Stream Processing for Optical Network Monitoring with Streaming Telemetry and Video Analytics

Jesse E. Simsarian[(1)], Matthew Nance Hall[(2)], Gurudutt Hosangadi[(1)], Jurgen Gripp[(3)], Wolfgang Van Raemdonck[(4)], Jiakai Yu[(5)], and Theodore Sizer[(1)]

[(1)] Nokia Bell Labs, Murray Hill, New Jersey, USA, jesse.simsarian@nokia-bell-labs.com,
[(2)] University of Oregon, Computer and Information Science, Eugene, OR, USA,
[(3)] Nokia ION Optics, Murray Hill, New Jersey, USA,
[(4)] Nokia Bell Labs, Antwerp, Belgium,
[(5)] College of Optical Sciences, University of Arizona, Tucson, AZ, USA

Abstract *We use a cloud platform to demonstrate the processing of streaming data from diverse sources for infrastructure monitoring. Telemetry data from an optical network testbed is processed for alarm generation by anomaly detection. The optical alarm is correlated with machine-learning person detection by video analytics.*

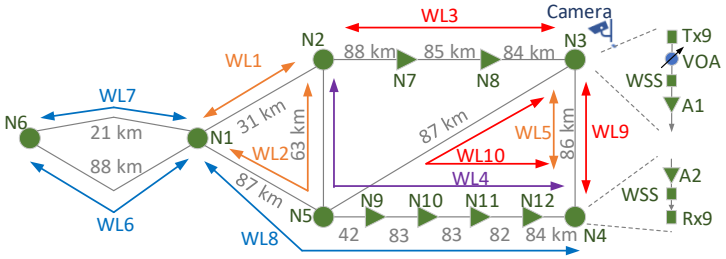
Introduction

As optical networks become more dynamic, real-time monitoring of performance metrics from coherent transponders and the network infrastructure will be important for achieving high reliability. We are undergoing a shift in optical-network monitoring by moving away from 15-minute intervals to streaming telemetry with gNMI/gRPC[1] for faster updates[2]. With 5G, which supports many internet-of-things devices, we can correlate data from diverse sources, e.g., weather data[3] or video monitoring of the network infrastructure. We need a scalable, distributed, platform that allows event and alarm correlation from the diverse sources. Prior work utilized a database[4]–[6] for optical telemetry data storage and retrieval, which is useful for optical software-defined networking controllers[7] as well as post-failure and long-term trend analysis. However, telemetry data is now generated as streams of information, and it is natural to process the data as streams for flexible and deterministic alarm generation and correlation, but such a paradigm shift in network monitoring has not yet occurred. Research has been done on streaming analysis at the IP layer[8].

In this work, we use stream processing to generate impairment (soft failure) alarms from an optical network with streaming telemetry and correlate it with a person detection alarm from a video stream that monitors a network element. The optical network is vulnerable to attacks such as vandalism and fiber tapping[9] as well as unintentional impairment caused by craft personnel, and the correlated alarm provides insight for network operations. We use World Wide Streams (WWS)[10], a real-time stream processing cloud platform, to flexibly process data and media streams. WWS implements the distributed deployment of stream processing units and stream forwarding using RabbitMQ[11] and RTMP[12]/WebRTC[13]. WWS gives different network operations teams access to monitoring data to create stream processing pipelines without expert knowledge in each domain. For video, the stream processing uses machine learning (ML) to classify objects in the images, so that video frames do not need to be stored in a database. We demonstrate that the streaming platform has low latency for real-time alarm generation.

Optical Network Testbed

As diagrammed in Fig. 1, the optical network testbed consists of six commercial Nokia 1830



Wavelength	Length (km)	Freq. (THz)
WL1	31	193.67
WL2	150	193.67
WL3	257	193.73
WL4	437	193.61
WL5	86	193.67
WL6*	88	193.55
WL7*	21	193.55
WL8*	461	193.55
WL9	86	193.73
WL10	461	193.73

* Not monitored

Fig. 1: Optical network diagram with optical attenuator and camera monitor on node N3.

