Enabling Tenant-driven Network Debugging with Partial Packet Provenance

Alexander Wolosewicz (IL Tech), Raffay Atiq (SRI), Nishanth Shyamkumar (IL Tech), Vinod Yegneswaran (SRI), Ashish Gehani (SRI), Nik Sultana (IL Tech)

ILLINOIS TECH
College of Computing

Short Bio

- 2nd year Ph.D. student
- Bachelor's in CS
- Working on this project since ~September 2024.

ILLINOIS TECH
College of Computing

Introduction

- Virtualization is present in many modern networks
  - Cloud, campus, telecom, testbed
- Problems can manifest both in the virtual (tenant) network and in the physical network (underlay)
- When an issue occurs, vantage points enable diagnosing and debugging
  - Network operators have many vantage points
  - Tenants have significantly less

Problem

- Tenants encounter issues in their virtual networks
  - May be self-induced, may be virtual bug, may be underlay
- Limited vantage point is an impediment
- When in doubt, escalate to operator
  - Operator support is a limited resource
- **Motivation:** improving tenant visibility conserves operator time
- **Research Question:** How can tenants diagnose/debug without operator support, preserving virtualization and permissions?

ILLINOIS TECH
College of Computing

Stargate

- A debugging architecture for tenants which is flexible and general
  - No reliance on specific hardware or protocols
  - Provides comprehensive network debugging
- Formal design for reasoning about network faults with per-packet provenance
- Working, user-evaluated implementation run on FABRIC
  - Virtual network testbed we do not operate

ILLINOIS TECH
College of Computing

Network Problems

- Packet loss and reachability
- Performance anomalies
- Correct forwarding, loop detection, path conformance
- Packet provenance

