Self-Adaptive Network Monitoring in IP-over-EONs: When Multilayer Telemetry is Flexible and Driven by Data Analytics

Shaofei Tang[1], Jianquan Peng[1], Siqi Liu[1], Jiawei Kong[1], and Zuqing Zhu[1]
1. University of Science and Technology of China, Hefei, Anhui 230027, China, Email: zqzhu@ieee.org

Abstract: We propose and experimentally demonstrate a self-adaptive network monitoring system for IP over elastic optical networks, which can coordinate and adjust in-band and out-of-band monitoring schemes intelligently based on the data analytics on multilayer telemetry data.

OCIS codes: (060.1155) All-optical networks; (060.4261) Networks, protection and restoration.

1. Introduction

Recently, due to rapid development of network services, the network control and management (NC&M) of backbone networks becomes increasingly complicated [1]. Hence, as a key enabling technique of NC&M, network monitoring is under the pressure to be realtime, precise, and efficient. Moreover, since backbone networks usually take the multilayer architecture, we need to monitor their packet and optical layers jointly. Therefore, people have designed several multilayer in-band network telemetry (ML-INT) techniques to realize fine-grained, realtime, and programmable network monitoring [2, 3]. However, as ML-INT inserts telemetry data into packets as header fields, it causes noticeable bandwidth overheads, which, if not controlled well, will affect the quality-of-service (QoS) of network services. The bandwidth overheads of ML-INT can be reduced by inserting telemetry data in packets in a selective [4] or probabilistic [5] way (*i.e.*, the flexible ML-INT). Nevertheless, the previous studies in [4, 5] did not consider how to optimize the actual operation scheme of ML-INT (*e.g.*, the locations to invoke ML-INT, types of ML-INT data to collect, and ratios of packets to insert ML-INT data) according to the status of a dynamic network. In other words, the question of how to spend the overheads of ML-INT intelligently to realize self-adaptive network monitoring has not been addressed yet.

As ML-INT conveys rich telemetry data to the control plane and enables deep learning (DL) based data analytics there, it can provide us not only the information on how to optimize the operation of a packet-over-optical network, but also the insights on how to adjust its own operation scheme adaptively. Hence, in this work, we propose to drive multilayer telemetry with the data analytics on ML-INT data, for realizing self-adaptive monitoring in IP over elastic optical networks (IP-over-EONs). Specifically, our self-adaptive network monitoring system can coordinate and adjust the operation schemes of ML-INT and out-of-band optical monitoring intelligently based on the closed-loop control driven by DL-based data analytics. We implement our proposal in a real IP-over-EON testbed, and experimentally demonstrate its effectiveness with two representative use-cases related to the IP and EON layers, respectively.

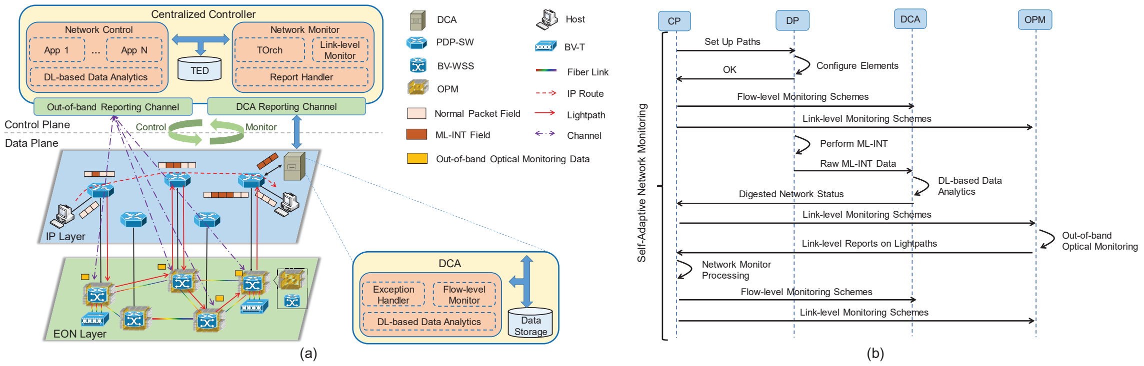


Fig. 1. (a) Architecture of our self-adaptive network monitoring system, and (b) Operation procedure.

