PayLess: A Low Cost Network Monitoring Framework for Software Defined Networks

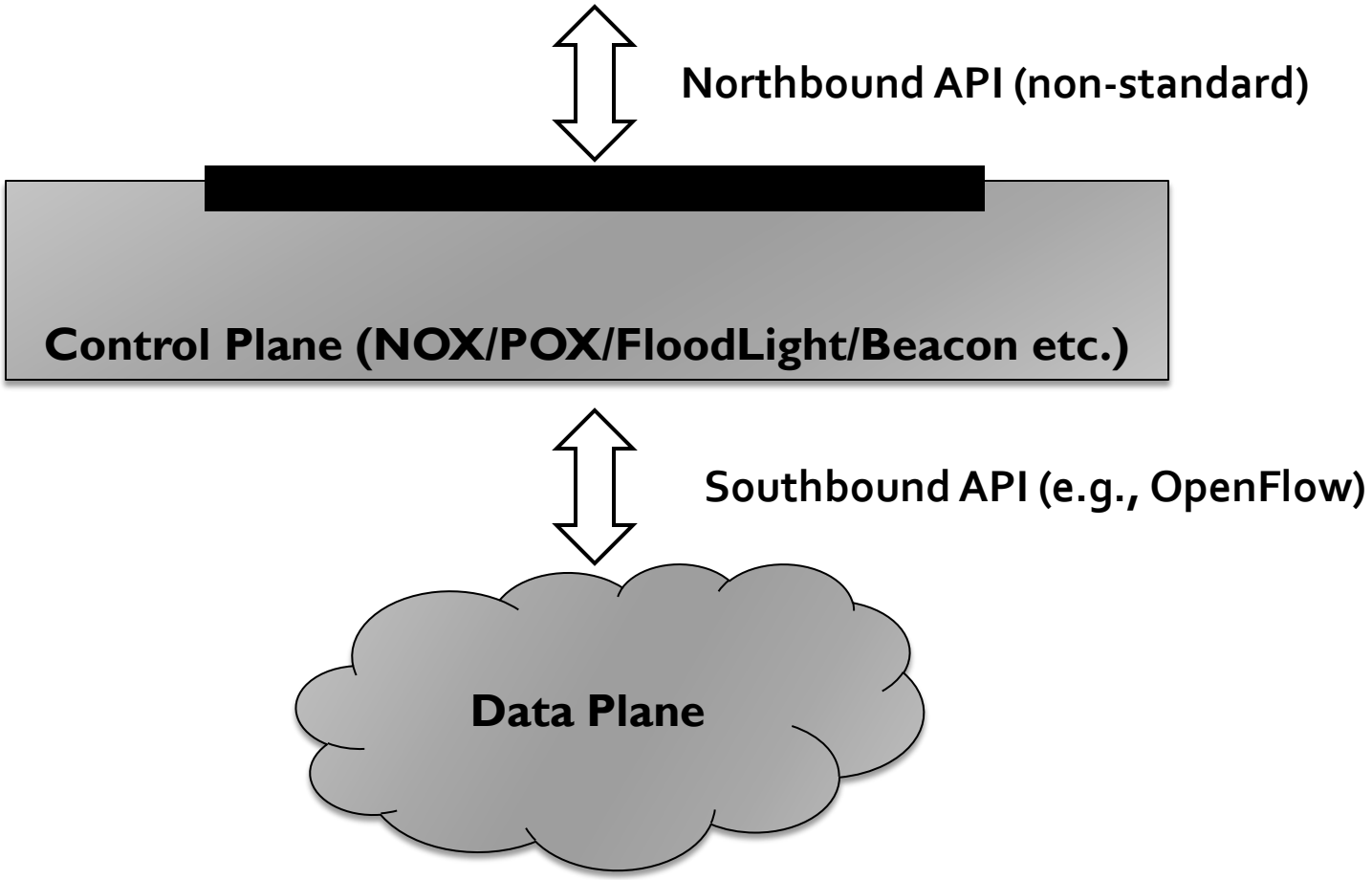
Shihabur R. Chowdhury, Md. Faizul Bari, Reaz Ahmed and Raouf Boutaba