2020 European Conference on Optical Communications (ECOC) | 978-1-7281-7361-0/20/$31.00 ©2020 IEEE | DOI: 10.1109/ECOC48923.2020.9333169

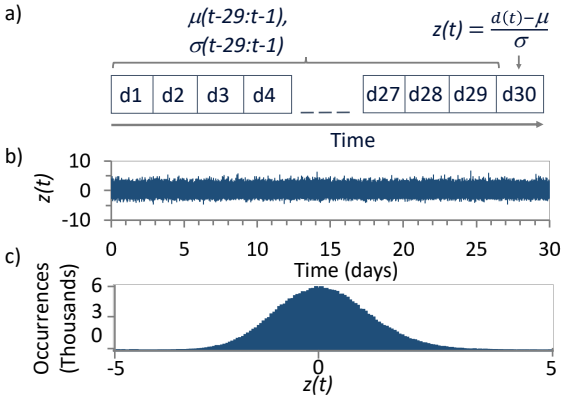


Fig. 2: a) Window averaging method b) z(t) for preFEC BER versus time measured over 30 days c) Histogram of b)

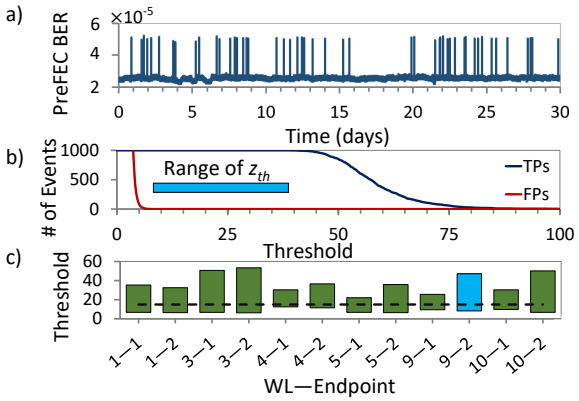


Fig. 3: a) preFEC BER of WL9 EP2 for 30 days with artificial anomalies b) TP and FP versus z_{th} with a range of possible z_{th} c) z_{th} ranges for anomaly detection for different endpoints

PSS optical nodes (N1-N6) with flexgrid reconfigurable optical add drop multiplexers and 2200 km of fiber. Seven of the ten bidirectional wavelengths (WLs) use Nokia flexible line cards with variable bitrate operated at 200 Gb/s with 8QAM modulation format, and we monitor their performance metrics using gNMI with 10 s period. Nodes N1, N2, N3, and N4 terminate the monitored wavelengths and are the sources of the optical streaming telemetry data. The path lengths and optical frequencies are given in the table of Fig. 1. More detail on one direction of WL9 is shown where a variable optical attenuator (VOA) in the path attenuates the optical signal causing optical signal-to-noise ratio degradation and the preFEC bit error ratio (BER) to rise. N3 has a camera pointed at it with a machine learning based person identification process running on WWS. In this work we correlate the person detection alarm with BER impairment on wavelengths that terminate at N3, specifically, both directions of WL3, WL5, WL9, and WL10.

Stream Processing

WWS implements real-time processing on a broad set of stream types, including event and media streams. Dataflow pipeline processing applications are written in XStream[14], a typed domain-specific functional-programming language on top of TypeScript[15].

We detect point anomalies in the incoming optical data stream by using a time-window detection method as shown in Fig. 2a. The mean, μ, and standard deviation, σ, of the previous 29 data points are calculated over the window, and $z(t)$ of the most recent data point, $d(t)$, is given by $z(t) = (d(t) - \mu)/\sigma$. Since the telemetry data arrives every 10 s, the window duration is 5 min, which removes any slower wander in the data. Fig. 2b shows $z(t)$ over 30 days of recorded preFEC BER data for one of the wavelengths, with the histogram shown in Fig. 2c. We use $z(t)$ to identify point anomalies in the incoming data when $z(t)$ crosses a threshold value, z_{th}. To determine an initial default value of z_{th}, we inject artificial point anomalies into the data streams for each of the monitored wavelength endpoints. Fig. 3a shows a trace of WL9 at node N4 (endpoint EP2) over 30 days with 50 artificial anomalies created by adding 2x the mean BER (mean Q degradation of 0.35 dB) to randomly selected data points. We perform a scan of z_{th} over a range of values and calculate the number of false positives (FPs) and true positives (TPs) for detecting 1000 anomalies, shown in Fig. 3b for the same endpoint. The bar in Fig. 3b indicates the range of threshold values where no false positives, and 100% of true positives were detected. Fig. 3c shows the results of applying the method to six of the wavelengths. The value of $z_{th} = 15$ allows for identification of all the anomalies without false positives, while one of the wavelengths (not shown) requires a larger anomaly amplitude of 3x the mean. While a fixed z_{th} applied to every endpoint is convenient, we can adjust the threshold for each endpoint over time to change the sensitivity of the alarm. For example, wavelength 9,

