AmLight ExP
Americas Lightpaths Express & Protect

2023 Internet2 Technology Exchange

Sep 21st, 2023 // Minneapolis, MN

A Scalable Solution to Detect Microbursts

Jeronimo Bezerra , Italo Valcy, Renata Frez, Julio Ibarra

<jbezerra@fiu.edu>, <idasilva@fiu.edu>, <renata.frez@rnp.br>, <julio@fiu.edu>

Outline

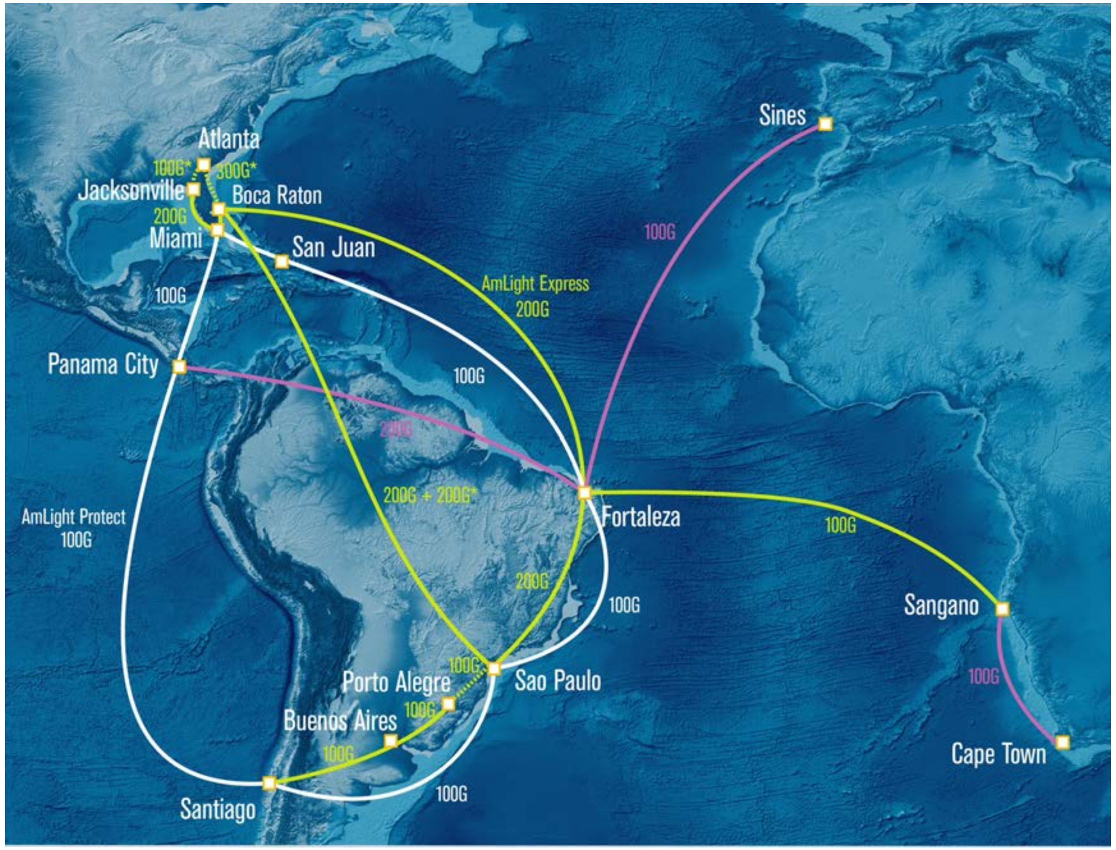
- The Motivation
- The Challenge
- The Solution
- Conclusion & Future Work

AmLight ExP
Americas Lightpaths Express & Protect

The Motivation.

