Orna Agmon Ben-Yehuda

Presents

Communication-Efficient Online Detection of Network-Wide Anomalies

Ling Huang* XuanLong Nguyen*

Minos Garofalakis§ Joe Hellerstein*

Michael Jordan* Anthony Joseph* Nina Taft§

*UC Berkeley §Intel Research

Coming on Spring 2011 to a

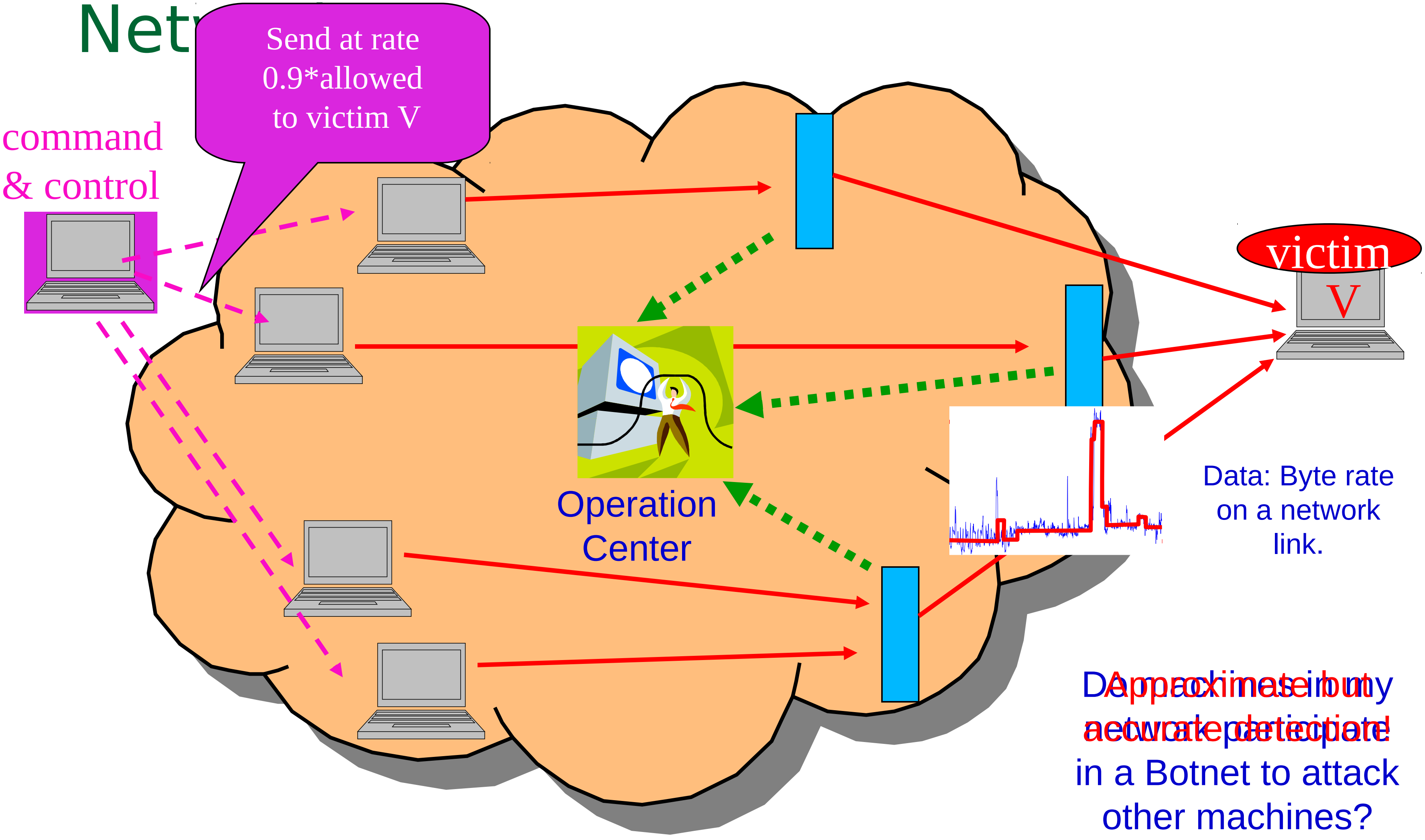
Seminar 236803 on Processing and Mining Distributed Data

Near you

Network-Wide Anomalies

- Are bad:
  - Router mis-configurations
  - Border Gateway Protocol (BGP) policy modifications
  - Device failures
- Or even malicious:
  - DDOS attacks
  - Viruses, spam sending
  - Port scanning
- But also just unpredictable
  - Flash Crowds (mob) supercomputing

Detection Problems in Enterprise Net


command & control
Send at rate 0.9*allowed to victim V
victim
V
Operation Center
Data: Byte rate on a network link.

Do machines in my network participate in a Botnet to attack other machines?

Approximate but accurate detection!

For efficient and scalable detection, push data processing to the edge of network!

We shall talk about:

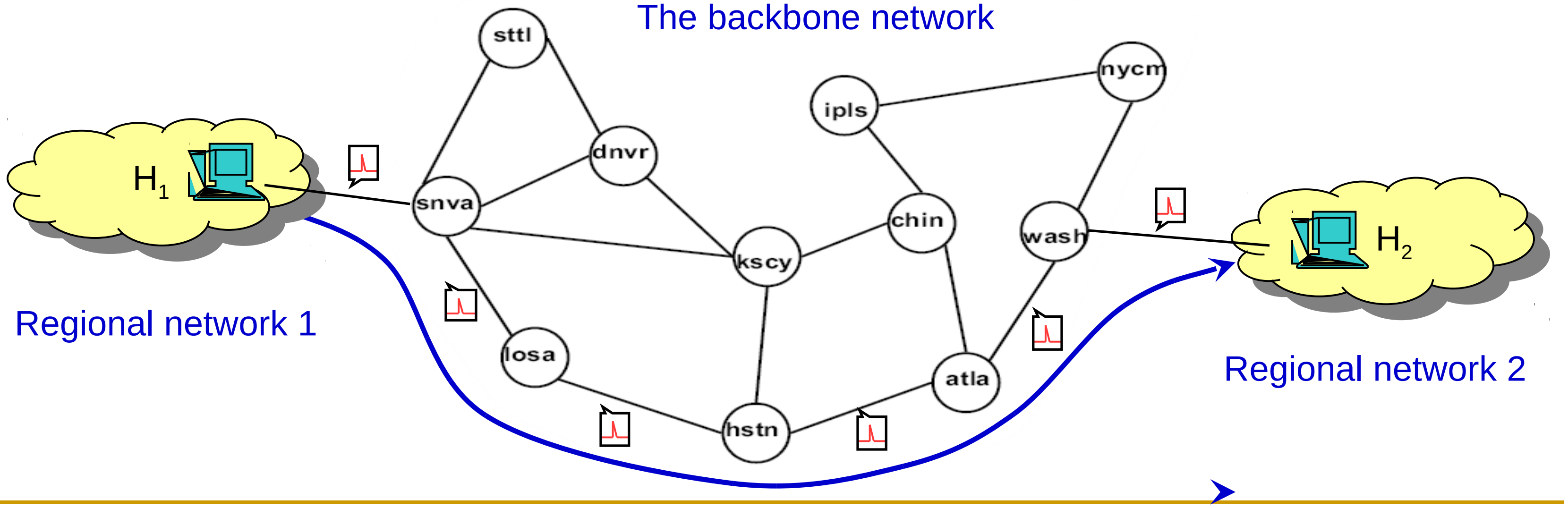
- Lakhina et al.'s centralized algorithm
- Decentralized anomaly detection
- Slack determination
- Evaluation
- Open Discussion

Towards Decentralized Detection

- Lakhina et al.: Distributed Monitoring & Centralized Computation
  - Stream-based data collection
  - *Periodically* evaluate detection function over collected data
  - Doesn’t scale well in network size, timescale, detection delay
- Huang et al.: Decentralized Detection
  - *Continuously* evaluate detection function in a decentr. way
  - Low-overhead, rapid response, accurate and scalable
  - Detection accuracy controllable by a “tuning knob”
    - Provable guarantee on detection error (false alarm rate)
    - Flexible tradeoff between overhead and accuracy

