Efficient Constraint Monitoring Using Adaptive Thresholds

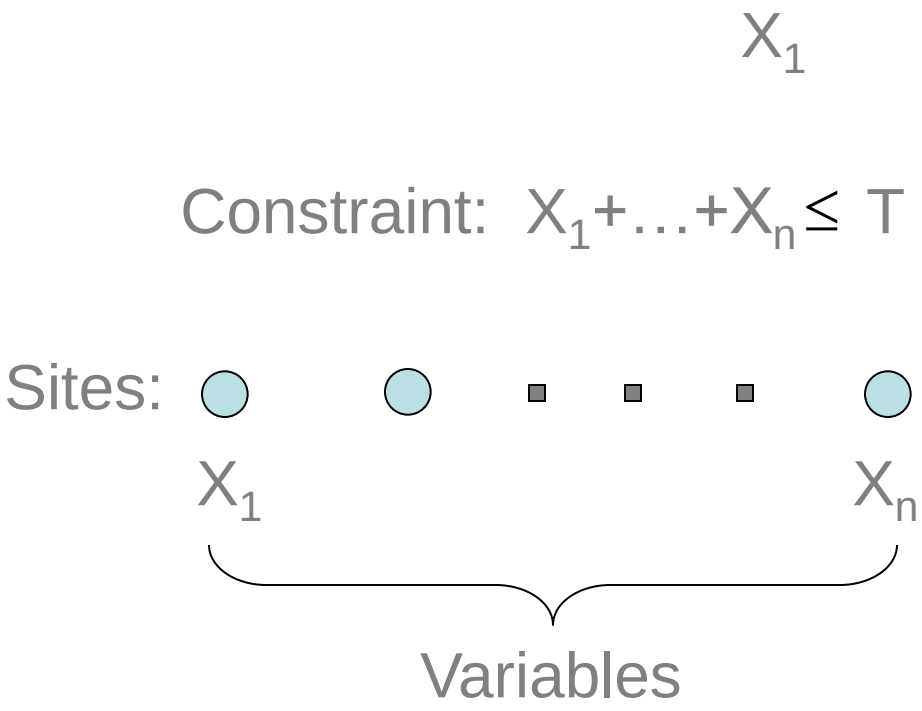
Srinivas Kashyap, IBM T. J. Watson Research Center
Jeyashankar Ramamirtham, Netcore Solutions
Rajeev Rastogi, Yahoo! Labs Bangalore
Pushpraj Shukla, Univ of Texas at Austin

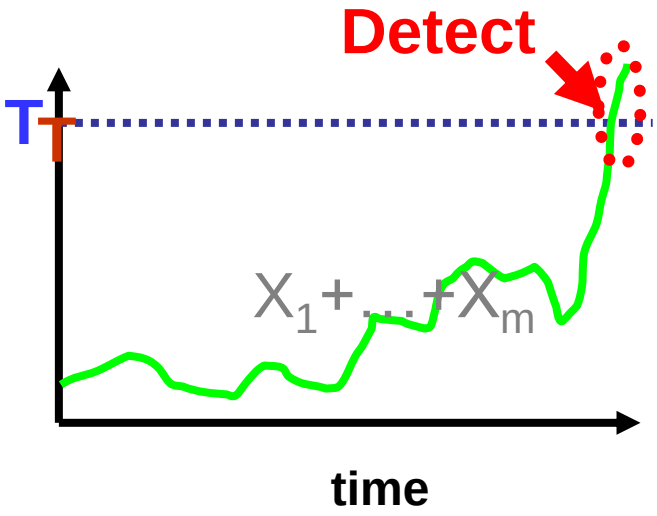
Talk Outline

- Motivation
- Constraint monitoring architecture
- Existing approaches
- Problem formulation
- Markov-based algorithm
- Reactive algorithm
- Experimental results
- Conclusions

Constraint Monitoring Problem

- Detect violation of distributed SUM constraints
  - Distributed Triggers [Jain et al. 04]