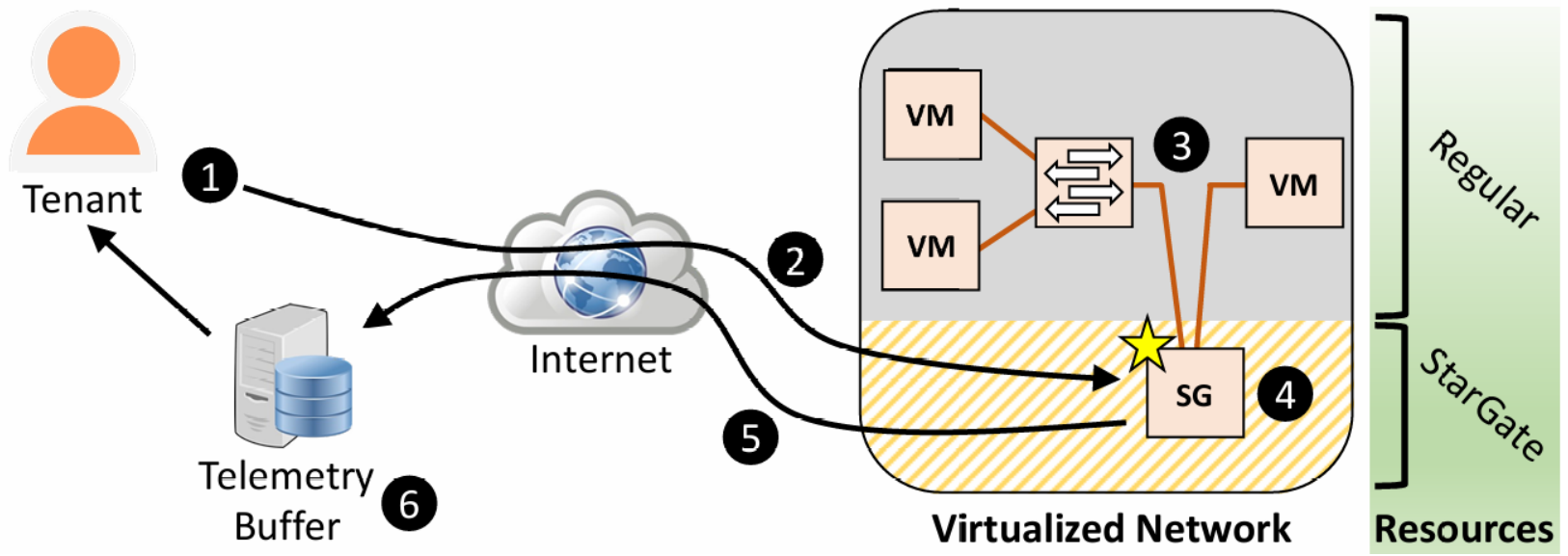
Debugging Requirements

- Secure and private: a tenant's debugger is contained to the tenant
- Modification-free: a debugger functions on production traffic without modification to workloads, topology, VM images, software
- Scalable: a debugger scales to large networks and concurrent tenant usage
- Flexible: a debugger covers a full range of network problems
- Measurement-capable: can report on the performance (and culpability) of the underlay

ILLINOIS TECH
College of Computing

Design of Stargate

1. Tenants access their resources over the internet
2. Tenants use an API to create, manage, query SG
3. SG nodes are placed to bisect virtual links
4. SG nodes are custom lightweight VMs that capture information needed for debugging
5. Debugging information is streamed to a telemetry buffer
6. The buffer is queried by the tenant



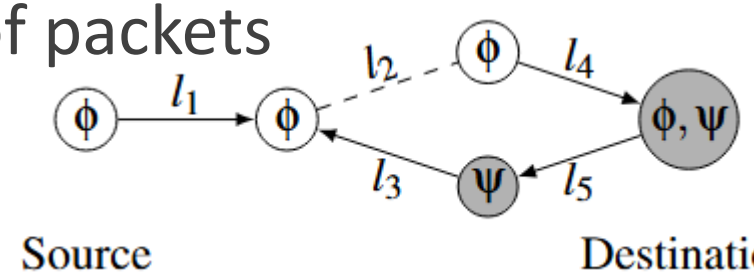
ILLINOIS TECH
College of Computing

Debugging API

- Create: specify two virtual nodes to place a SG node between
- Manage: control filtering of captured information
- Query: retrieve provenance output filtered on specific criteria

Formalism

- Partial Packet Provenance (3P): histories of packets
  - Partial – does not require full visibility
- Relation P: frames (ϕ) over links (l)
- Reach: where a ϕ is present
- Repeat: same ϕ and l, different time
- Duplicate: ϕ leaves more times on egress links L' than it entered on l.



$$\text{Reach}(\phi, l) \stackrel{def}{=} \{l : (\_, \_, l) \in \text{hist}(\phi)\}$$

$$\text{Repeat}(\phi, l) \stackrel{def}{=} |\text{Reach}(\phi, l)| > 0$$

$$\text{Duplicate}(\phi, l, L') \stackrel{def}{=} |\text{Reach}(\phi, l)| < \Sigma_{l' \in L'} |\text{Reach}(\phi, l')|$$

ILLINOIS TECH
College of Computing

Debugging with 3P Primitives

- Reach covers drops (negation) as well as intended and unintended reachability; packet goes where it should, and not where it shouldn't
- Reach can reveal loops (all instances where duplication not present)
- Duplication strengthens the two to handle the edge cases frame duplication can introduce

ILLINOIS TECH
College of Computing

Implementation

- Virtual network: the FABRIC national networking testbed
- API: Python 3 extension of FABRIC's API, Fablib
- SG nodes: Ubuntu 22 VMs running a C application using DPDK
- Buffer uses SPADE as the provenance recorder and database manager (PostgreSQL backend)

ILLINOIS TECH
College of Computing

SG Nodes

- Receives frame, checks if it has been tagged, tags if not
  - In-band filtering determines which packets get this treatment – those not meeting the filter are just forwarded
- Forwards frame (same order as received – goal is transparency)
- Sends frame ID, node info, time, and frame metadata to buffer

