Troubleshooting Chronic Conditions in Large IP Networks

Ajay Mahimkar, Jennifer Yates, Yin Zhang,
Aman Shaikh, Jia Wang, Zihui Ge, Cheng Tien Ee

UT-Austin and AT&T Labs-Research
mahimkar@cs.utexas.edu

ACM CoNEXT 2008

Network Reliability

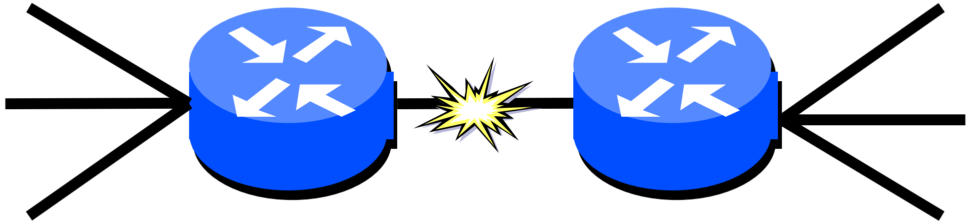
- Applications demand high reliability and performance
  - VoIP, IPTV, Gaming, …
  - Best-effort service is no longer acceptable

- Accurate and timely troubleshooting of network outages required
  - Outages can occur due to mis-configurations, software bugs, malicious attacks
    - Can cause significant performance impact
    - Can incur huge losses

Hard Failures

- Traditionally, troubleshooting focused on hard failures
  - E.g., fiber cuts, line card failures, router failures
  - Relatively easy to detect
  - Quickly fix the problem and get resource up and running

Link failure

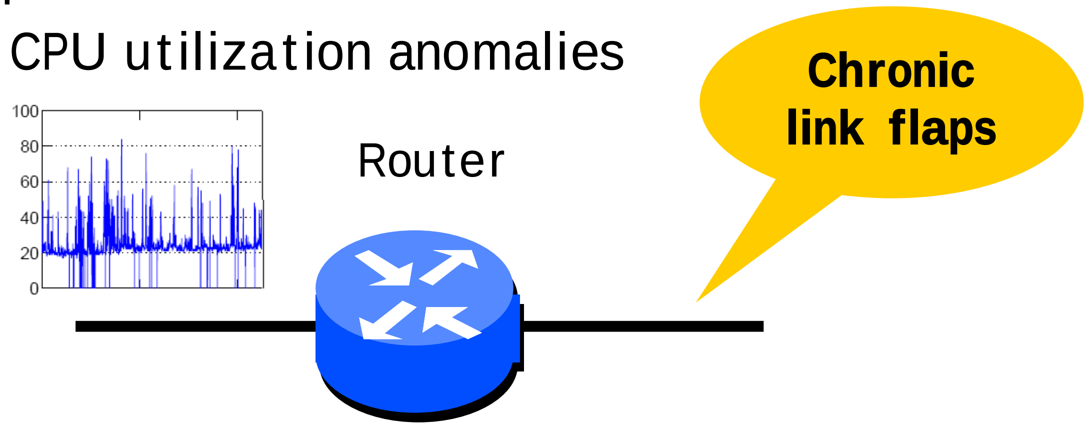


Lots of other network events flying under the radar,
and potentially impacting performance

Chronic Conditions

- Individual events disappear before an operator can react to them
- Keep re-occurring
- Can cause significant performance degradation
  - Can turn into hard failure
- Examples
  - Chronic link flaps
  - Chronic router CPU utilization anomalies

