Contents

[IPCCC-2021]PUFF: A passive and universal learning-based framework for intra-domain failure detection

Lianjin Ye § *, Qing Li ‡ *, Xudong Zuo § , Jingyu Xiao, Yong Jiang § *,
Zhuyun Qi*, Chunsheng Zhu*

§ Tsinghua Shenzhen International Graduate School, Shenzhen, China
‡ Southern University of Science and Technology, Shenzhen, China
*Pengcheng Laboratory, Shenzhen, China

Background

- The rapid development of network applications puts forward higher requirements for network reliability.
- Network failures are inevitable and occur more frequently.
- Network fault detection methods based on probe packets are facing bottlenecks.

Existing Tools

- Active detection based on probe packet connection
  - ◆ OSPF
  - ◆ BFD
- Active detection based on packet statistics
  - ◆ PingMesh
  - ◆ NetBouncer
- Passive detection based on indicators
  - ◆ Netpoirot
- Log-based passive detection
  - ◆ Prefix

Table I
COMPARING PUFF WITH EXISTING METHODS OF NETWORK FAILURE DETECTION

	Type	Failure Coverage	Sampling Period	Bottleneck
BFD	proactive	general	ms	bandwidth
Pingmesh	proactive	data center	10s	storage
007	passive	data center	s	deployment
ML-LFIL	passive	general	s	collection
PUFF	passive	general	ms	-

Active detection Bottleneck: The bandwidth and storage of the detection packet require more overhead, and the scene is single.
Passive detection Bottleneck: equipment support is required, end-side deployment requires additional support

Passive detection on programmable switch

- **Passive detection based on programmable switch and Machine Learning**

◆ Design idea: Based on the programmable switch, the data collection task is transferred from the end side to the switch side.

◆ Low Overhead: Customized hardware and software design reduces passive detection overhead.

◆ Network IntelliSense: A machine learning method based on in-network data to perceive network status.

Outline

- PUFF motivation
- PUFF design
- PUFF Implementation
- Evaluation

Motivation

- Comprehensive and in-network packet history helps locate the malfunctions.
- Continuous changes in traffic of TCP reflect network failure without resource-consuming end-to-end metrics