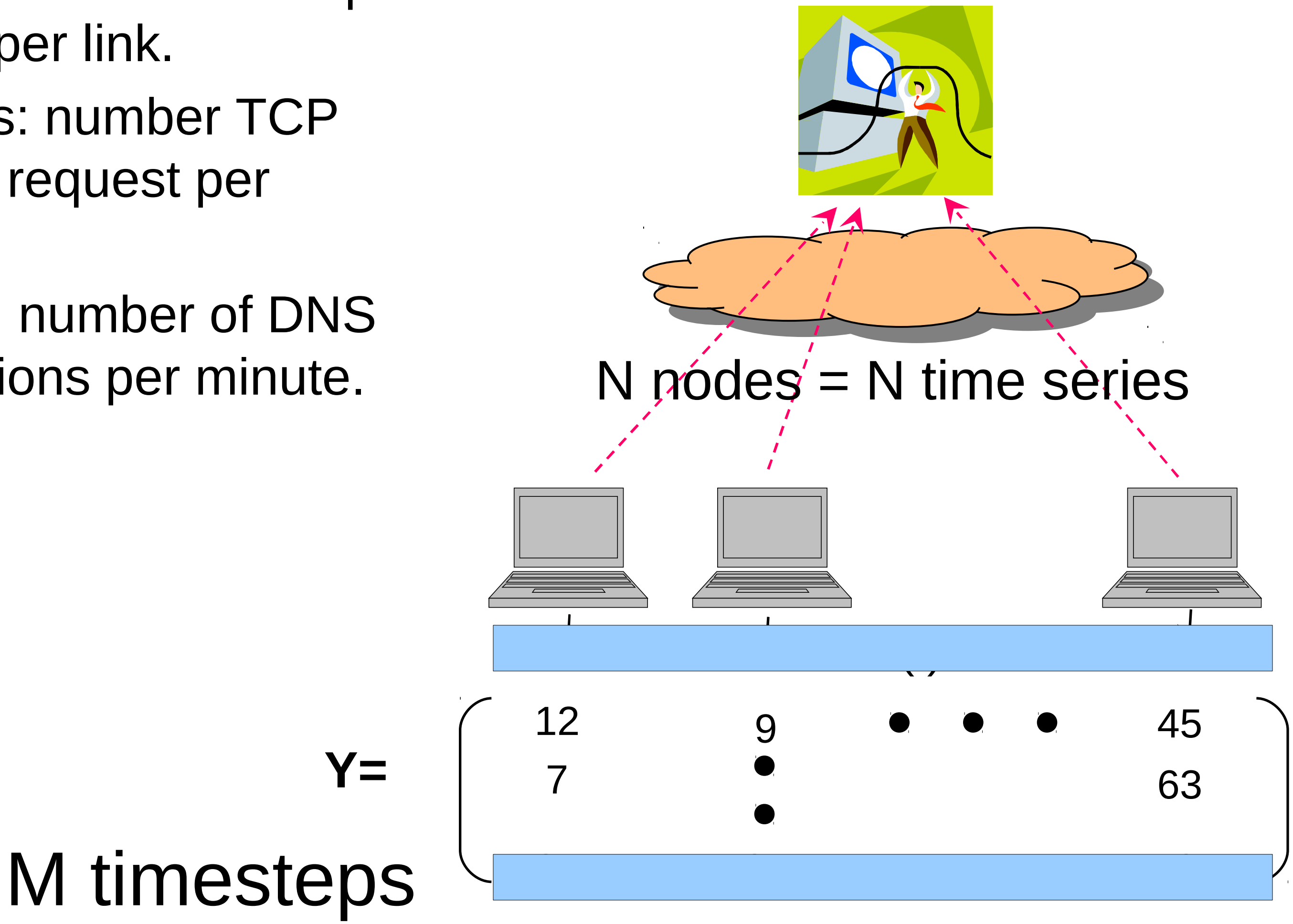
Detection of Network-wide Anomalies

- A ***volume anomaly*** is a sudden change in an Origin-Destination flow (*i.e.*, point to point traffic)
- Given link traffic measurements, ***detect*** the volume anomalies



The Data Collected by Monitors

- Routers: volume traffic per second per link.
- Firewalls: number TCP connect request per second.
- Servers: number of DNS transactions per minute.

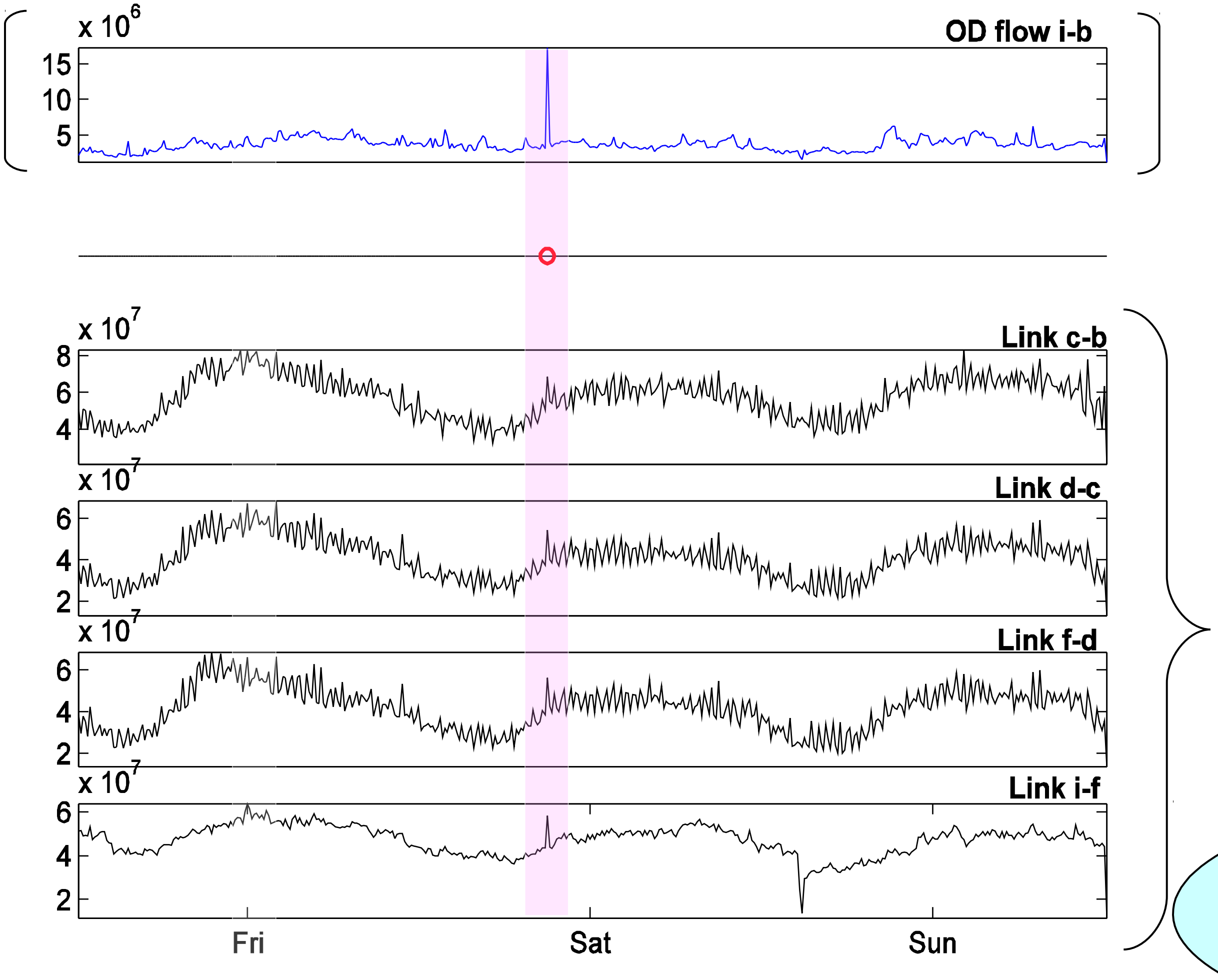


N nodes = N time series

$$Y = \begin{pmatrix} 12 & 9 & \bullet\ \bullet\ \bullet & 45 \\ 7 & \vdots & & 63 \end{pmatrix}$$

M timesteps

Flow vs. Link (Lakhina et al.)



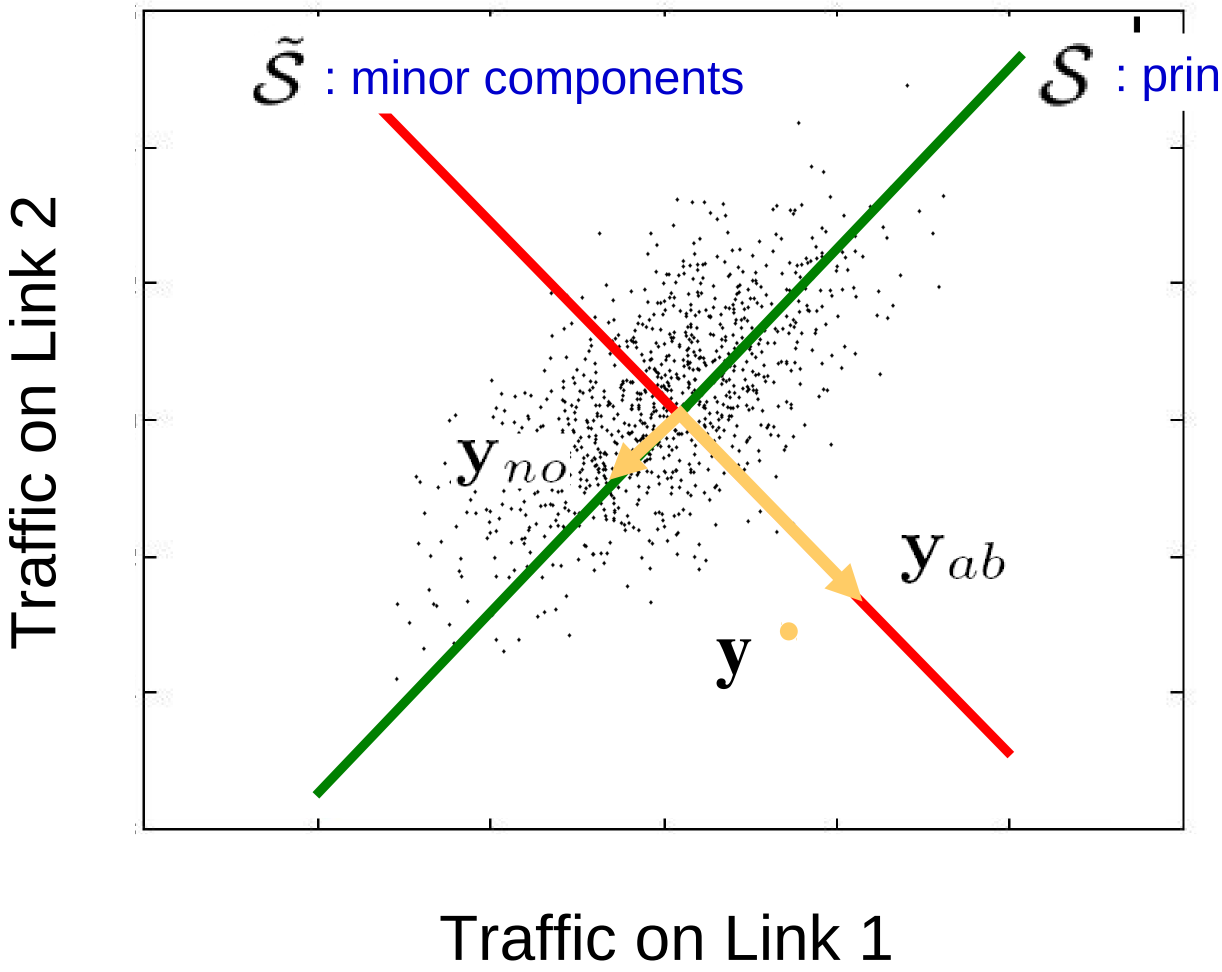
Anomalies in (unobserved) flow data

Observed network link data = aggregate of application-level flows

Each link is a dimension

Finding anomalies in high-dimensional, noisy data is difficult!

