The 16th IEEE LAN/MAN Workshop,
September 3-6, 2008, Cluj-Napoca, Romania

DIPStorage: Distributed Architecture for Storage of IP Flow Records

Cristian Morariu, Thierry Kramis, Burkhard Stiller

Department of Informatics IFI, Communication Systems Group CSG,
University of Zürich
Zurich, Switzerland

Motivation
Design and Implementation
Evaluation
Concluding Remarks

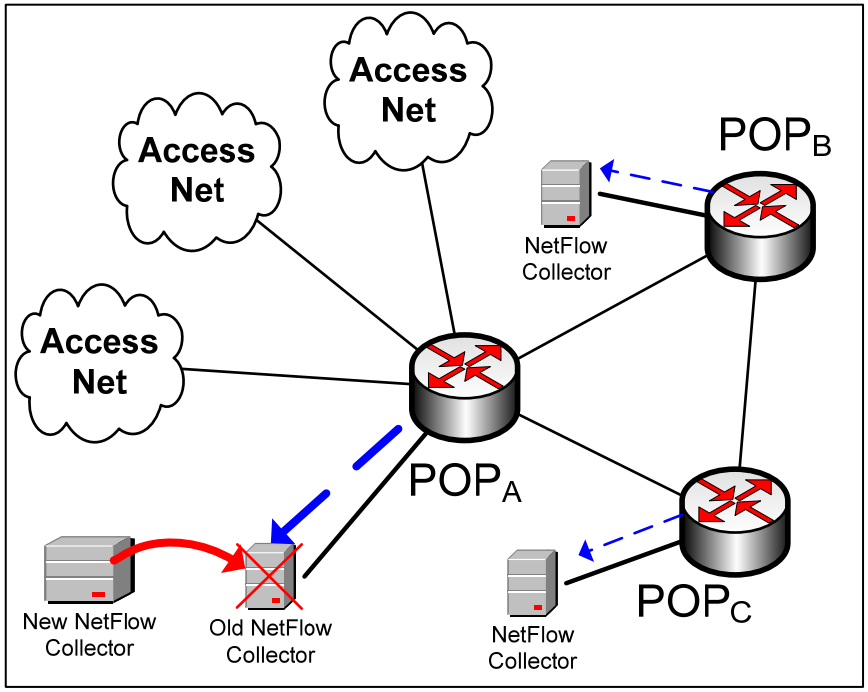
ifi

Introduction

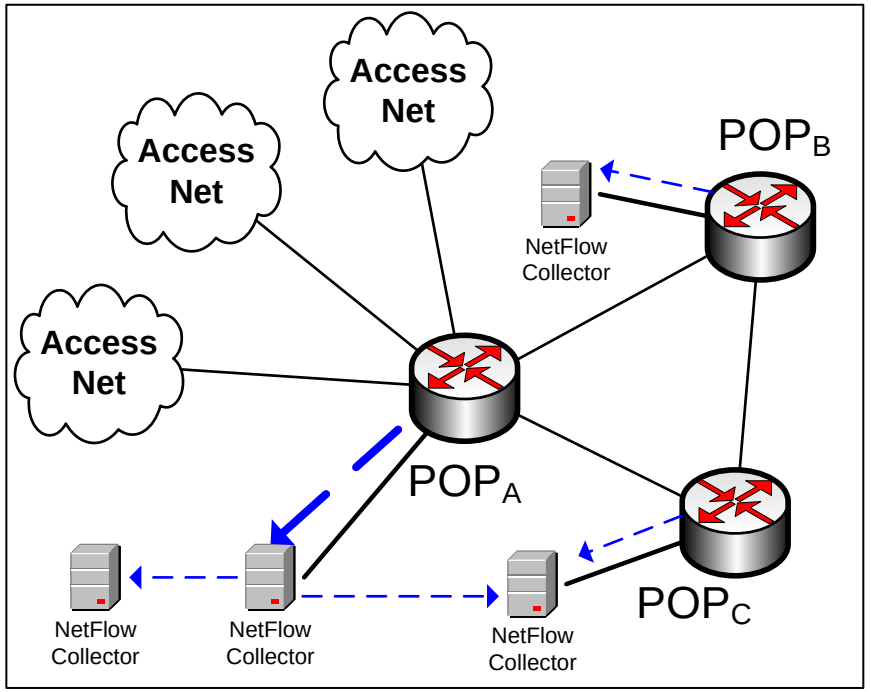
- Link speeds increase 50%-75% every year
- High-speed links generate several Gigabytes of flow records every day
- Centralized solutions for flow records storage:
  - Bad scalability
  - Poor performance in querying flow records
- Existing solutions: packet sampling and flow sampling
  - Problem: measurement accuracy
    - Lower sampling rate → lower accuracy, less storage required
- *IP Flow*: unidirectional stream of data between two endpoints
- *IP Flow Record*: quantitative description of a flow
- *Flow Keys*: IP header fields that define an IP flow