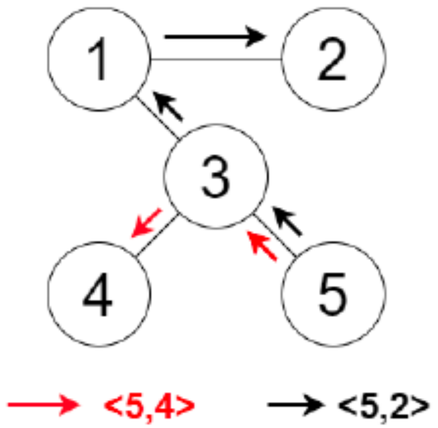


Figure 1. Using packet history to locate failures

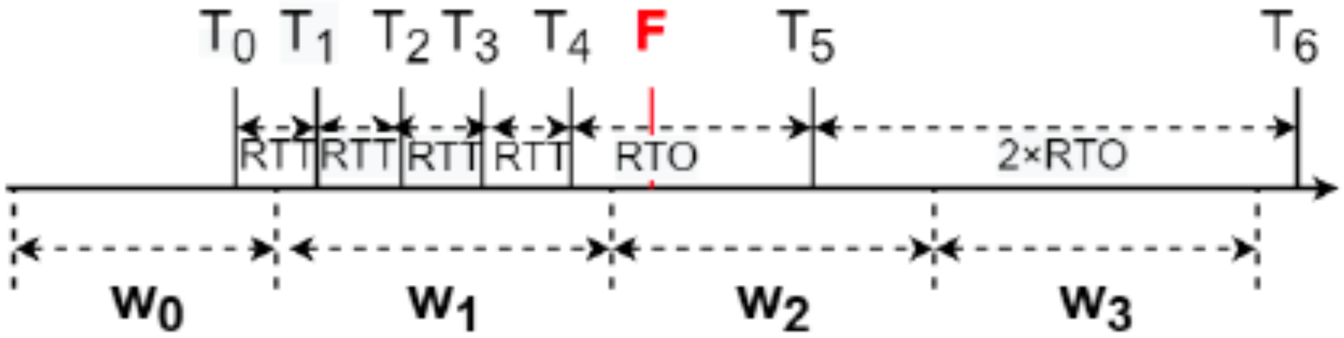


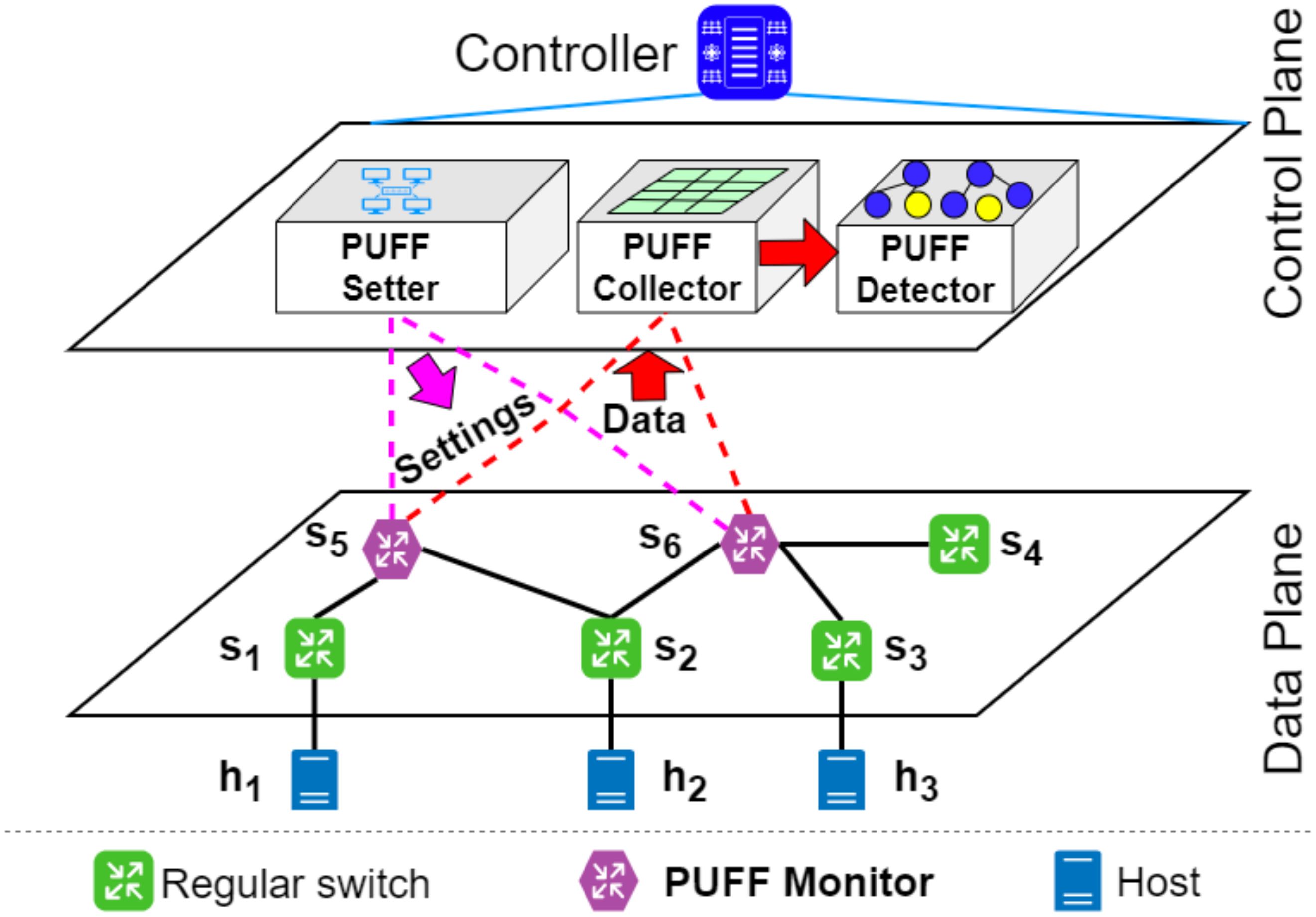
Figure 2. Example of a given TCP stream when one node is down