2. Self-adaptive Network Monitoring in IP-over-EONs

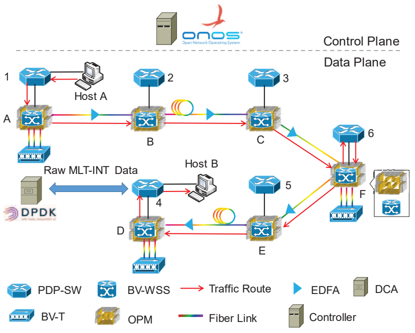
Fig. 1(a) shows the architecture of our self-adaptive network monitoring system, whose data plane (DP) is a multilayer IP-over-EON. The EON layer is built with bandwidth-variable wavelength-selective switches (BV-WSS'), and optical line systems (OLS') that include bandwidth-variable transponders (BV-Ts). On each node in the EON layer, we implement optical performance monitoring (OPM), which provides the status about active lightpaths (*e.g.*, bit-error-rate (BER), optical spectrum, optical signal-to-noise-ratio (OSNR)). Here, the telemetry data can be reported in either in-band or out-of-band way. The in-band way uses the ML-INT in [3], *i.e.*, the OPM sends the telemetry data to the local programmable data plane switch (PDP-SW), which will encode the data as header fields and insert them selectively in the packets of application traffic. Hence, the in-band way achieves realtime and application-specific monitoring of the EON layer, but since the telemetry data regarding a lightpath can only be collected and inserted in packets at its end nodes, link-level monitoring might not always be possible. In other words, if a lightpath goes through multiple fiber links, the in-band way is not applicable at the intermediate node(s) because packets on the lightpath bypass their local PDP-SW(s). The out-of-band way lets OPMs report telemetry data directly to the control plane (CP). Therefore, it can report the status of lightpaths at arbitrary nodes, but the monitoring will not be real-time or application-specific.

The IP layer consists of hosts, PDP-SWs and data collection agents (DCAs). The DCAs are introduced to process the ML-INT data collected with the in-band way, and to only report digested information to the control plane. Hence, the DCAs leverage distributed data analytics to achieve realtime and flow-level monitoring and troubleshooting (*i.e.*, offloading the processing of ML-INT data from the control plane). The hosts generate/receive application traffic, which are routed in the IP layer by the PDP-SWs. In the process, the PDP-SWs realize the in-band collection of ML-INT data, which also include the data about their own status (*e.g.*, port bandwidth usage, and packet processing latency).

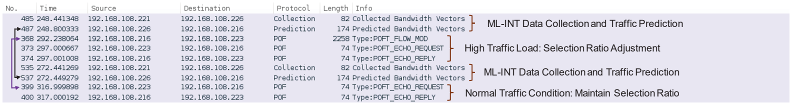
Fig. 1(a) also explains the design of the centralized controller. The traffic engineering database (TED) stores the multilayer provisioning schemes of application traffic. The network monitor utilizes three modules to achieve self-adaptive network monitoring. The report handler checks the reports from the DCAs to keep track of the flow-level monitoring, the link-level monitor coordinates the out-of-band optical monitoring, and the telemetry orchestrator (TOrch) manages the network monitoring schemes in the IP-over-EON. Specifically, the TOrch analyzes applications' multilayer provisioning schemes, their QoS demands, and the current settings and results of in-band and out-of-band monitoring, to determine whether the monitoring schemes need to be adjusted and how to adjust them accordingly. With the suggestion from the TOrch, the controller then updates the monitoring schemes to make them adapt to network status and application demands. The operation procedure of our self-adaptive network monitoring system is shown in Fig. 1(b).