Use Case

- Traditional approach:
  - Traffic increase → replace traffic analysis hardware



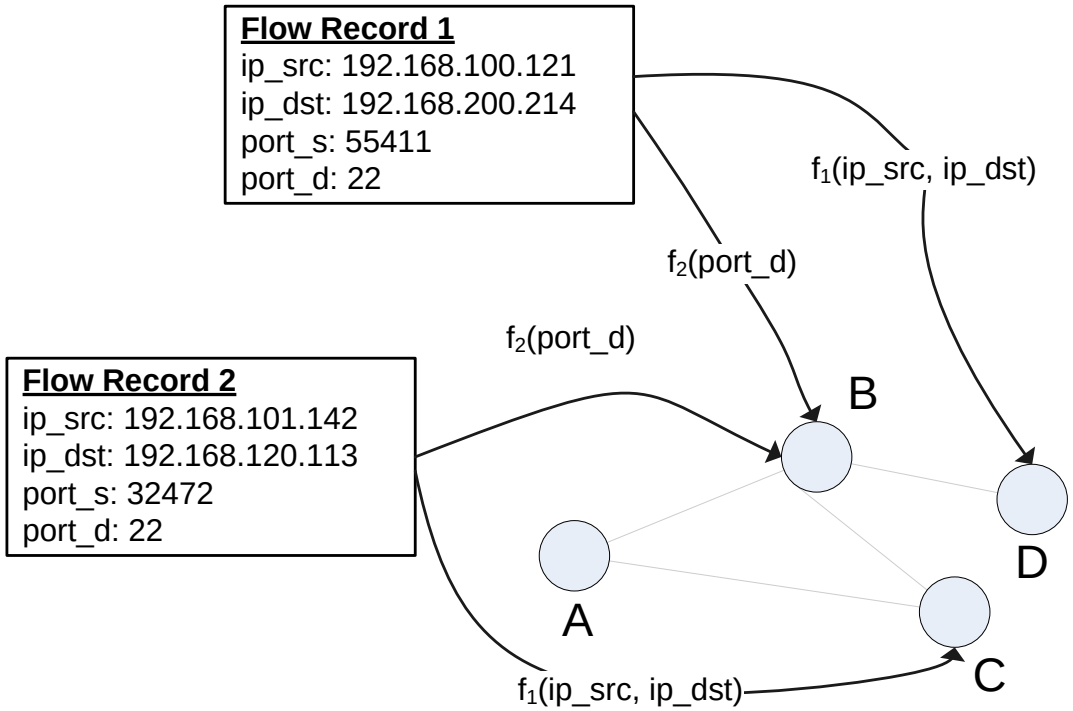
- Distributed approach:
  - Traffic increase → make use of available resources



Motivation and Proposed Approach

- Motivation:
  - Existing solutions for flow records storage lack of scalability
  - Data retention laws are enforced in many countries
- Requirements:
  - Scalable storage for IP flow records
  - Fast query response
- Idea: Store the flow records within a distributed, decentralized storage network.

Flow Record Storage



- DipStorage nodes are organized in a P2P overlay.
- A routing function is applied on the *Flow Keys* of each flow record.
- Result of the routing function returns the storage node of that flow record.
- The flow record is routed through the P2P network to the node responsible with its storage.
- Several routing functions can be used in parallel for redundancy.

