# Decentralized Real-Time Monitoring of Network-Wide Aggregates

**Rolf Stadler**
**Mads Dam, Alberto Gonzalez, Fetahi Wuhib**

KTH Royal Institute of Technology
Stockholm, Sweden
*www.ee.kth.se/~stadler*

*Large-scale Distributed Systems and Middleware (LADIS '08)*
*IBM TJ Watson Research Lab, NY, Sept 15-17, 2008*

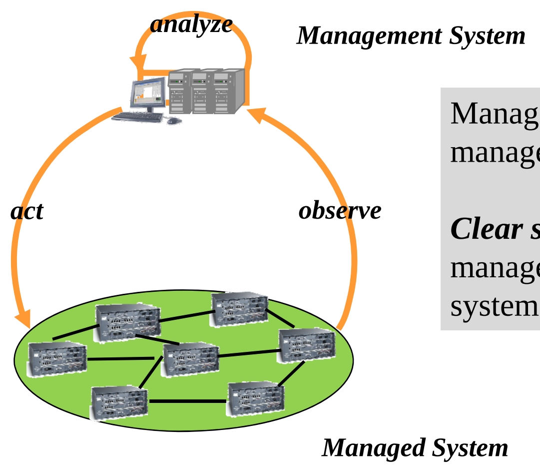
## Outline

A self-organizing Monitoring Layer

Continuous Monitoring of Aggregates with Accuracy Objectives (A-GAP)

Performance comparison gossip vs. tree-based monitoring (GAP vs. G-GAP)

# Today's Management Systems for Today's Network Technologies

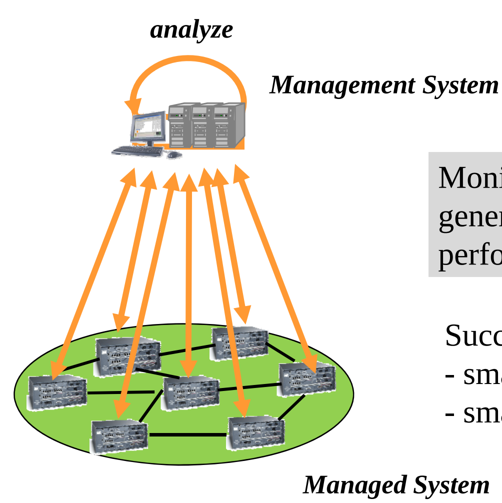



Management intelligence ***outside*** managed system.

***Clear separation*** between management system and managed system, ***by design.***



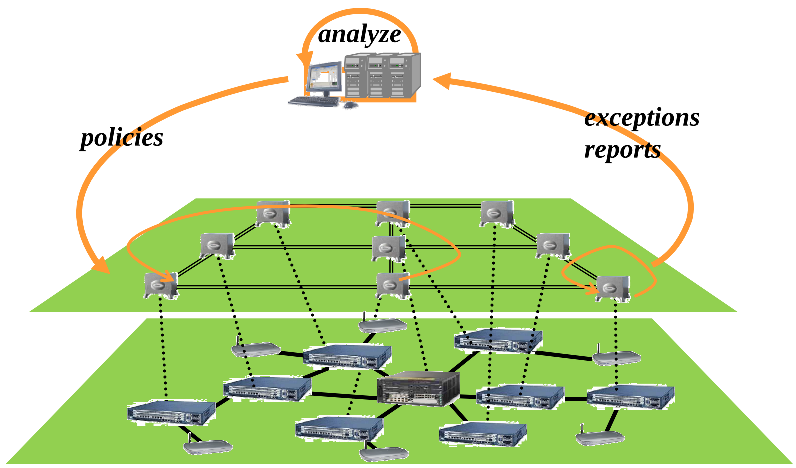
# Today's Management Systems for Today's Network Technologies (2)




Monitoring and configuration, generally FCAPS functions, performed on a ***per-device basis***.

Successful for
- small number of components
- small rate of change.