Outline

- PUFF motivation
- PUFF design
- PUFF Implementation
- Evaluation

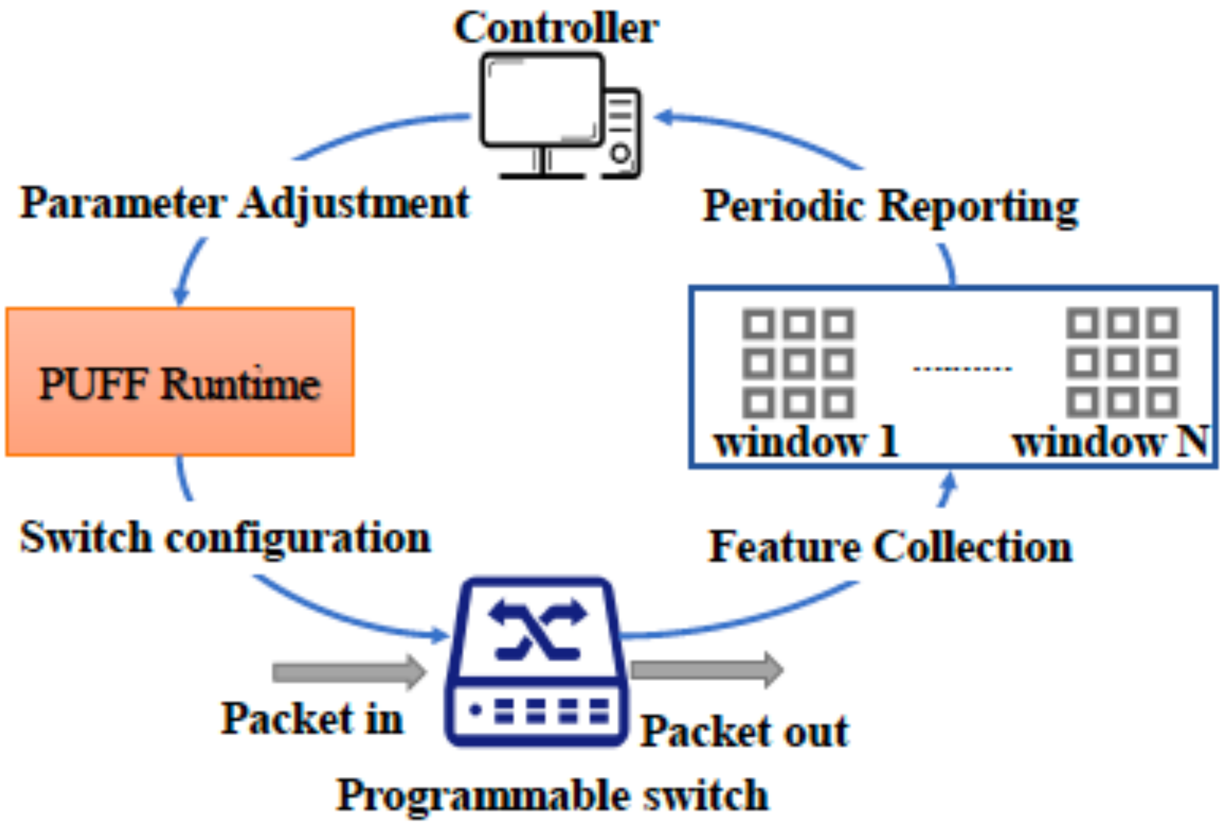
Overview

- Control plane
  - PUFF Setter
  - PUFF Collector
  - PUFF Detector
- Data plane
  - PUFF Monitor



Implementation

- PUFF Controller
  - ◆ DPTP Ryu controller
- PUFF Runtime
  - ◆ P4 code and parameter distribution.
- PUFF Data plane.
  - ◆ P4 implementation
  - ◆ DPTP for time synchronization