The Motivation

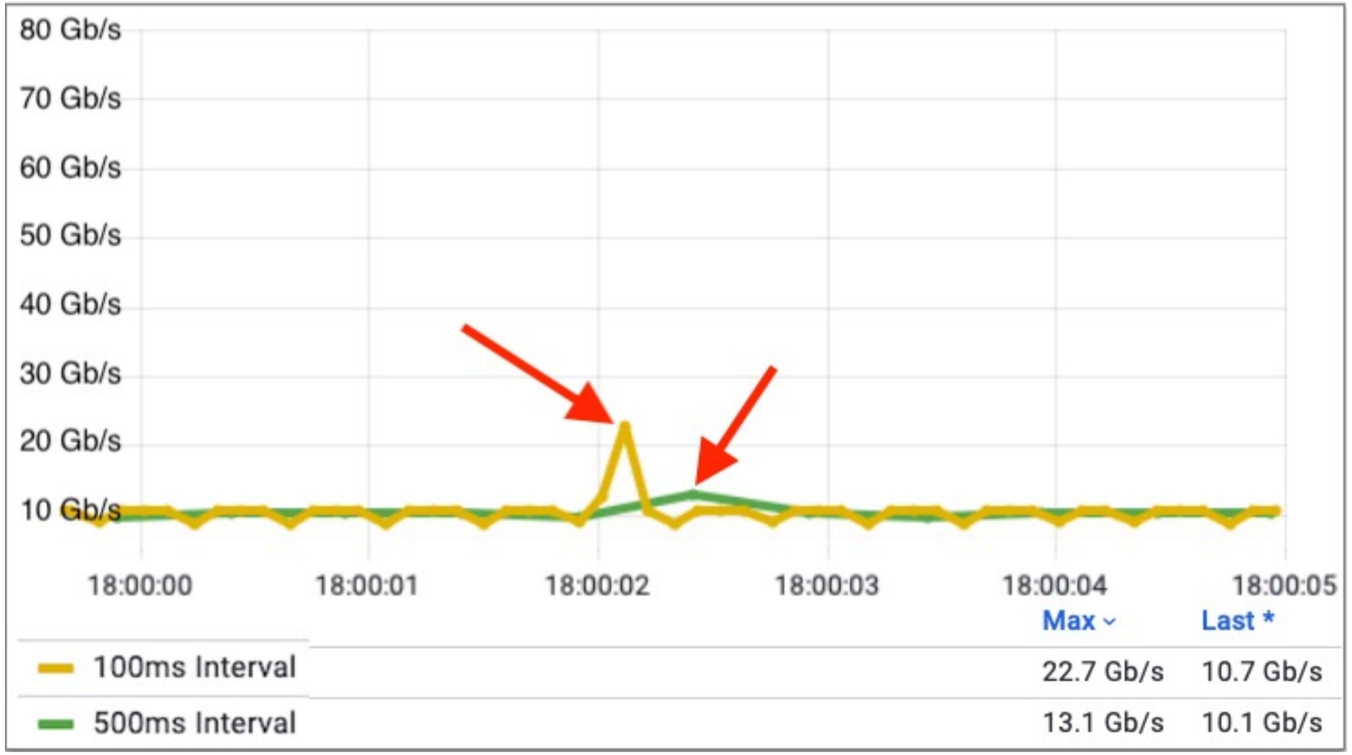
- AmLight is an international research and education network funded by NSF with a SLA-driven application
- The SLA (Service Level Agreement) is specific about the performance metrics
- Packet drops and bottlenecks caused by microbursts can lead to poor performance and SLA penalties
- Perfect motivation to build a new network monitoring solution to enable sub-second network visibility.



AmLight ExP
Americas Lightpaths Express & Protect

Running a small proof-of-concept of a microburst

- First: Let's assess the challenge: Simulating a microburst
- A microburst traffic of 70Gbps for 40ms
- And a constant 10Gbps background flow
  - Total: 80Gbps
- Two instances of the "current" NMS:
  - One gathering counters every 500 ms (default)
  - One gathering counters every 100 ms (5x faster)
- The results are provided in the figure.
  - None of the instances came close to 80 Gbps
  - *Found the gap in our monitoring!*



The Challenge.