## A Management Layer inside the Network



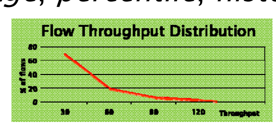


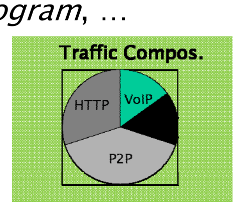

## A Monitoring System for Large-scale Dynamic Environments

1. Engineer a self-organizing monitoring layer inside the managed system.
2. Support monitoring of aggregates in real-time.
   across neighborhood, domain, network
   *sum*, *max*, *average*, *percentile*, *histogram*, …

**Top K flows**

| Source | Destination | Throughput |
|---|---|---|
| 10.10.3.17:898 | 10.10.9.3:240 | 120 |
| 10.10.1.52:578 | 10.10.7.9:150 | 117 |
| 10.10.7.15:201 | 10.10.6.98:200 | 80 |




3. Provide primitives for polling,
   continuous monitoring,
   detection of threshold crossings.
4. Support controlling the performance trade-offs.
   accuracy, overhead, execution time, robustness

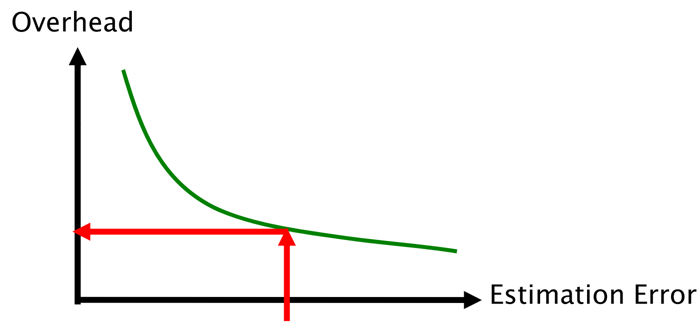
## Continuous Monitoring of Aggregates with Accuracy Objectives (A-GAP)



## The Problem

- Find an efficient solution for ***continuous monitoring*** of ***aggregates*** in large-scale dynamic network environments
  - Aggregation functions: SUM, MAX and AVERAGE, …
  - Sample aggregates: total number of VoIP flows, maximum link utilization, histogram of current load across routers in a network domain
- Key Application Areas: Network Supervision, Quality Assurance, Proactive Fault Management

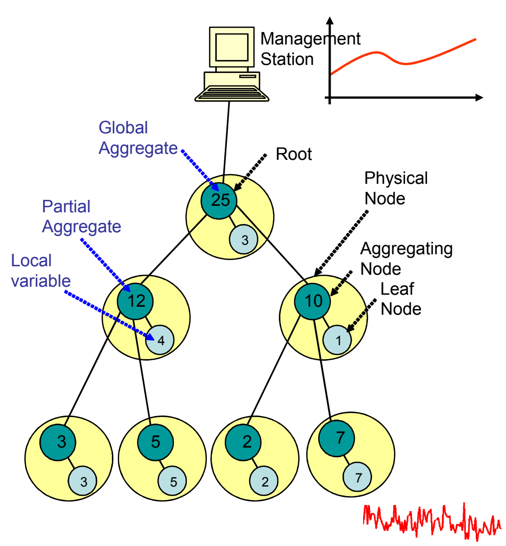
## Tradeoff between Estimation and Overhead




- Management solutions deployed today usually provide qualitative control of the accuracy
- Goal: Control trade-off through error objective



## Decentralized in-Network Aggregation



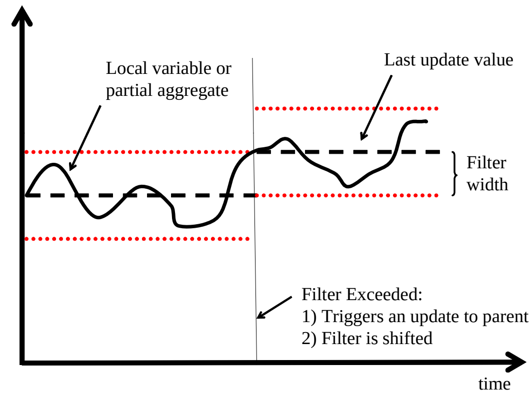

***Computing Aggregates***

- Self-stabilizing spanning tree
- Incremental, in-network aggregation
- Push-based