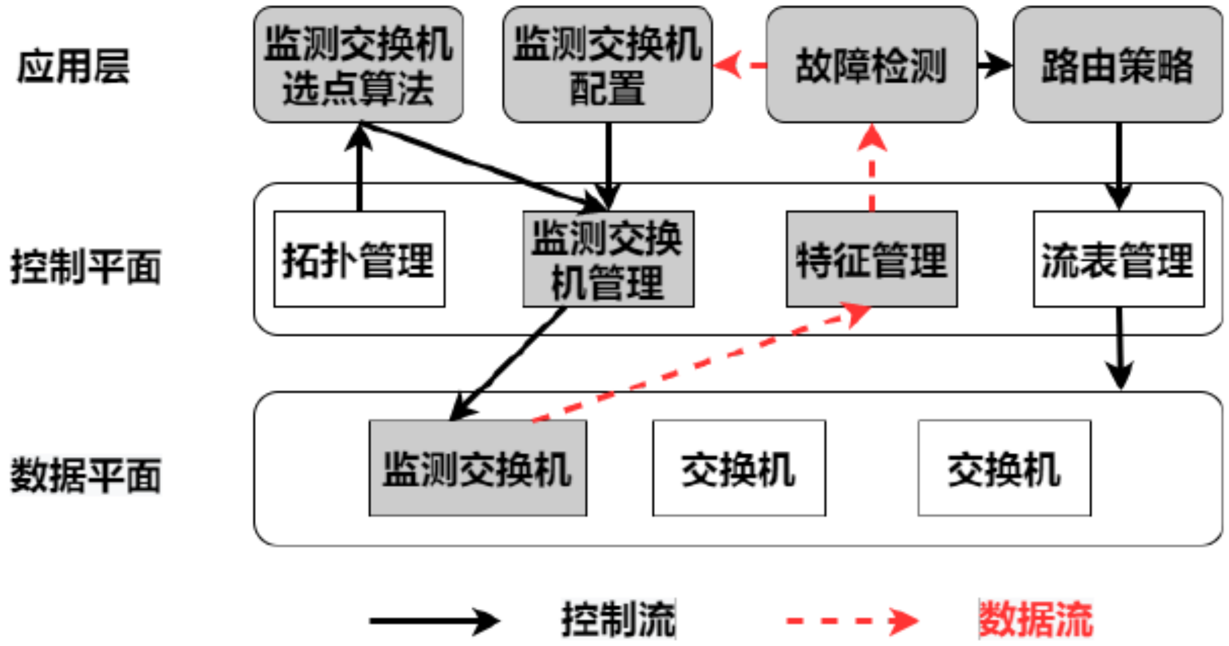


Outline

- PUFF motivation
- PUFF design
- PUFF Implementation
- Evaluation

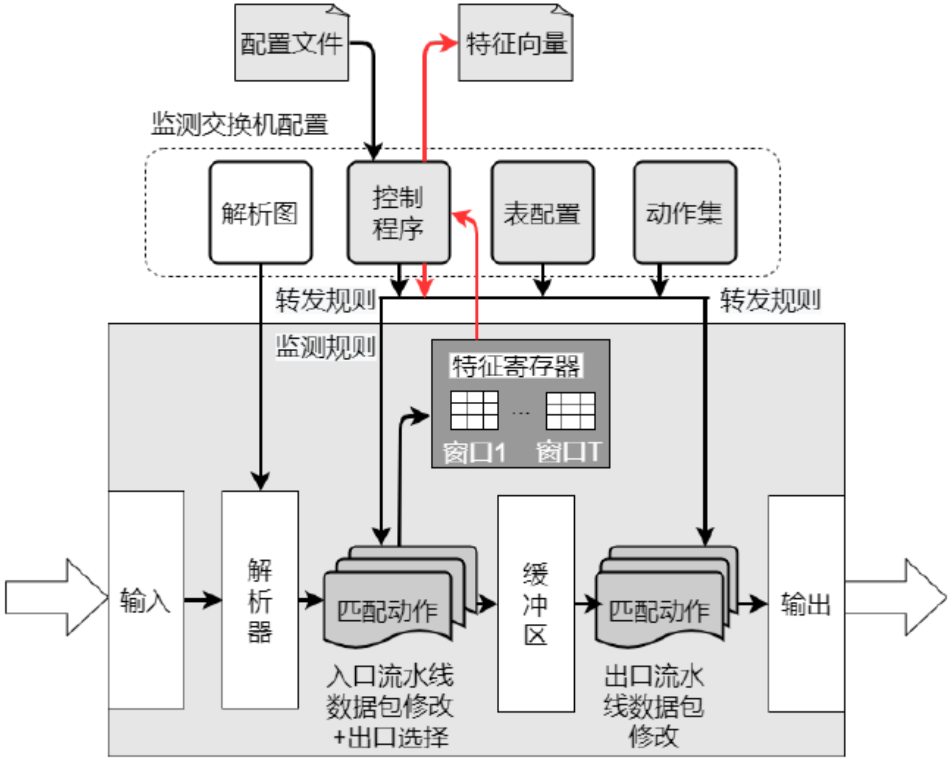
PUFF Control plane

- Monitor Selection Module
  - ◆ Generate monitoring switch deployment nodes by reading topology management.
- Monitor configuration module
  - ◆ The best hyperparameters (time window size, number of time windows) in the topology are generated through integrated learning of tagged fault detection modules.
- Monitor management:
  - ◆ Deploy monitoring switch settings



PUFF Data plane

- Control program
  - Set the number of observation data packets and characteristic registers by reading the issued configuration file.When receiving the control plane access, read the register data and return.
- Feature register
  - Store the corresponding characteristics.
- Matching action
  - Store eligible data packets



Detection Algorithm

- Feature Extractor
  - Link Feature
  - Node Feature
- Link Classifier
  - Learning-Based Model.
- Node Classifier
  - Threshold method.