Storage Strategies

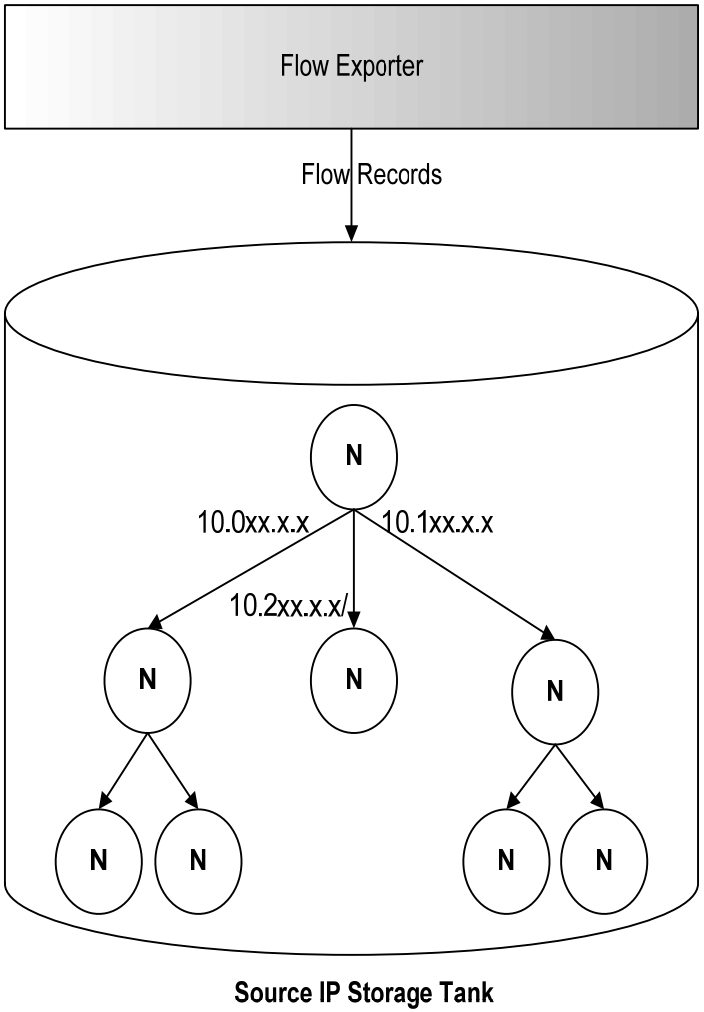
Random storage

- Routing function is a pseudo-random function.
- All storage nodes receive approximately the same number of flow records.
- Can be built on top of a fully-decentralized P2P overlay.

Structured storage

- Stores together flow records with similar properties.
- Needs a structured P2P network.
- Routing function needs to generate “similar” results for flow records with “close” characteristics.
- Some nodes may become more loaded than others.