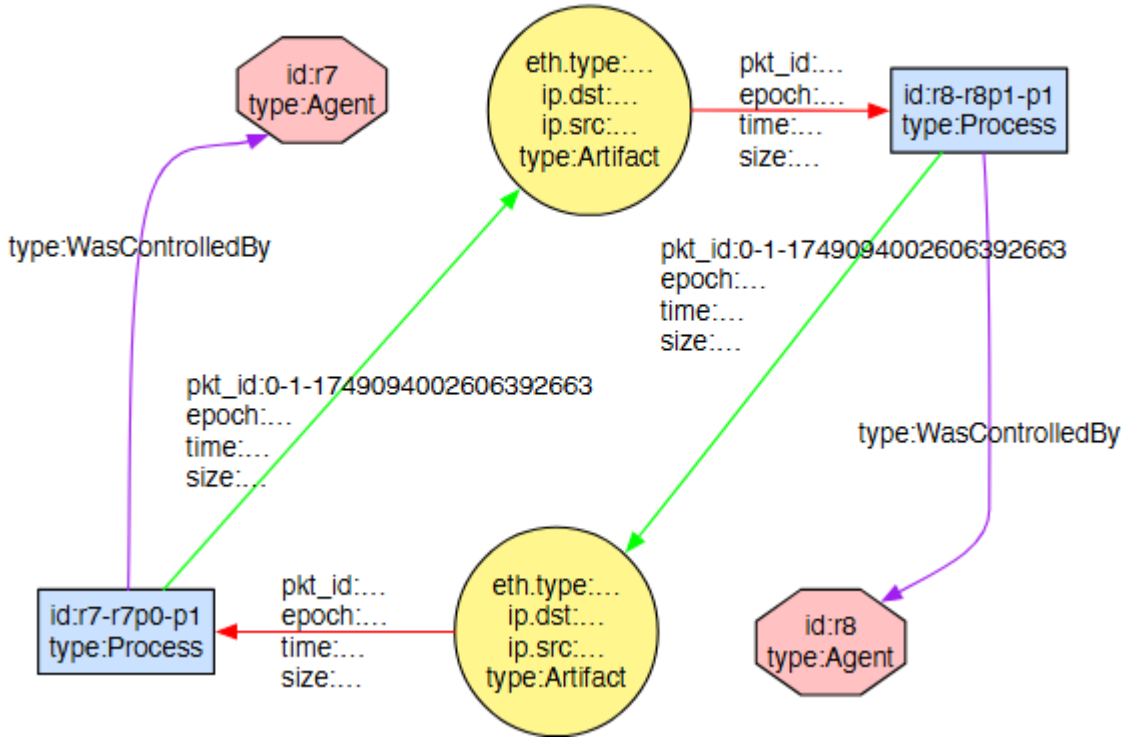
Frame Trailers

- Per the design, need to uniquely identify frames while maintaining the modification-free property
- Frame trailers are invisible to process not actively looking for them – no length increment, so appear as padding
- 16 bytes:
  TimeStamp (64 bits) | Node (8) | Port (8) | TS lower 8 (8) | 0x6587
- Limitation: must avoid fragmentation

ILLINOIS TECH
College of Computing

Provenance

- Conforms to Open Provenance Model, adapted to networking
- With frame IDs, shows where it went, when, and what it was
- Sufficient information to debug a complete set of problems