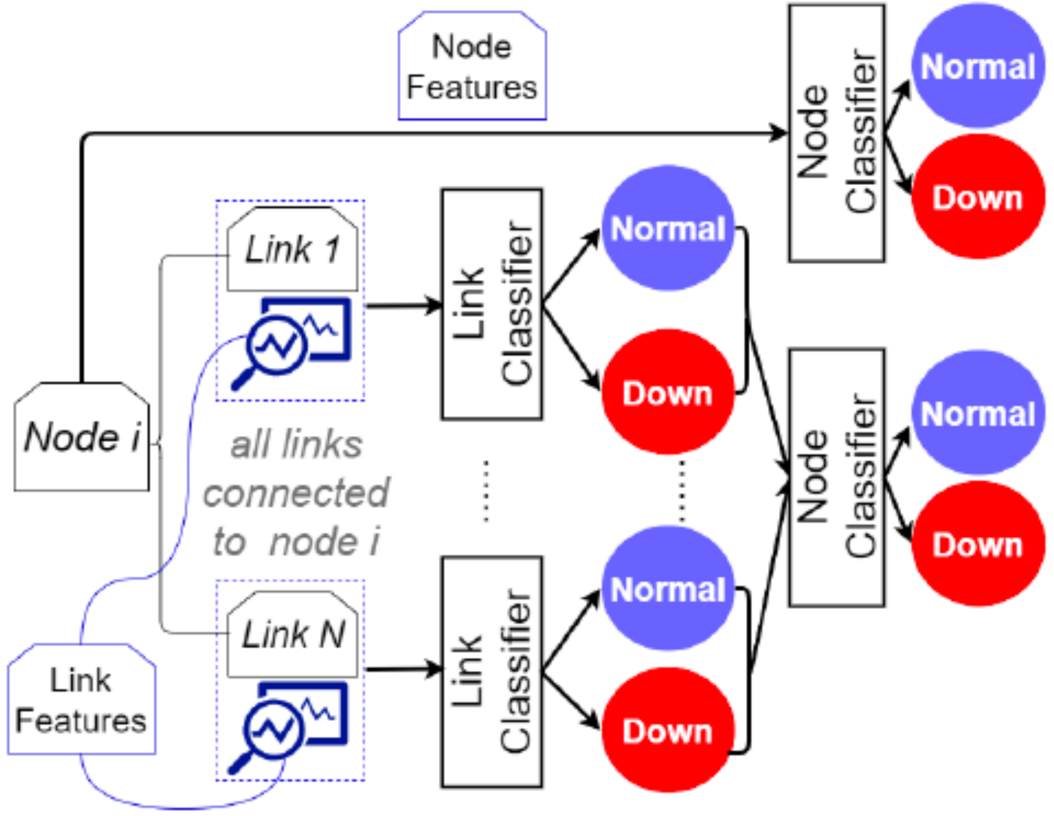


Figure 4. Failure detection model of nodes and links

Outline

- PUFF motivation
- PUFF design
- PUFF Implementation
- Evaluation

Evaluation

- Evaluation of monitor deployment

Table III
TOPOLOGY SETTINGS

Topology	Node	Link	RTT Median
GEANT	40	61	21ms
Tinet	53	88	72ms
AS1221	104	306	28ms

Table IV
COVER FLOW INDEX

	Monitor Counts							
Topology	1	2	3	4	5	6	7	8
GEANT	67.1	85.6	87.6	89.4	89.4	90.4	90.4	90.4
Tinet	64.5	74.6	76.3	76.7	80.4	81.7	82.2	88.0
AS1221	15.8	21.6	26.9	31.6	33.2	36.2	38.5	40.6

Evaluation

- Evaluation of two-stage feature design
  - ◆ Evaluation of feature design

Table VI
EXAMPLE OF FEATURE IN GEANT WHEN W=105MS

Node Type	Broken Node			Normal Node		
Position	-1	0	+1	-1	0	+1
ts_t^i	34.6	25.7	23.9	128.8	149.2	154.3
td_t^i	39.5	33.0	30.0	140.5	156.3	162.5
bs_t^i	1.8	1.7	2.8	6.3	5.2	3.9
bd_t^i	0.7	4.8	6.2	7.7	5.2	3.8
as_t^i	7.8	4.9	6.3	3.4	2.7	2.8
ad_t^i	6.6	7.9	2.6	2.4	2.8	4.7

Table VII
EXAMPLE OF FEATURE IN GEANT WHEN W=210MS

Node Type	Broken Node			Normal Node		
Position	-1	0	+1	-1	0	+1
ts_t^i	68.1	120.4	37.1	194.3	315.9	394.9
td_t^i	69.6	117.4	41.6	209.2	342.5	436.9
bs_t^i	34.9	10.8	2.4	57.5	35.8	12.3
bd_t^i	32.0	13.4	3.2	60.9	36.9	12.5
as_t^i	4.8	53.3	22.8	2.1	5.5	2.6
ad_t^i	2.6	10.3	8.1	1.4	2.5	2.7

Evaluation

- Evaluation of two-stage feature design
  - Evaluation of classifier in detection algorithm

Table VIII
COMPARISON OF MACHINE LEARNING ALGORITHMS

Machine Learning Method	Failures	F1-score	Time Per link (in μs)
Logistic Regression	Link	0.75	2.5
SVM	Link	0.80	808
GBDT	Link	0.81	4.4
Random Forest	Link	0.79	12.3
Logistic Regression	Node	0.71	1.3
SVM	Node	0.63	305
GBDT	Node	0.76	7.8
Random Forest	Node	0.73	8.9