Principal Component Analysis (PCA)



Principal components are top eigenvectors of covariance matrix.

$$Y\,Y^T$$

They are also directions of maximal variance.

They form the subspace projection matrices C_{no} and C_{ab}

$$\mathbf{y}_{no} = \mathbf{C}_{no}\mathbf{y}$$

$$\mathbf{y}_{ab} = \mathbf{C}_{ab}\mathbf{y}$$

Anomalous traffic usually results in a large value of $\mathbf{y}_{ab}$

The Subspace Method (Lakhina'04)

- An approach to separate normal from anomalous traffic based on Principal Component Analysis (PCA)
- Normal Subspace $\mathcal{S}$: space spanned by the top *k* principal components
- Anomalous Subspace $\tilde{\mathcal{S}}$: space spanned by the remaining components
- Then, decompose traffic on all links by projecting onto $\mathcal{S}$ and $\tilde{\mathcal{S}}$ to obtain:

$$\mathbf{y} = \mathbf{y}_{no} + \mathbf{y}_{ab}$$

Traffic vector of all links at a particular point in time

Normal traffic vector

Residual traffic vector

Link Traffic Variance of Principle Components

- Link matrices have low dimensionality

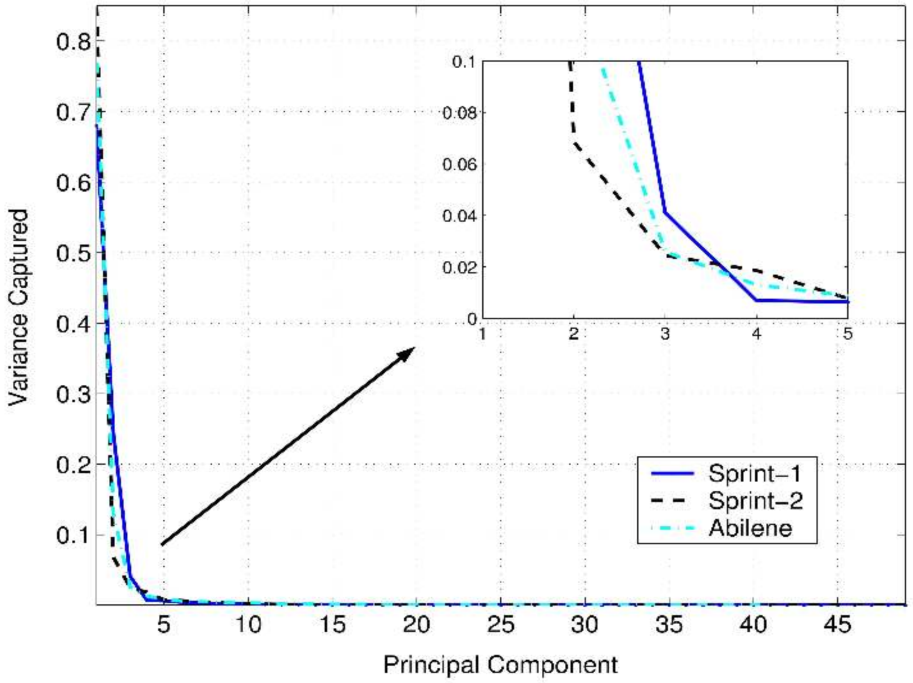
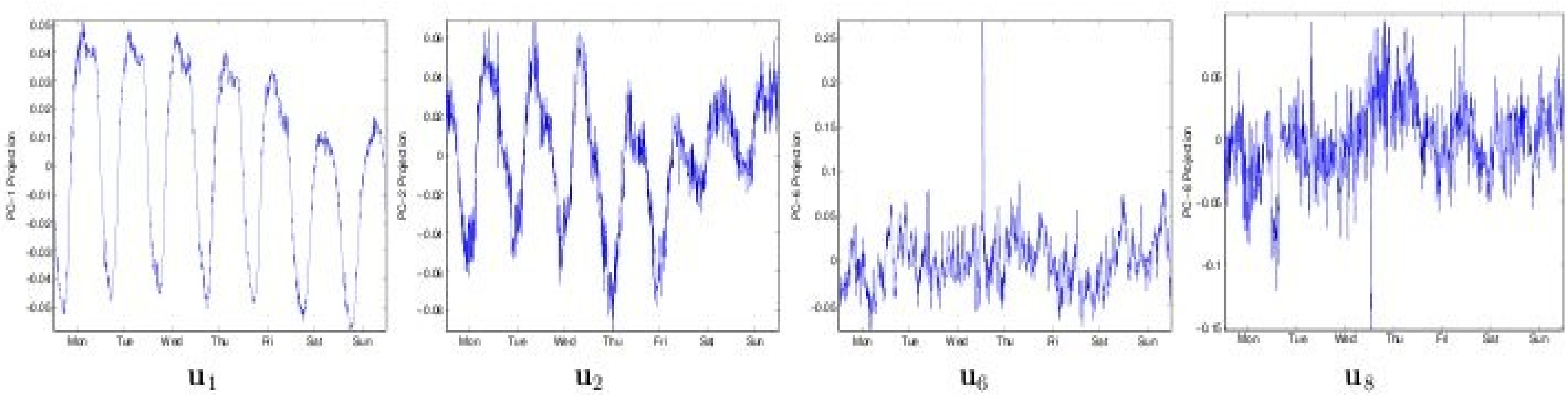


Figure 2: Fraction of total link traffic variance captured by each principal component.

Lakhina et al.,

Projections onto Principle Components –
normal and abnormal traffic variation



u_1 u_2

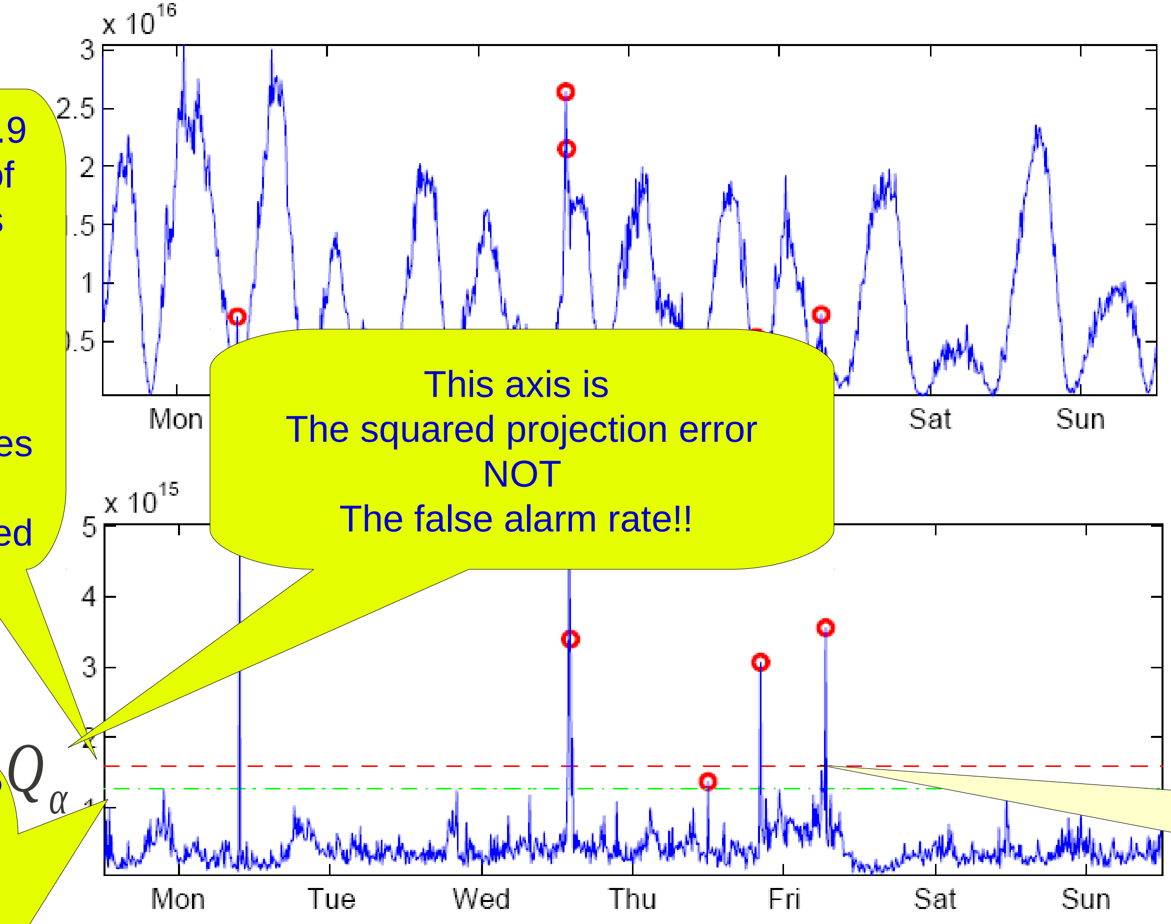
(a) Normal Behavior

u_6 u_8

(b) Anomalous Behavior

Figure 3: Projections onto principal components showing normal and anomalous traffic variation.

