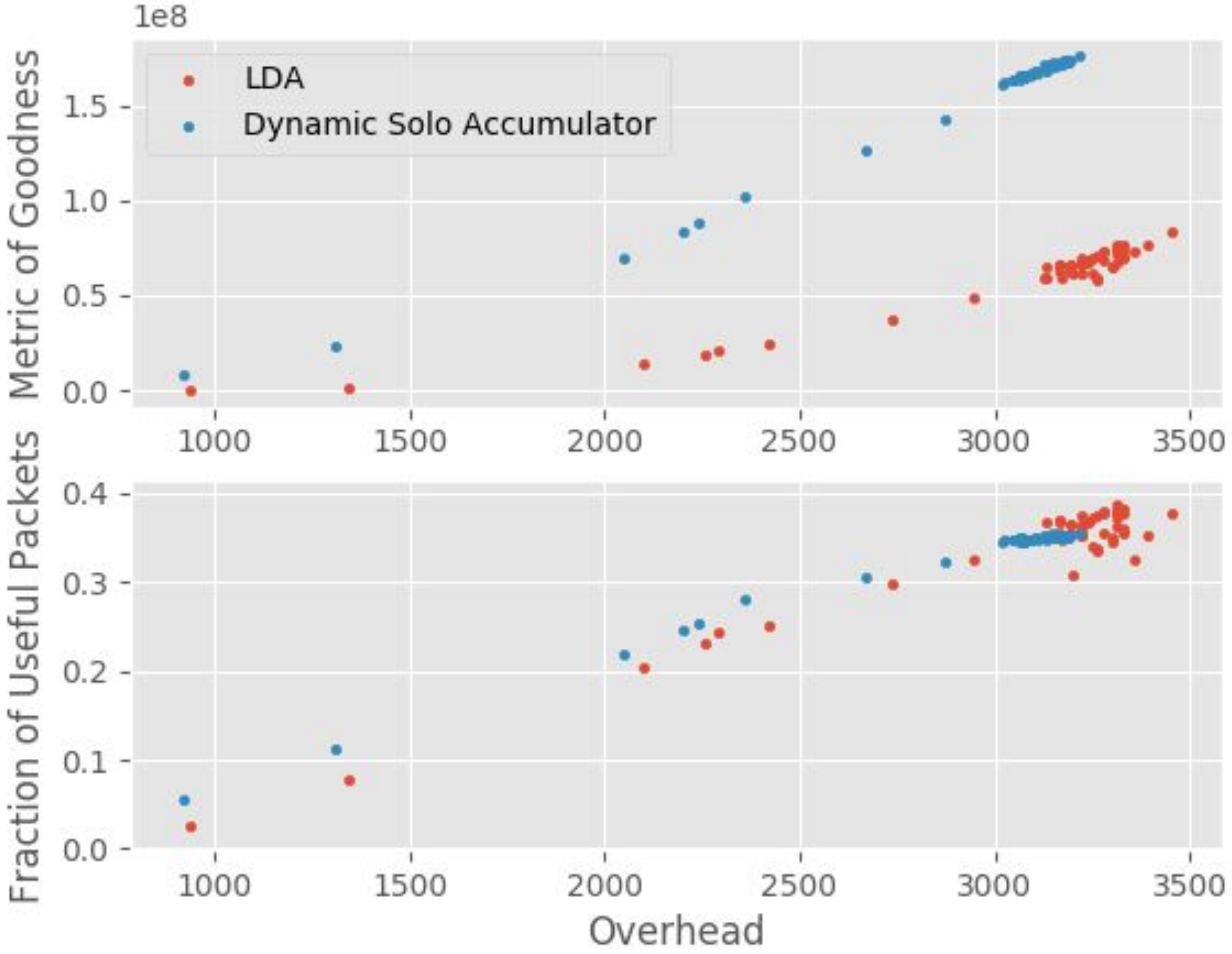
SpiNet

Microsoft

Window Adaptation (Single DSA)