What is a microburst?

- Microbursts are sporadic bursts of traffic that occurs in *very short* time-scales:
  - Most publications suggest a **millisecond** time-scale (1/1000th of a second)
  - However most network monitoring solutions (commercial and open-source) work on a **second** time-scale
    - Extreme challenging to detect millisecond-based microbursts with traditional tools!

- Why should we monitor microbursts:
  - Microbursts deplete bandwidth, fill out buffers, and introduce delay, jitter, and packet drops

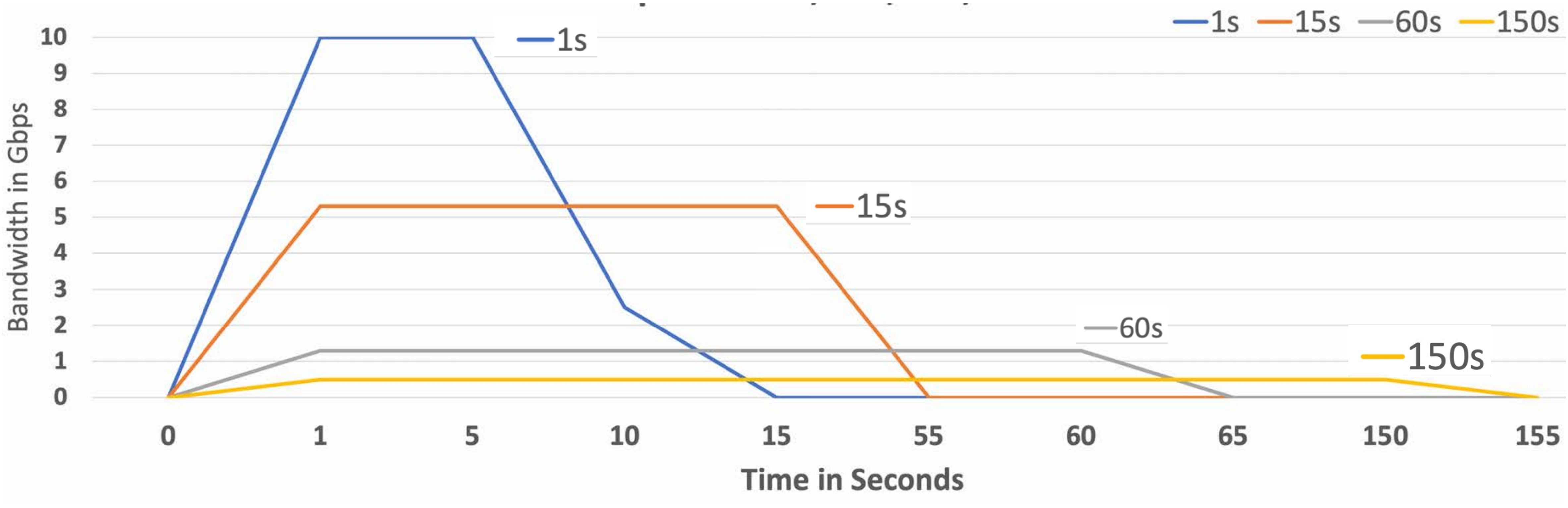
The Challenges of detecting a microburst

- How fast should we query network devices for interface/flow counters?
  - 60s? 30s? 1s? Sub-second? (assuming the network device supports these short intervals)
  - In our experiment, even every 100ms wasn’t enough to detect some microbursts
  - SNMP, Automation, API queries
- How small should be the interval or sample to export interface/flow counters?
  - 1:1000? 1:500?
  - sFlow, JTI
- Is disk space utilization a concern?
  - The more granular the measurement (smaller gathering interval or streaming frequency), more data must be stored.
  - To avoid excessive disk space consumption over time, retention policies delete “old” data (losing historical data) or create *trends* (losing accuracy)

How fast should we query network devices for interface/flow counters?

