Programmable Multilayer INT: An Enabler for AI-assisted Network Automation

Shaofei Tang, Jiawei Kong, Bin Niu, and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—Recently, the fast development of backbone networks has made the traffic, services and infrastructure of packet-over-optical networks increasingly complicated. This stimulates the research and development on fine-grained and realtime performance monitoring and troubleshooting. In this paper, we propose a programmable multilayer in-band network telemetry (ProML-INT) system, which can visualize a packet-over-optical network in realtime, and enable customized performance monitoring and troubleshooting. We introduce the system design in detail, and explain how to control the overhead of multilayer INT (ML-INT) by inserting INT fields in packets selectively. The ProML-INT system is experimentally demonstrated in a small-scale but real packet-over-optical network testbed. The experimental results confirm that our proposal can monitor packet and optical layers jointly in realtime, and the home-made data analyzer in it can leverage artificial intelligence (AI) to identify the root-causes of exceptions in packet-over-optical networks correctly and timely.**

***Index Terms*—Multilayer in-band network telemetry (ML-INT), Programmable data plane (PDP), Packet-over-optical networks, Artificial intelligence (AI), Data analytics.**

I. Introduction

DRIVEN by the ever-increasing pressure from the edge for many years, backbone networks, which usually take the multilayer architecture of "packet-over-optical", have undergone continuous changes in terms of traffic, services and infrastructure. Consequently, the conventional network control and management (NC&M) for packet-over-optical networks, which was designed to address slowly-varying network status with semi-permanent configurations, is facing great challenges to adapt to the new network environments [1]. Meanwhile, recent advances on knowledge-defined networking (KDN) have suggested that the symbiosis of software-defined networking (SDN) and artificial intelligence (AI) will facilitate AI-assisted network automation [2, 3]. Hence, future NC&M for packet-over-optical networks should include AI-assisted network automation, such that intelligent NC&M decisions can be made automatically and timely to arrange highly-dynamic traffic flows over packet and optical layers coordinately, for high resource utilization and enhanced quality-of-service (QoS) [4].

The AI-assisted network automation can hardly be realized without fine-grained and realtime performance monitoring and troubleshooting. This is because various failures can happen in a packet-over-optical network frequently and irregularly, and to identify their root-causes, one needs to analyze fine-grained telemetry data timely [5–7]. However, the existing techniques cannot fulfill these requirements on monitoring and troubleshooting. The limitations come from three perspectives. First of all, the monitoring and troubleshooting on a packet-over-optical network should be multilayer-capable to consider its packet and optical layers jointly. This will rule out the mainstream techniques that were designed for single-layer operations. Secondly, due to the complexity of a backbone network, the monitoring and troubleshooting on it should be fine-grained (*i.e.*, at flow level) and realtime. This will make the techniques that only collect telemetry data in the out-of-band manner (*e.g.*, the one in [8]) inapplicable. Finally, the monitoring and troubleshooting should provide the programmability to customize data collection and to balance the tradeoff between the accuracy and overhead of monitoring.

S. Tang, J. Kong, B. Niu, and Z. Zhu are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, P. R. China (Email: zqzhu@ieee.org).

Manuscript received on June 5, 2019.

Recently, the progress on programmable data-plane (PDP) [9, 10] has promoted the idea of in-band network telemetry (INT) [11], which encodes realtime statistics regarding packet switches in transient packets, and aggregates and processes the statistics at network edges in the distributed way. INT opens up new opportunities to realize fine-grained and realtime monitoring and troubleshooting on packet-over-optical networks. This inspires us to consider multilayer INT (ML-INT), which can collect statistics from the network elements (NEs) in both packet and optical layers, and encode them in packets.

In this article, we first review the background of this study and explain the motivations and challenges of realizing ML-INT. Then, we lay out the architecture of our programmable ML-INT (ProML-INT) system, which introduces relatively small overhead to visualize a packet-over-optical network in realtime and enables customized performance monitoring and troubleshooting. Next, we elaborate on the system design of ProML-INT and discuss how to implement it. Experimental demonstrations are also presented to show the advantages of our proposal. Finally, we summarize the article.

II. Review of Background

A. Programmable Data-Plane (PDP)

The major improvements brought by PDP are that a PDP switch will not be restricted by the existing network protocols anymore, and it is capable of defining packet fields, match actions and packet processing procedure, all in the arbitrary manner. Programming protocol-independent packet processor (P4) based PDP [9] provides a high-level programming language to customize packet formats and the processing procedure of packets in PDP switches. With the support from commercial hardware, P4-based PDP has recently attracted intensive interests from both academia and industry. A

