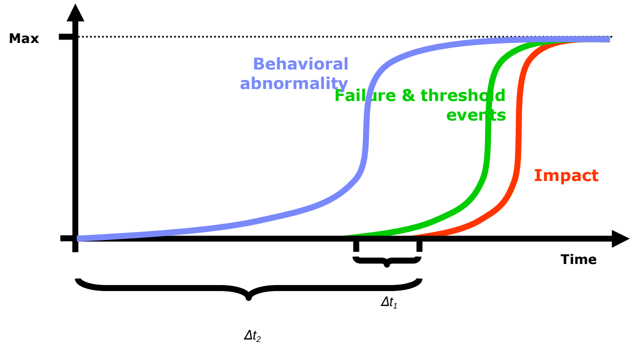


Id	Time	Figure	Divergence	Archetypes
0	2004-08-27 15:00		0.360	
1	2004-08-27 10:00		0.354	
2	2004-08-26 15:00		0.350	
3	2004-08-26 14:00		0.348	
4	2004-08-27 14:00		0.312	
5	2004-08-27 11:00		0.298	
6	2004-08-26 13:00		0.296	
7	2004-08-27 13:00		0.280	
8	2004-08-24 15:00		0.222	
8	2004-08-25 09:00		0.222	


Step 1
Step 2
Initialization
System in normal operation
Training set of patterns
. . .
Select Archetypes
Determine threshold
Threshold
Archetypes
. . .
Start
System in current operation
Determine distance of current pattern from archetypes
Get current pattern
Distance is above threshold?
no
yes
Mark pattern OK
Mark pattern not OK
Step 3

Future Work

- Continue the theoretic and empirical work on this approach
- Experiment with different distance functions and clustering algorithms
- Prove the time advantage of behavioral network problem prediction



Avalanche analogy

- Close integration with IBM's flow-based network profiling system
- Use approach with server workloads
- Visualization with force-directed graphs (ie, attractive/repulsive forces)
  - ip08, duration, …

THANKS!

http://www.zurich.ibm.com/aurora