***Efficient Operation***

- Local filters conform to error objective
- Adapt dynamically to network statistics

# Local Adaptive Filters




Local filter on a node

- **Controls the management overhead** by filtering updates
- **Drops updates** with small change to partial aggregate
- **Periodically adapts** to the dynamics of network environment



# Problem Formalization

Find ***filter widths*** to monitor aggregate
for a given accuracy objective, with minimal overhead

Overhead:
maximum processing load $\omega^n$
over all management processes

Accuracy objective:

average error Minimize $\underset{n}{Max}\{\omega^n\}$ s.t. $E[|E^{root}|] \leq \varepsilon$

percentile error Minimize $\underset{n}{Max}\{\omega^n\}$ s.t. $p(|E^{root}| > \gamma) \leq \theta$

maximum error Minimize $\underset{n}{Max}\{\omega^n\}$ s.t. $|E^{root}| \leq \kappa$

## A-GAP: A Distributed Heuristic

- The global problem is mapped onto a ***local problem for each node***

$$\text{Minimize } \underset{\pi}{Max}\{\omega^{\pi}\} \quad \text{s.t.} \quad E\left(\left|E^{n}_{out}\right|\right) \le \varepsilon^{n}$$

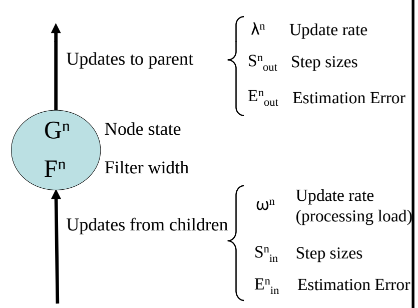
- Attempts to minimize the maximum processing load over all nodes by minimizing the load within each node's neighborhood
- Filter computation: ***decentralized*** and ***asynchronous***
- Each node independently runs a control cycle:

```
every τ seconds {
    request model variables from children
    compute new filters and accuracy objectives for children
    compute model variables for local node }
```



## An Stochastic Model for the Monitoring Process

- Model based on discrete-time Markov chains
- It relates for each node *n*
  - –the error of its partial aggregate
  - –evolution of the partial aggregate
  - –the rate of updates *n* sends
  - –the width of the local filter
- It permits to compute for each node
  - –the distribution of estimation error
  - –the protocol overhead



## Stochastic Model (leaf node)

Estimating step size (MLE) $X^n$

Evolution of local variable
$$j^n = \begin{cases} i^n + X^n & -F^n \le i^n + X^n \le F^n \\ 0 & otherwise. \end{cases}$$

Transition Matrix
$$t_{ij}^n = \begin{cases} P(X^n = j^n - i^n) & |j^n| \le F^n, j^n \ne 0 \\ P(X^n = -i^n) + P(F^n - i^n < X^n < -F^n - i^n) & j^n = 0 \end{cases}$$

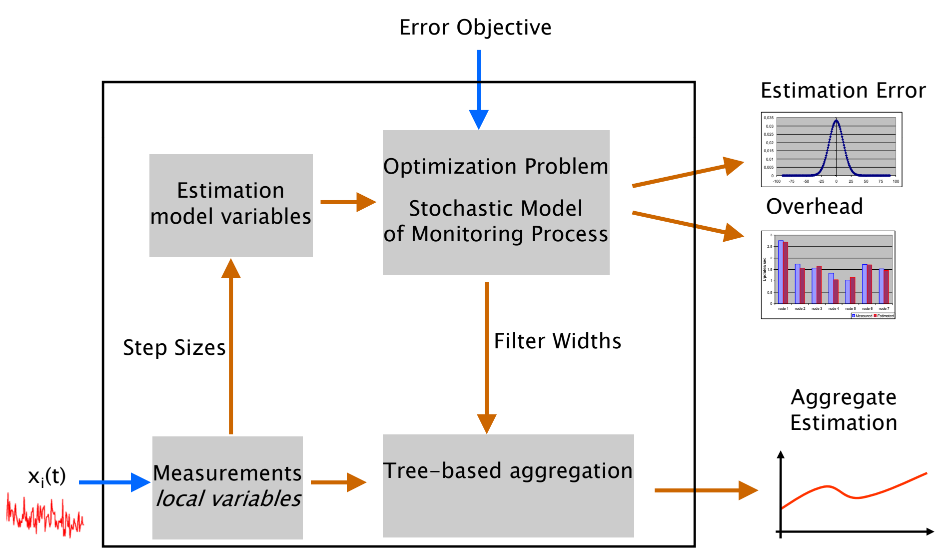
Step Size
$$P(S_{out}^n = s) = \begin{cases} \sum_{z=s-F^n}^{s+F^n} P(X^n = z) P(G^n = s - z) & |s| > F^n \\ \sum_{d=-F^n}^{F^n} \sum_{z=d-F^n}^{d+F^n} P(X^n = z) P(G^n = d - z) & s = 0 \\ 0 & otherwise. \end{cases}$$