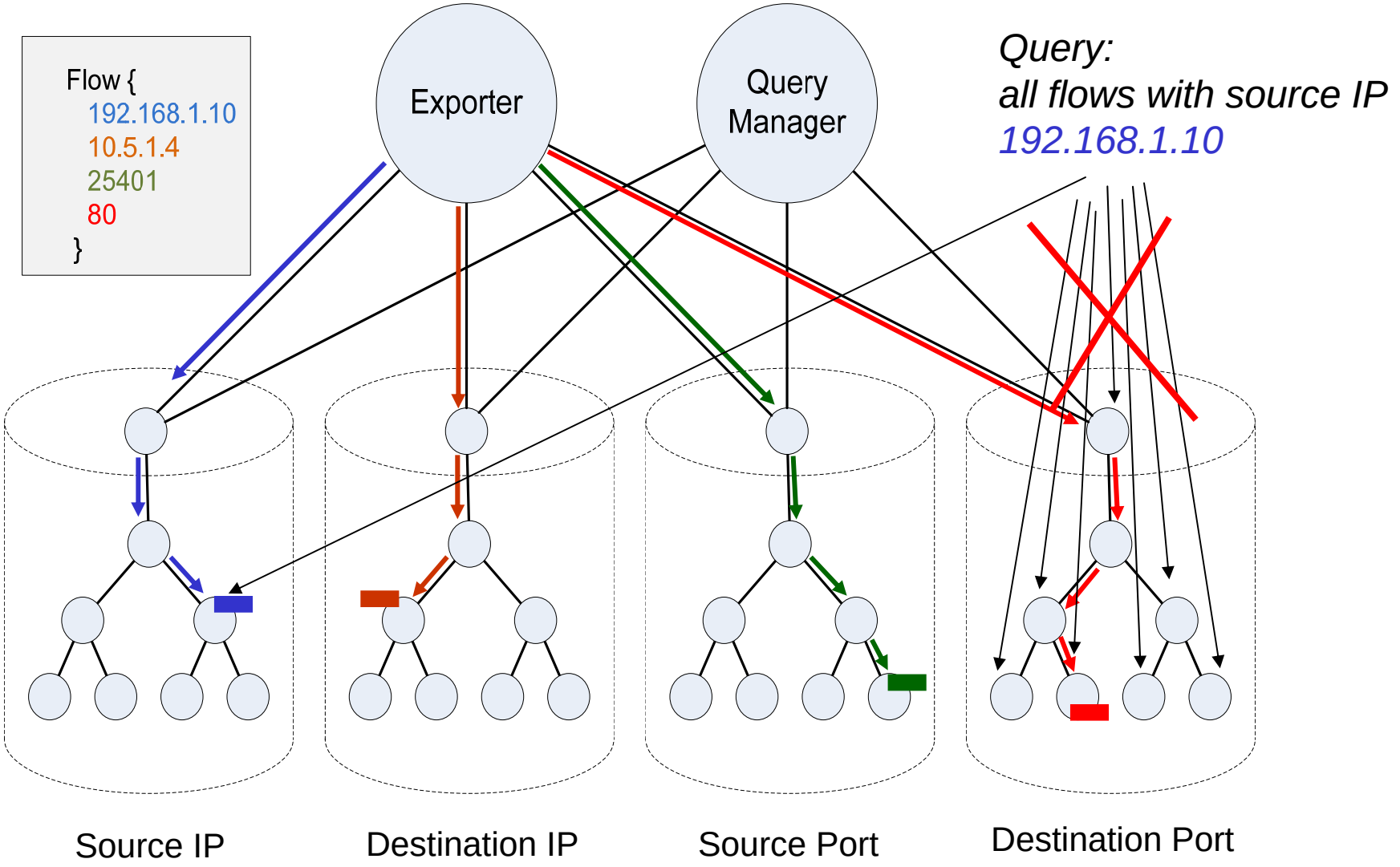
DIPStorage Tank



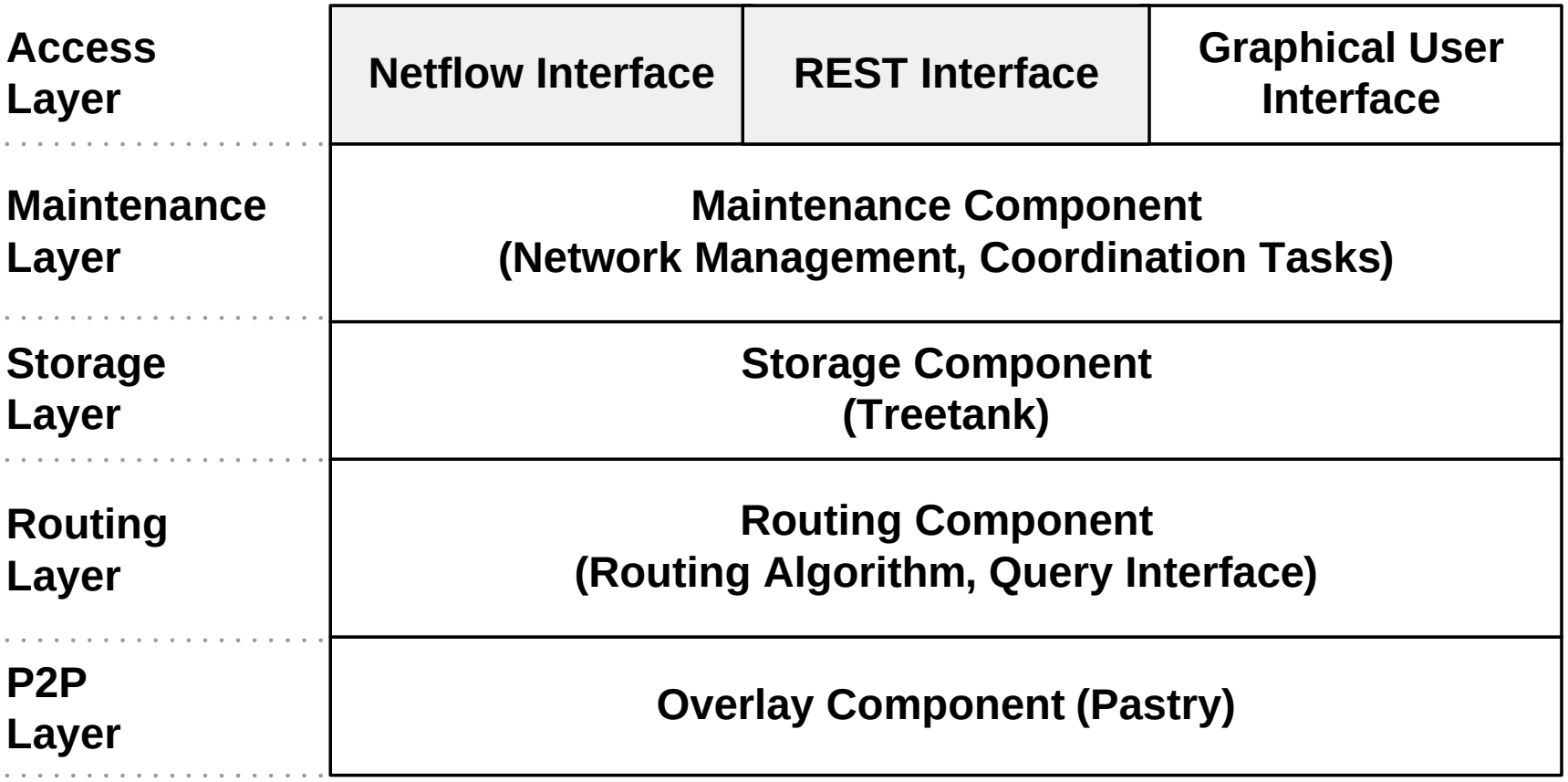
DIPStorage Tank:

- A set of nodes organized in a structured P2P overlay.
- All nodes have the same routing function.
- Flow records are routed based on a particular characteristic (*e.g.* source IP address)

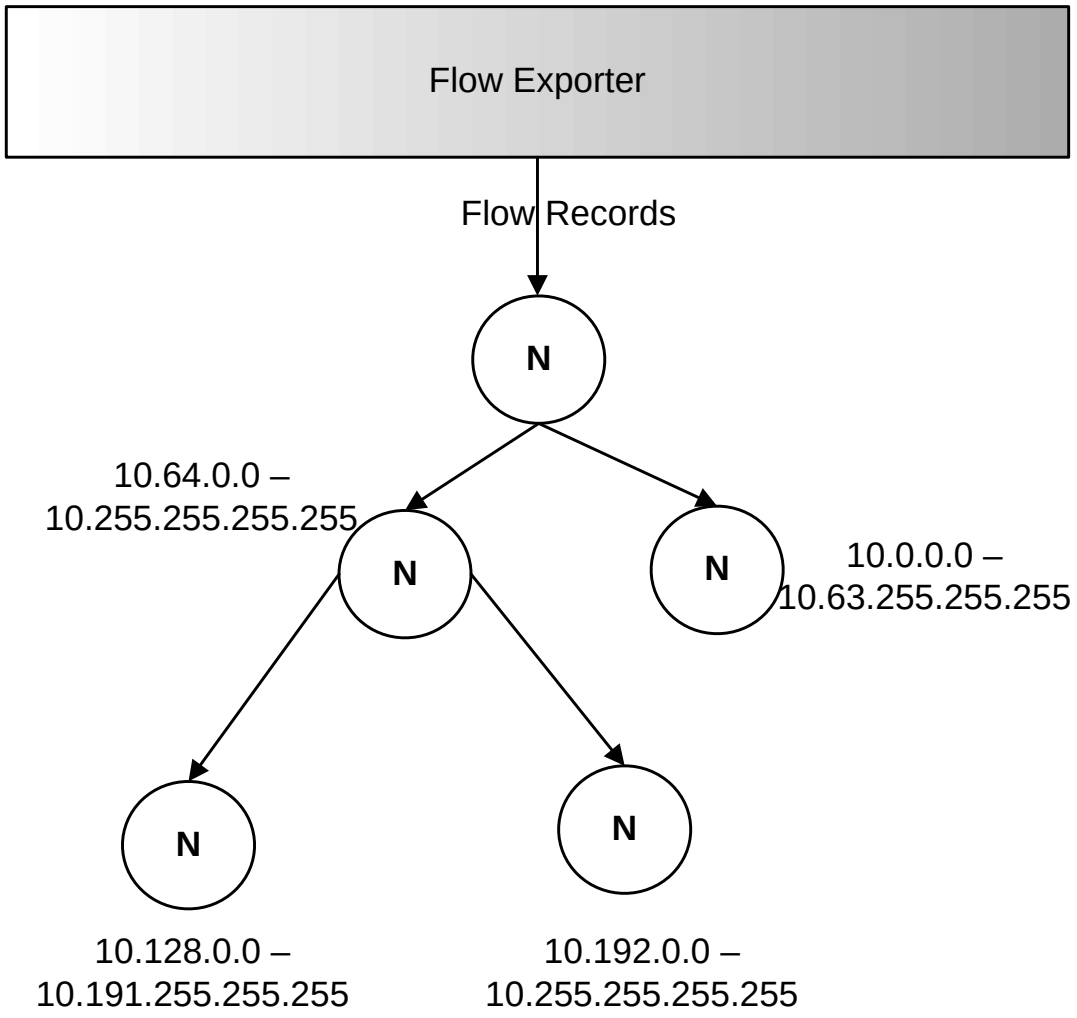
DIPStorage Architecture