Router CPU Spikes



Troubleshooting Chronic Conditions

- Detect and troubleshoot before customer complains

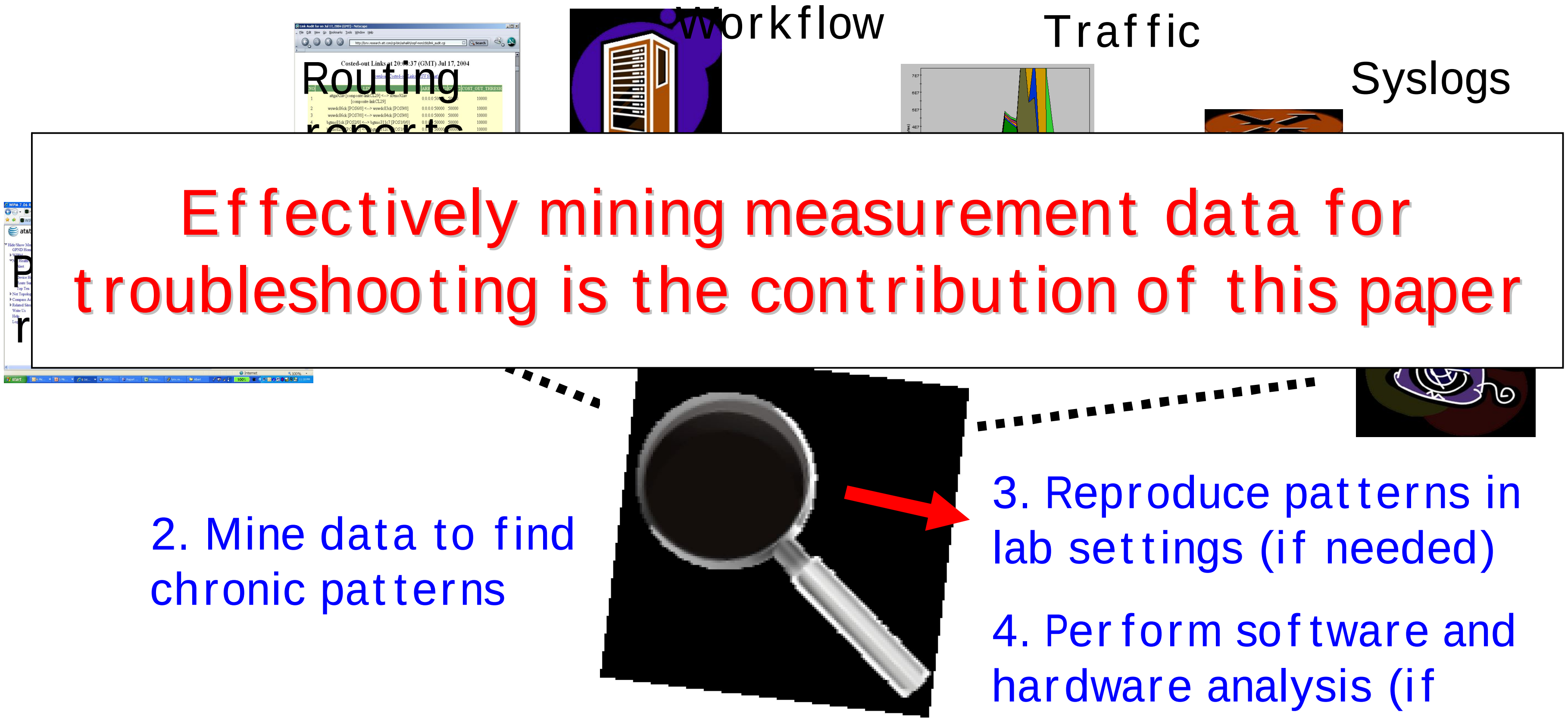
- State of art
  - Manual troubleshooting

- **N**etwork-wide **I**nformation **C**orrelation and **E**xploration (NICE)
  - First infrastructure for automated, scalable and flexible troubleshooting of chronic conditions
  - Becoming a powerful tool inside AT&T
    - Used to troubleshoot production network issues
    - Discovered anomalous chronic network conditions

Outline

- Troubleshooting Challenges
- NICE Approach
- NICE Validation
- Deployment Experience
- Conclusion

Troubleshooting Chronic Conditions is hard



Effectively mining measurement data for troubleshooting is the contribution of this paper