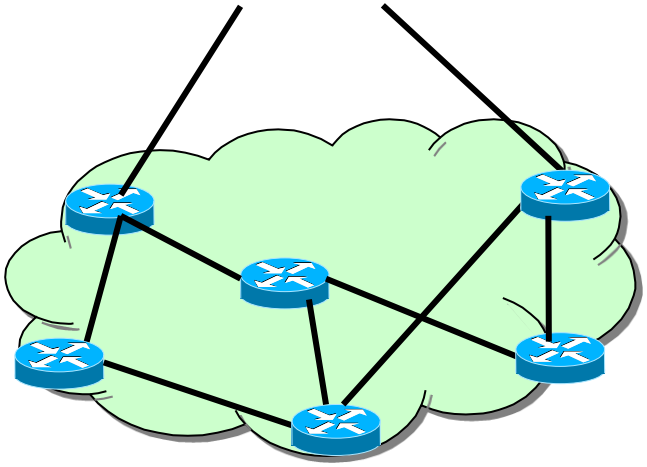
Applications: Network Monitoring

Alert when sum of link delays along a Voice over IP path exceeds 200msec

Identify all destinations that receive more than 2GB of traffic from the monitored network in a day, and report their transfer totals

Network Operations Center (NOC)

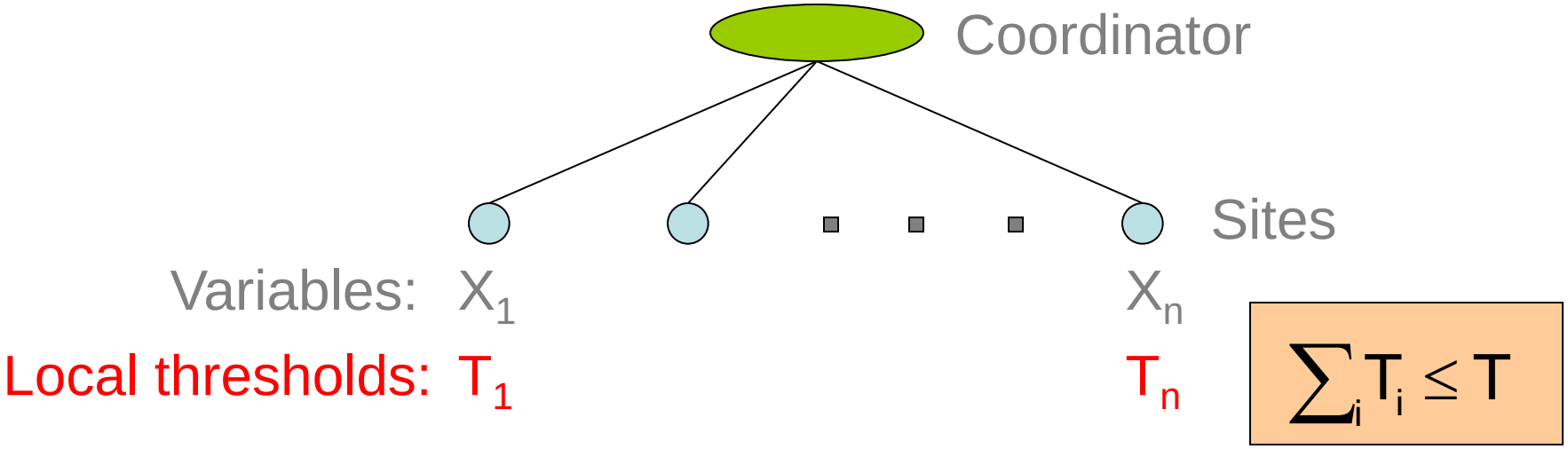
Monitor the volume of remote login (telnet, ssh, ftp etc.) requests received by hosts within the organization that originate from the external hosts.



Source	Destination	Duration	Bytes	Protocol
10.1.0.2	16.2.3.7	12	20K	http
18.6.7.1	12.4.0.3	16	24K	http
13.9.4.3	11.6.8.2	15	20K	http
15.2.2.9	17.1.2.1	19	40K	http
12.4.3.8	14.8.7.4	26	58K	http
10.5.1.3	13.0.0.1	27	100K	ftp
11.1.0.6	10.3.4.5	32	300K	ftp
19.7.1.2	16.5.5.8	18	80K	ftp