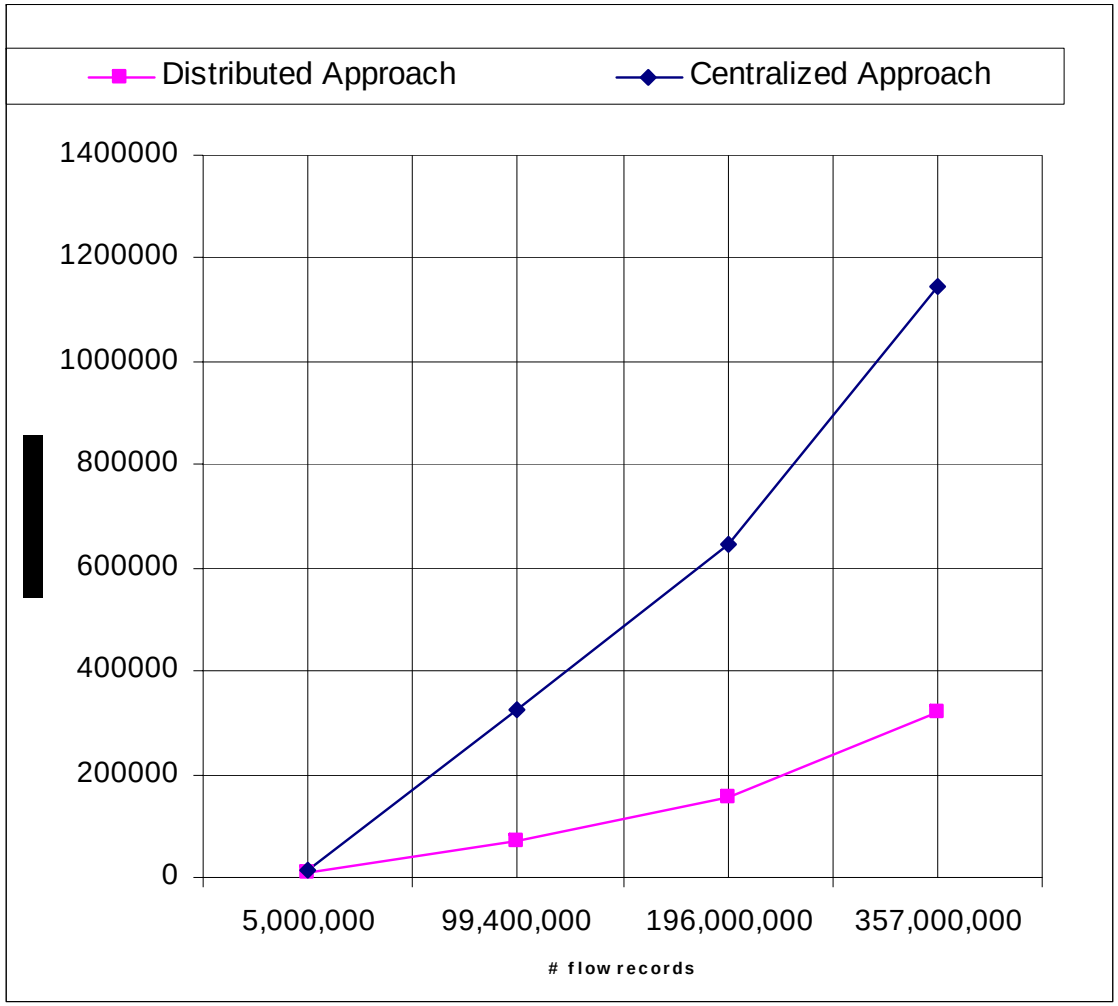
Implementation Architecture



Evaluation Scenario



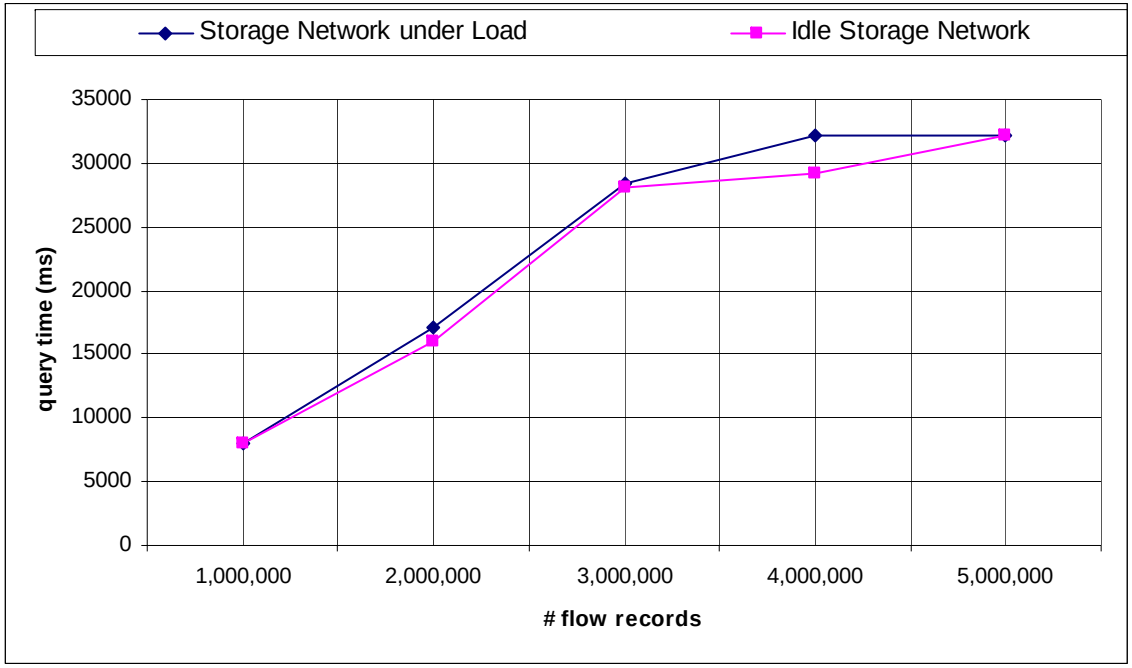
DipStorage Evaluation



Performance evaluation

- 4 storage nodes used