- Time and space efficient
- Coarse grained measurement



Packet Burst/Uniform Loss

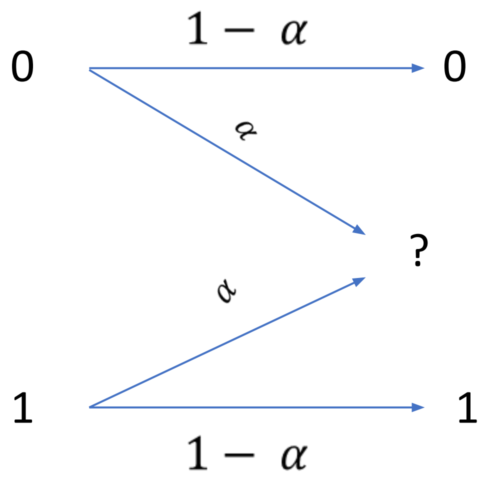
- 1. Trivial Case (single flow)

Packet Burst/Uniform Loss

- 2. Slightly looser (multiple-flow round robin case)

Packet Burst/Uniform Loss

- 3. Pure randomness case
- Modeled as a binary erasure channel



- Gödel number sequencing
- Cantor pairing
- Approximation?

Packet Burst/Uniform Loss

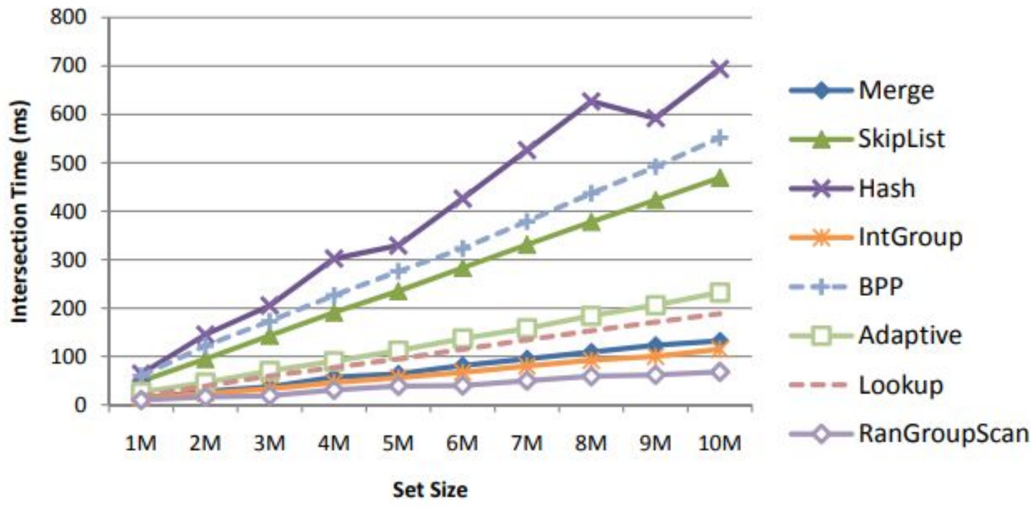
- Modify outgoing packet with ‘in-link’ sequence number (1 Mb/s overhead). Problem reduces to [1].

Window Adaptation (Multiple DSA)