3. Experimental Demonstrations

We implement a proof-of-concept of the self-adaptive network monitoring system in the small but real IP-over-EON testbed in Fig. 2(a). The EON layer consists of optical switches based on Finisar 1×9 BV-WSS', each of which enables flexible-grid spectrum allocation [6], fiber links with inline erbium-doped fiber amplifiers (EDFAs), and an OLS that is based on Juniper BTI7800 platform and deploys its BV-Ts on *Nodes A*, *D* and *F* in the testbed. The BV-Ts support line-rates within $[100, 400]$ Gbps, and their working status (*e.g.*, power-level, OSNR and BER) are monitored by the OLS in realtime. Hence, the OLS assists our system to realize the in-band collection of lightpath status. Meanwhile, we place a commercial optical channel monitor (OCM), which can scan the whole C-band with a fine resolution of 312.5 MHz, on each optical port to enable the link-level monitoring (*i.e.*, the out-of-band collection of lightpath status). We emulate each host with a commercial traffic analyzer that can generate/receive data packets at 10 Gbps. The DCA is home-made and can process packets with ML-INT data at a speed up to 2 million packets per second (Mpps). The PDP-SWs use 10 GbE ports to connect to the client side of the OLS. The controller is developed based on the ONOS platform, and we implement the control plane modules in Fig. 1(a) in it for self-adaptive network monitoring. We set up two lightpaths (*A-B-C-F* and *F-E-D*) to transmit data packets from *Host A* to *Host B*. Hence, the multilayer provisioning scheme of the application traffic uses the two lightpaths in the EON layer and gets switched in the IP layer in *PDP-SW 6*. With this setting, we experimentally demonstrate the effectiveness of our proposal with two use-cases.



(a)



```
No.   Time          Source            Destination       Protocol    Length Info
  485 248.441348    192.168.108.221   192.168.108.226   Collection      82 Collected Bandwidth Vectors
  487 248.800333    192.168.108.226   192.168.108.216   Prediction     174 Predicted Bandwidth Vectors
  368 292.238064    192.168.108.216   192.168.108.223   POF           2258 Type:POFT_FLOW_MOD
  373 297.000667    192.168.108.223   192.168.108.216   POF             74 Type:POFT_ECHO_REQUEST
  374 297.001008    192.168.108.216   192.168.108.223   POF             74 Type:POFT_ECHO_REPLY
  535 272.441269    192.168.108.221   192.168.108.226   Collection      82 Collected Bandwidth Vectors
  537 272.449279    192.168.108.226   192.168.108.216   Prediction     174 Predicted Bandwidth Vectors
  399 316.999898    192.168.108.223   192.168.108.216   POF             74 Type:POFT_ECHO_REQUEST
  400 317.000192    192.168.108.216   192.168.108.223   POF             74 Type:POFT_ECHO_REPLY
```

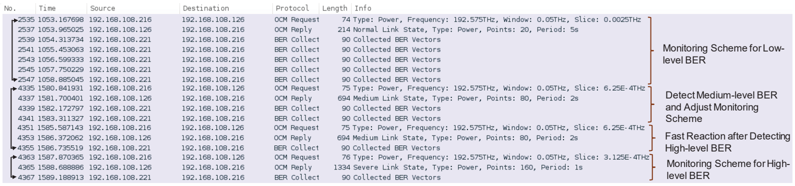
ML-INT Data Collection and Traffic Prediction