Example NetFlow IP Session Data

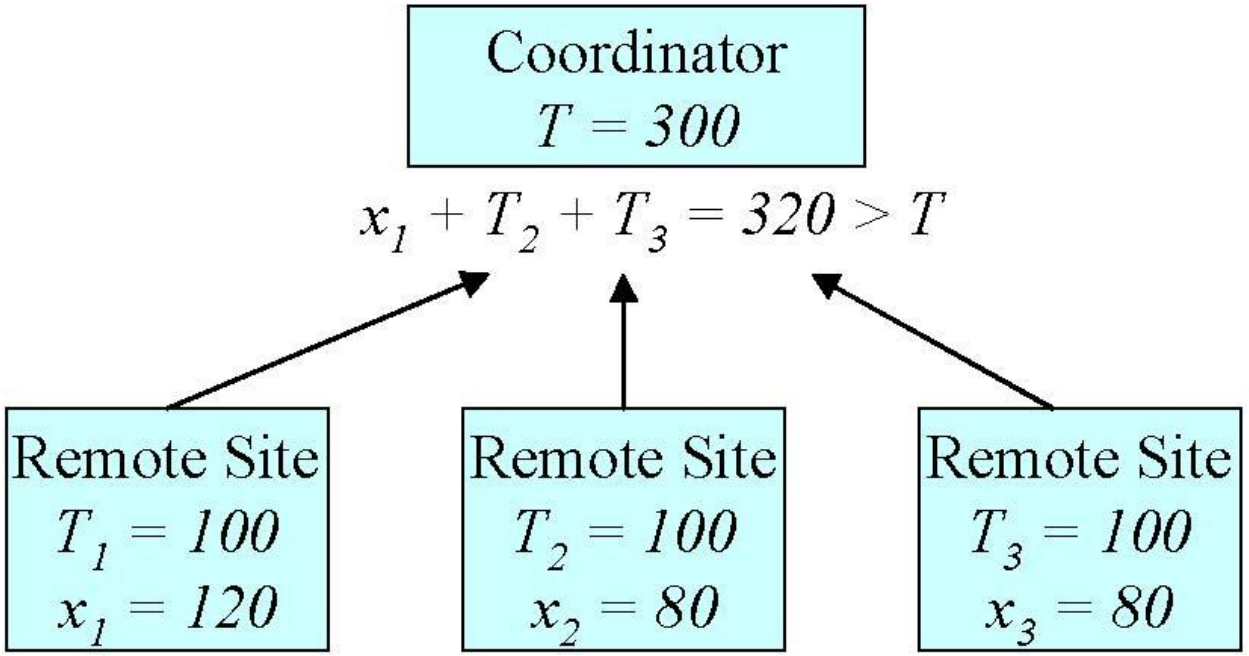
Constraint Monitoring Architecture



- At site j:
  - if $X_j > T_j$: send alarm to coordinator (with X_j value)
- At coordinator:
  - if $X_j + \sum_{i \neq j} T_i > T$: poll X_i values to check if constraint is violated (global poll)

Existing Approaches: Zero Slack

- Local thresholds satisfy: $\sum_i T_i = T$
- Drawback: every alarm → global poll $(X_j + \sum_{i \neq j} T_i > T)$



Existing Approaches: Zero Slack

- Static local thresholds [Jain et al. 04]