- 1. Fine grained measurement

- Header space analysis
- Locality sensitive hashing
- Accumulate counters

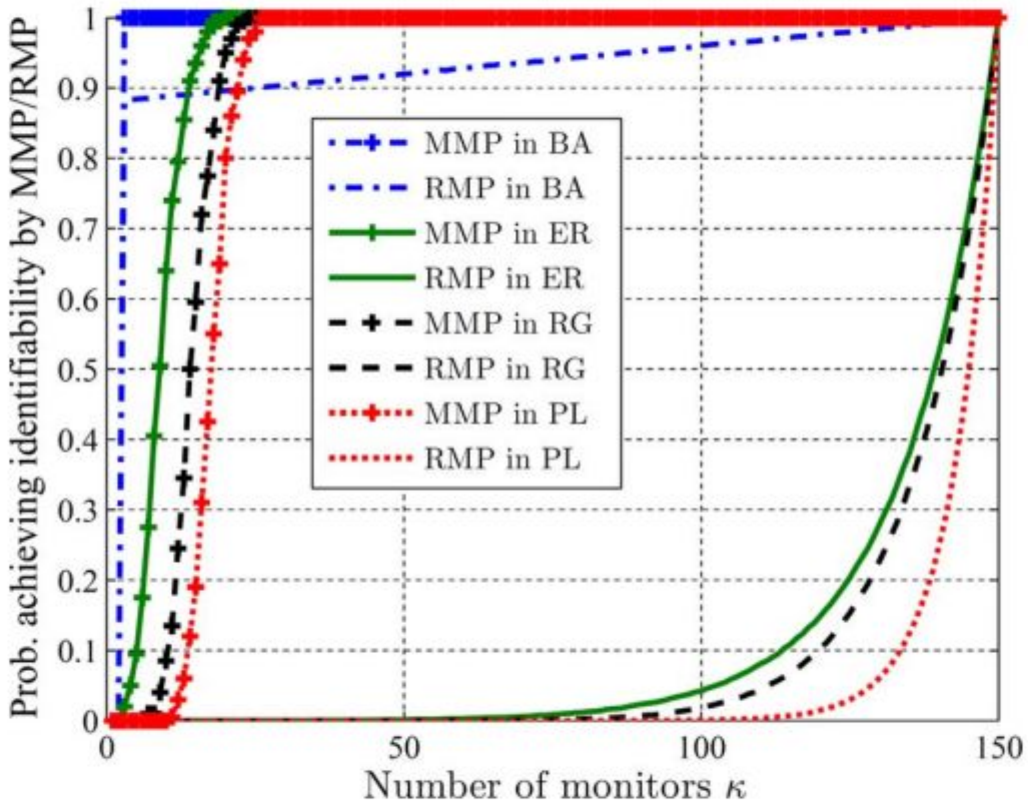
- 2. Fast Set intersection



Optimal Deployment

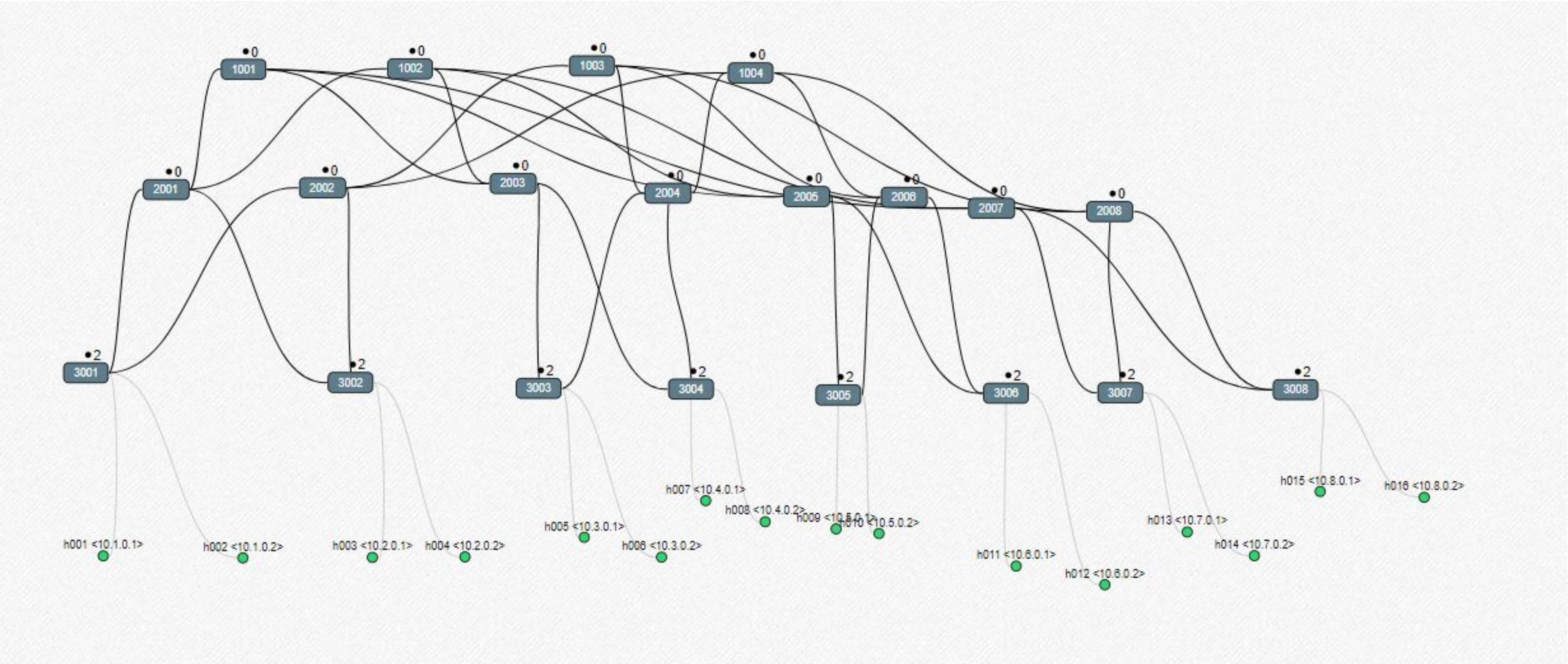
Algorithm 1: Minimum Monitor Placement (MMP)

input: Connected graph $\mathcal{G}$
output: A subset of nodes in $\mathcal{G}$ as monitors
choose all the nodes with degree less than 3 as monitors;
partition $\mathcal{G}$ into biconnected components $\mathcal{B}_1, \mathcal{B}_2, \ldots$;
foreach *biconnected component* $\mathcal{B}_i$ *with* $|\mathcal{B}_i| \geq 3$ **do**
    partition $\mathcal{B}_i$ into triconnected components $\mathcal{T}_1, \mathcal{T}_2, \ldots$;
    foreach *triconnected component* $\mathcal{T}_j$ of $\mathcal{B}_i$ *with* $|\mathcal{T}_j| \geq 3$ **do**
        If $0 < s_{\mathcal{T}_j} < 3$ **and** $s_{\mathcal{T}_j} + M_{\mathcal{T}_j} < 3$ **then**
            randomly choose $3 - s_{\mathcal{T}_j} - M_{\mathcal{T}_j}$ nodes in $\mathcal{T}_j$ that are neither separation vertices nor monitors as monitors;
        end
    end
    if $0 < c_{\mathcal{B}_i} < 3$ **and** $c_{\mathcal{B}_i} + M_{\mathcal{B}_i} < 3$ **then**
        randomly choose $3 - c_{\mathcal{B}_i} - M_{\mathcal{B}_i}$ nodes in $\mathcal{B}_i$ that are neither cut-vertices nor monitors as monitors;
    end
end
If *the total number of monitors* $\kappa < 3$ **then**
    randomly choose $3 - \kappa$ nonmonitor nodes as monitors;
end



Erdös–Rényi (ER) graphs, Random Geometric (RG) graphs, Barabási–Albert (BA) graphs, and Random Power Law (PL) graphs

Evaluation



- What is the memory/bandwidth usage, and how do we compare to existing solutions (FlowRadar, LDA, NetSight, etc.)?
- What is the memory usage per switch for different deployment scenarios?
- How fast is the loss detection?
- Is the information provided useful (can we use this data to improve performance and improve rules)?

- <u>Extra:</u>
  - Under realistic network traffic, how effective is the inference?