Estimation Error $E_{out}^n = G^n$

Management Overhead $\lambda^n = (1 - P(S_{out}^n = 0))$



## A-GAP: Model-based Monitoring

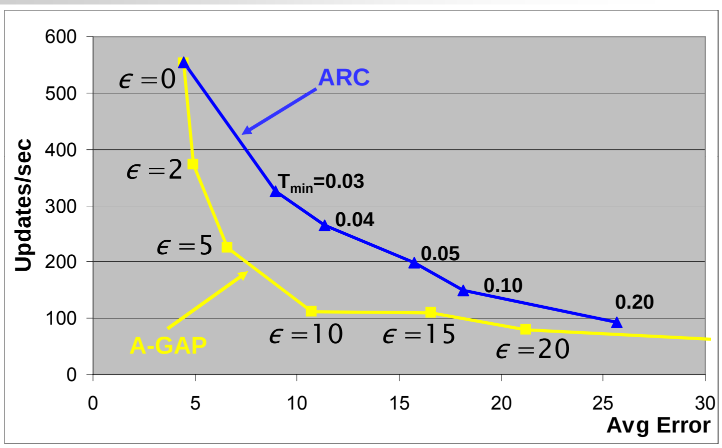


## A-GAP: Evaluation through Simulation

- Overlay topologies
  - Abovenet: 654 nodes, 1332 links
  - Grids: 25, 85, 221, 613 nodes
- Aggregate: ***Number of http flows*** in the domain
- Traces
  - From two 1 Gbit/s links that connect University of Twente to a research network
- Control cycle
  - $\tau$=1 sec

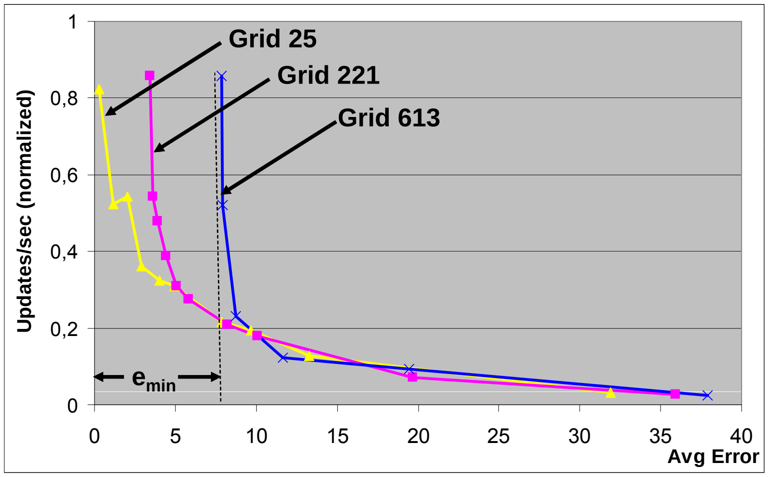


## Tradeoff: Accuracy vs Overhead




- Overhead **decreases monotonically**
- Overhead depends on the **changes of the aggregate**, not on its value.
- A-GAP **outperforms** a rate-control scheme (ARC)