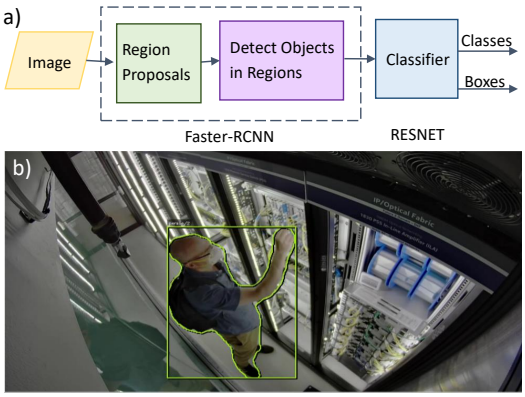


Fig. 4: a) Processing pipeline for person detection b) Video image with ML person detection overlay indication

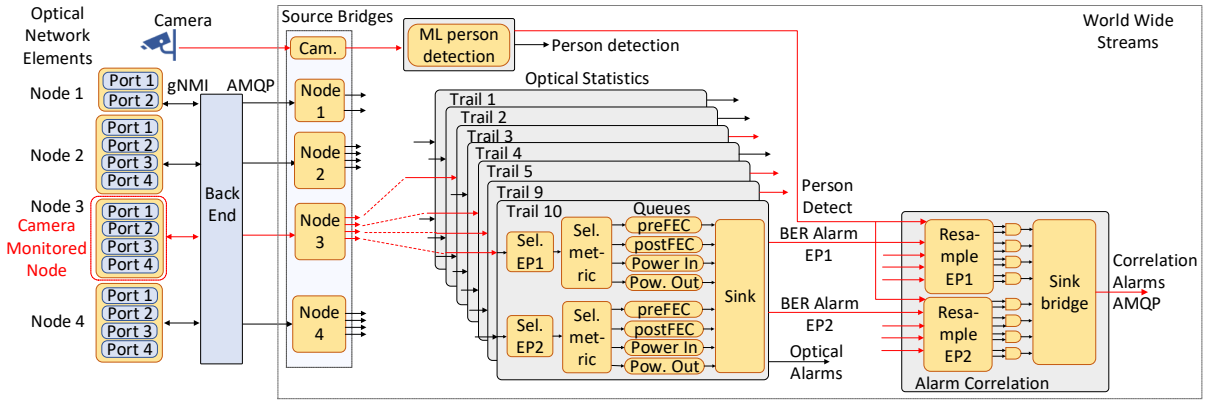


Fig. 5: Diagram of the WWS stream processing where grey boxes are XStream processes

EP2 can be operated at a more sensitive level of $z_{th} = 10$ for 100% true positive detection and no false positives.

We use a two-stage deep learning network (mask-rcnn-resnet-101)[16], for the person detection at N3, see Fig. 4a. WWS can be extended with stream processing functions written in any programming language. We implemented the ML in Python, Keras and TensorFlow using the code at[17] as a starting point. This is Dockerized[18] to provide a GPU-based person-detection microservice to WWS via a ZeroMQ socket. The outputs of the classifier are the classes and locations of the objects in the frame with a latency of about 8 s and mean frame rate of 91 ms. Fig. 4b shows a person at N3 with a bounding box and outline imposed by the neural network.

Alarm Correlation Experiment

We use the WWS platform to correlate the optical telemetry anomaly detection alarms of the wavelengths that terminate at N3 with the person detection alarm. Fig. 5 shows a diagram of the XStream processes on the WWS platform. The backend program makes the gNMI connection to the optical nodes and sends the data directly to the WWS source bridges using AMQP[19]. The optical statistics functions on WWS process the optical performance data according to the windowing method previously described. The output of the wavelength monitors of interest and ML person detection are sent to the alarm correlation XStream process. Since the period of the optical wavelength monitor updates is 10 s, and the person detection frame rate is 91 ms, we resample all the data streams with a period of 150 ms, selected to be lower than the camera frame rate. When new data is available from both sources, the samples are output synchronously from the resampler and combined with a logical and operation.