Detection Illustration



1-α=99.9
99.9% of
Alarms
Are
Real,
But
More
Anomalies
Go
undetected

This axis is
The squared projection error
NOT
The false alarm rate!!

1-α=99.5
Only
99.5% of
Alarms
Are real
But many
Anomalies
Are
detected

Value of $\|\mathbf{y}\|^2$
over time
(all traffic)

Value of $\|\mathbf{C}_{ab}\mathbf{y}\|^2$
over time

This small spike
is not an anomaly
we wished to detect

Red dots: anomalies

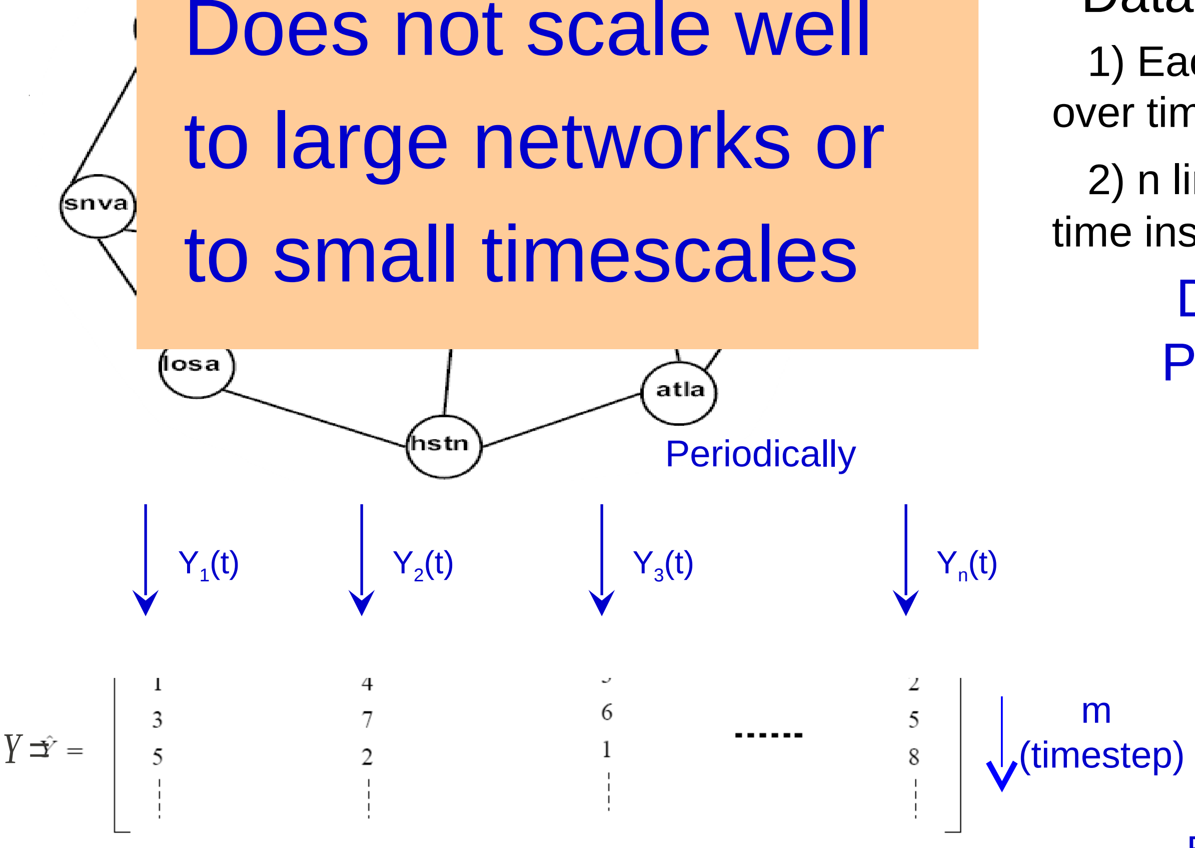
Blue curve: traffic data

Detection Threshold

$$\|C_{\mathrm{ab}}\, y\|^2 > Q_\alpha$$

- Q_α is a threshold on the Squared Projection Error (SPE). It guarantees a false alarm rate of less than α.
- Jackson & Mudholkar: computed threshold based on the abnormal eigenvalues of the covariance matrix.
  - No matter where the distinction is made (how many components are considered normal).
  - No matter what the mean amount of traffic is.
  - For multivariate Gaussian distribution only.
- Jensen & Solomon: In practice, holds for different distributions.
- Lakhina et al. Believe traffic is multivariate Gaussian.
  - but have not verified this.

The Centralized Algorithm



Does not scale well to large networks or to small timescales

n (nodeID)

Operation center

- Data matrix *Dat*

  1) Each link produces a column of m data over time.

  2) n links produce a row data y at each time instance.

Detection by Squared Prediction Error (SPE):

$$\|C_{ab}\, y\|^2 > Q_\alpha$$

Projection C_{ab}

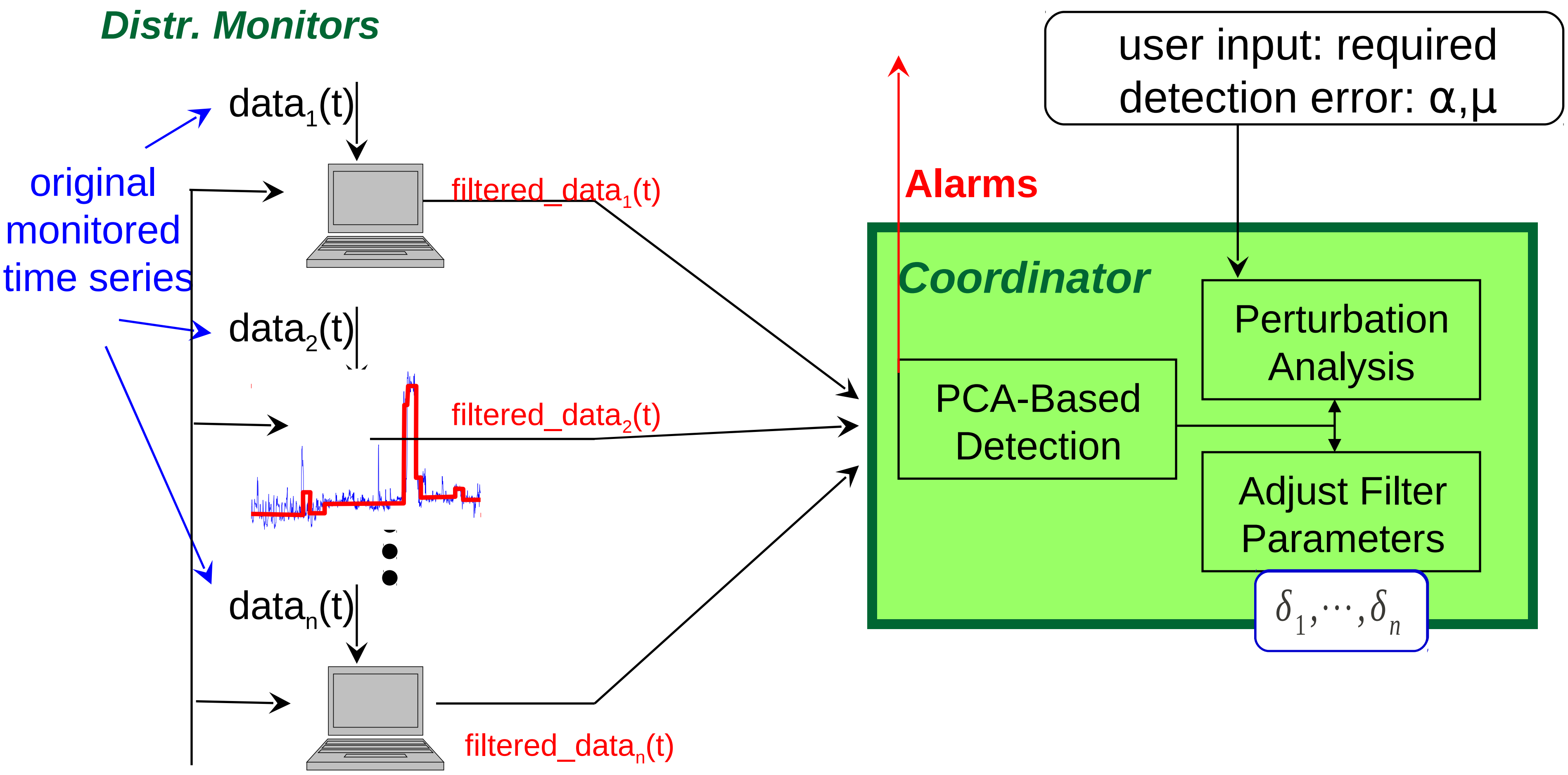
Threshold Q_α

Eigen vectors

Eigen values

PCA on Y

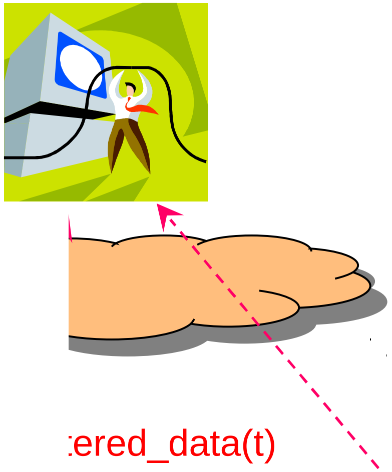
Huang et al.: In-Network Detection Framework



The Communication and Error Tradeoff

Approximate Info. ← PCA on $\hat{Y}$ ← $\hat{Y}$

$$\|\hat{C}_{ab}\,\hat{y}\|^2 > \hat{Q}_\alpha$$

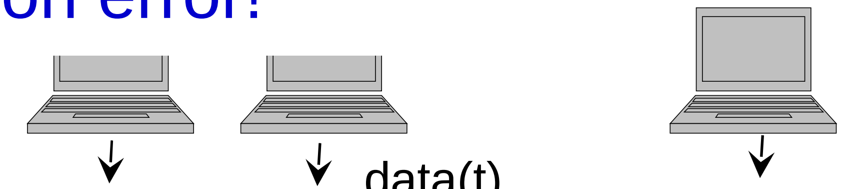


The bigger the filtering parameter δ_i,

the less the communication overhead,

but the more the detection error!

$$\|C_{ab}\,y\|^2 > Q_\alpha$$



PCA on *Y* ← **Y=**

$$\begin{pmatrix} 12 & 9 & \bullet\ \ \bullet\ \ \bullet & 45 \\ 7 & \bullet & & 63 \\ & \bullet & & \\ 24 & 31 & & 72 \end{pmatrix}$$

The coordinator computes a set of good δ_1, …, δ_n to manage this difference.