David R. Cheriton School of Computer Science, University of Waterloo

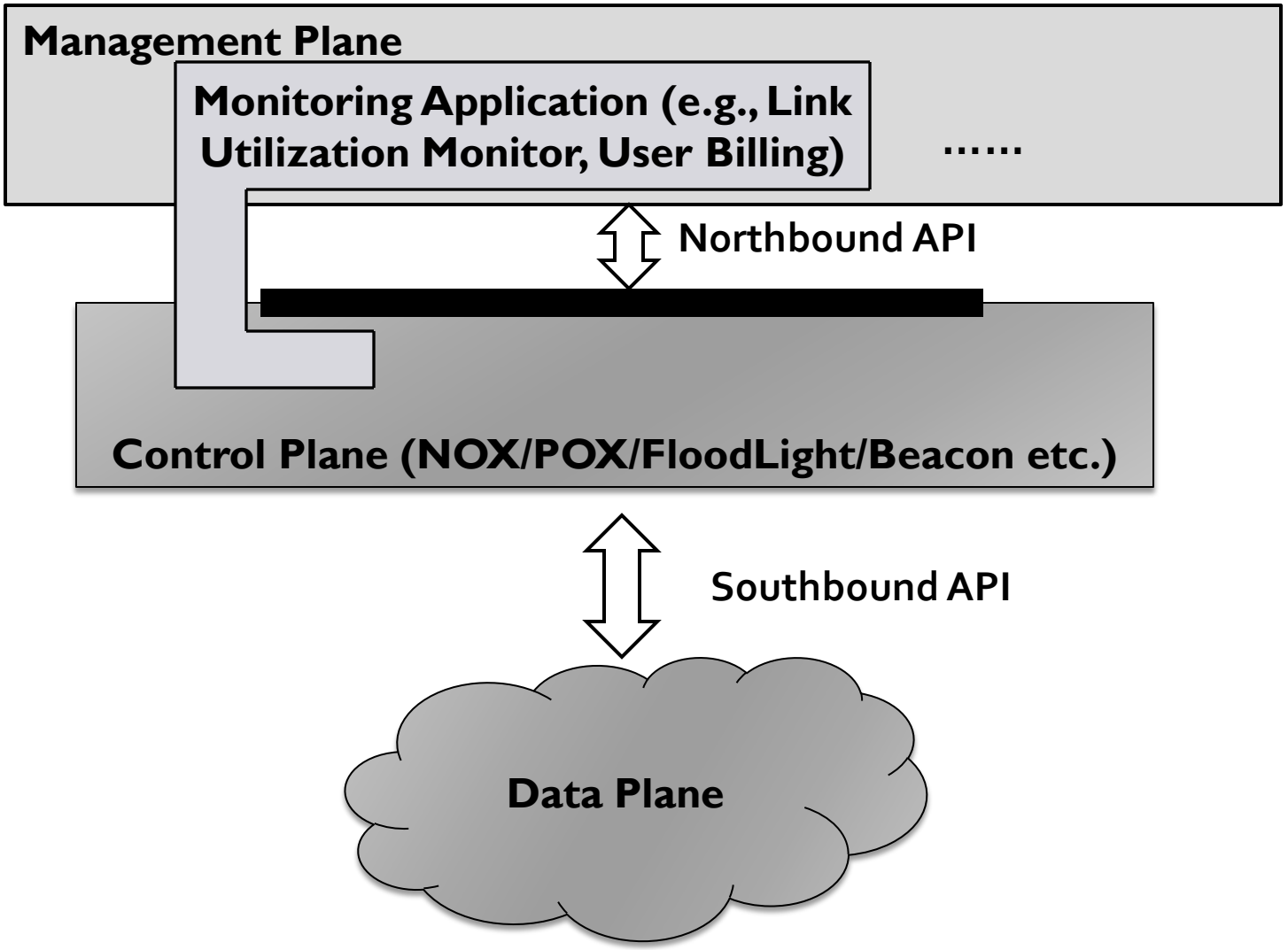
This work has been supported by NSERC discovery grant and SAVI research network