Sending 10GB of data over 10Gbps link: 8 seconds

Let's query a network interface's counters every 1s, 15s, 60s, and 150s:



AmLight ExP
Americas Lightpaths Express & Protect

Is disk space utilization a concern?

A typical interface counter has 4 bytes for timestamp (8 bytes if nanosec is required) plus 8 bytes for the counter (octets or packets)

For bandwidth, we query for incoming and outgoing traffic, per bytes and per packets (at least)

Interval	# of queries in a day per monitored item	Amount of data collected in a day (in Bytes) for ONE monitored item	Amount of Data Collected in a DAY for AmLight for all monitored items
Every 1s	86,400	1.04 Mbytes	69.97 Gbytes
Every 15s	5,760	0.07 Mbytes	4.6 Gbytes
Every 60s	1,440	0.01 Mbytes	1.2 Gbytes
Every 150s	576	0.007 Mbytes	0.5 Gbytes

More Accuracy -> More Disk Space, I/O, and CPU utilization

The Solution.

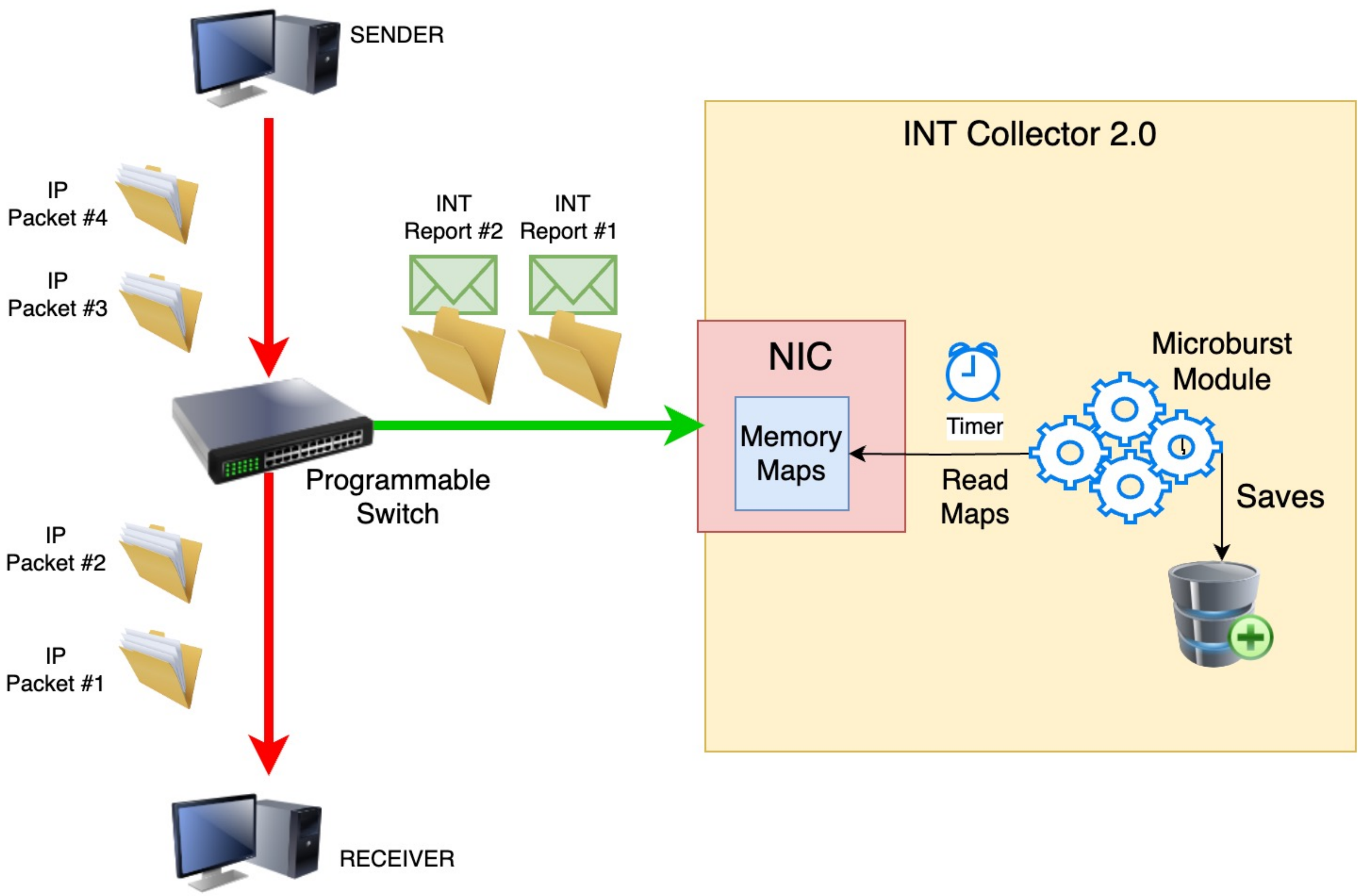
AmLight INT Collector 2.0 – Microburst edition