$$T_i = \frac{T}{n}$$

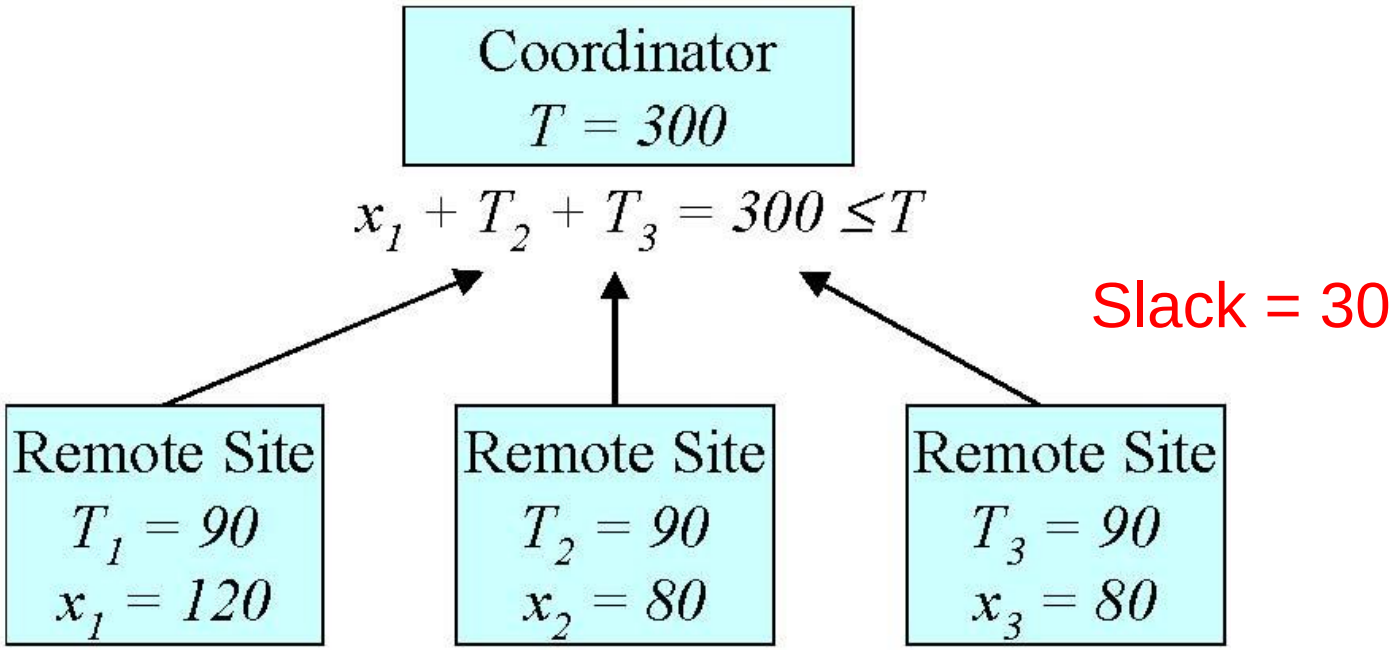
- Dynamic local thresholds [Sharfman et al. 06]
  - Thresholds reset each time alarm generated

$$\text{Slack } S = T - \sum_i X_i$$

$$T_i = X_i + \frac{S}{n}$$

Non-zero Slack [Dilman and Raz 01]

- Threshold setting *with* slack: $T - \sum_i T_i > 0$



- Slack leads to fewer global polls

Non-zero Slack: Key Questions

- How to set local threshold values so that constraint violations can be detected with minimal communication overhead?
  - $\sum_i T_i$ too low → too many local alarms
  - $\sum_i T_i$ too high→ too many global polls

- How to adapt thresholds for changing data distributions?

Communication Cost Model

- Define $Y_i = X_i$ if $X_i > T_i$
  $= T_i$ otherwise
- Coordinator's SUM estimate $Y = \sum_i Y_i$
- Probability of local alarm $P_l(i) = Pr[X_i > T_i]$
- Probability of global poll $P_g = Pr[Y > T]$
- Local alarm is O(1) messages, global poll is O(n) messages
- Expected cost = $n * P_g + \sum_i P_l(i)$

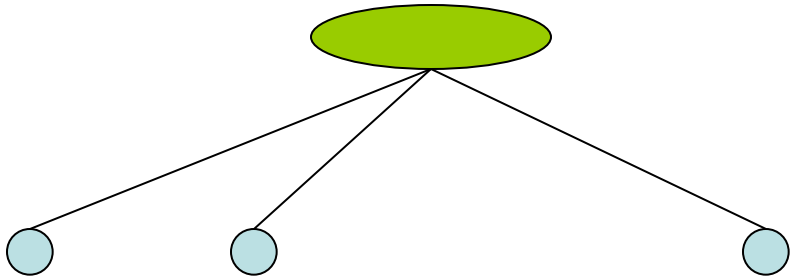
Problem Formulation

- Given threshold T and variables X_i at sites, select local thresholds T_i such that the cost $n * P_g + \sum_i P_l(i)$ is minimized.

Key Challenge

Computing $P_g = Pr[\sum_i Y_i > T]$

- Depends on Ti values at all sites
- Optimal value computation requires enumerating all Ti value combinations



- Each site maintains histogram: $H_i(v) = Pr[X_i = v]$
- Then $R(i) = 1 - \sum_{v=0}^{T_i} H_i(v)$
  - Can be computed locally for specific T_i value

Markov-based Algorithm

- Key idea: Use Markov's inequality to decompose Pg into components that can be computed locally

$$P_g \leq \frac{E[\sum_i Y_i]}{T} \leq \frac{\sum_i E[Y_i]}{T}$$

$$Cost \leq \sum_i \left(\frac{n * E[Y_i]}{T} + R(i) \right)$$

$$E[Y_i] = \sum_{v=0}^{T_i} T_i * H(i, v) + \sum_{v=T_i+1}^{T} v * H(i, v)$$

- Each site can independently determine T_i value that minimizes its contribution to the total cost