2. Mine data to find chronic patterns

3. Reproduce patterns in lab settings (if needed)

4. Perform software and hardware analysis (if needed)

Troubleshooting Challenges

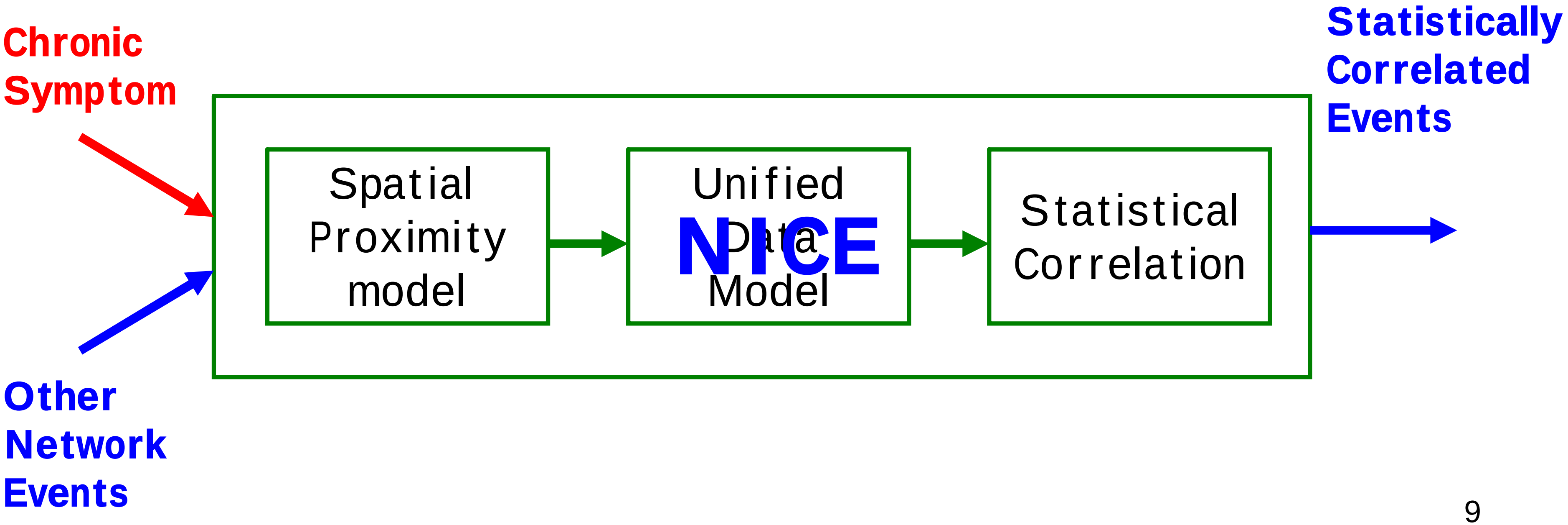
- Massive Scale
  - Potential root-causes hidden in thousands of event-series
  - E.g., root-causes for packet loss include link congestion (SNMP), protocol down (Route data), software errors (syslogs)

- Complex spatial and topology models
  - Cross-layer dependency
  - Causal impact scope
    - Local versus global (propagation through protocols)

- Imperfect timing information
  - Propagation (events take time to show impact – timers)
  - Measurement granularity (point versus range events)

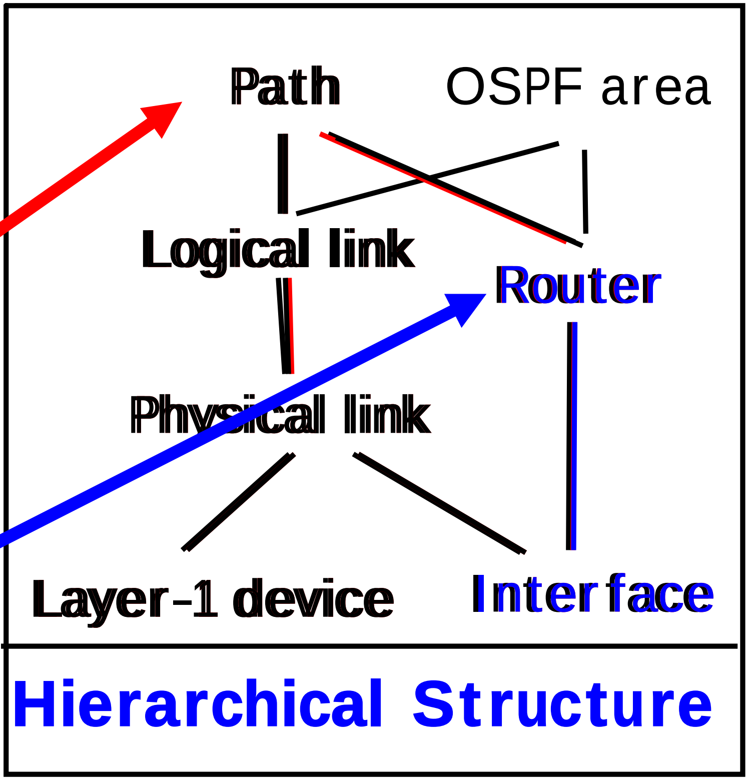
NICE

- Statistical correlation analysis across multiple data
  - Chronic condition manifests in many measurements
- Blind mining leads to information snow of results
  - NICE starts with symptom and identifies correlated events