The Protocol At Monitors

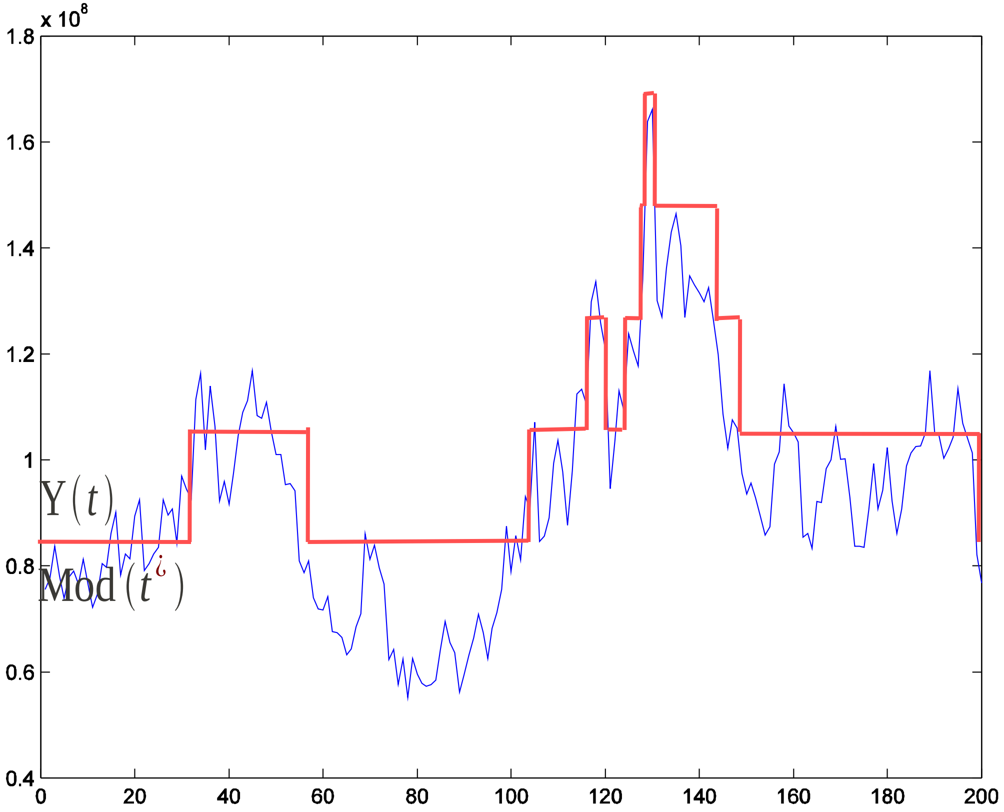
- Monitor i updates information if

  $$|Y_i(t) - \mathrm{Mod}_i(t^{\text{¿}})| > \delta_i$$

  $\delta_1, \cdots, \delta_n$ are the *filtering parameters*

- $\mathrm{Mod}_i(t^{\text{¿}})$ can be based on any prediction **mod**el built on historical data.
  - The prediction model is known to both monitor and coordinator.
  - For example, the average of last 5 communicated signal values.

The Protocol At Monitors



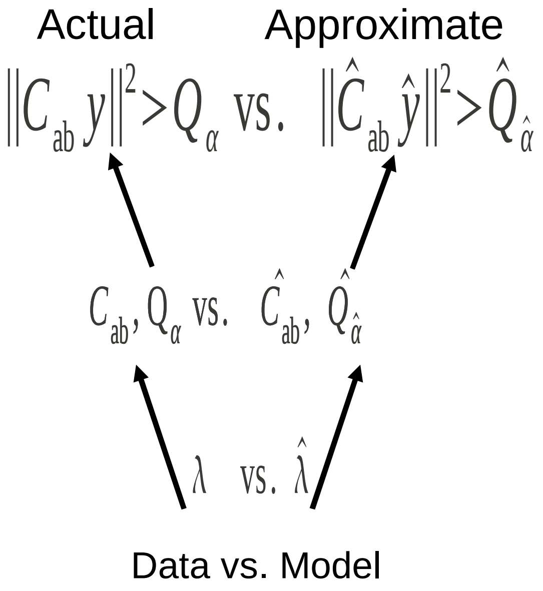
-
    - Simple but enough to achieve 10x data reduction

The Protocol at the Coordinator

- Create new time data from communication and predictions
- Update (cyclic) matrix: add new data, lose oldest
- Re-compute PCA (residual projection matrix, threshold)
- Detect anomalies, fire warnings
- Update slacks when needed (no details...)

Parameter Design and Error Control

- Users specify an upper bound on false alarm rate, then we determine the filtering parameters δ's



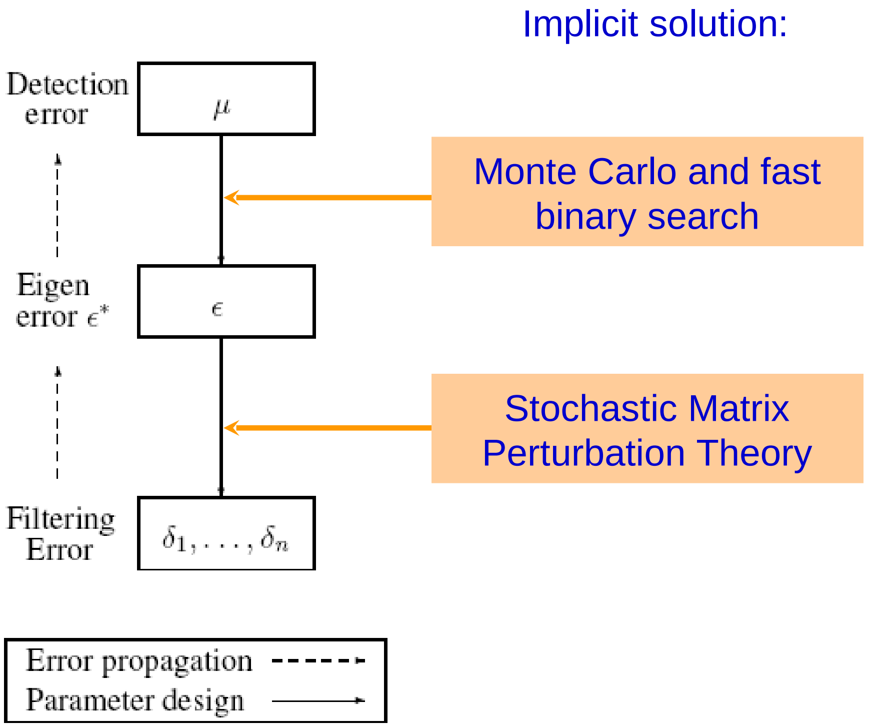
Actual Approximate

$$\|C_{ab}y\|^2 > Q_\alpha \text{ vs. } \|\hat{C}_{ab}\hat{y}\|^2 > \hat{Q}_{\hat{\alpha}}$$

$$C_{ab}, Q_\alpha \text{ vs. } \hat{C}_{ab}, \hat{Q}_{\hat{\alpha}}$$

$$\lambda \text{ vs. } \hat{\lambda}$$

Data vs. Model



Implicit solution:

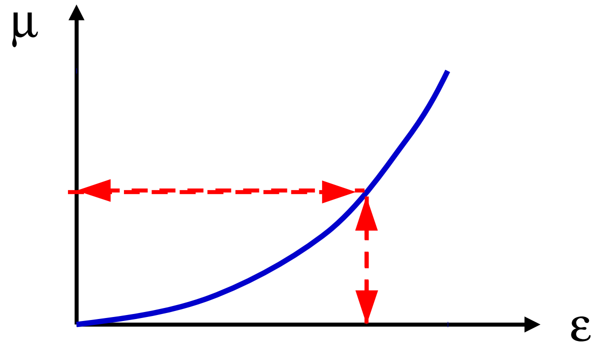
Monte Carlo and fast binary search

Stochastic Matrix Perturbation Theory

Eigen error: L_2 norm of the difference between the approximate eigenvalues and the actual ones

Parameter Design and Error Control (II)

- Detection Error μ → Eigen-Error ε
  - Monte Carlo simulation to find the mapping from ε to μ



  - For the given μ, a fast binary search to find an ε

From Eigen-Error to detection Deviation

$$\Pr\left[\|\mathbf{C}_a\mathbf{y}\|^2 > Q_\alpha\right] = \Pr\left[X > c_\alpha\right] = \alpha,$$

Normalized form of

$\|C_{\mathrm{ab}}\, y\|^2$

(Jensen & Solomon)