Drawback of the Markov Algorithm

- Markov's inequality over-estimates global poll probability P_g
  - computed thresholds T_i' *lower* than optimal

Reactive Algorithm

- Key idea: Use local alarms and global polls to adjust T_i values
- Let $\lambda_i = \frac{P(\alpha, i)}{P_g}$ at thresholds T_i' computed by Markov
- On local alarm: With probability $\frac{1}{\lambda_i}$, set $T_i = \alpha * T_i$
- On global poll: With probability λ_i, set $T_i = \frac{Ti}{\alpha}$

Analysis

- At stable state, $\lambda_i = \frac{\#\ \text{of local alarms}}{\#\ \text{of global polls}}$

- Since Markov inequality over-estimates P_g, at T_i'

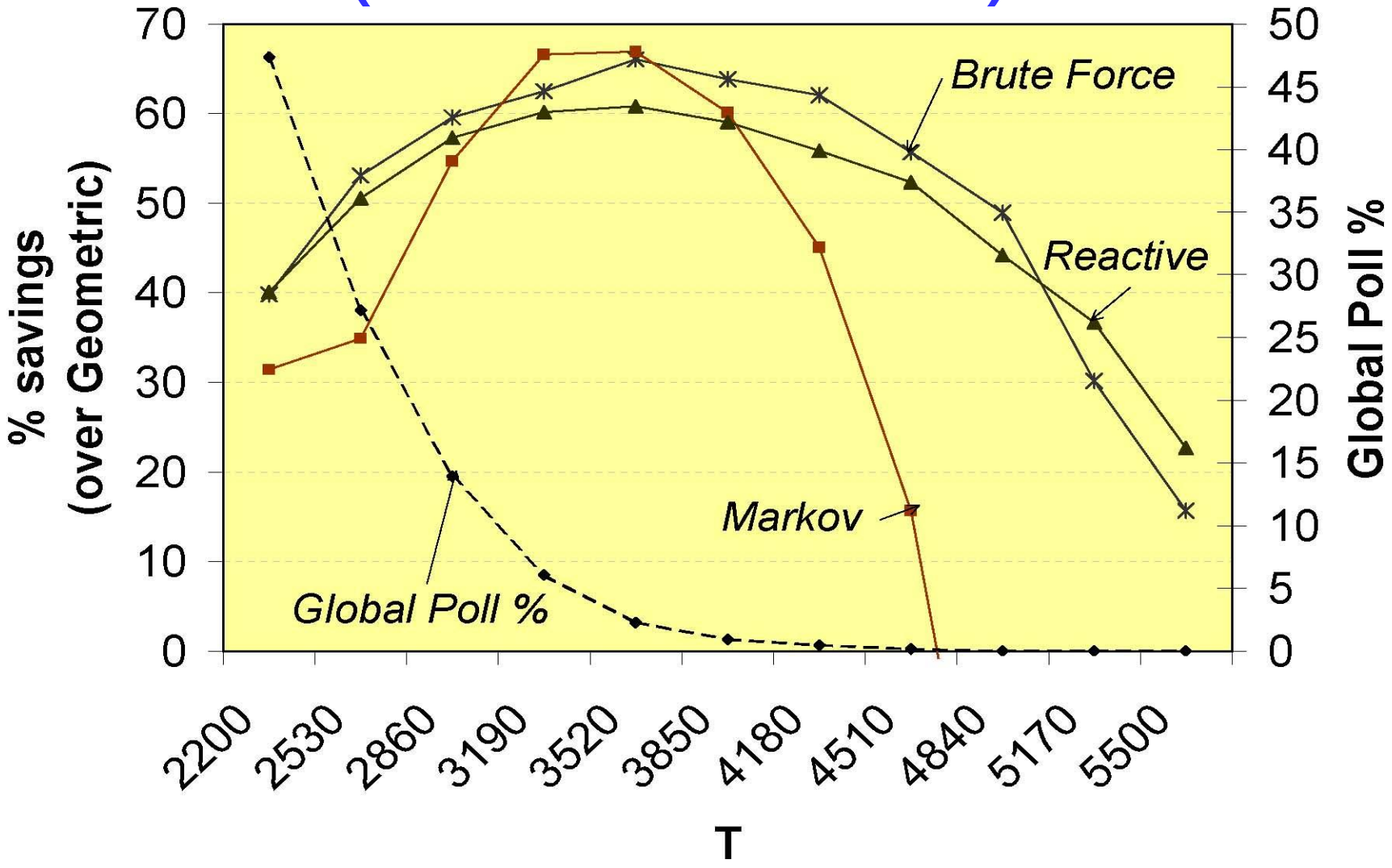
$$\lambda_i < \frac{\#\ \text{of local alarms}}{\#\ \text{of global polls}}$$