- Speedup increase by a factor of 4 at high number of flow records

DipStorage Load Resilience



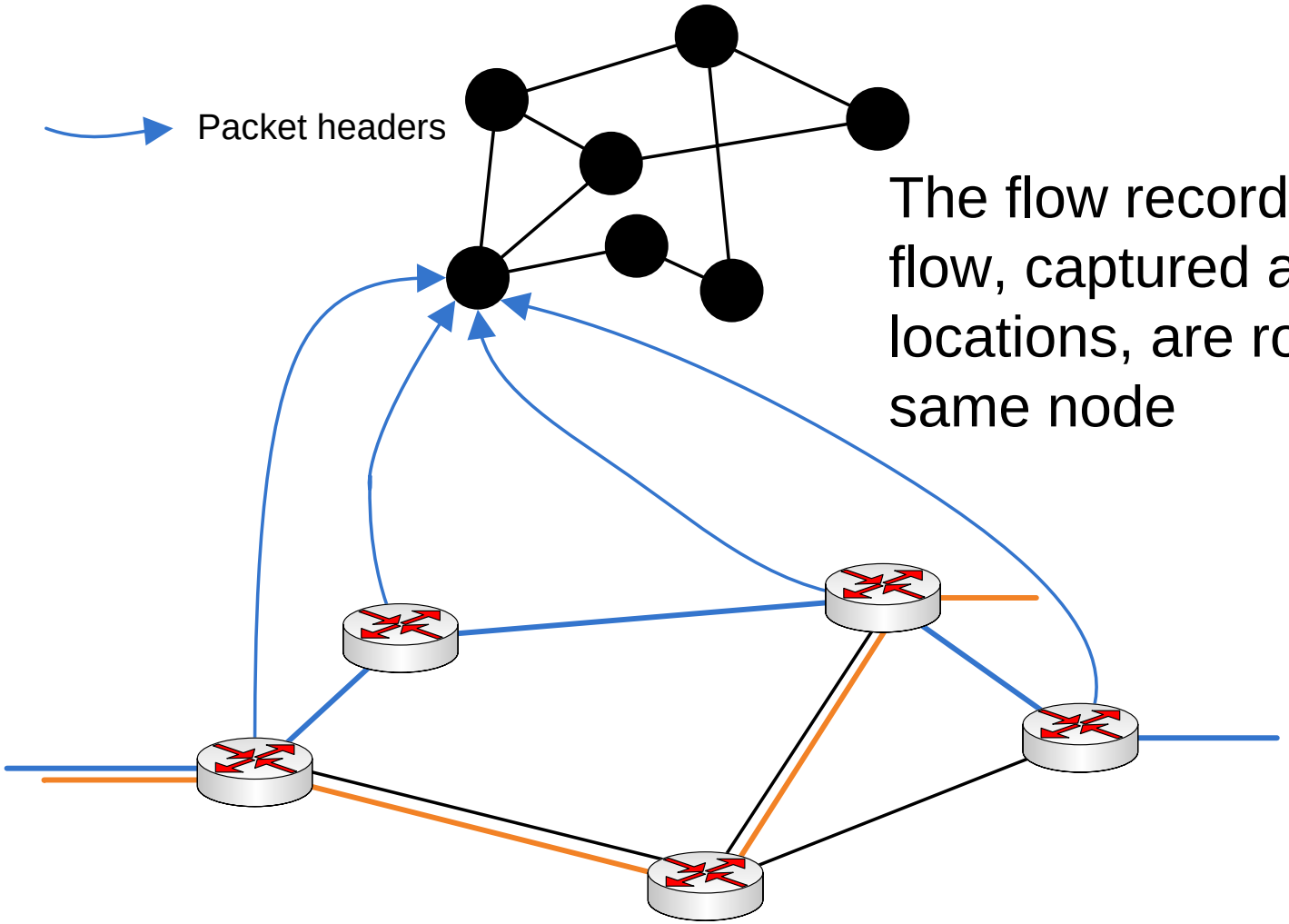
Purpose:

Test the query performance when the storage network is idle and when it stores flow records at a rate of ~10.000 flows/second.