Evaluation

- Evaluation of link failure detection
  - Analysis of parameters

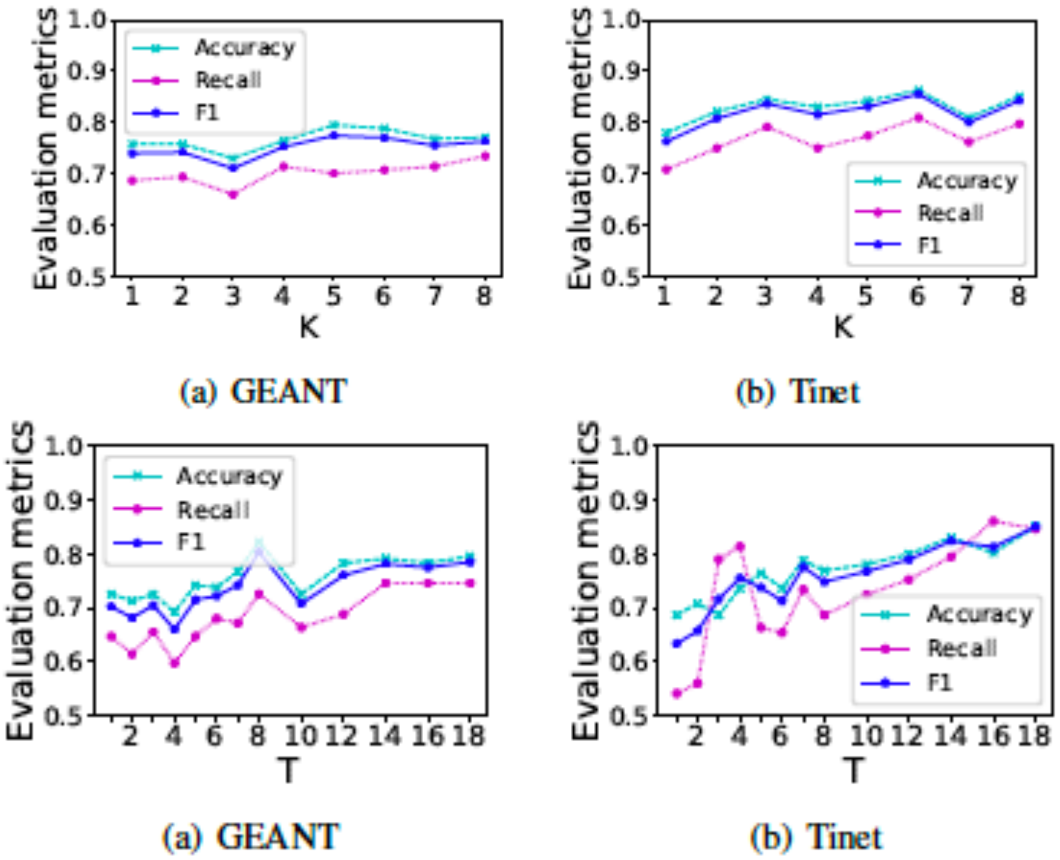


Figure 8. Effects of the amount of time windows

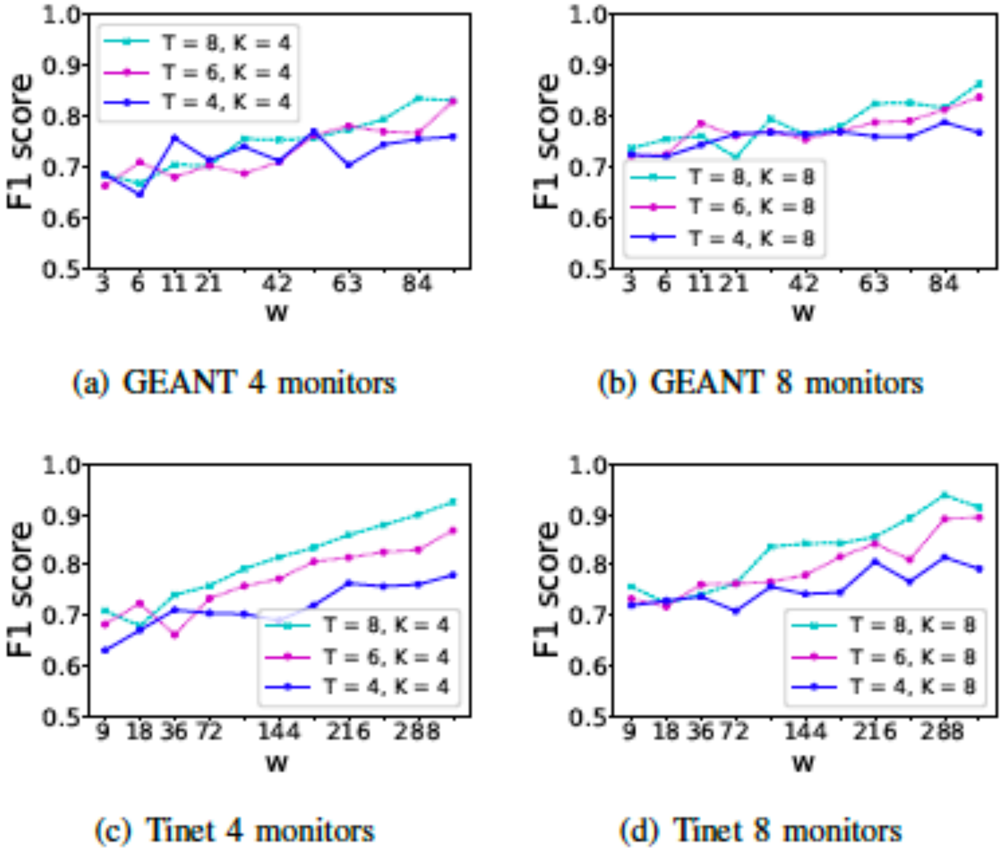
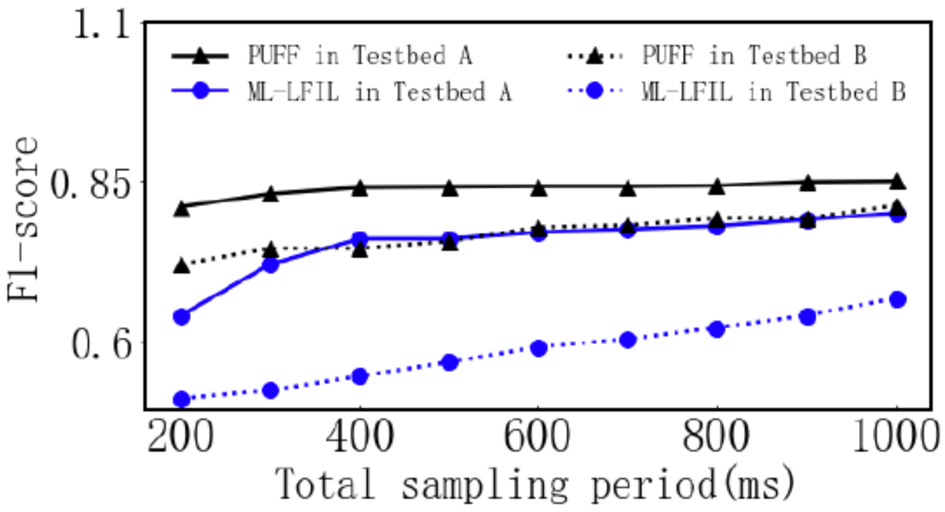