- So thresholds T_i will converge to values > T_i'

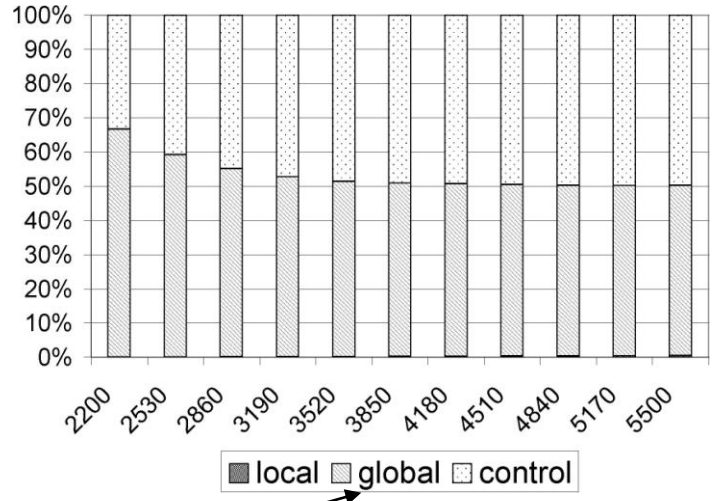
Experimental Results

- Real-life datasets
  - Netflow traces from Abilene network: 73 million packets across 11 routers
  - Link traces from NLANR: 21 million packets
- Distributed constraint: Total amount of traffic flowing into network across ingress links $\leq T$
- Schemes considered
  - Geometric [Sharfman et al. 06], Markov-based, Reactive

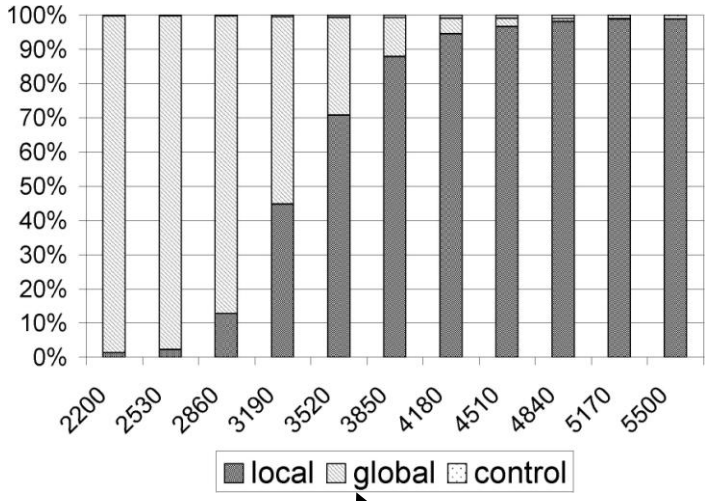
Communication Savings (Abilene Dataset)



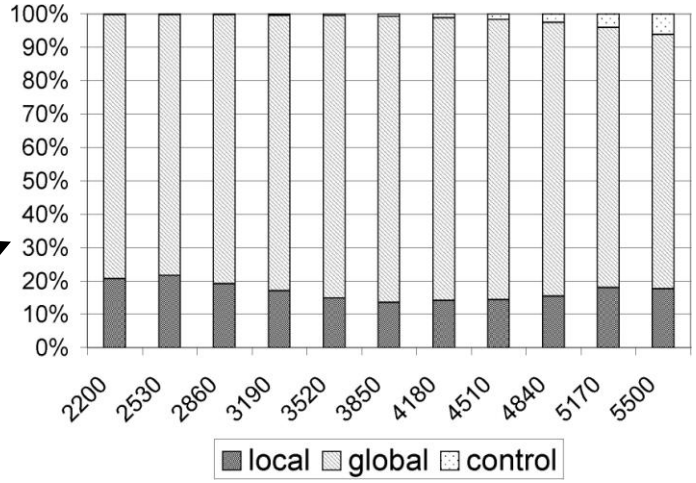
Breakup of Message Overhead (Abilene Dataset)



