OmniMon: Re-architecting Network Telemetry with Resource Efficiency and Full Accuracy

Qun Huang, Haifeng Sun, Patrick P. C. Lee
Wei Bai, Feng Zhu, Yungang Bao
Flow-level Network Telemetry

Flow Statistics

Flow 1
Pkt count ...........

Flow 2
Pkt count ...........

Flow 3
Pkt count ...........

...

End-hosts

Hardware Switches

Packet: (flowkey, packet values)
Goal

Flow Statistics

- Flow 1
  - Pkt count
- Flow 2
  - Pkt count
- Flow 3
  - Pkt count
  - ...

Controller

End-hosts

Hardware Switches

Full Accuracy

Resource Efficiency
Full Accuracy

Flow Statistics

<table>
<thead>
<tr>
<th>Flow 1</th>
<th>Flow 2</th>
<th>Flow 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pkt count</td>
<td>Pkt count</td>
<td>Pkt count</td>
</tr>
<tr>
<td>..........</td>
<td>..........</td>
<td>..........</td>
</tr>
<tr>
<td>..........</td>
<td>..........</td>
<td>..........</td>
</tr>
<tr>
<td>..........</td>
<td>..........</td>
<td>..........</td>
</tr>
</tbody>
</table>

1. Always-on: all time intervals
2. Network-wide: all devices
3. Complete: all flows
4. Correct: zero per-flow error
Resource Efficiency

Flow Statistics

<table>
<thead>
<tr>
<th>Flow 1</th>
<th>Pkt count</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow 2</td>
<td>Pkt count</td>
<td>...</td>
</tr>
<tr>
<td>Flow 3</td>
<td>Pkt count</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

1. Always-on: all time intervals
2. Network-wide: all devices
3. Complete: all flows
4. Correct: zero per-flow error

End-Hosts
- Sufficient memory 😊
- Programmability 😊
- Slow CPU 😞
- Limited visibility 😞

Controller
- Sufficient CPU and memory 😊
- Global visibility 😊
- Limited bandwidth 😞

Hardware Switches
- Fast ASIC 😊
- Limited memory 😞
- Limited programmability 😞
Existing Approaches: Trade-offs

- Resource Efficiency
- Full Accuracy
Existing Approaches: Trade-offs

- Resource Efficiency
  - Course-grained
  - SNMP
Existing Approaches: Trade-offs

- SNMP: Course-grained
- Hash tables: High Overheads

Axes:
- Resource Efficiency
- Full Accuracy
Existing Approaches: Trade-offs

- Resource Efficiency
- Full Accuracy

Course-grained
- SNMP
- Sampling
- Event Matching
- Top-k Counting

High Overheads
- Hash tables
Existing Approaches: Trade-offs

- Resource Efficiency vs. Full Accuracy

- Course-grained
  - SNMP
  - Sampling
  - Event Matching
  - Only Partial Flows

- Top-k Counting

- High Overheads
  - Hash tables
Existing Approaches: Trade-offs

- Full Accuracy
- Resource Efficiency

**Course-grained**
- SNMP
- Event Matching
- Only Partial Flows

**Top-k Counting**
- Sampling

**Sketch Approximate Results**
- Hash tables
- High Overheads

- Approximate Results
- High Overheads
Existing Approaches: Trade-offs

- Resource Efficiency
- Full Accuracy
- High Overheads

- Course-grained
- Only Partial Flows
- SNMP
- Sampling

- Event Matching

- Top-k Counting

- Sketch
  - Approximate Results

- Our Goal
Operators are executed individually with limited collaboration
Operators are executed individually with limited collaboration

Operators have to be heavy and sacrifice accuracy
Operators are executed individually with limited collaboration.

Operators have to be heavy and sacrifice accuracy.
Re-architect network telemetry by distributed design

Question 1
Coordinate different entities for network telemetry?

Question 2
Reliable guarantees for the coordination?
Re-architect network telemetry by distributed design

Question 1
Coordinate different entities for network telemetry?