Presented By: Shihabur Rahman Chowdhury

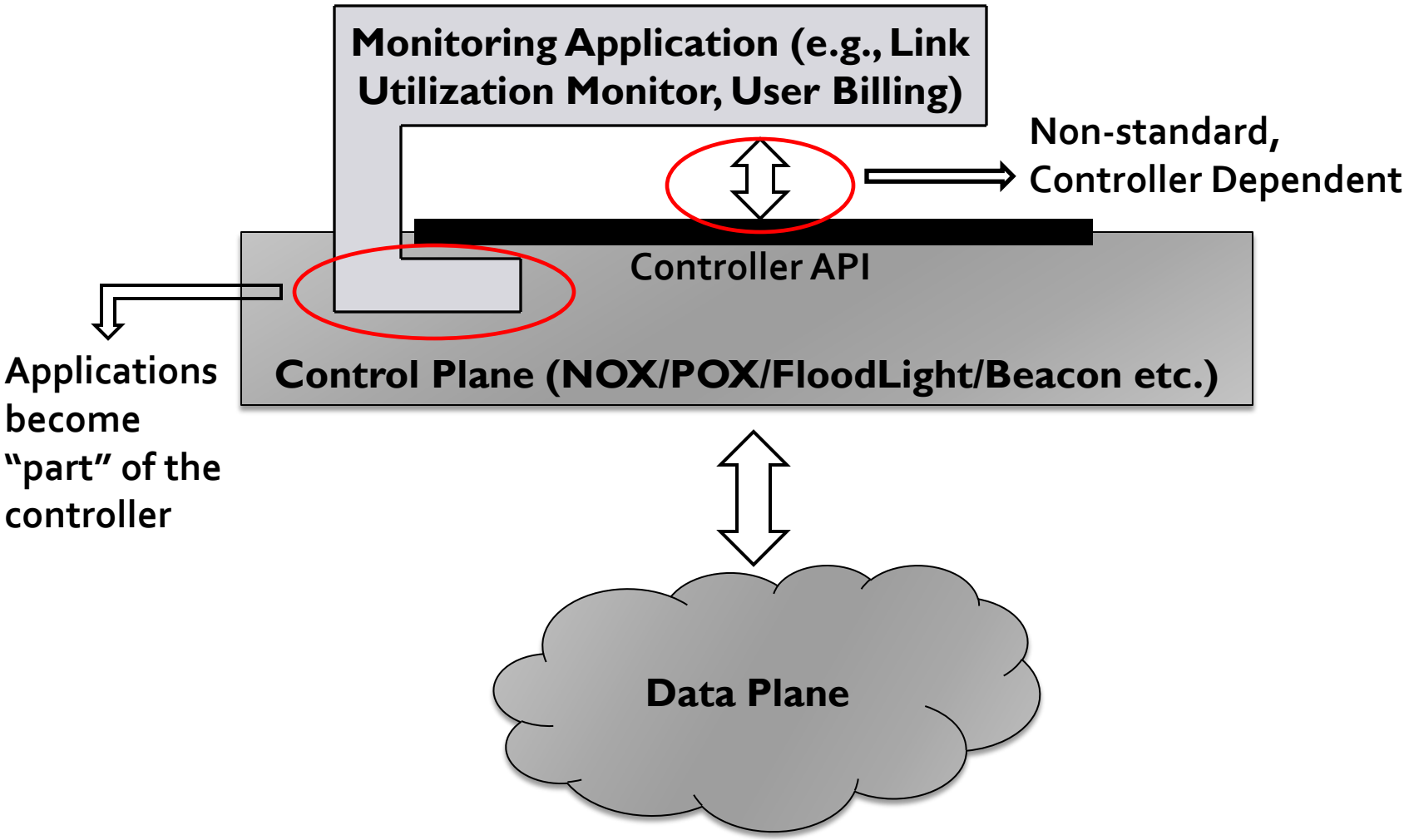
Typical SDN Scenario



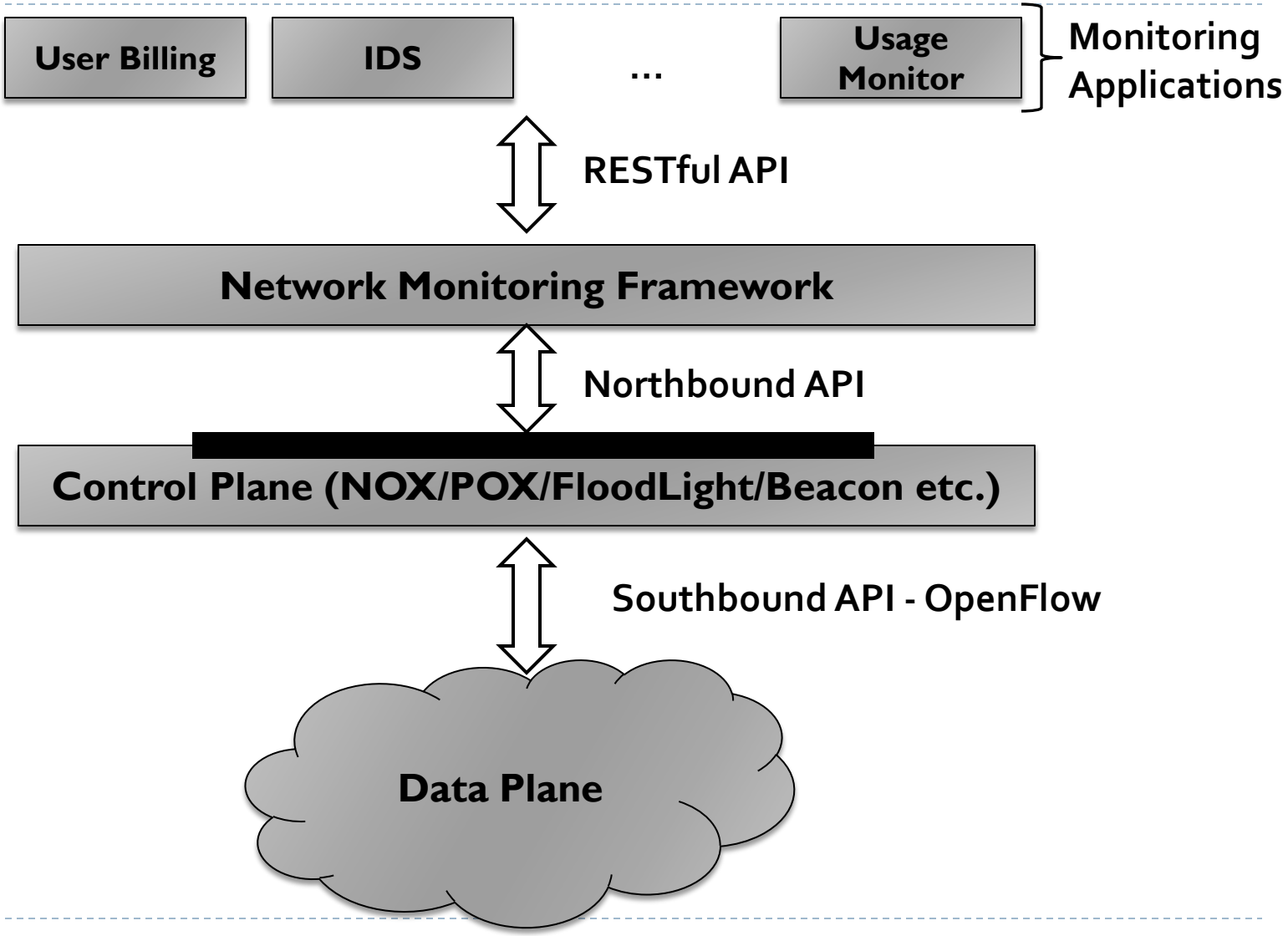
Application Development over SDN: Current Scenario



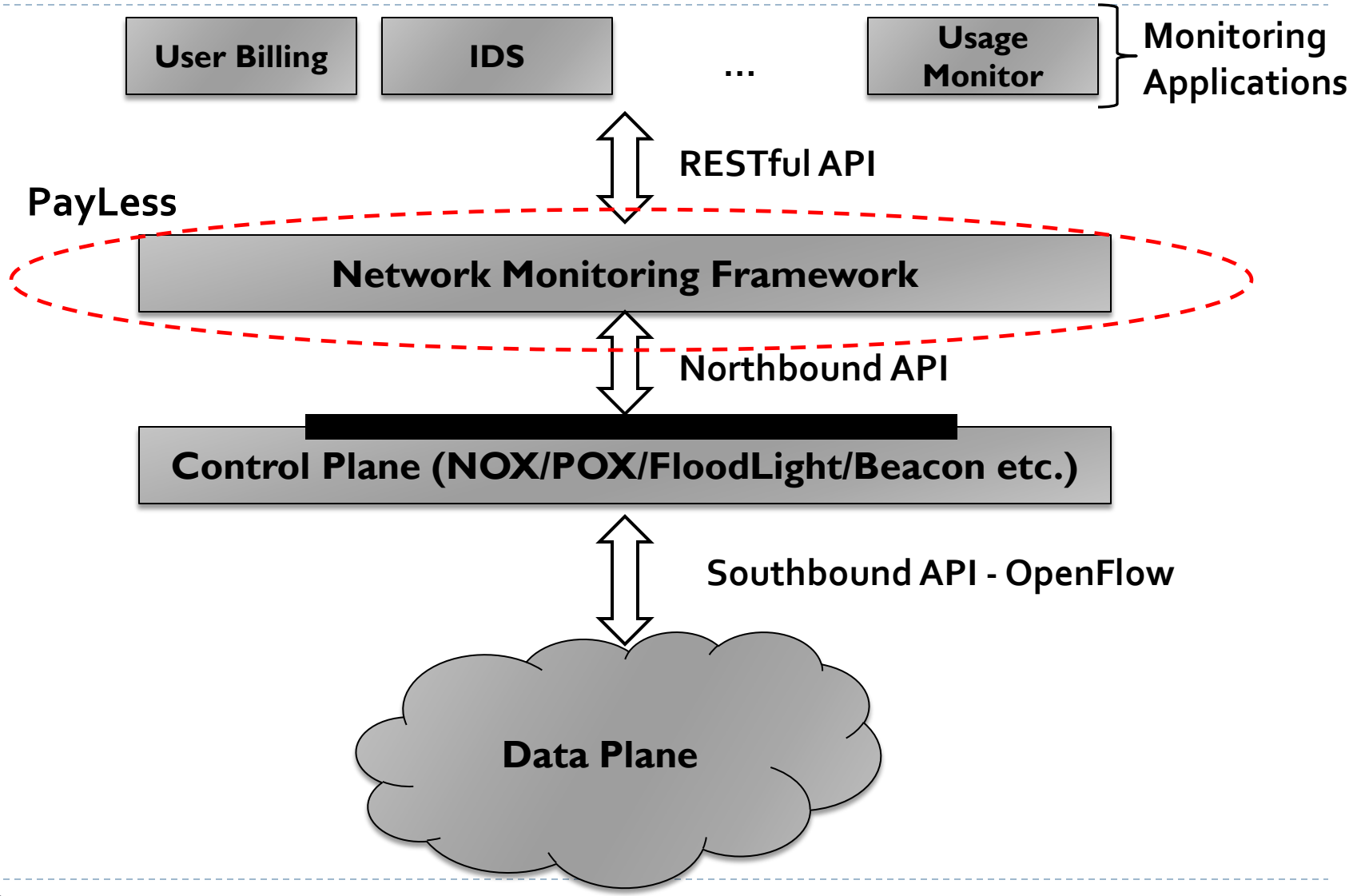
Application Development over SDN: Current Scenario (contd…)