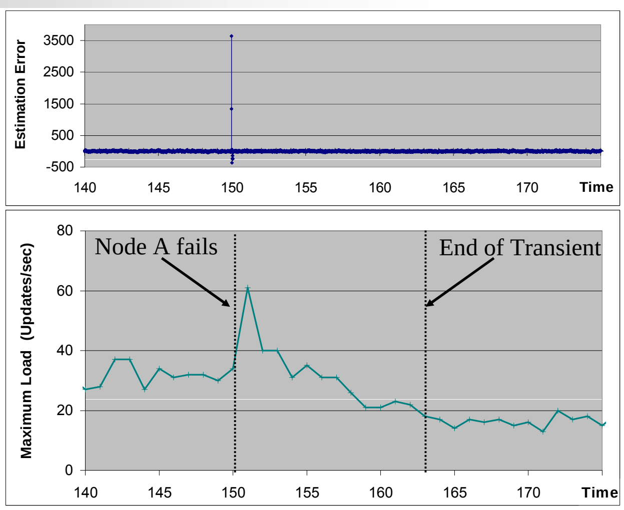
## Scalability




- Minimum error $e_{min}$ increases with the network size
- Overhead increases linearly with network size for same error objective

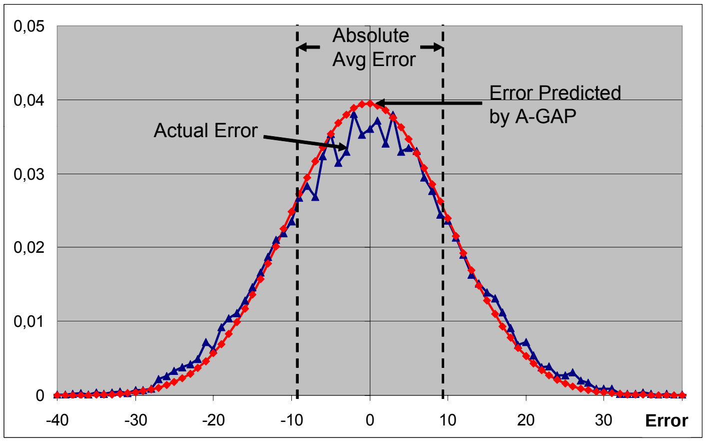


## Robustness




- Estimation error: several spikes during sub-second transient period
- Overhead: single peak with a long transient

## Error Prediction by A-GAP vs Actual Error



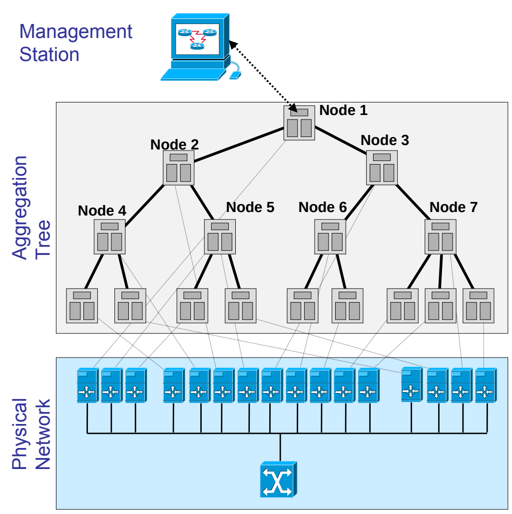

- **Accurate prediction** of the error distribution
- Maximum error >> average error (one order of magnitude)