- Goal 1: Leveraging In-band Network Telemetry (INT) to detect microbursts
- Goal 2: No more static gathering interval (every *N* milliseconds)
- Goal 3: Monitoring bandwidth utilization in a very short time interval but saving the bandwidth utilization counter only in case of *pattern changes* (user-defined)

The Solution

- Version 1.0:
  1. Reads counters from NIC every 500 ms
  2. Saves countera to disk

- Version 2.0:
  1. Reads counters from NIC every 20 ms
  2. Search for microbursts and pattern changes.
  3. If found, saves a summary to disk.



AmLight ExP
Americas Lightpaths Express & Protect

Algorithm 1: Detecting microbursts

Algorithm 1 searches for *pattern changes*: when current bandwidth grows above a predefined β factor (microburst factor) using the last Δ values as reference.

Algorithm 1: Detecting Microbursts

```
isBurst ← False;
while True do
    BW_curr ← GetCurrBW();
    if isBurst = False then
        if BW_curr > max(γ, BW_avg(Δ) ∗ β) then
            isBurst ← True;
            SaveLastValues(Δ);
            SaveCurrBW(BW_curr);
            BW_target ← BW_curr;
    else
        SaveCurrBW(BW_curr);
        if not BW_curr > BW_target ∗ θ then
            isBurst = False;
            SendToDatabase();
    Sleep(α);
end
```