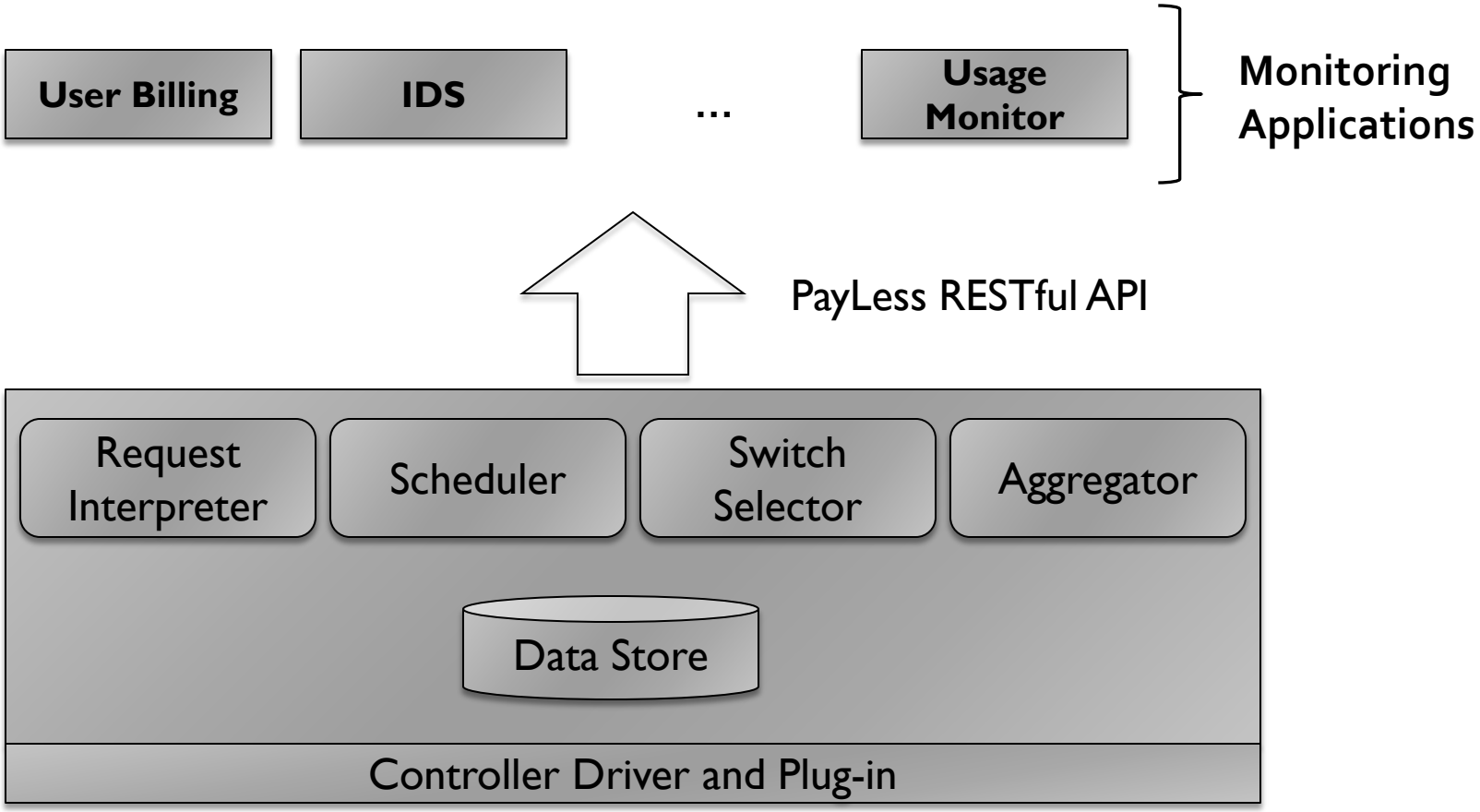
Application Development over SDN: Proposal