High Traffic Load: Selection Ratio Adjustment

ML-INT Data Collection and Traffic Prediction

Normal Traffic Condition: Maintain Selection Ratio

(b) Wireshark Capture of Control Messages in First Use-case



```
No.   Time          Source            Destination       Protocol    Length Info
 2535 1053.167698   192.168.108.216   192.168.108.126   OCM Request     74 Type: Power, Frequency: 192.575THz, Window: 0.05THz, Slice: 0.0025THz
 2537 1053.965025   192.168.108.126   192.168.108.216   OCM Reply      214 Normal Link State, Type: Power, Points: 20, Period: 5s
 2539 1054.313734   192.168.108.221   192.168.108.216   BER Collect     90 Collected BER Vectors
 2541 1055.453063   192.168.108.221   192.168.108.216   BER Collect     90 Collected BER Vectors
 2543 1056.599333   192.168.108.221   192.168.108.216   BER Collect     90 Collected BER Vectors
 2545 1057.750229   192.168.108.221   192.168.108.216   BER Collect     90 Collected BER Vectors
 2547 1058.885045   192.168.108.221   192.168.108.216   BER Collect     90 Collected BER Vectors
 4335 1580.841931   192.168.108.216   192.168.108.126   OCM Request     75 Type: Power, Frequency: 192.575THz, Window: 0.05THz, Slice: 6.25E-4THz
 4337 1581.700401   192.168.108.126   192.168.108.216   OCM Reply      694 Medium Link State, Type: Power, Points: 80, Period: 2s
 4339 1582.172797   192.168.108.221   192.168.108.216   BER Collect     90 Collected BER Vectors
 4341 1583.311327   192.168.108.221   192.168.108.216   BER Collect     90 Collected BER Vectors
 4351 1585.587143   192.168.108.216   192.168.108.126   OCM Request     75 Type: Power, Frequency: 192.575THz, Window: 0.05THz, Slice: 6.25E-4THz
 4353 1586.372062   192.168.108.126   192.168.108.216   OCM Reply      694 Medium Link State, Type: Power, Points: 80, Period: 2s
 4355 1586.735519   192.168.108.221   192.168.108.216   BER Collect     90 Collected BER Vectors
 4363 1587.870365   192.168.108.216   192.168.108.126   OCM Request     76 Type: Power, Frequency: 192.575THz, Window: 0.05THz, Slice: 3.125E-4THz
 4365 1588.688886   192.168.108.126   192.168.108.216   OCM Reply     1334 Severe Link State, Type: Power, Points: 160, Period: 1s
 4367 1589.188913   192.168.108.221   192.168.108.216   BER Collect     90 Collected BER Vectors
```

Monitoring Scheme for Low-level BER

Detect Medium-level BER and Adjust Monitoring Scheme

Fast Reaction after Detecting High-level BER

Monitoring Scheme for High-level BER

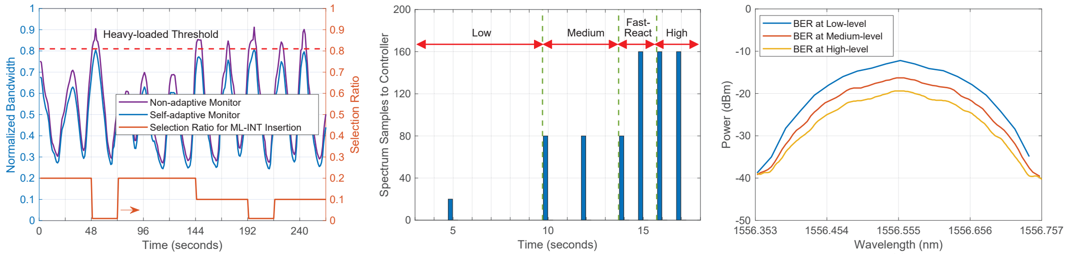
(c) Wireshark Capture of Control Messages in Second Use-case

Fig. 2. (a) Experimental setup, (b) Wireshark capture for first use-case, and (c) Wireshark capture for second use-case.

The first use-case is about the IP layer, where our system makes the bandwidth overheads of ML-INT be adaptive to the fluctuation of application traffic. As the in-band way lets PDP-SWs insert ML-INT data as header fields in packets, the induced bandwidth overheads can cause heavy-loaded or even congested traffic conditions if the monitoring scheme is not adaptive. Our proposal overcomes this issue by implementing a traffic predictor in each DCA, and letting the DCA report an alarm to the controller when the traffic prediction indicates that the current monitoring scheme will not be suitable in the future. The bandwidth overheads of ML-INT can be reduced, if we select a smaller portion of packets to insert ML-INT data [3]. Therefore, upon receiving the alarm from a DCA, the controller will leverage TOrch to determine the new selection ratio and update the monitoring scheme accordingly. The experiment makes *Host A* generate dynamic traffic whose data-rate varies within $[2, 10]$ Gbps according to a realistic trace. Fig. 2(b) shows the Wireshark capture of the control messages used in the system to achieve self-adaptive monitoring. The bandwidth usage at the output of *PDP-SW 6* is plotted in Fig. 3(a), which indicates that our system always adjusts the selection ratio in advance to adapt to future traffic. Fig. 3(a) also shows the bandwidth usage of non-adaptive monitoring, where the bandwidth overheads of ML-INT can cause heavy-loaded traffic conditions from time to time.