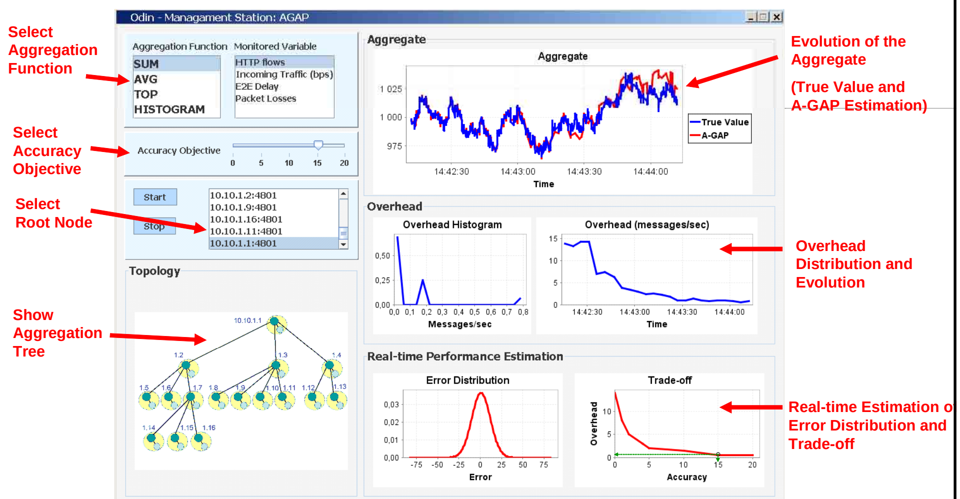
## A-GAP Prototype

### Lab testbed at KTH

- 16 monitoring nodes
- 16 Cisco 2600 Series routers
- Smartbits 6000 traffic generator
- A-GAP implemented in Java

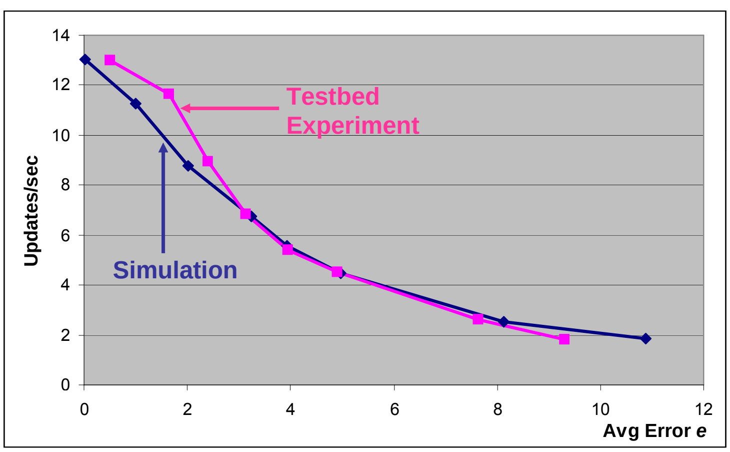


## Prototype: Management Station Interface



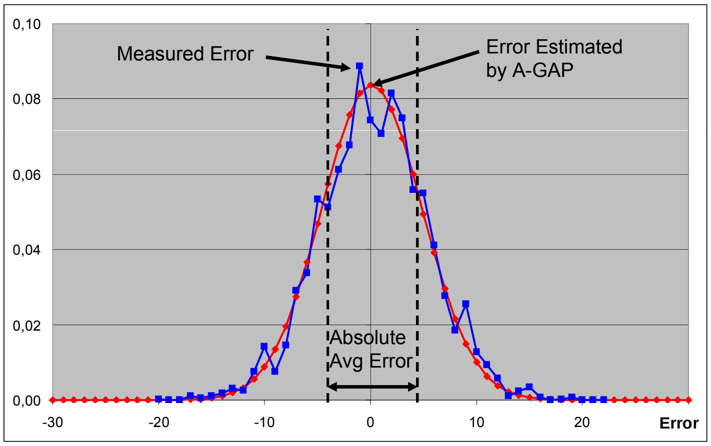



## Simulation vs Testbed Measurements




- Curves are close: difference in overhead below 3,5%
- Prototype validates simulation mode

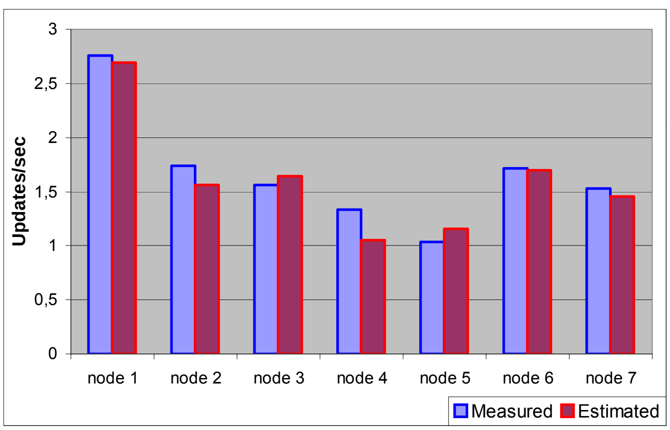
## Prototype: Error Estimation by A-GAP vs Actual Error