Spatial Proximity Model

- Select events in close proximity
- Hierarchical structure
  - Capture event location
- Proximity distance
  - Capture impact scope of event
- Examples
  - Path packet loss - events on routers and links on same path
  - Router CPU anomalies - events on same router and interfaces



Path

OSPF area

Logical link

Router

Physical link

Layer-1 device

Interface

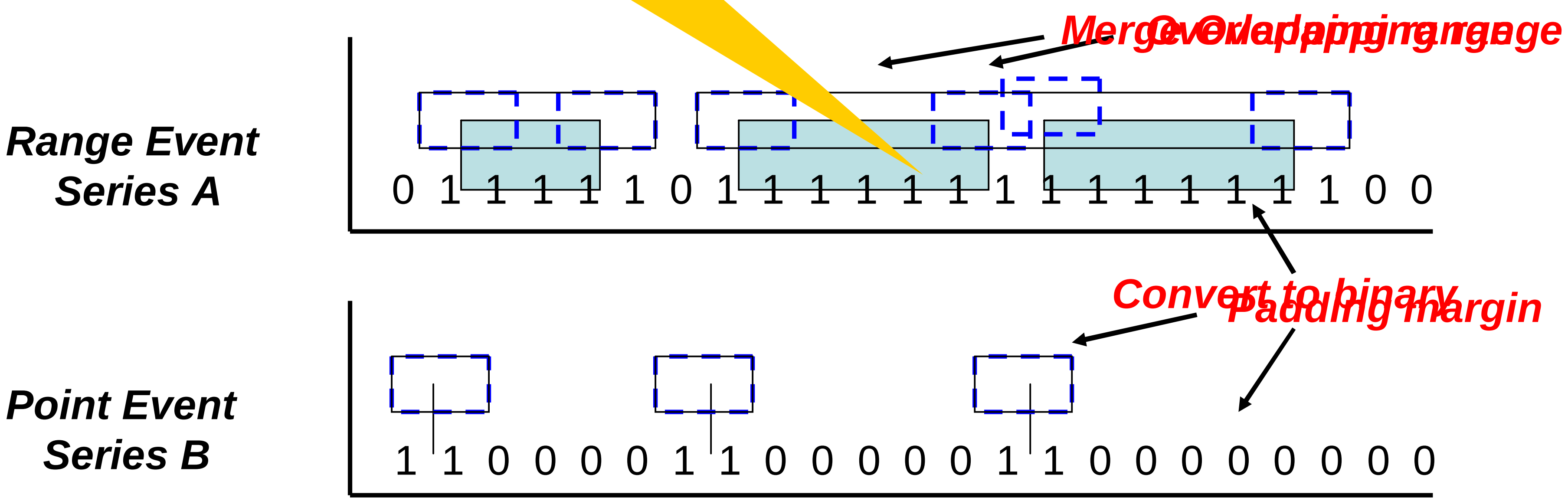
Hierarchical Structure

Network operators find it flexible and convenient to express the impact scope of network events

Unified Data Model

- Facilitate easy cross-event correlations
- Padding time-margins to handle diverse data
  - Convert any event-series to range series
- Co [illegible] y correlations
  - Convert range-se [illegible] to binary time-series

Auto-correlation


Merge overlapping range
Overlapping range
Range Event Series A
0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
Convert to binary
Padding margin
Point Event Series B
1 1 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0

Statistical Correlation Testing

- Co-occurrence is not sufficient
- Measure statistical time co-occurrence
  - Pair-wise Pearson's correlation coefficient
- Unfortunately, cannot apply the classic significance test
  - Due to auto-correlation
    - Samples within an event-series are not independent
    - Over-estimates the correlation confidence: high false alarms
- We propose a novel circular permutation test
  - Key Idea: Keep one series fixed and shift another
    - Preserve auto-correlation
    - Establishes baseline for null hypothesis that two series are independent

NICE Validation

- Goal: Test if NICE correlation output matches networking domain knowledg
  - Validation using 6 months of d