The second use-case is about the EON layer, where our system makes the out-of-band optical monitoring be adaptive to the status of lightpaths. Note that, the out-of-band way uses an OCM to scan the spectrum at each optical port and analyzes/reports the results to the controller. However, there are tradeoffs between the resolution and frequency of the scanning and the accuracy and timeliness of the monitoring, *e.g.*, using a finer resolution brings more accurate spectral results, but it also increases the amount of data reported to the controller. Our system analyzes the ML-INT data collected with the in-band way, and utilizes the data analytics results to guide the adjustment of out-of-band optical monitoring. Specifically, the experiment uses the OLS to collect the pre-forward-error-correction (pre-FEC) BER of *Lightpath A-B-C-F*, and lets *PDP-SW 6* insert the BER results in packets as ML-INT data. Then, the DCA analyzes the data and reports an alarm to the controller when it sees abnormal BER changes. Upon receiving the alarm, the controller will change the resolution and frequency of the spectrum scanning for out-of-band optical monitoring accordingly. Here, the lightpath's channel width is 50 GHz, and we classify its BER as low, medium and high levels.

Initially, the lightpath works normally and its BER stays at $\sim 4.5 \times 10^{-8}$ (low level), and thus the scanning can just use a coarse granularity of 2.5 GHz and sample the spectrum every 5 seconds. Then, we introduce slight power loss on link *C-F*, which makes the BER increase to $\sim 1.3 \times 10^{-7}$ (medium level). Hence, the DCA reports an alarm to the controller, which will in turn ask the out-of-band optical monitoring to provide higher resolution spectra more frequently for anomaly detection. Specifically, as shown in Fig. 3(b) for $t \in [10, 14]$ seconds, the out-of-band monitoring gets changed to scan with a granularity of 625 MHz and sample the spectrum every 2 seconds. Next, we keep increasing the power loss to change the BER to $\sim 1.0 \times 10^{-6}$ (high level), and the DCA and the controller respond quickly at $t = 15$ seconds to change the granularity and period of the scanning to 312.5 MHz and one second, respectively. Fig. 2(b) shows the the control messages used to achieve the self-adaptive monitoring, while Fig. 3(c) plots the optical spectra collected when the BER results are in different levels. It can be seen that when the BER level is higher, the lightpath spectrum from the out-of-band optical monitoring has more spectral details.



(a) (b) (c)

Fig. 3. Experimental results on self-adaptive network monitoring in two use-cases.

4. Summary

This work proposed to multilayer telemetry with the data analytics on ML-INT data, for realizing self-adaptive network monitoring in IP-over-EONs. We implemented the proposed system and demonstrated its effectiveness experimentally.

References

[1] P. Lu *et al.*, "Highly-efficient data migration and backup for Big Data applications in elastic optical inter-data-center networks," *IEEE Netw.*, vol. 29, pp. 36-42, Sept./Oct. 2015.

[2] F. Paolucci *et al.*, "Network telemetry streaming services in SDN-based disaggregated optical networks," *J. Lightw. Technol.*, vol. 36, pp. 3142-3149, Aug. 2018.

[3] B. Niu *et al.*, "Visualize your IP-over-optical network in realtime: A P4-based flexible multilayer in-band network telemetry (ML-INT) system," *IEEE Access*, vol. 7, pp. 82413-82423, 2019.

[4] S. Tang *et al.*, "Sel-INT: A runtime-programmable selective in-band network telemetry system," *IEEE Trans. Netw. Serv. Manag.*, vol. 17, pp. 708-721, Jun. 2020.

[5] B. Ran *et al.*, "PINT: Probabilistic in-band network telemetry," *in Proc. of ACM SIGCOMM 2020*, pp. 662-680, Aug. 2020.

[6] Z. Zhu *et al.*, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15-22, Jan. 2013.