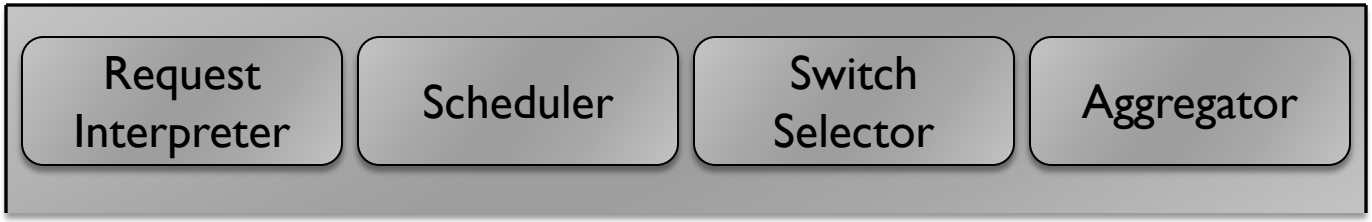
Application Development over SDN: Proposal



PayLess Architecture

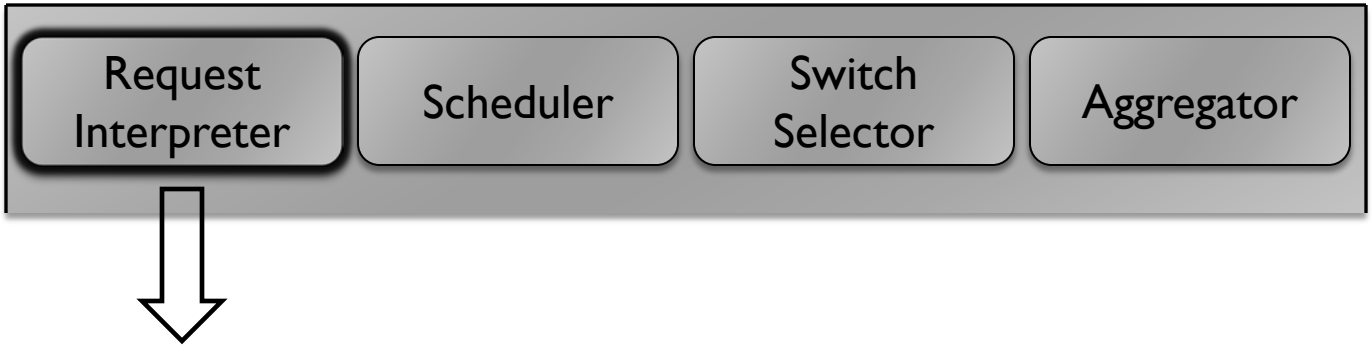


PayLess Architecture (contd…)



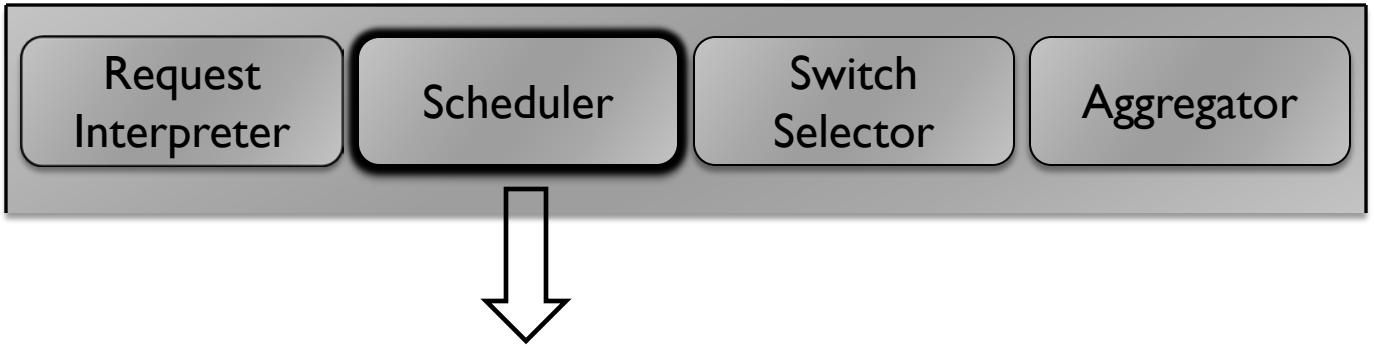
- Each component has well-defined interface
  - Can be easily replaced by a custom implementation
    - i.e., data aggregation level, sampling algorithm etc. can be customized

PayLess Architecture (contd…)



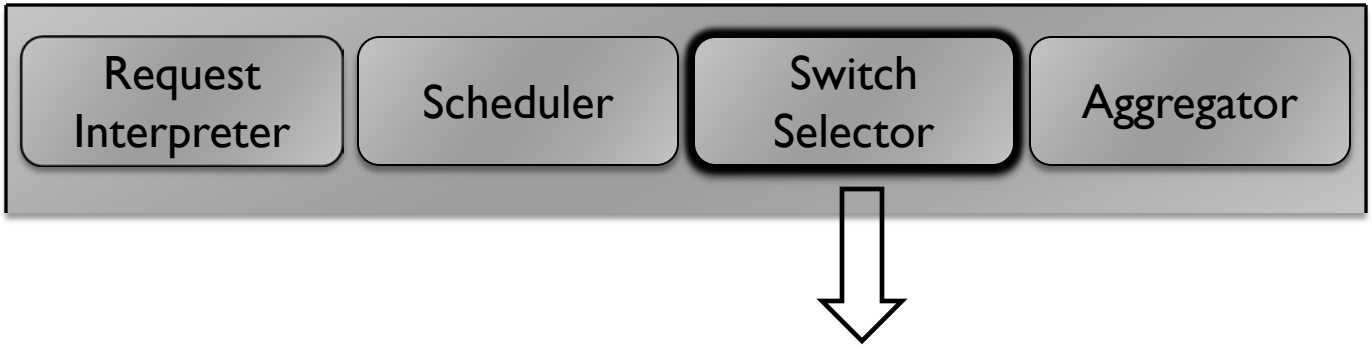
- Determines ***what*** to monitor by translating monitoring requests from the applications
  - Requests are sent in JSON format
    - May contain:
      - Type (performance, security, fault etc.)
      - Monitoring metric (what to measure)
      - Entity (which network entity to monitor)
      - Aggregation level (per switch / user / flow etc.)
      - Priority (real-time, medium, low)
      - Logging (format of the log)