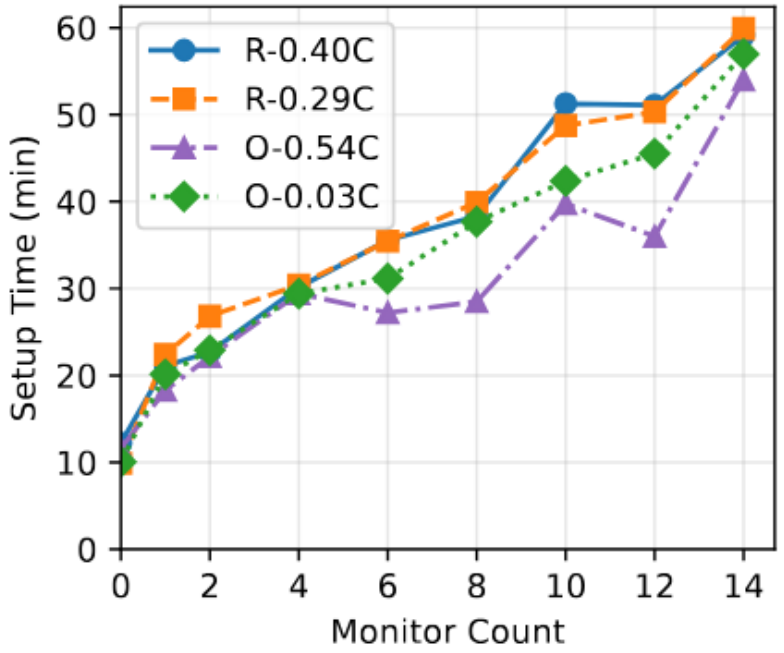
ILLINOIS TECH
College of Computing

Evaluation

- SG nodes use DPDK version 25.07 (latest at the time), 4 cores (2.4 GHz), 1 GB RAM, Mellanox Connect-X 6 SR-IOV virtual function NICs
- Buffer: 4 cores, 16 GB RAM (for DB queries)
- Fablib version 1.9.2