Expected to correlate, NICE marked uncorrelated

Expected to not correlate, NICE marked correlated

			NICE Correlat Results		
Pairs for correlation testing	**Expected not to correlate**	**Expected to correlate**	**Matched outputs**	**Unexpected Correlations**	**Missed Correlations**
1785	1592	193	1732	24	29

- For 97% pairs, NICE correlation output agreed with domain knowledge
- For remaining 3% mismatch, their causes fell into three categories
  - Imperfect domain knowledge
  - Measurement data artifacts
  - Anomalous network behavior

Anomalous Network Behavior

- Example – Cross-layer Failure interactions
  - Modern ISPs use failure recovery at layer-1 to rapidly recover from faults without inducing re-convergence at layer-3
    - i.e., if layer-1 has protection mechanism invoked successfully, then layer-3 should not see a link failure

- Expectation: Layer-3 link down events should not correlate with layer-1 automated failure recovery
  - Spatial proximity model: SAME LINK

- Result: NICE identified strong statistical correlation
  - Router feature bugs identified as root cause
  - Problem has been mitigated

Troubleshooting Case Studies

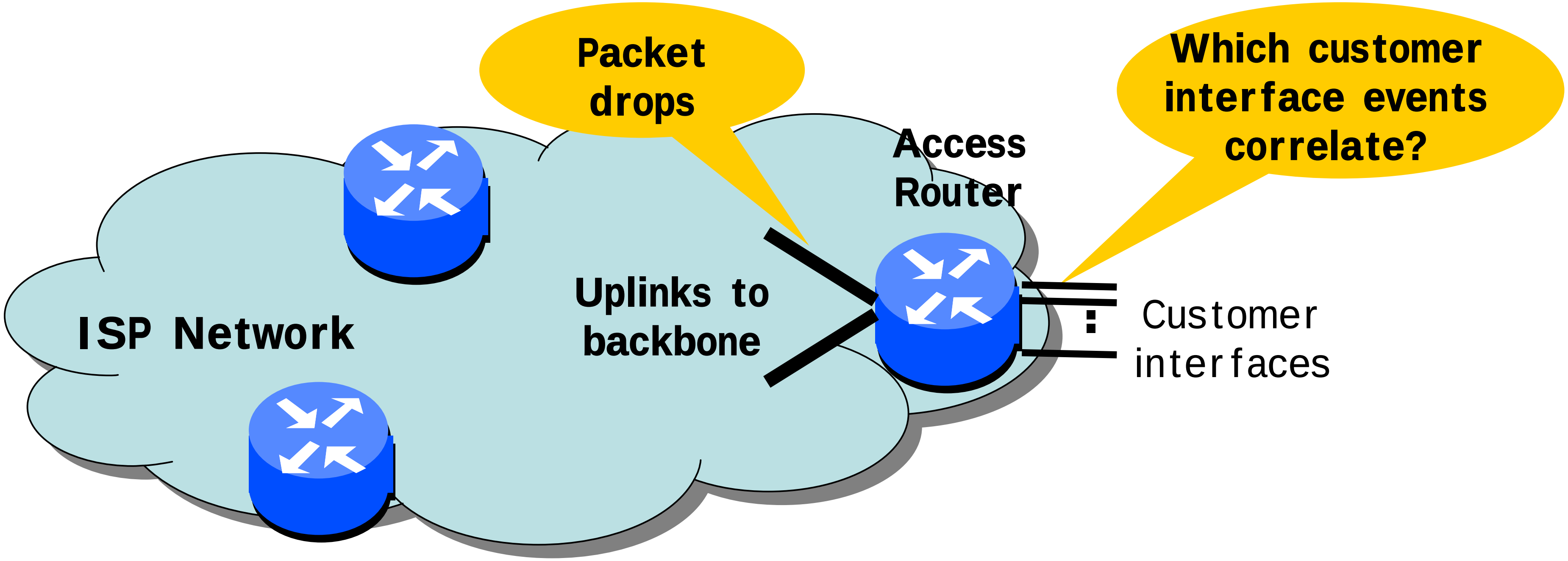
AT&T Backbone Network

- Uplink packet loss on an access router
- Packet loss observed by active measurement between a router pair
- CPU anomalies on routers

Data Source	Number of Event types
Layer-1 Alarms	130
SNMP	4
Router Syslogs	937
Command Logs	839
OSPF Events	25
Total	1935