Question 2
Reliable guarantees for the coordination?
Split-and-Merge Architecture

Flowkey Tracking
Value Updating
Resource Management
Collective Analysis

Network telemetry
Break heavy operators

Controller

Network-wide coordination
Flowkey Tracking

Network telemetry

Flowkey Tracking
Value Updating
Resource Management
Collective Analysis

Controller

Flowkeys

Flowkeys
Value Update

Network telemetry

Flowkey Tracking  Value Updating  Resource Management  Collective Analysis

Flowkeys

Flowkeys

Controller
Mapping at End-Host

Network telemetry

Flowkey Tracking | Value Updating | Resource Management | Collective Analysis

Controller

Different strategies of slot mapping in end-hosts and switches
Mapping at End-Host (Egress)

Network telemetry

Flowkey Tracking
Value Updating
Resource Management
Collective Analysis

Controller

Slots (Ingress) Flowkeys Slots (Egress)

Slots (Ingress) Flowkeys Slots (Egress)
Mapping at End-Host (Ingress)

Network telemetry

Flowkey Tracking | Value Updating | Resource Management | Collective Analysis

Slots (Ingress) Flowkeys Slots (Egress) Slots (Ingress) Flowkeys Slots (Egress)

No Flowkeys No Flowkeys

Controller

Packet

1. Embed Location
2. Locate Slot
Mapping at Switch

Network telemetry

1. Global Coordination
2. Embed Index
3. Extract & Update

Flowkey Tracking | Value Updating | Resource Management | Collective Analysis
Collective Analysis

Network telemetry

Flowkey Tracking  Value Updating  Resource Management  Collective Analysis

Collect results from end-host and switches to form final flow statistics
Collective Analysis

Network telemetry

Flowkey Tracking | Value Updating | Resource Management | Collective Analysis

Slots (Ingress) Flowkeys Slots (Egress)

Controller

Slots (Ingress) Flowkeys Slots (Egress)

Exploit end-host information to decompose switch slots
Collective Analysis (Detail)

Source End-Host

Flowkeys | Slots (Egress)
--- | ---
Flow 1 | 13

Flowkeys | Slots (Egress)
--- | ---
Flow 2 | 11

Switch Index

Switch

Flow 1: 13

Flow 2: 11
Putting It Together

Network telemetry

Flowkey Tracking  Value Updating  Resource Management  Collective Analysis

End-Hosts: Hash Table
Exact Per-flow Tracking 😊
Affordable Operations 😊

Switches: Shared Slots
Low Memory Usage 😊
Simple Updates 😊

Controller: Global Info.
Switch Index Mapping 😊
Collective Analysis 😊
Re-architect network telemetry by distributed design

Question 1
Coordinate different entities for network telemetry?

Question 2
Reliable guarantees for the coordination?
Unreliable Events

- Lack of global clock
  - Devices reside in different intervals

- Packet loss
  - Flow values are missing in some devices
Impact of Unreliable Events

Flow 1: 13
Flow 2: 11

Long Delay
Packet Loss

Recorded by Other Intervals
Not Recorded

Expect: 24
Impact of Unreliable Events

Flow 1: ???
Flow 2: ???

Long Delay
Packet Loss

Recorded by Other Intervals
Not Recorded

Expect: 24
Reliability Guarantees

Lack of global clock

- Hybrid consistency model
  - Network-wide synchronization

- Guarantees
  - Each packet is included in the same intervals by all devices
  - All end-hosts reside in the same time intervals in most time

Packet Loss

- Loss inference
  - Linear system with DCN-specific optimizations
  - Flow mapping algorithm

- Guarantees
  - Per-switch, per-flow loss interference in common cases

More details in the paper
Implementation

- Testbed
  - End-hosts: DPDK
  - Switch: P4
  - Controller: C++

- Simulator: 8-ray fat-tree
Host Overheads

- <10% overheads when adding telemetry functionalities to PktGen
- Hash lookup dominates the overheads
Switch Overheads

- Less resources than sketch-based techniques
- Only OmniMon achieves zero errors

**Sketch techniques**
- FR: FlowRadar (NSDI 16)
- UM: OmniMon (SIGCOMM 16)
- ES: Elastic Sketch (SIGCOMM 18)
- SL: SketchLearn (SIGCOMM 18)

Each sketch only monitors packet count

**OmniMon**
- OS: OmniMon that monitors only packet count
- OF: OmniMon that monitors 9 statistics
More Results

- Controller overheads
- Synchronization efficiency
- Accountability
- Scalability
- User case: anomaly detection
- Use case: network failure diagnosis
- Use case: load balance evaluation
Conclusion

- OmniMon architecture: split-and-merge design
  - Four partial operations
  - Network-wide coordination

- Consistency guarantee
  - Network-wide synchronization with hybrid consistency model

- Accountability guarantee
  - Packet loss inference with linear systems
  - Flow mapping algorithm

- Prototype: DPDK + P4

- Results: compare with 11 state-of-the-art solutions in various aspects

Source Code Available: https://github.com/N2-Sys/Omnimon