- **Accurate estimation** of the error distribution
- Maximum error >> average error (one order of magnitude)



## Prototype: Overhead Estimation by A-GAP vs Actual Overhead




- **Accurate estimation** of the overhead
- Estimation tends to be more accurate for **nodes close to the root**

## Gossip vs. Tree-based Aggregation



## Gossip protocols

Gossip protocols are round-based,
during each round a node randomly selects a subset of neighbors and interacts with them.

Applications

- information dissemination
- database replication
- failure detection
- resource discovery
- *computing aggregates*
- …

## Computing aggregates with gossiping

**Push Synopses**
**[Kempe et al. '03]**

- The protocol computes AVERAGE of the local variables $x_i$.
- After each round a new estimate of the aggregate is computed as $s_i/w_i$.
- ***Exponential convergence*** for uniform gossip and complete graphs
- ***Protocol Invariants***:

$$\sum_i s_{r,i} = \sum_i x_{r,i} \ , \ \sum_i w_{r,i} = n_r$$

*Round* $0$ {
  1. $s_i = x_i$;
  2. $w_i = 1$;
  3. send $(s_i, w_i)$ to self }

*Round* $r+1$ {
  1. Let $\{(s_i^*, w_i^*)\}$ be all pairs sent to $i$ during round $r$
  2. $s_i = \sum s_i^*$; $w_i = \sum w_i^*$
  3. choose shares $\alpha_{i,j} \geq 0$ for all nodes $j$ such that $\sum_j \alpha_{i,j} = 1$
  4. for all $j$ send $(\alpha_{i,j} * s_i, \alpha_{i,j} * w_i)$ to each $j$ }

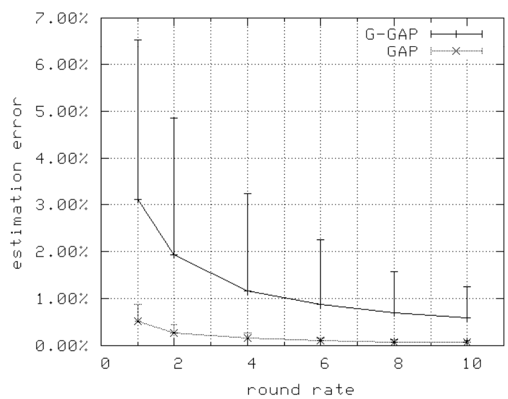


## The G-GAP protocol

*Round 0* {
  1. $s_i = x_i$ ;
  2. $w_i = 1$ ;
  3. $L_i = \{i\}$ ;
  4. for each node $j$ $(rs_{i,j}, rw_{i,j}) = (0,0)$ ;
  5. for each node $j$ $(srs_{i,j}, srw_{i,j}) = (0,0)$ ;
  6. send $(s_i, w_i, 0,0,0,0)$ to self;
  7. for all $j \neq i$ send $(0,0,0,0,0,0)$ to $j$ }