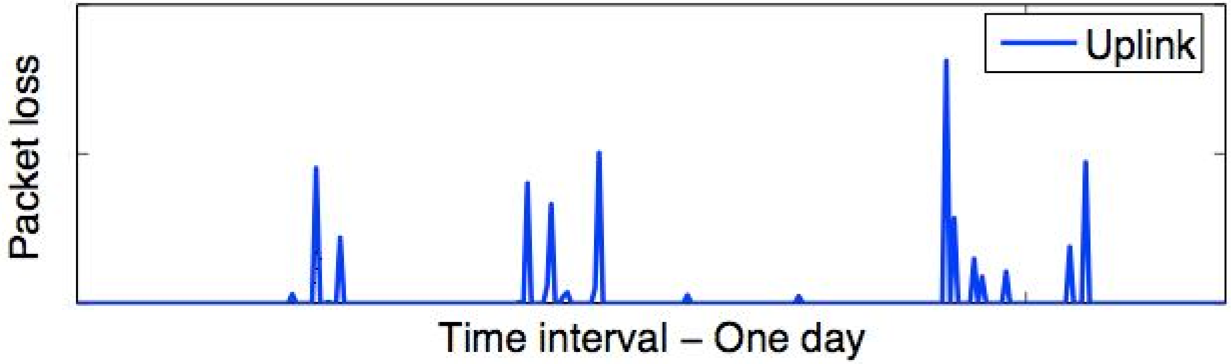
All three case studies uncover interesting correlations with new insights

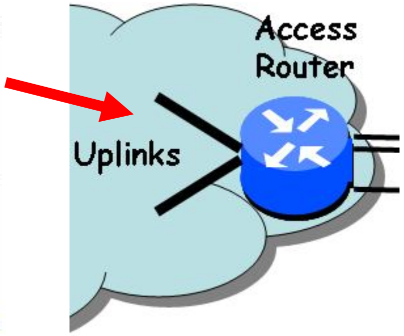
Chronic Uplink Packet loss

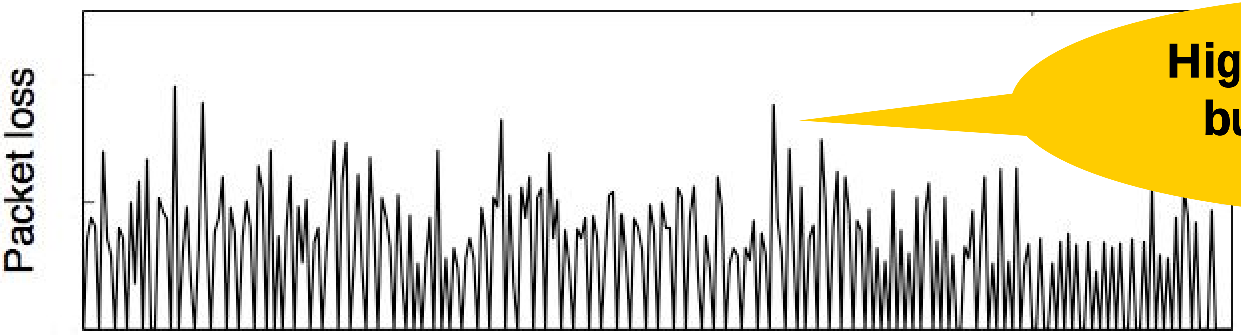


- Problem: Identify strongly correlated event-series with chronic packet drops on router uplinks
  - Significantly impacting customers
- NICE Input: Customer interface packet drops (SNMP) and router syslogs