pattern changes

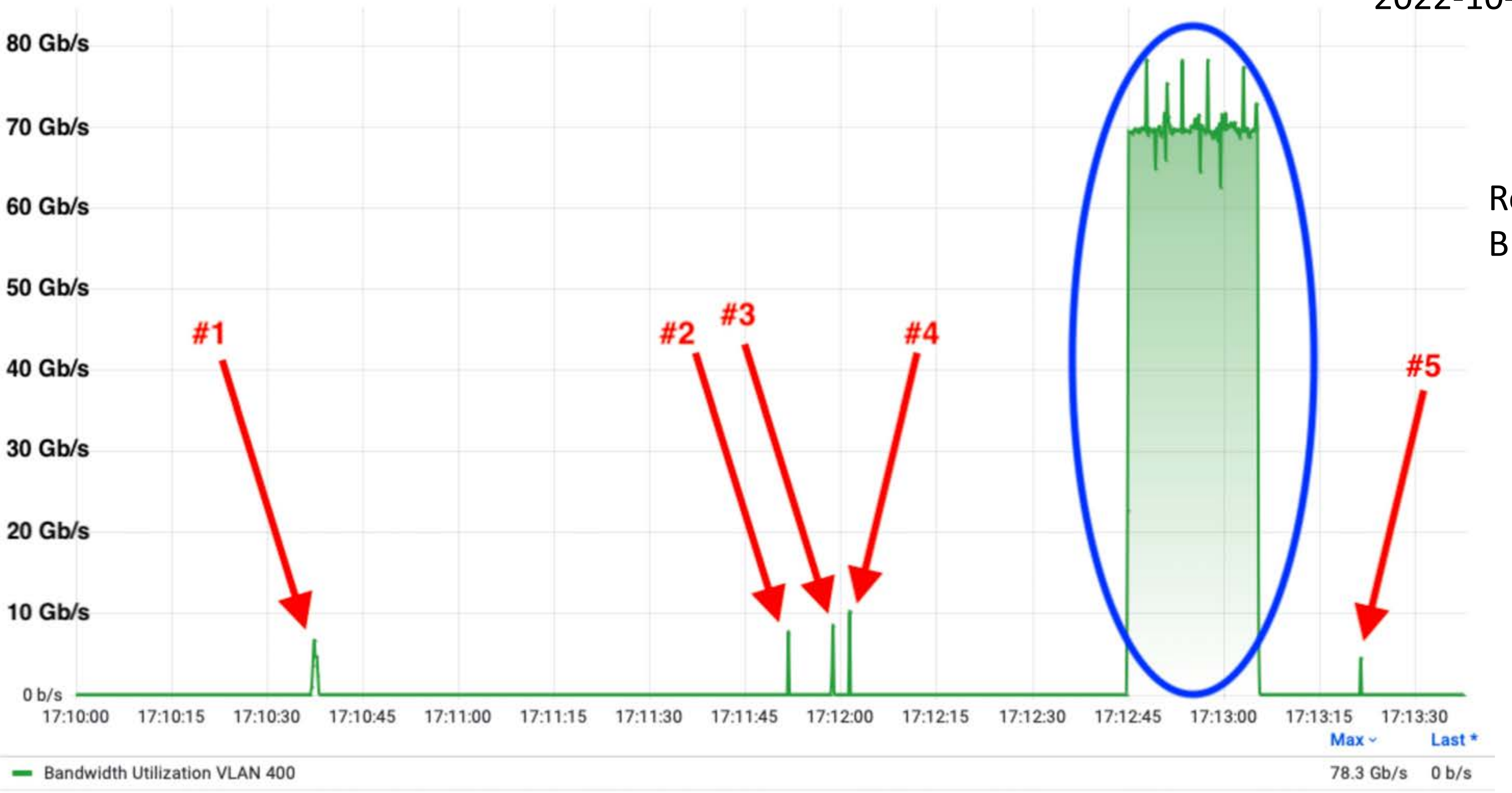
Saves state pre-microburst

Once microburst is over, store it in disk

α = loop interval to collect counters
γ = minimal bandwidth
β = growing factor
Δ = number of previous measurements
θ = microburst margin

Algorithm 1: Field Trial in Production

2022-10-09 17:10:00 to 17:32:30



Red Arrows: Microburst
Blue Ellipse: Burst

$\alpha = 20$ ms
$\gamma = 14$ Gbps
$\beta = 4$
$\Delta = 3$
$\theta = 60\%$

AmLight ExP
Americas Lightpaths Express & Protect

Innovation 1: Field Trial in Production [2]

Zoom in on Microburst #1:

Start Time (UTC)	Duration (s)	Max BW (Gbps)
2022-10-09T13:10:37.304385768	0.02	16.35
2022-10-09T13:10:37.400937960	0.02	17.44
2022-10-09T13:10:37.499991784	0.02	18.88
2022-10-09T13:10:37.598316288	0.02	19.01
2022-10-09T13:10:37.696891136	0.02	18.97
2022-10-09T13:10:37.795097088	0.02	18.91
2022-10-09T13:10:37.893028608	0.02	19.09
2022-10-09T13:10:37.992322792	0.02	18.66
2022-10-09T13:10:38.091456256	0.02	18.91
2022-10-09T13:11:51.925128192	0.04	37.87
2022-10-09T13:11:58.794430952	0.06	53.41
2022-10-09T13:12:01.507265768	0.04	41.48
2022-10-09T13:13:21.666561768	0.04	20.83