*Round r+1* {
  1. Let M be all messages received by $i$ during round $r$
  2. $s_i = \sum_{m \in M} s(m) + (x_{r,i} - x_{r-1,i})$; $w_i = \sum_{m \in M} w(m)$
  3. for all j $(acks_{i,j}, ackw_{i,j}) = (0,0)$
  4. $L_i = L_i \bigcup orig(M)$
  5. for all $j \in Neighbors$ {
     a. $(rs_{i,j}, rw_{i,j}) = (rs_{i,j}, rw_{i,j}) + \sum_{m:orig(m)=j} ((rs(m), rw(m) - acks(m), ackw(m)))$
     b. $(acks_{i,j}, ackw_{i,j}) = (srs_{i,j}, srw_{i,j}) + \sum_{m:orig(m)=j} (s(m), w(m))$
     c. if (detected_failure(j)) {
        i. $(s_i, w_i) = (s_i, w_i) + (rs_{i,j}, rw_{i,j})$
        ii. $(rs_{i,j}, rw_{i,j}) = (srs_{i,j}, srw_{i,j}) = (0,0)$
        iii. $L_i = L_i \setminus j$
        }
     }
  6. for all $j \in L_i$ {
     a. choose $\alpha_{i,j} \geq 0$ such that $\sum_j \alpha_{i,j} = 1$
     b. choose $\beta_{i,j} \geq 0$ such that $\sum_j \beta_{i,j} = 1$ and $\beta_{i,i} = 0$
     c. $(srs_{i,j}, srw_{i,j}) = \beta_{i,j}(\alpha_{i,i} s_i - x_i), \beta_{i,j}(\alpha_{i,i} w_i - 1)$
     d. send $(\alpha_{i,j} s_i, \alpha_{i,j} w_i, srs_{i,j}, srw_{i,j}, acks_{i,j}, ackw_{i,j})$ to $j$
     e. $(rs_{i,j}, rw_{i,j}) = (rs_{i,j} + \alpha_{i,j} s_i, rw_{i,j} + \alpha_{i,j} w_i)$
     }
}

## Accuracy vs. Overhead
## gossip- and tree-based aggregation protocol




***GAP*** and ***G-GAP***

654 node network
GoCast overlay, connectivity 10

aggregation: AVERAGE

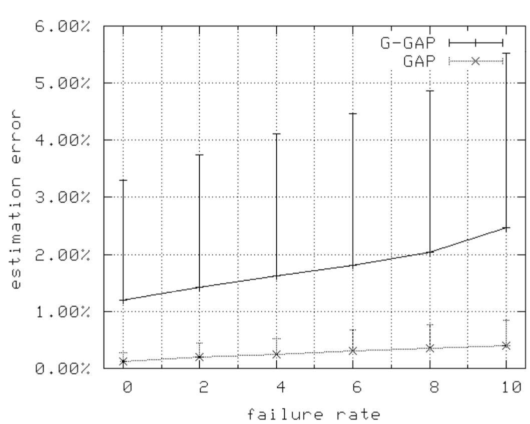
UT trace

4 rounds/sec

no failures



## Accuracy vs. Failure Rate
## gossip- and tree-based aggregation protocol




***GAP*** and ***G-GAP***

654 node network
GoCast overlay, connectivity 10

aggregation: AVERAGE

UT trace

4 rounds/sec

nodes fail randomly, recover after 10 sec

## Summary

- A self-organizing monitoring layer inside the managed system
  - Monitoring network-wide aggregates.
  - Polling, continuous monitoring, threshold detection.
  - Controlling the performance trade-offs.

- Continuous monitoring of aggregates with accuracy objectives
  - Efficient, scalable and adaptable monitoring using aggregation trees is feasible.
  - Model-based monitoring allows for performance prediction.

- Tree-based vs. gossip-based continuous monitoring
  - In a traditional wireline networking scenario, tree-based aggregation outperforms gossip-based aggregation



## References

F. Wuhib, M. Dam, R. Stadler: "Decentralized Detection of Global Threshold Crossings Using Aggregation Trees," Computer Networks, Vol. 52, No. 9, pp 1745-1761, 2008.

A. Gonzalez Prieto, R. Stadler: "A-GAP: An Adaptive Protocol for Continuous Network Monitoring with Accuracy Objectives," IEEE Transactions on Network and Service Management (TNSM), Vol. 4, No. 1, June 2007.

F. Wuhib, M. Dam, R. Stadler, A. Clemm: "Robust Monitoring of Network-wide Aggregates through Gossiping," 10th IFIP/IEEE International Symposium on Integrated Management (IM 2007), Munich, Germany, May 21-25, 2007.

K.S. Lim and R. Stadler: "Real-time views of network traffic using decentralized management," 9th IFIP/IEEE International Symposium on Integrated Network Management (IM 2005), Nice, France, May 16-19, 2005.