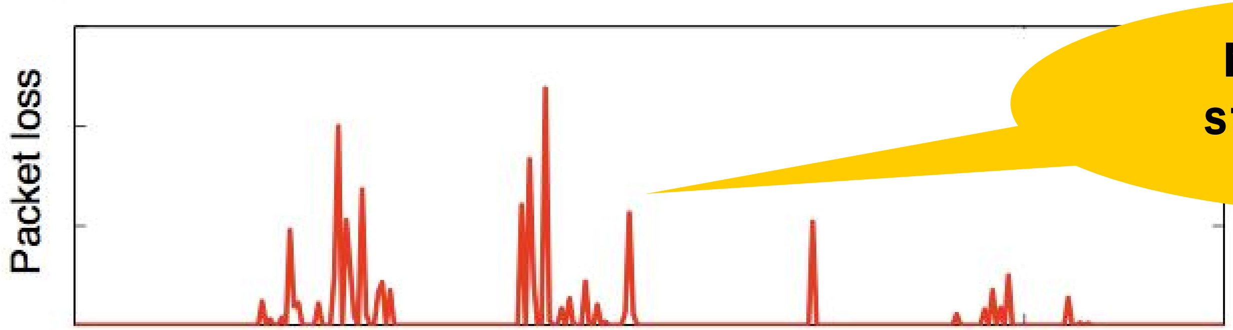
Chronic Uplink Packet loss







High co-occurrence, but no statistical correlation



NICE identifies strong statistical correlation

Chronic Uplink Packet loss

- NICE Findings: Strong Correlations with
  - Packet drops on four customer-facing interfaces (out of 150+ with packet drops)
    - All four interfaces from SAME CUSTOMER
  - Short-term traffic bursts appear to cause internal router limits to be reached
    - Impacts traffic flowing out of router
    - Impacting other customers
  - Mitigation Action: Re-home customer interface to another access router

Conclusions

- Important to detect and troubleshoot chronic network conditions before customer complains

- NICE – First scalable, automated and flexible infrastructure for troubleshooting chronic network conditions
  - Statistical correlation testing
  - Incorporates topology and routing model

- Operational experience is very positive
  - Becoming a powerful tool inside AT&T

- Future Work
  - Network behavior change monitoring using correlations
  - Multi-way correlations

Thank You !

Backup Slides ...

Router CPU Utilization Anomalies

- Problem: Identify strongly correlated event-series with chronic CPU anomalies as input symptom

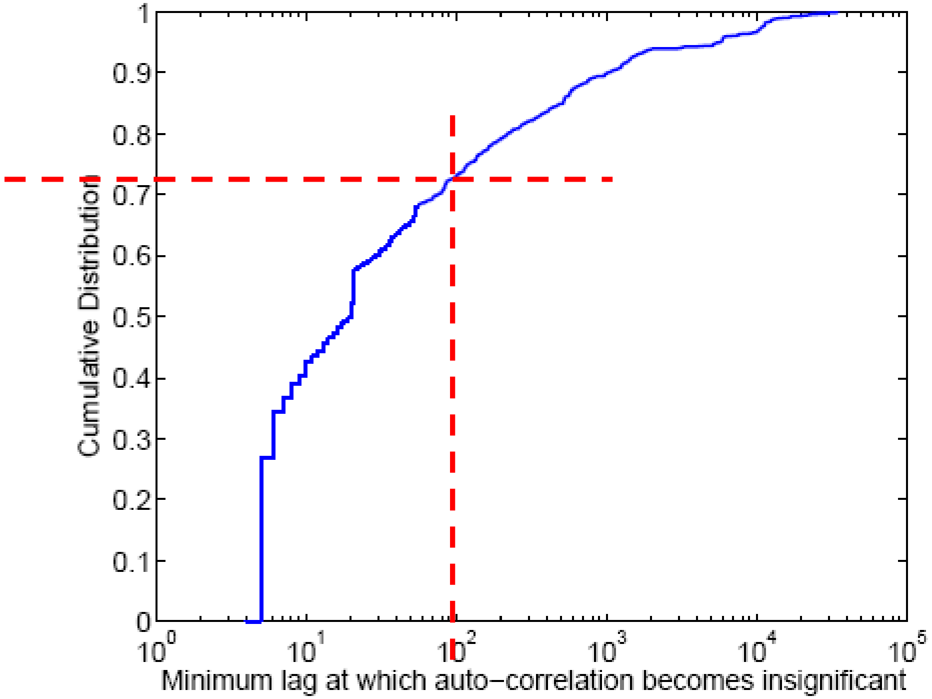
- NICE Input: Router syslogs, rou
logs and layer-1 alarms

Consistent with earlier operations findings

- NICE Findings: Strong Correlations with
  - Control-plane activities
  - Commands such as viewing routing protocol states
  - Customer-provisioning
  - SNMP polling
    - Mitigation Action: Operators are working with router polling systems to refine their polling mechanisms