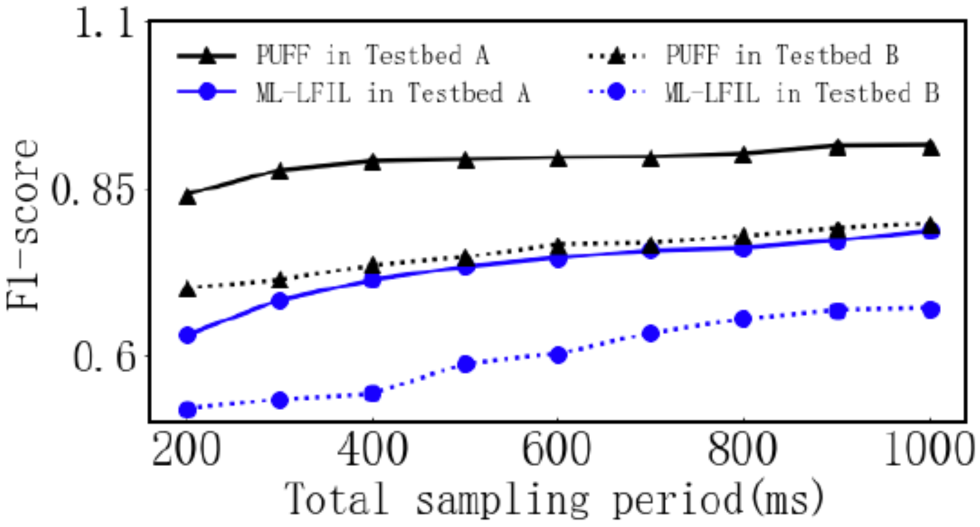
Figure 7. Effects of the size of time window

Evaluation

- Evaluation of link failure detection
  - ◆ Comparison with end-to-end passive detection



(a) GEANT



(b) Tinet

Evaluation

- Evaluation of node failure detection
  - Analysis of threshold

Table IX
PROPORTION OF LINKS CONNECTED TO NORMAL NODE BEING LABELED AS BROKEN LINK

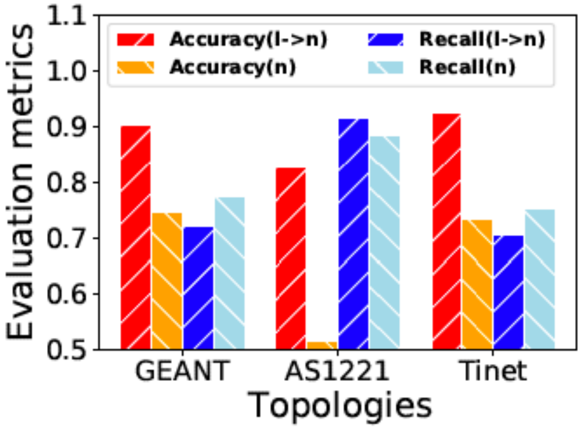
Node Type		Normal Node			
Topology	w(ms)	[0,0.25]	(0.25,0.5]	(0.5,0.75]	[0.75,1]
GEANT	21	0.8	0.1	0.02	0.08
GEANT	42	0.77	0.1	0.03	0.1
Tinet	72	0.75	0.13	0.05	0.07
Tinet	144	0.82	0.10	0.02	0.05
AS1221	28	0.70	0.10	0.03	0.17
AS1221	56	0.73	0.10	0.03	0.14