PayLess Architecture (contd…)



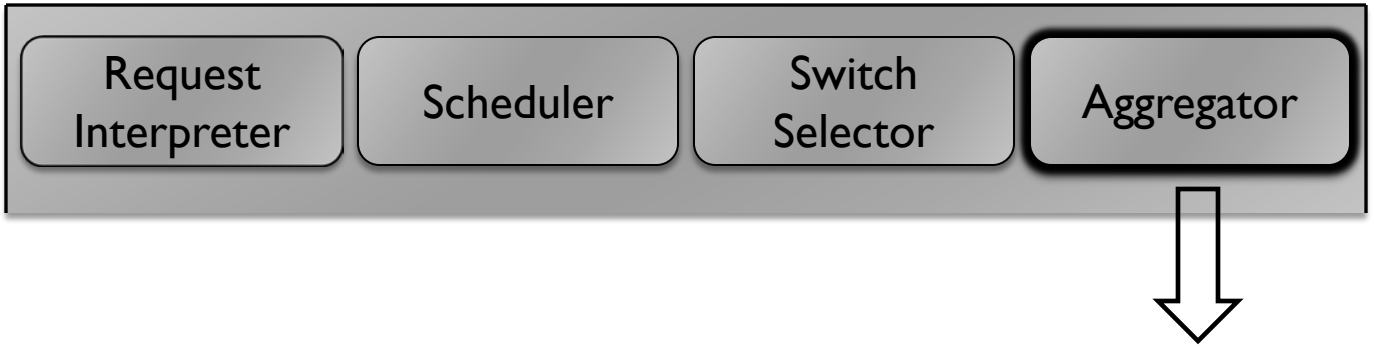
- Determines ***when*** to collect statistics from the network.
  - Periodic
  - At specific events
  - Adaptive sampling
  - *etc.*

PayLess Architecture (contd…)



- Determines ***which*** subset of network elements should be probed for statistics.
  - Only the ingress and egress switches
  - Switches with maximum connectivity
  - *etc.*

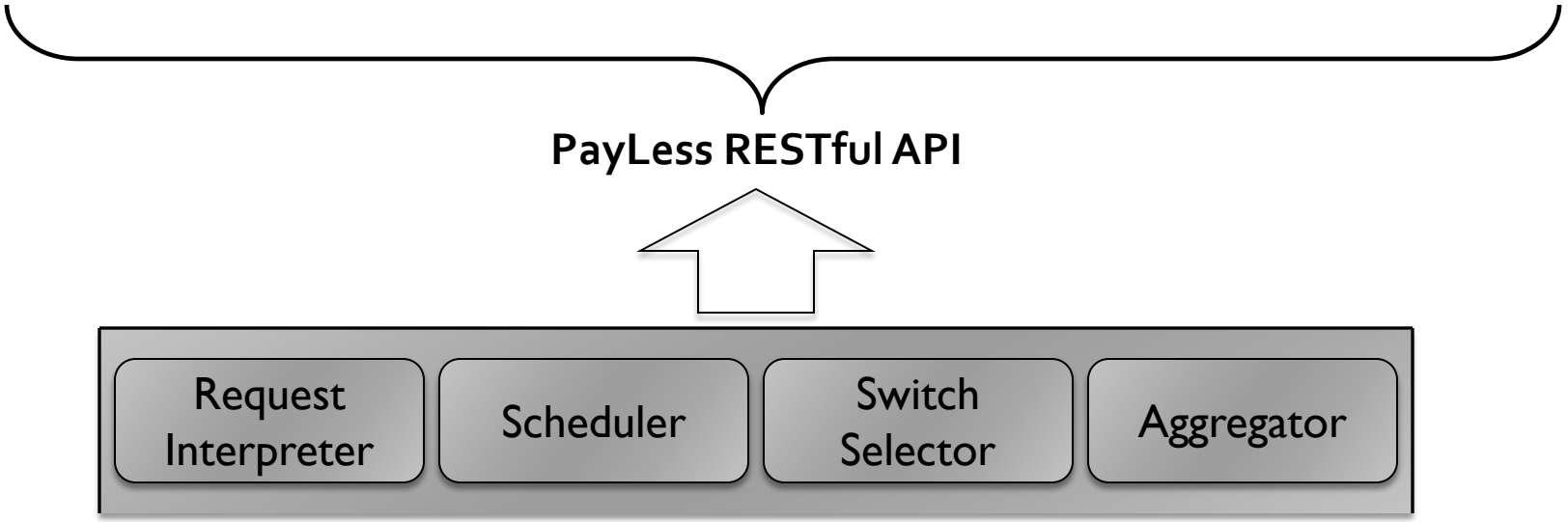
PayLess Architecture (contd…)



- Aggregates the raw monitoring data as per application requirement
  - Per link aggregation
  - Per user aggregation
  - Per switch aggregation
  - *etc.*

PayLess Architecture (contd…)

- RESTful API for developing monitoring applications
  - Applications can be oblivious of control plane technology
  - Applications can be written in any Language
  - Despite of the control plane technology, applications have the same set of services available