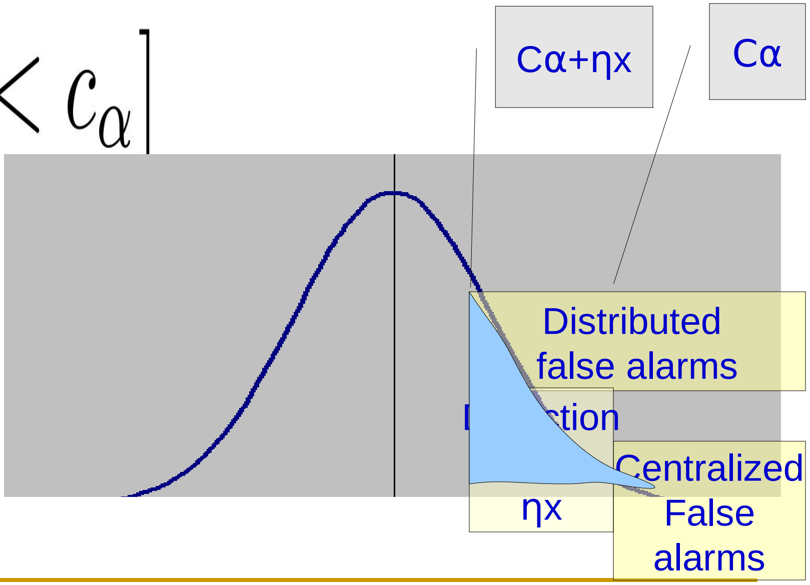
(1-α)-th percentile

$$\mu = \Pr\left[c_\alpha - \eta_X < N(0,1) < c_\alpha\right]$$

Upper bound on

$|\hat{X} - X|$

Estimated using max of Monte Carlo results



Parameter Design and Error Control (III)

Eigen-Error ε → Filtering parameters δs

- Error Matrix: $W = Y - \hat{Y}$
- Elements of column vector W_i bound by δ_i
- Assumptions:
  - W_i are independent, radially symmetric random vectors
  - For each i, all elements of a column vector are i.i.d random variables with mean 0 and variance σ^2
- The variance σ^2 is a function of the slacks δ_i

Parameter Design and Error Control (III)

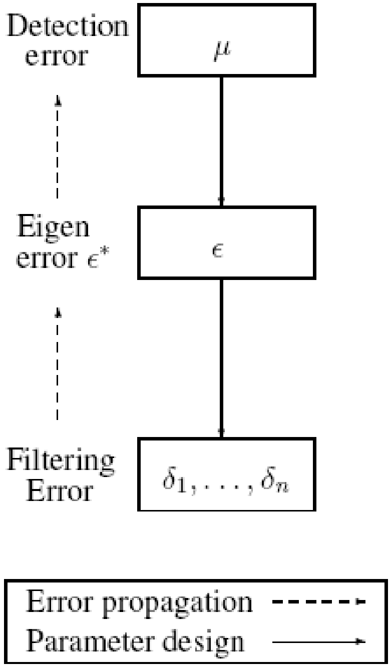
Tolerable Eigen-Error

Theorem: Setting δ_i to satisfy:

Average of Perturbed eigenvalues

$$2\sqrt{\frac{\bar{\lambda}}{m}\cdot\sum_{i=1}^{n}\sigma_i^2}+\sqrt{\left(\frac{1}{m}+\frac{1}{n}\right)\sum_{i=1}^{n}\sigma_i^4}=\epsilon$$

Guarantees $\epsilon^{¿} < \epsilon$ with high probability.



Detection error

μ

Eigen error ϵ^*

ϵ

Filtering Error

$\delta_1, \ldots, \delta_n$

Error propagation

Parameter design

Absent:

A connection between local variances and local slacks

Slack Allocation Methods

1. Homogeneous slack allocation: uniform distribution of errors in range $[-\delta_i, \delta_i]$
   - $\sigma_i = \frac{\delta_i^2}{3}$, results in closed expression for δ
2. Homogeneous slack allocation: local variance estimation
   - $\sigma_i = \sigma_i(\delta)$, monitors approximate locally by fitting an (e.g., quadratic) function according to a recent window of data. Approximation sent to coordinator.
3. Heterogeneous slack allocation.
   - Assume uniform distribution of errors in range
   - Minimize communication; Solve using Lagrange multipliers.

Evaluation:Accuracy and Cost

- Given user-specified false alarm rate, evaluate the actual detection accuracy and communication overhead
- Experiment setup
  - Abilene backbone network data of one week:
    - 121 flows, 41 links, 1008 10 minute periods
  - Traffic matrices of size 1008 X 41
  - Set uniform slack $\delta_i = \delta$ for all monitors
  - Injected: 60 small “bursts” +60 large “anomalies”
  - Threshold corresponding 0.5% false alarm rate
  - How many experiments (repetitions)?

Evaluation Metrics

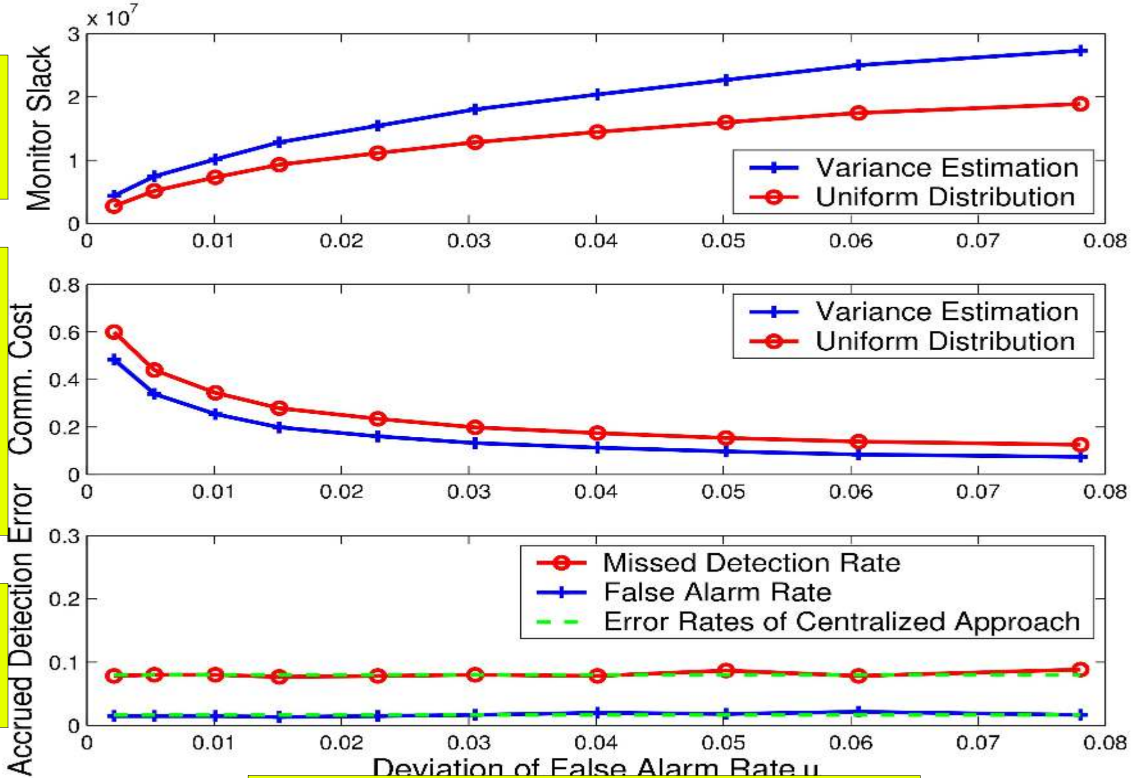
- False alarm rate = false alarms/ bursts
- Missed detection rate = missed detections/anomalies
- Cost = num/(n*m) = messages per monitor per sampled time points
  - num = all exchanged messages
  - n = number of monitors
  - M = number of time series points

Evaluation Results

Monitor Slack

Communication Cost (fraction of centralized completely updated)