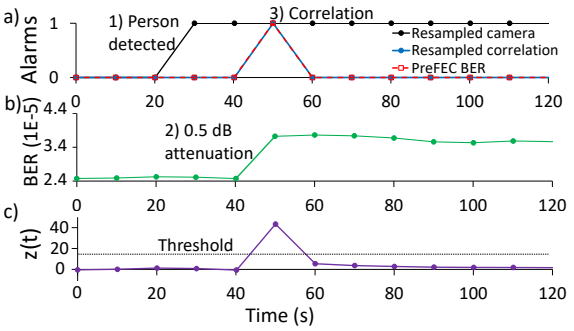


Fig. 6: a) BER and person detection alarm correlation b) preFEC BER vs time c) $z(t)$ for the anomaly detection

A typical measurement of the alarm correlation running on WWS is shown in Fig. 6a, where the person detection alarm initially goes high as the person approaches the optical node. Once at the optical node N3, the VOA attenuation is increased by 0.5 dB on WL9 EP2, resulting in the increase in preFEC BER at node N4 (Fig. 6b). The corresponding increase in $z(t)$ is shown in Fig. 6c, and the preFEC BER alarm is raised in Fig. 6a due to the crossing of z_{th}. Following the resampling, the correlation alarm is raised in Fig. 6a. Note that the delay between the optical BER alarm and correlation alarm is not noticeable on this timescale. The latencies through the backend and WWS of the optical and correlation alarms are measured to be 40$\pm$17 ms and 133$\pm$36 ms, respectively, referenced to the timestamp given by the network element when the BER was measured. The correlation alarm latency is largely determined by the resampling period.

Conclusions

We demonstrated stream processing for the flexible generation and correlation of optical alarms with ML-based person detection to improve network awareness and give insight to operators when optical networks are impaired. The stream processing platform is scalable and cloud based, and we measured low deterministic latency.

References

[1] (2020), [Online]. Available: https://www.grpc.io/.

[2] F. Paolucci, A. Sgambelluri, F. Cugini, and P. Castoldi, "Network telemetry streaming services in sdn-based disaggregated optical networks", *J. Lightwave Technol.*, vol. 36, no. 15, pp. 3142–3149, 2018.

[3] D. Charlton *et al.*, "Field measurements of sop transients in opgw, with time and location correlation to lightning strikes", *Optics Express*, vol. 25, no. 9, pp. 9689–9696, 2017.

[4] L. Velasco, L. Gifre, J.-L. Izquierdo-Zaragoza, F. Paolucci, A. P. Vela, A. Sgambelluri, M. Ruiz, and F. Cugini, "An architecture to support autonomic slice networking", *J. Lightwave Technol.*, vol. 36, no. 1, pp. 135–141, 2018.

[5] S. Yan, A. Aguado, Y. Ou, R. Wang, R. Nejabati, and D. Simeonidou, "Multilayer network analytics with sdn-based monitoring framework", *J. Opt. Commun. Netw.*, vol. 9, no. 2, A271–A279, 2017.

[6] Q. Pham Van, D. Verchere, P. Layec, A. Dupas, F. Ilchmann, L. Dembeck, H. Tran-Quang, and D. Zeghlache, "Monitoring intent for optical channel defragmentation in software-defined elastic optical networks", in *Proc. ICTON'18*, Bucharest, Romania, Jul. 2018, Mo.C3.2.

[7] N. Sambo, F. Cugini, A. Sgambelluri, and P. Castoldi, "Monitoring plane architecture and oam handler", *J. Lightwave Technol.*, vol. 34, no. 8, pp. 1939–1944, 2016.

[8] A. Gupta, R. Birkner, M. Canini, N. Feamster, C. MacStoker, and W. Willinger, "Network monitoring as a streaming analytics problem", in *Proc. HotNets '16*, Atlanta, GA, Nov. 2016, pp. 106–112.

[9] T. Hughes, *Attacks show fiber optic internet cables vulnerable*, https://www.usatoday.com/story/news/2015/09/16/attacks-show-fiber-optic-internet-cables-vulnerable/32502785/, 2015.

[10] (2020), [Online]. Available: https://www.worldwidestreams.io.

[11] (2020), [Online]. Available: https://www.rabbitmq.com/.

[12] (2012), [Online]. Available: https://wwwimages2.adobe.com/content/dam/acom/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf.

[13] (2020), [Online]. Available: https://webrtc.org/.

[14] (2019), [Online]. Available: https://www.bell-labs.com/our-research/publications/296844/.

[15] (2020), [Online]. Available: https://www.typescriptlang.org/.

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn", in *Proc. IEEE International Conference on Computer Vision (ICCV'17)*, Venice, Italy, Oct. 2017, pp. 2980–2988.

[17] W. Abdulla, *Mask r-cnn for object detection and instance segmentation on keras and tensorflow*, https://github.com/matterport/Mask_RCNN, 2017.

[18] (2020), [Online]. Available: https://www.docker.com/.

[19] (2020), [Online]. Available: https://www.amqp.org/.