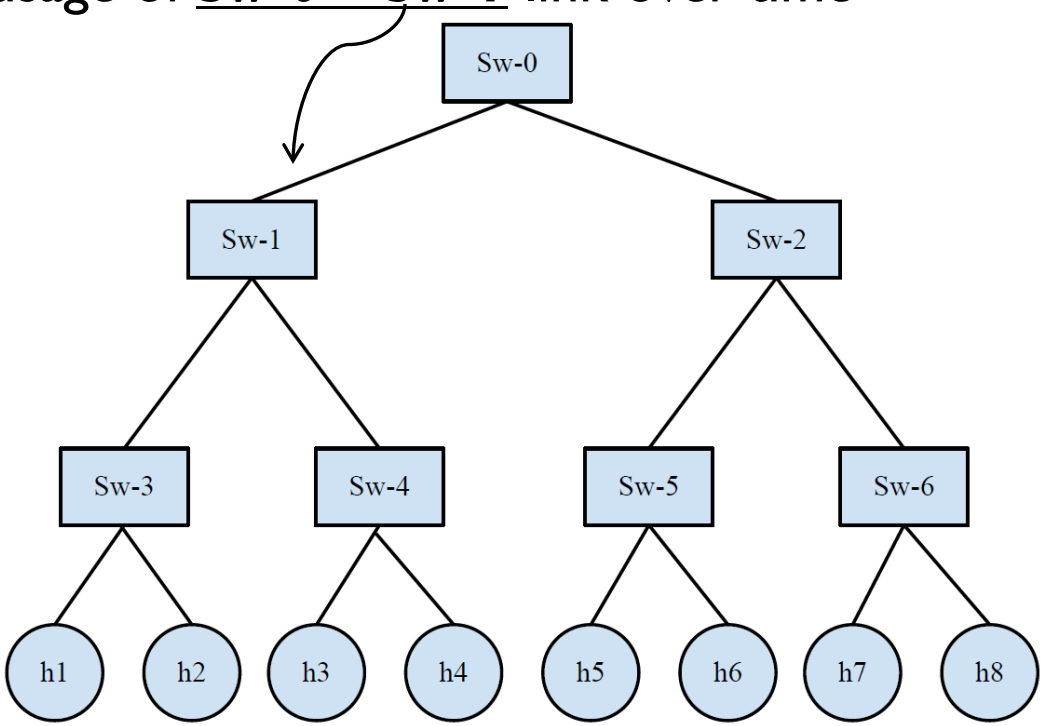


Implementation

- Application
  - Adaptive Link Usage Monitoring
- Scheduler
  - We propose an adaptive sampling algorithm
    - Adjust the monitoring frequency according to *network load*.
    - Assign a *monitoring time out* to each flow
    - Query the switch(es) for flow statistics *when timeout expires*
    - If no significant traffic change (<= α), increase the timeout (up to T_{max})
    - If change in traffic is significant (>= β), decrease the timeout (up to T_{min})
- Switch selector:
  - Query all the switches
- Aggregator
  - Aggregate data per link

Evaluation: Setup

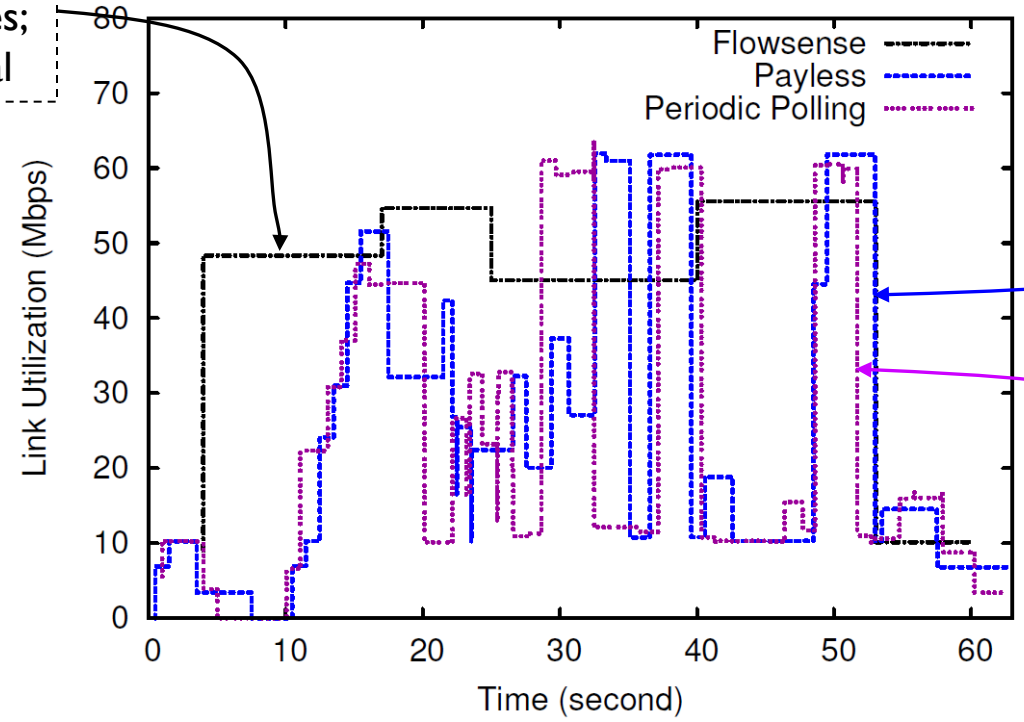
- Simulation using Mininet and Floodlight controller
- Topology
  - Hierarchical topology to emulate behavior of a scaled down data center
- α = β =100MB; T_{max} = *5s*, T_{min} = *500ms*
- Monitor the usage of ***Sw-0 – Sw-1*** link over time



Evaluation: Utilization

- Comparison with FlowSense*, and Periodic polling (every 500ms)