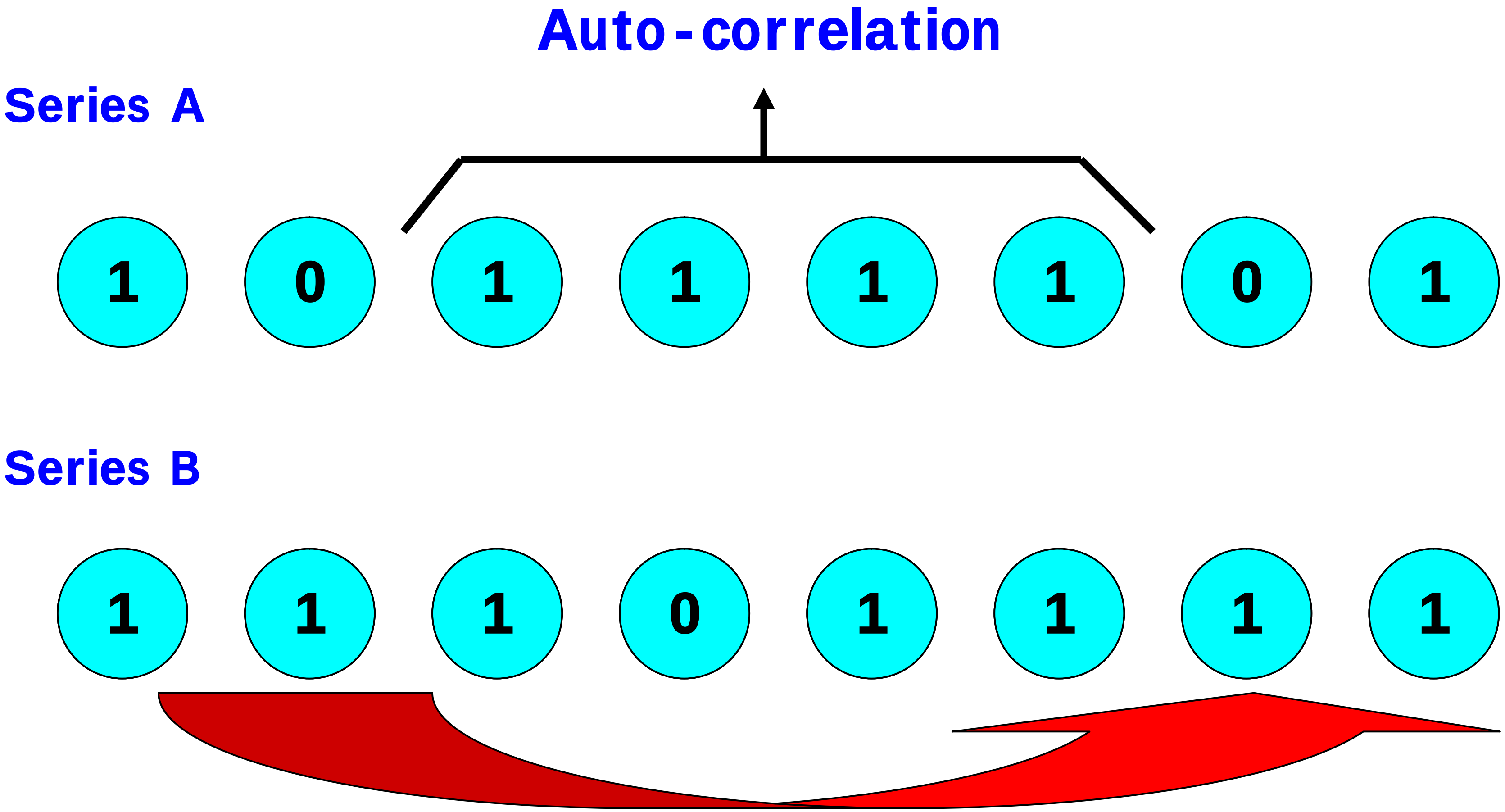
New

Auto-correlation



About 30% of event-series have significant auto-correlation at lag 100 or higher

Circular Permutation Test



Auto-correlation

Series A

1 0 1 1 1 1 0 1

Series B

1 1 1 0 1 1 1 1

Permutation provides correlation baseline to test hypothesis of independence

Imperfect Domain Knowledge

- Example – one of router commands used to view routing state is considered highly CPU intensive

- We did not find significant correlation between the command and CPU value as low as 50%
  - Correlation became significant only with CPU above 40%
  - Conclusion: The command does cause CPU spikes, but not as high as we had expected
    - Domain knowledge updated !