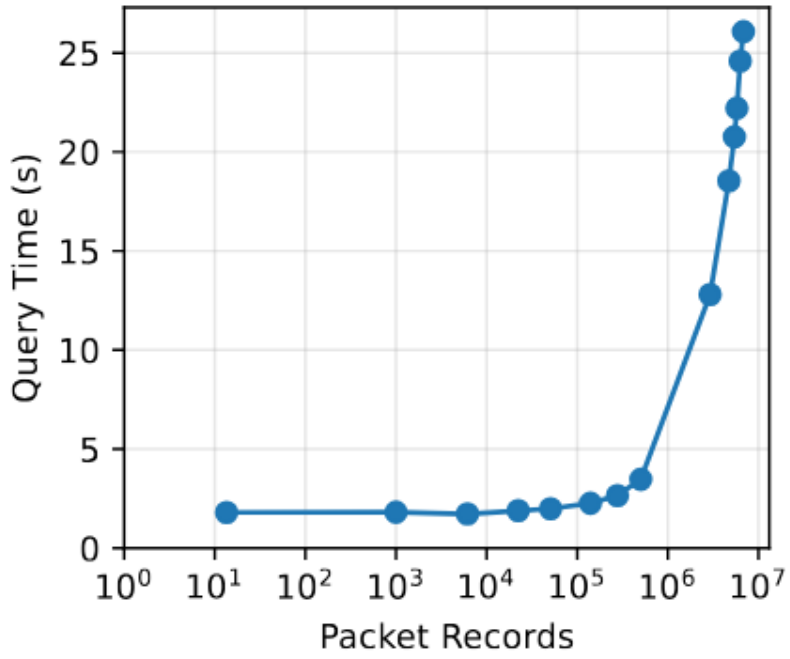
ILLINOIS TECH
College of Computing

Scalability



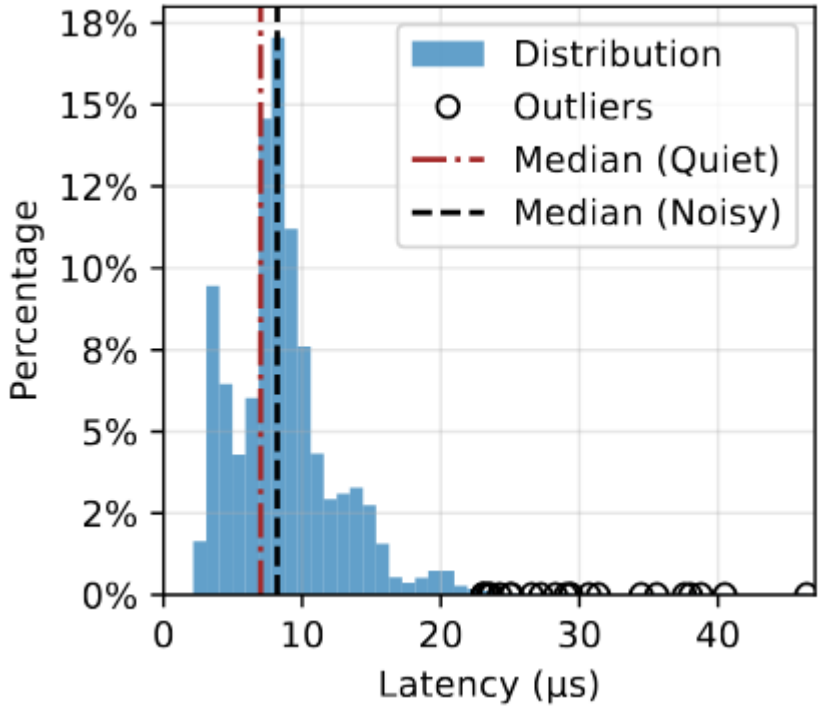
Adding SG nodes to 11-node, 14-link slice on 4 sites
R: on terabit ring vs Off-ring (size of site), C: ratio avail. CPU (use of site)

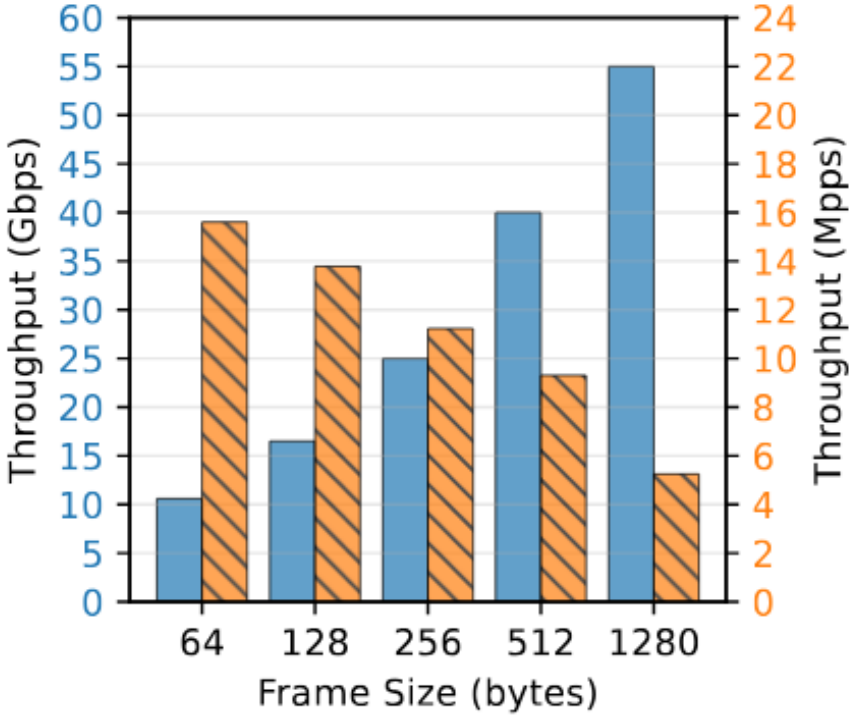
Querying provenance database of 5-hop packet histories for one specific history (5 records/history)



ILLINOIS TECH
College of Computing

In-Band Performance





Left: Latency distribution in quiet environment (6,692 samples)
Right: SG node throughput at varying frame sizes