Misses the spikes;
Far from original



Closer to original;
Non-zero but less overhead

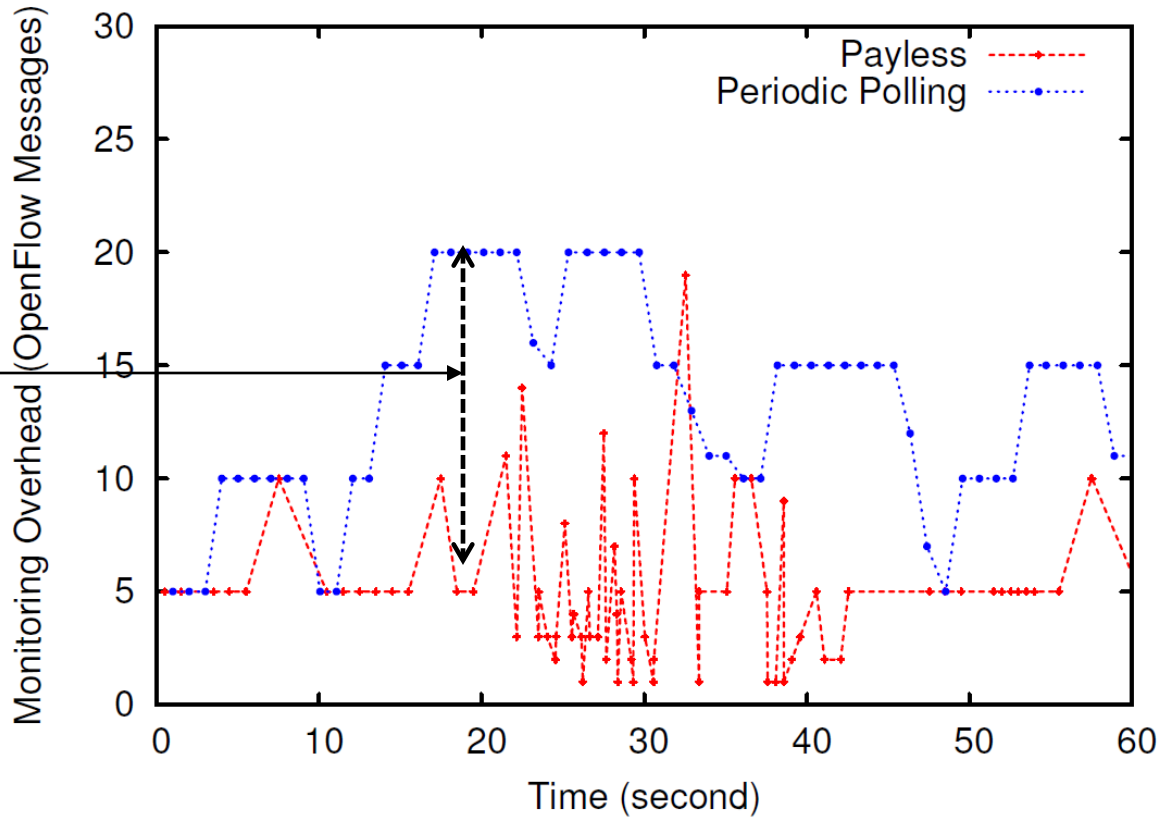
Closest to original;
More overhead

*Yu, C. *et al.* FlowSense: Monitoring Network Utilization with Zero Measurement Cost. *Passive and Active Monitoring (PAM) 2013*

Evaluation: Overhead

Overhead is measured by the number of monitoring queries.

PayLess can reduce the messaging overhead **up to 50%** compared to periodic polling.



Conclusion

- Summary
  - State-of-the art controllers offer different northbound APIs. We need an uniform API for network applications
  - Payless is a step to provide unified API for monitoring application development
- Future Works
  - Full fledged implementation
  - Develop a QoS policy enforcement application* over PayLess

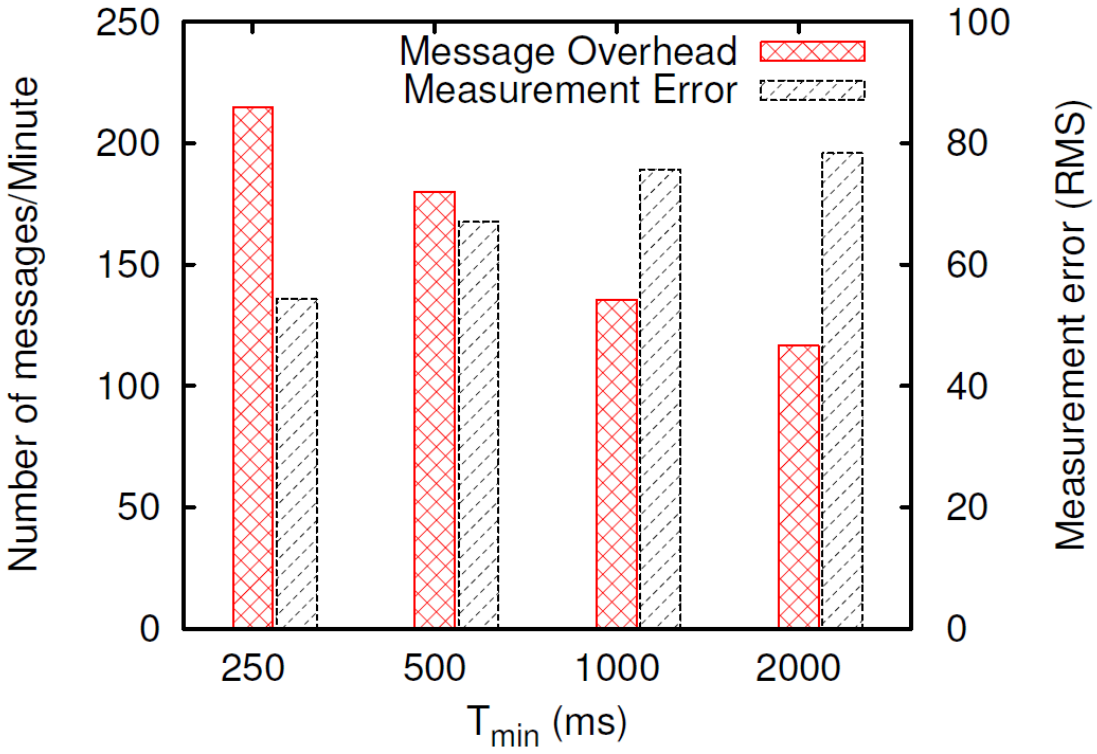
*Bari *et al.*, PolicyCop: An Autonomic QoS Policy Enforcement Framework for Software Defined Networks. *IEEE SDN4FNS 2014*

Questions

?

Evaluation: Effect of T_{min}

- T_{min} is the minimum polling timeout.
- T_{min} is varied ***(250ms – 2s)*** to observe its effect on accuracy and overhead
  - Accuracy was measured as rms error between PayLess and periodic polling over 250ms interval



Related Works

- OpenTM (PAM '10)
  - Heuristics on **which** OpenFlow switches to query for measuring traffic matrix
- FlowSense (PAM '13)
  - Event based link utilization monitoring in SDN.
  - No additional measurement overhead.
- OpenSketch (NSDI '13)
  - Clean slate redesign of data plane to support monitoring in SDN