AmLight ExP
Americas Lightpaths Express & Protect

Algorithm 2: Adaptive Monitoring

- The adaptive approach is keep processing the bandwidth counters and dynamically deciding whether to save them.
- Our strategy compares the observed current bandwidth **BWcurr** to previous value **BWprevious** to understand if bandwidth has varied “significantly” using a user-defined margin **BWFactor**.
- If bandwidth has not varied significantly, the strategy decreases or increases the interval via user-defined **DecreaseFactor** and **IncreaseFactor**.
- We consider that bandwidth varies when it increases or decreases more than a user-defined threshold, for instance, more than 15%.

Algorithm 2: Adaptive Measurement Interval

while *True* **do**
- $BW_{curr} \leftarrow GetCurrentBW()$;
- $BW_{previous} \leftarrow GetPreviousBW()$;
- **if** $BW_{curr} > (BW_{previous} * BW_{Factor})$ ***or*** $BW_{previous} > (BW_{curr} * BW_{Factor})$ **then**
  - **if *not*** $(Interval - DecreaseFactor) < MinInterval$ **then**
    - $Interval \leftarrow Interval - DecreaseFactor$;
- **else**
  - **if *not*** $(Interval - IncreaseFactor) > MaxInterval$ **then**
    - $Interval \leftarrow Interval + IncreaseFactor$;
- $SaveCounter(BW_{curr})$;
- $Sleep(Interval)$;

end

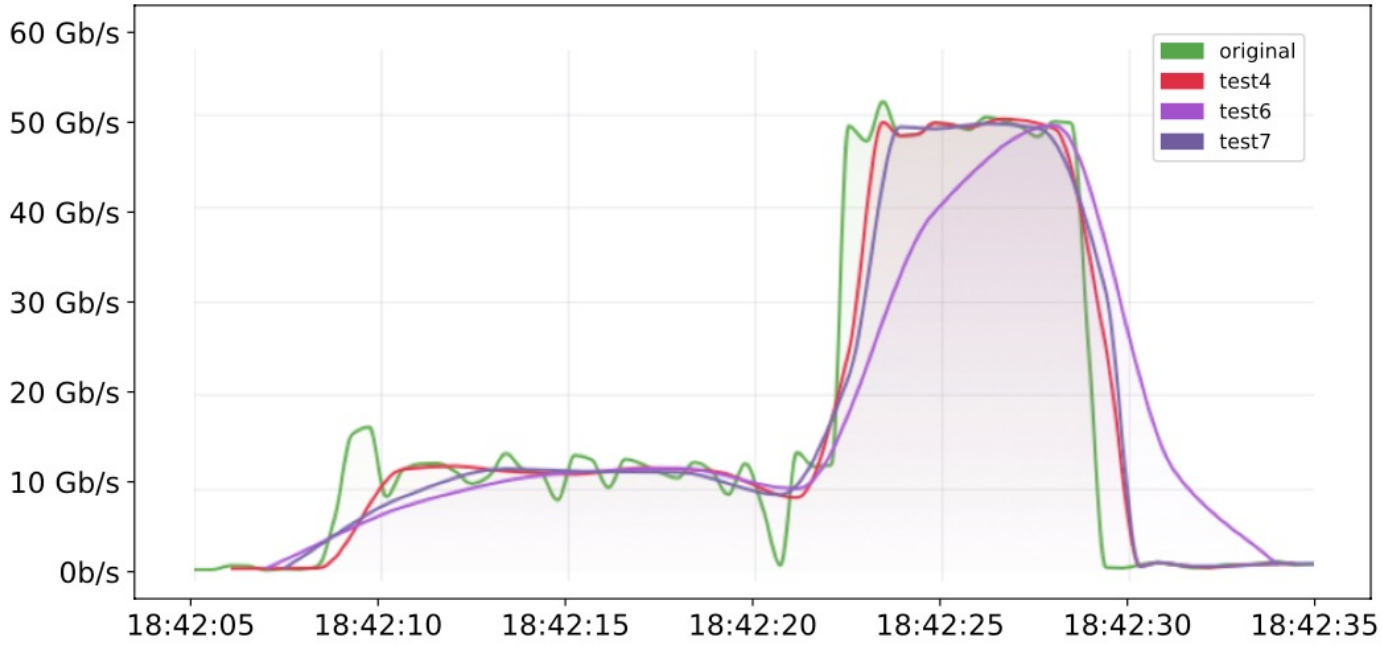
DecreaseFactor and IncreaseFactor are key variables since they define how fast to react to bandwidth variation.