Geometric

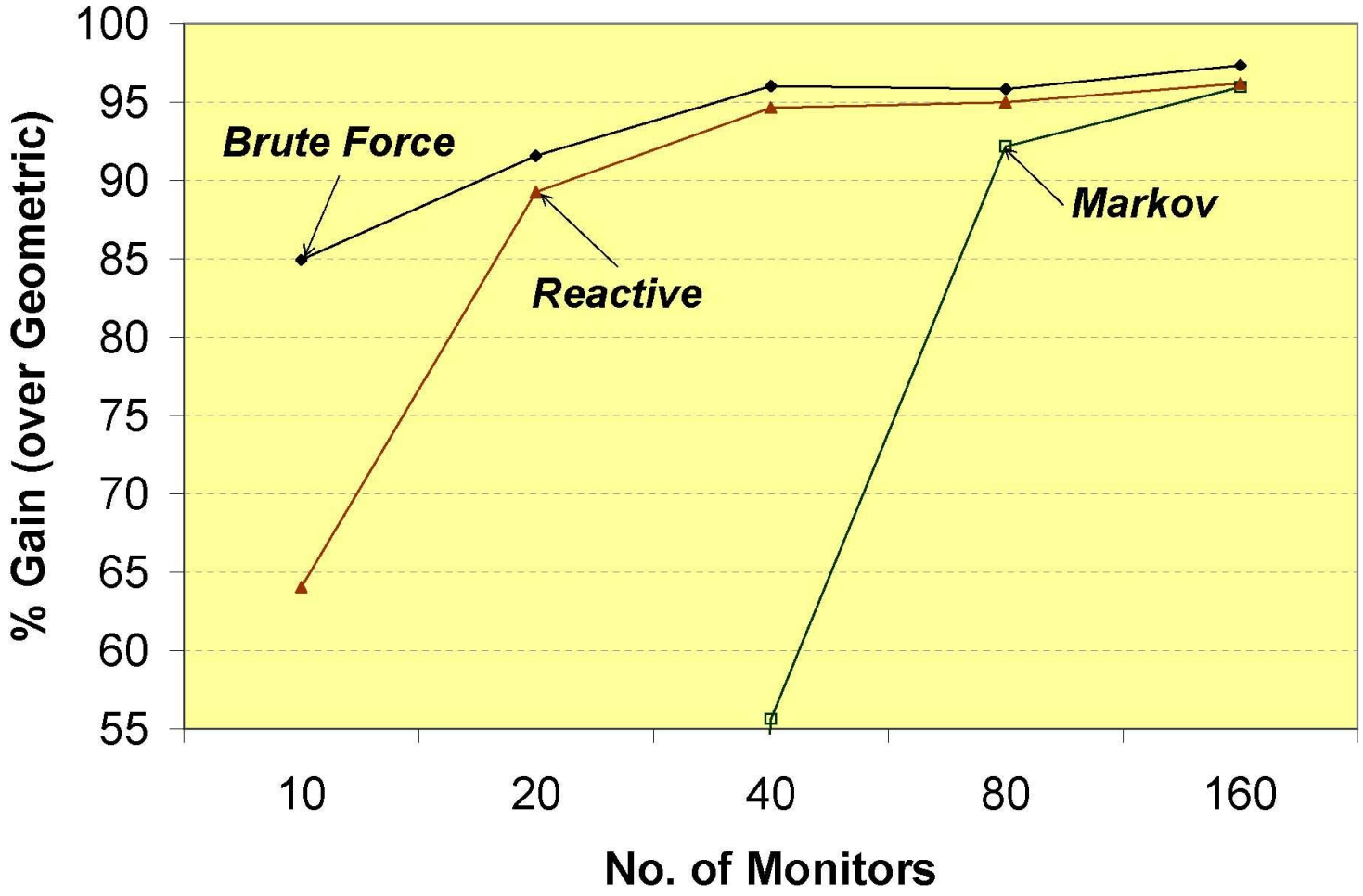


Markov



Reactive

Effect of scale (NLANR Dataset)



Summary

- Reactive algorithm for setting local thresholds in non-zero slack setting
  - Uses on Markov's inequality to simplify global poll probability estimation
  - Adjusts thresholds in response to local alarm and global poll events
  - Adapts to changing data distributions
- Reactive algorithm incurs 60% less communication overhead compared to the state-of-the-art zero slack scheme