Result:

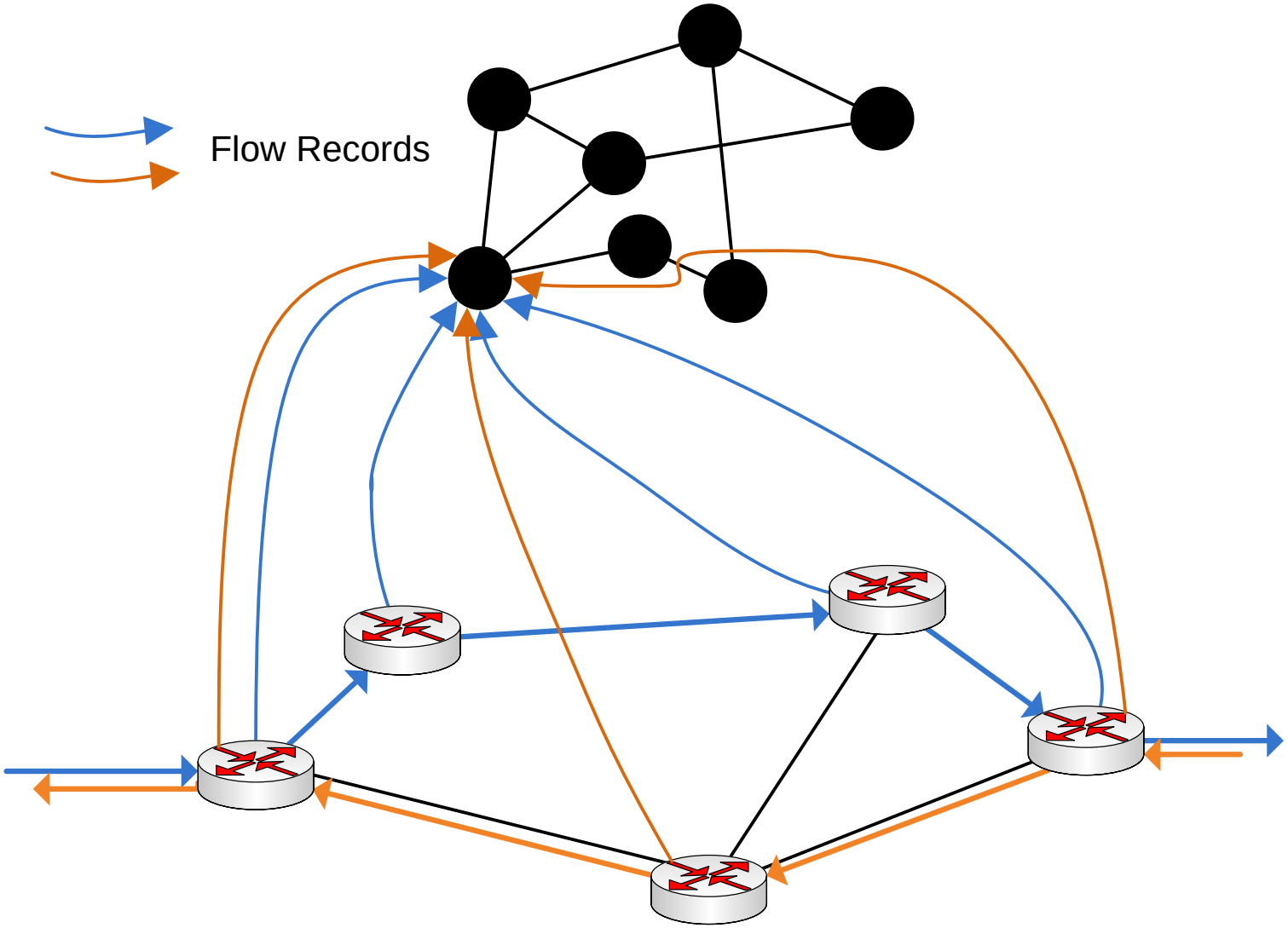
Query time is not significantly influenced by the routing/storage process.

Applications: Multi-Point Delay Calculation (1)



The flow records for the same flow, captured at different locations, are routed to the same node

Applications: Asymmetric Route Detection



Conclusion Remarks

- Existing distributed flow storage approaches are based on fixed, inflexible configurations
- A P2P based approach was not yet investigated
- DIPStorage allows:
  - Scalability
  - Increased storage space for flow records
  - Faster query response for IP flow records repositories

Thank you for your attention!

Questions ?