Algorithm 2: Field Trial in Production

- To evaluate our solution, we used two well-known image comparison algorithms: PSNR-B and SSIM.
  - Peak Signal-to-Noise Ratio (PSNR-B) and Structural Similarity Index (SSIM) are used for video quality analysis and image comparison

Time Frame	Parameters	Similarity		Efficiency
		PSNR-B	SSIM	
Last 30 seconds	Setup 4	26.03	81.36%	61.64%
	Setup 6	26.23	81.48%	64.38%
	Setup 7	26.10	81.28%	64.38%
Last 30 days	Setup 4	34.88	97.92%	53.90%
	Setup 6	23.48	92.28%	67.48%
	Setup 7	33.70	99.24%	63.96%

- PSNR-B assigns higher values whenever images are more similar
  - > 37 – Excellent
  - 31 – 37 – Good
  - 25 – 31 – Acceptable



- SSIM returns the similarity percentage
  - > 90% – Excellent
  - 77–89% – Good
  - 61 – 76% – Acceptable

AmLight ExP
Americas Lightpaths Express & Protect

Conclusion & Future Work

- Field trials were performed and demonstrated how effectively we detect microbursts in a production network, down to 20 milliseconds, and how efficiently we reduced storage space (above 60%) while preserving good levels of accuracy.

- **Similar results can be achieved with port mirroring or fiber taps solutions!**

- Future work:
  - Dynamic tuning the parameters
  - Expanding the INT collector to enable additional specialized monitoring functions (e.g., per-flow microbursts detection).

	Commercial and OSS Tools	AmLight Innovations	
		Algorithm 1	Algorithm 2
Monitors Bandwidth Utilization	Full Support	N/A	Full Support
Fixed Data Gathering Interval	Full Support	N/A	N/A
Adaptive Data Gathering Interval	Not Supported	N/A	Full Support
Detects Bursts	Partial Support *	N/A	Full Support
Captures beginning of bursts	Partial Support *	Full Support	Full Support
Detects Microbursts	Not Supported	Full Support	N/A
Efficient Disk Space Consumption	Not Supported	Full Support	Full Support

AmLight ExP
Americas Lightpaths Express & Protect

Thank you! Any questions?

A Scalable Solution to Detection Microbursts

Jeronimo Bezerra , Italo Valcy, Renata Frez, Julio Ibarra

<jbezerra@fiu.edu>, <idasilva@fiu.edu>, <renata.frez@rnp.br>, <julio@fiu.edu>

Recent Presentations/References

- Understanding the impact of network microbursts to science drivers - 07/07/2023
    - https://youtu.be/_wronGw48os
    - CI Engineering Lunch & Learn Series
- Detecting Network Microbursts at AmLight - 04/21/2023
    - https://youtu.be/1x-aVZTyyiM
    - CI Engineering Lunch & Learn Series
- In-band Network Telemetry at AmLight - 03/18/2022
    - https://youtu.be/M6n_UZlhBQQ
    - CI Engineering Lunch & Learn Series
- Autonomic Network Architecture at AmLight - 02/25/2022
    - https://youtu.be/CRnKKuP9I3Y
    - CI Engineering Lunch & Learn Series
- Deploying per-packet telemetry in a long-haul network - 11/21/2021
    - https://www.youtube.com/watch?v=lVtY7dP7UGs&t=2s
    - INDIS Workshop