Table X
PROPORTION OF LINKS CONNECTED TO BROKEN NODE BEING LABELD AS BROKEN LINK

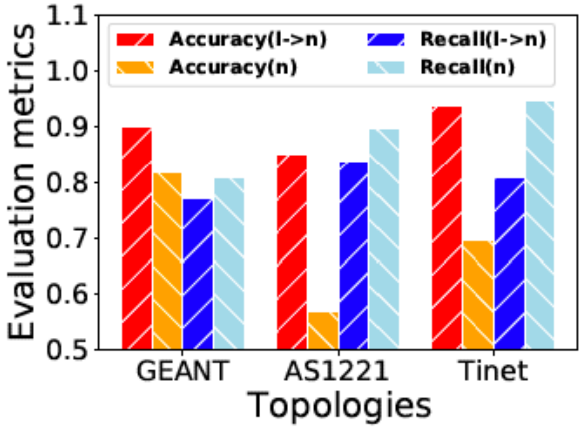
Node Type		Broken Node			
Topology	w(ms)	[0,0.25]	(0.25,0.5]	(0.5,0.75]	(0.75,1]
GEANT	21	0.23	0.04	0.01	0.72
GEANT	42	0.19	0.03	0.01	0.77
Tinet	72	0.22	0.04	0.03	0.71
Tinet	144	0.15	0.02	0.03	0.80
AS1221	28	0.07	0.01	0.01	0.91
AS1221	56	0.10	0.05	0.01	0.84

Evaluation

- Evaluation of node failure detection
  - Comparison in accuracy and failure localization time



(a) $w = 1 \times RTT_{median}$,T=8



(b) $w = 2 \times RTT_{median}$,T=8

Figure 10. Results of Node Failure Detection

Table XI
COMPARISON OF FAULT LOCALIZATION TIME

Methods	Task	Time (in μs)
Ping-based approach	link failures	1638000
ML-LFIL	link failures	202
PUFF	link failures	224
Ping-based approach	node failures	114500
PUFF	node failures	249

Evaluation

- Resource Usage

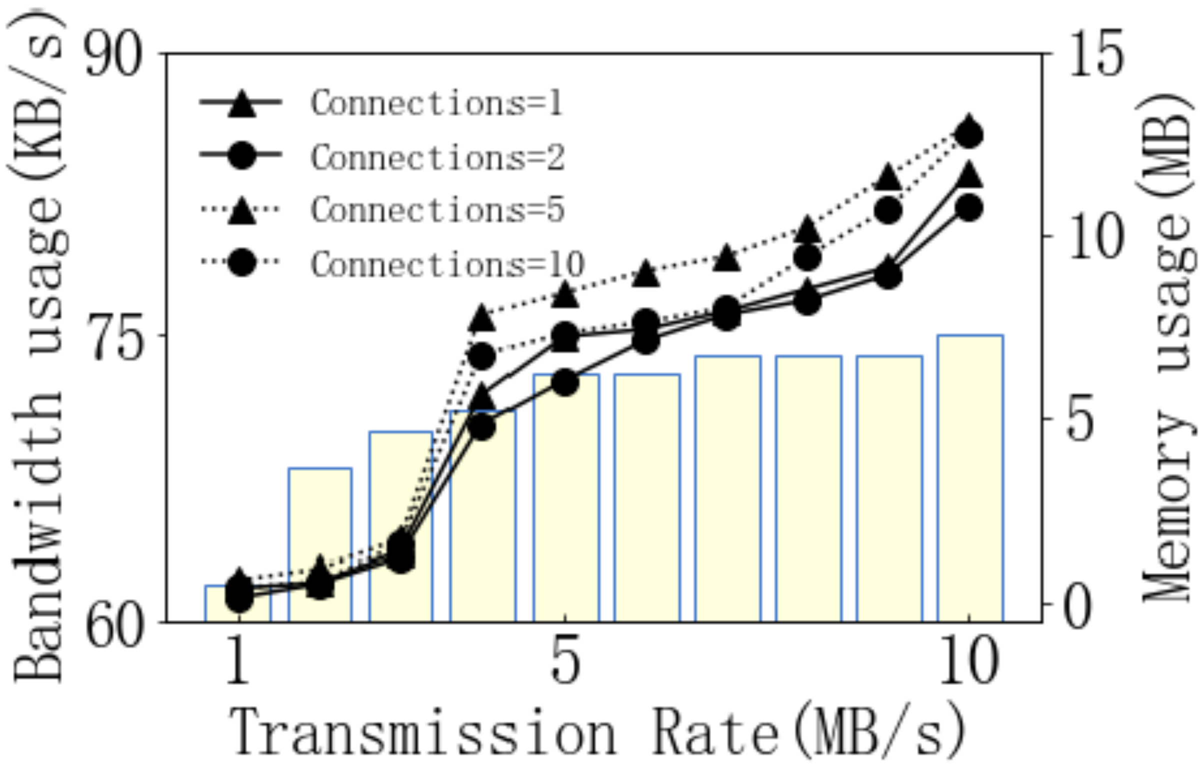


Figure 11. Resource usage of PUFF

Finally

THANKS!