ILLINOIS TECH
College of Computing

Trailer Preservation

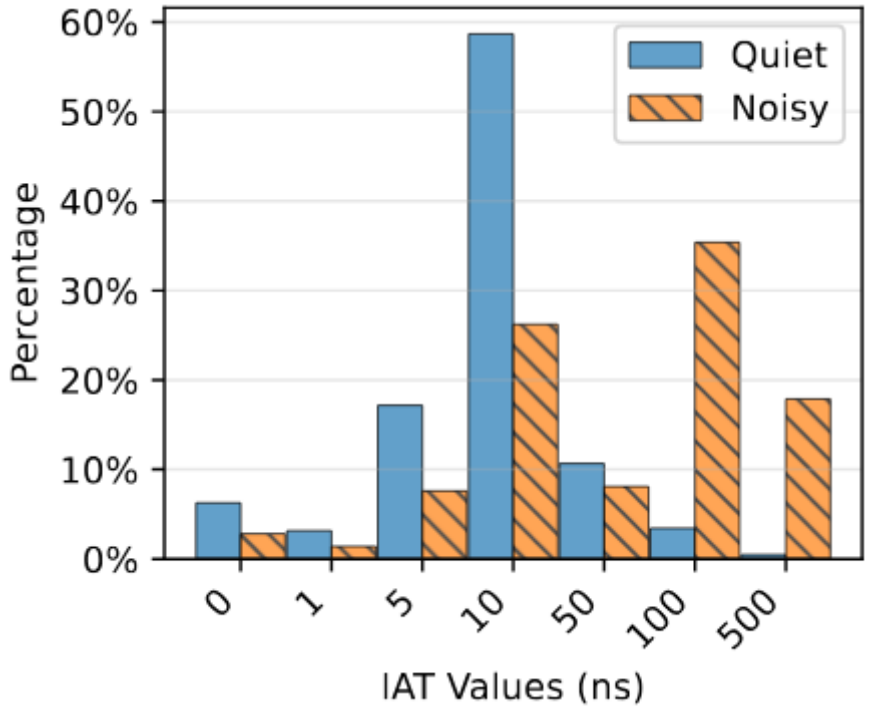
- Preserved in many contexts
- Linux/BSD IP forwarding rely explicitly on the IP length field
- Linux 5.15.0-143-generic
- BSD 14.0-RELEASE-p6
- OpenVSwitch 2.17.9

Element	Preserved	Element	Preserved
Linux Bridge	●	FABRIC L2	●
Linux IP Fwd	○	FABRIC L3	●
BSD Bridge	●	FABRIC Fac. Port	●
BSD IP Fwd	○	OpenVSwitch	●
Alveo-to-NIC	●	Tofino2	●

ILLINOIS TECH
College of Computing

Insight into Underlay

- Added traffic replaying to SG nodes
- Replay the same traffic multiple times to build a baseline of network consistency
- Experiment reveals impacts from high co-tenant traffic (overall less than link capacity)



ILLINOIS TECH
College of Computing

User Evaluation

- 11 outside users used SG to debug some networking issues while 11 used common tools like ping and tcpdump
- Those with medium FABRIC and network debugging experience could debug up to 37.5% faster on the second exercise, with only a 15-minute demo video and a previous exercise (two of three solved with a single API call)
- Small sample size, but shows potential

ILLINOIS TECH
College of Computing

Conclusions

- Stargate provides a highly-general, highly-capable debugging tool for virtual networks
- Readily available, used by outsiders with success

ILLINOIS TECH
College of Computing

Future Work

- Automation: SG node placement, deciding traffic filtering criteria, refining queries from more general prompts
- Capability: boosting monitor throughput through load-balancing. Requires work in coordinating the monitors to ensure overall network consistency
- Network consistency: expanding and refining metrics to quantify how consistent the network (visibility into underlay) is.

ILLINOIS TECH
College of Computing