Actual Detection Errors



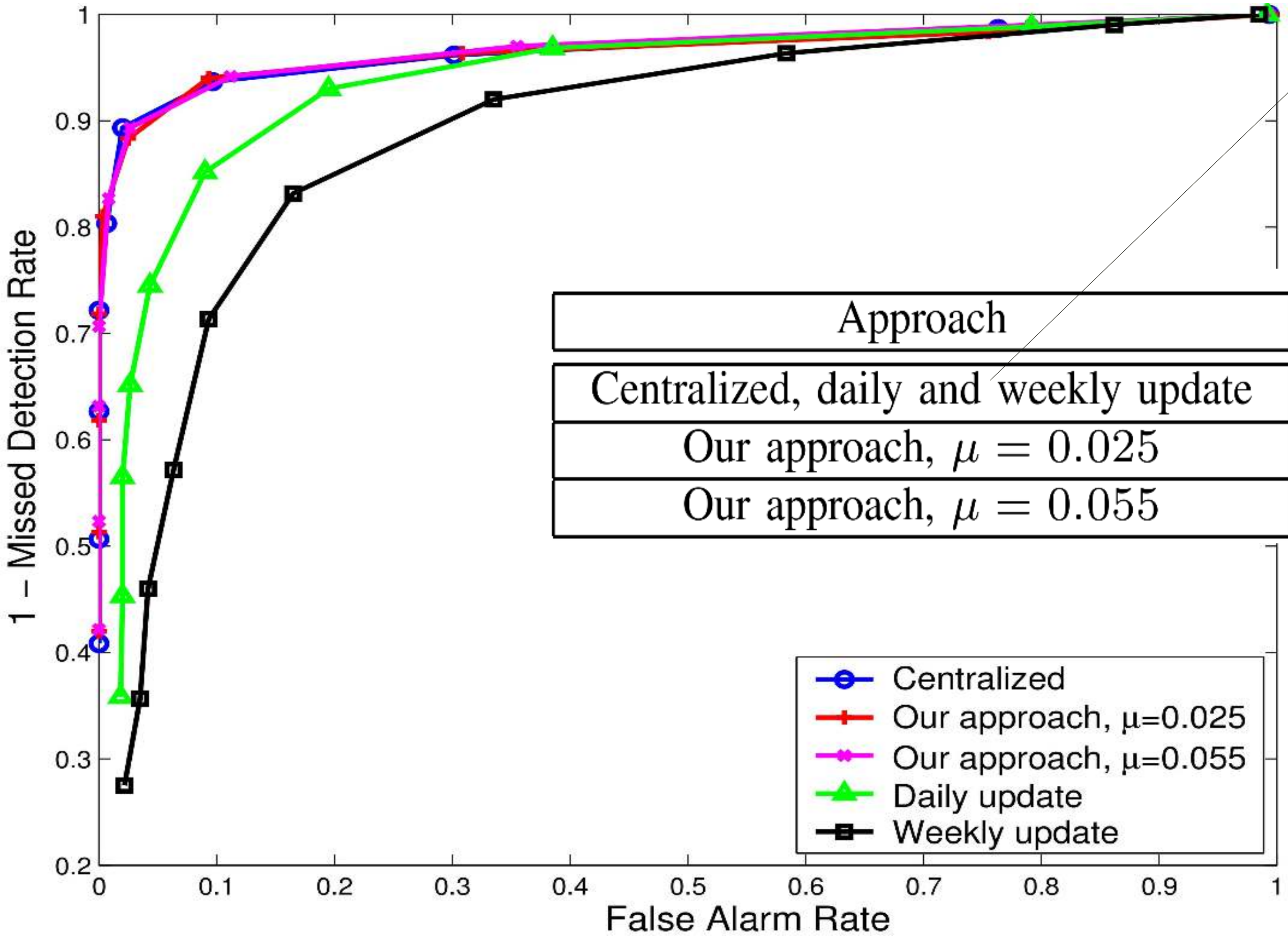
Deviation of false alarms rate μ

Fig. 6. Monitor slacks, communication cost and accrued detection. The dashed line is the detection error of centralized approach with complete data.

Observations

- Homogenous variance estimation outperforms Homogenous Uniform, but not by much (5%-10%).
- Homogenous Uniform method is simple.
- Homogenous Uniform might be “good enough”.
- 80%-90% of the transferred data can be saved without hurting performance.

ROC – Receiver Operating Characteristic Curve



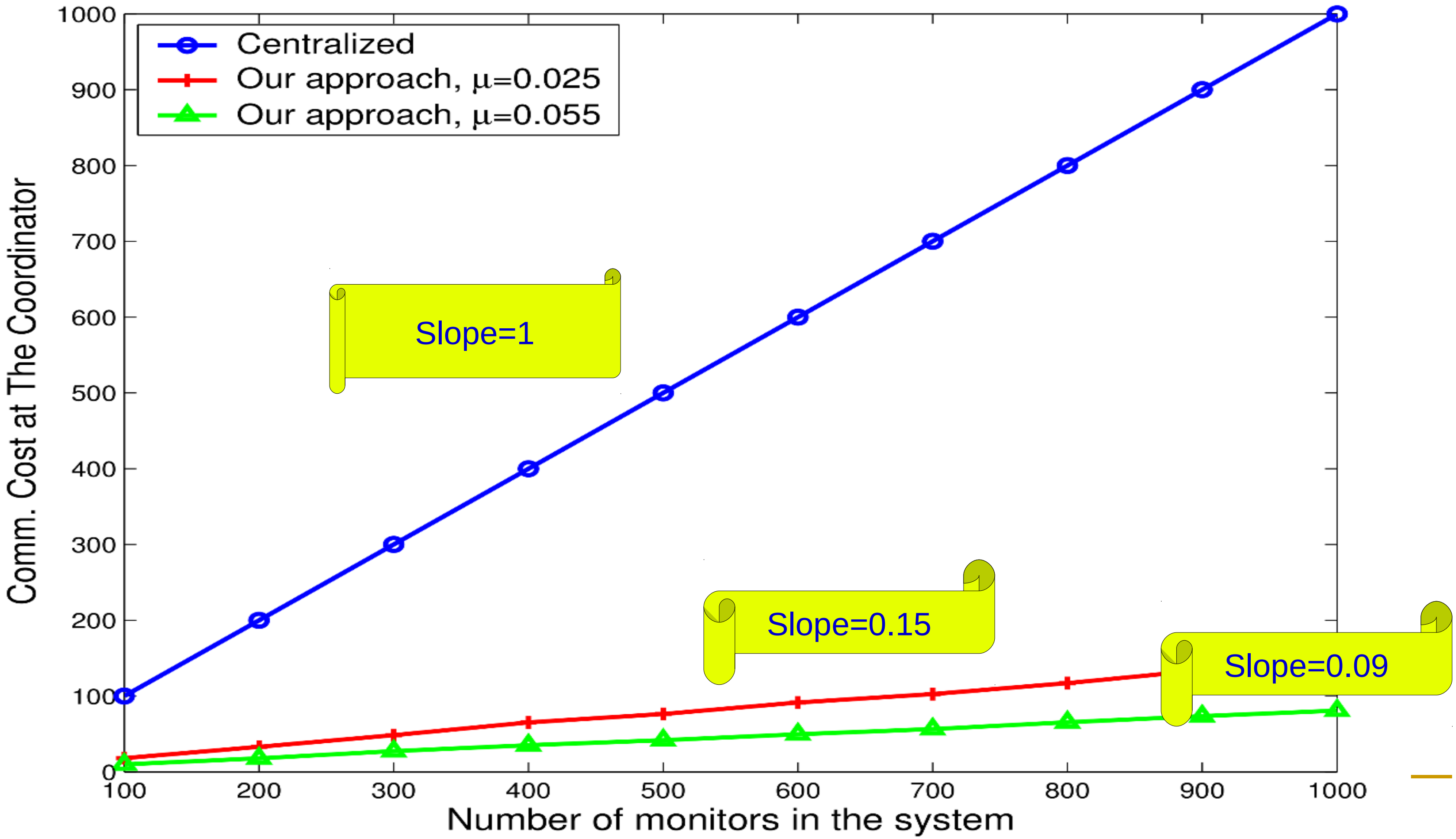
Approach	Communication Cost
Centralized, daily and weekly update	1.000
Our approach, $\mu = 0.025$	0.159
Our approach, $\mu = 0.055$	0.097

Update rate
Of PCA
(data is always full)

Evaluation of Scalability

- BRITE topology generator
- 100-1000 links
- Up to 500*500 Origin-Destination flows
- 4 weeks of realistic data, based of statistical characteristics of Abilene
- In each experiment on n nodes: 5 repetitions, on n randomly picked nodes.

Graceful Scalability by number of monitors: coordinator communication



Slope=1

Slope=0.15

Slope=0.09

Summary

- A communication-efficient framework that
  - detects anomalies at desired accuracy level
  - with minimal communication cost
- A distributed protocol for data processing
  - Local monitors decide when to update data to coordinator
  - Coordinator makes global decision and feedback to monitors
- An algorithmic framework to guide the tradeoff between communication overhead and detection accuracy

Discussion

References

[Huang’07] *Communication-Efficient Online Detection of Network-Wide Anomalies*. L. Huang, X. Nguyen, M. Garofalakis, J. Hellerstein, M. Jordan, A. Joseph and N. Taft. To appear in INFOCOM'07.

[Lakhina’04] *Diagnosing Network-Wide Traffic Anomalies*. A. Lakhina, M. Crovella and C. Diot. In SIGCOMM '04.

[Jensen & Solomon] *A Gaussian approximation for the distribution of definite quadratic forms*. D.R. Jensen and H. Solomon, In J. Amer. Stat. Assoc., 67:898-902 (1972).

[Jackson & Mudholkar] *Control Procedures for Residuals Associated with Principal Component Analysis*. J. E. Jackson and G. S. Mudholkar, Technometrics, pages 341–349, 1979.

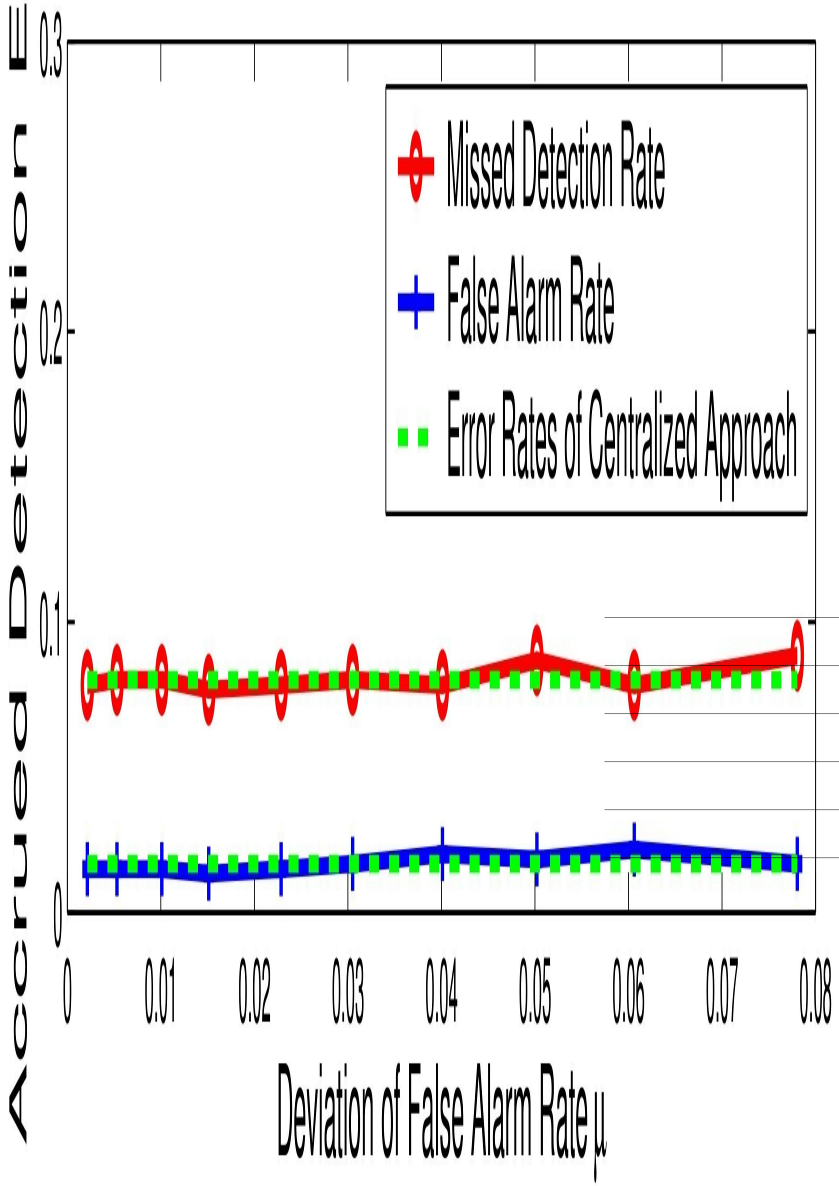
Weaknesses (My Opinion)

- Symmetry + Independence
- Experiments

symmetry + independence

- Is the symmetry + independence assumption valid?
- Correlation may result from simultaneous errors upon surprising data changes, or from (cyclic?) bursts induced by the updating algorithm.

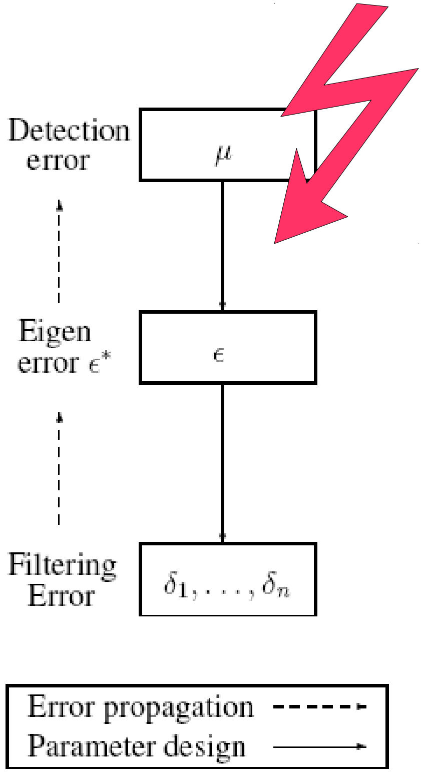
Experiments



Single Experiment
Quantization error:
1/60=0.016
(means one alarm)

Experiments: Lack of Trend

- Experiments do not show a statistically significant trend (dependency) of “tolerated deviation from false alarm rate”and actual false alarm rate.
- Estimations are too loose, or
- Experiments are too synthetic
- Between the lines: user is expected to trust experiment results.



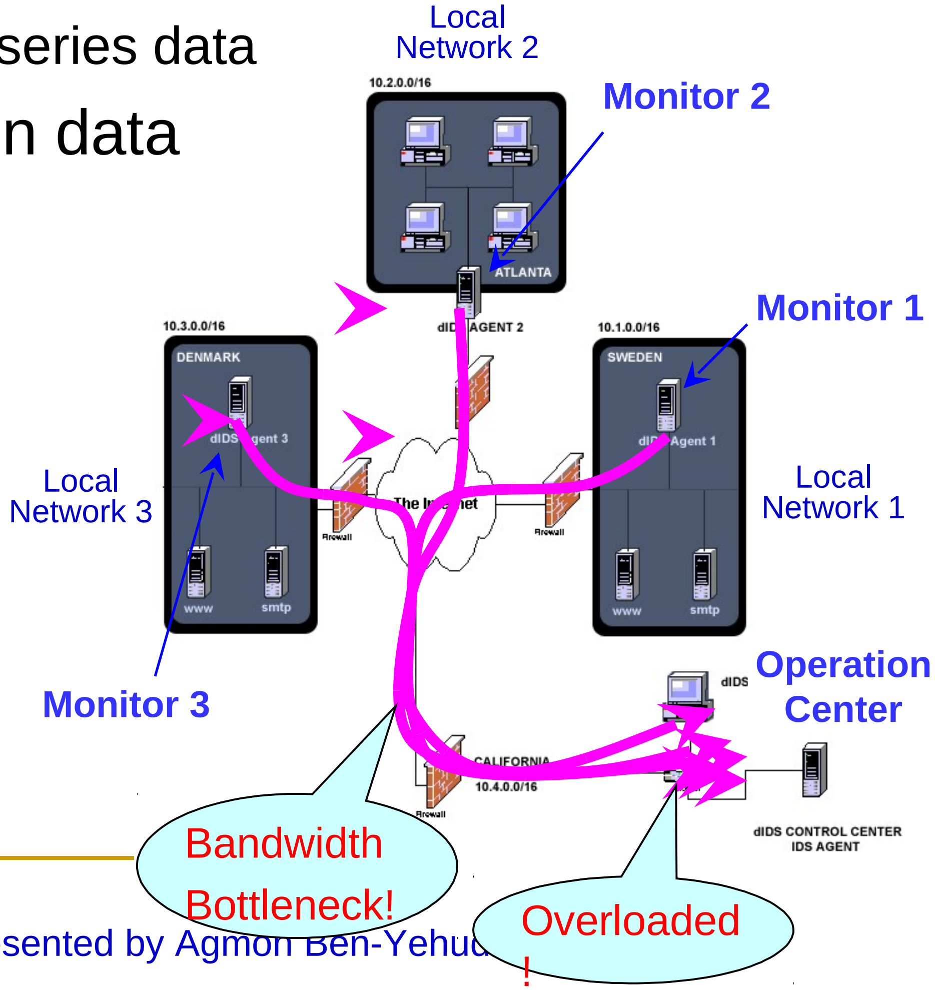
My Summary

- The decentralized algorithm works well in practice according to insufficient experiments.
- The tuning knob was not proved to work in experiments (to be connected to practical accuracy guarantees).
- Noisier experiments are needed.

Backup Slides

Traditional Distributed Monitoring

- Large-scale network monitoring and detection systems
  - Distributed and collaborative monitoring boxes
  - Continuously generating time series data
- Existing research focuses on data streaming
  - *Centrally* collect, store and aggregate network state
  - Well suited to answering approximate queries and continuously recording system state
  - Incur high overhead!



Local Network 2

Monitor 2

Monitor 1

Local Network 3

Local Network 1

Monitor 3

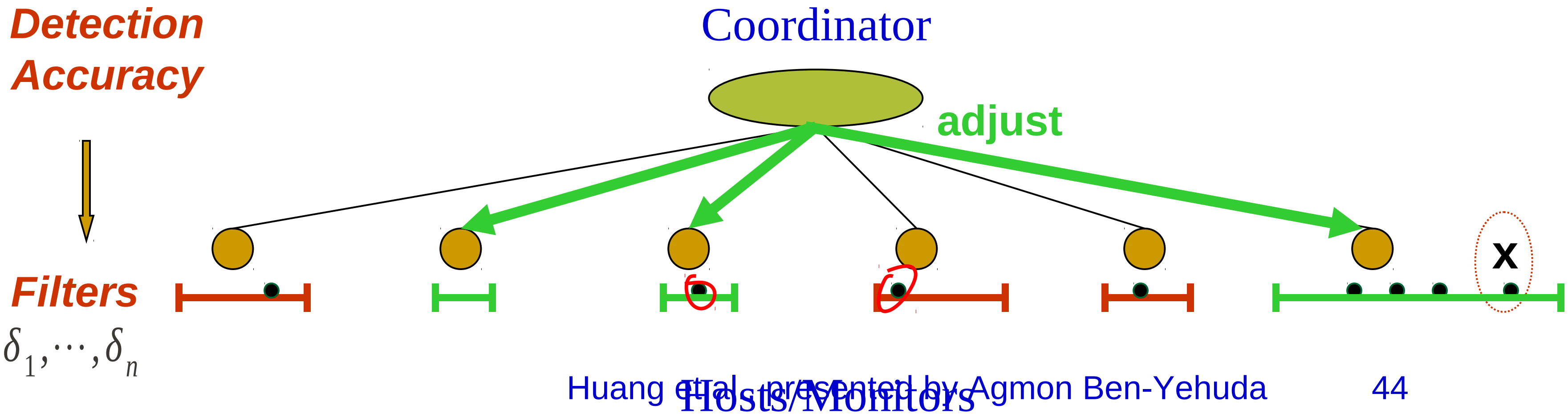
Operation Center

Bandwidth Bottleneck!

Overloaded!

Our Distributed Processing Approach

- A coordinator
  - Is aggregation, correlation and detection center
- A set of distributed monitors
  - Each produces a time series signals
  - Processes data locally, only sends needed info. to coordinator
  - No communication among monitors
  - *Coordinator tells monitors the level of accuracy for signal updates*



Performance

μ	Missed Detections		False Alarms		Data Reduction	
	Week 1	Week 2	Week 1	Week 2	Week 1	Week 2
0.01	0	0	0	0	75%	70%
0.03	0	1	1	0	82%	76%
0.06	0	1	0	0	90%	79%

error tolerance = upper bound on error

Data Used: Abilene traffic matrix, 2 weeks, 41 links.