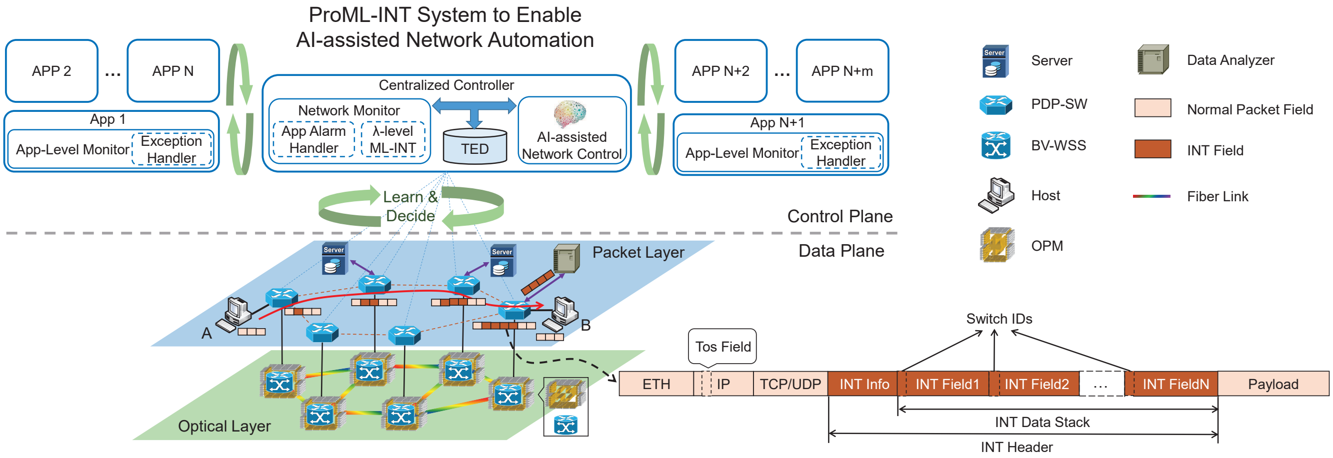

Fig. 1. Overall architecture of our ProML-INT system, App: application, TED: traffic engineering database, PDP-SW: PDP switch, BV-WSS: bandwidth-variable wavelength-selective switch, OPM: optical performance monitor.

PDP switch can also be realized by leveraging the protocol-oblivious forwarding (POF) [10]. POF refers to a packet field with the tuple <*offset*, *length*> (*i.e.*, *offset* denotes the start location of the field in a packet and *length* describes its length), and defines the protocol-oblivious forwarding instruction set (POF-FIS) to handle packet fields in the form of the tuple. One advantage of POF-based PDP switches is that they are runtime-programmable, *i.e.*, an SDN controller can modify the packet processing pipelines in them dynamically by updating the corresponding flow tables in runtime.

B. *In-band Network Telemetry (INT)*

INT was initially proposed to enable a network operator to monitor its packet network in realtime. Specifically, according to the technical specification published in [11], the INT-enabled switches along a flow's routing path can encode their statistics (*e.g.*, packet processing latency and port throughput) as INT fields and inserts them in each packet that belongs to the flow. Therefore, after aggregating and analyzing the statistics stored in INT fields, we can reveal how each packet is handled at each hop. In other words, INT realizes end-to-end monitoring at flow level and thus improves the realtime visibility on a packet network significantly. Both P4-based and POF-based PDP switches support INT. Despite the advantages, INT suffers from two drawbacks: 1) it may affect the packet processing throughput of a PDP switch due to the insertion of INT fields, and 2) it may generate excessively long packets when the flow has to be routed over many hops and/or the statistics to record for each hop are many.

C. *Recent Advances on Packet-over-Optical Networks*

Packet-over-optical networks integrate the benefits of packet switching and circuit switching to not only provision huge bandwidth but also handle bursty and dynamic traffic cost-effectively [4]. Specifically, in a packet-over-optical network, the optical layer establishes lightpaths to achieve high-throughput data transmission over long distances, while the switches/routers in the packet layer use the lightpaths as underlying pipes and switch packets among them. However, for the optical layer, the traditional fixed-grid wavelength-division multiplexing (WDM) networking is still rigid and spectrum-inefficient to support today's Internet services [1, 12]. Hence, flexible-grid elastic optical networking (EON) has been developed to bring the spectrum allocation granularity in the optical layer down to 12.5 GHz or even narrower [13–15]. To this end, a backbone network that takes the form of "packet-over-EON" would be promising, since adaptivity and agility are provided in both the packet and optical layers.

D. *Challenges for Realizing ML-INT*

Note that, extending INT to the ML-INT for packet-over-optical networks is not an easy task, for the following three challenges. Firstly, since ML-INT needs to have an optical performance monitor (OPM) to collect realtime statistics from optical network elements (NEs) and report them to PDP switches, we need to design and optimize the interaction between the OPM and the packet processing module in a PDP switch. More specifically, the interaction mechanism should arrange the statistics from electrical/optical NEs in the way that the time correlation between the packet and optical layers is preserved, and in the meantime, the mechanism should not affect the packet processing performance of a PDP switch or the negative effect should be minimized.

Secondly, as ML-INT needs to monitor many more statistics than those in the conventional INT, how to encode the statistics in packets should be carefully designed. In other words, we cannot simply insert all the statistics in each packet as the conventional INT does, because this would not only bring unbearable overheads but also generate excessively long packets. Note that, when the restriction from the maximum transmission unit (MTU) is present, excessively long packets introduce the complexity of packet fragmentation and concatenation.

Finally, the processing of the collected statistics is also a challenging task. We need to design a high-performance data analyzer to extract, parse and analyze the INT data carried by high-speed packet flows, and then introduce AI-assisted data analytics to abstract knowledge regarding NC&M from the data, for network automation. In our previous study [6], we presented the initial design of our ML-INT, but did not try to leverage AI-assisted data analytics to process the obtained telemetry data for network monitoring and troubleshooting. In the following, we will elaborate on our recent efforts to expand the ML-INT designed in [6] to a programmable ML-INT (ProML-INT) system that can utilize AI-assisted data analytics to detect and distinguish five types of network exceptions.

III. Programmable ML-INT for Visualizing Packet-over-Optical Networks

Fig. 1 shows the overall architecture of our ProML-INT system to visualize a packet-over-optical network in realtime, for enabling AI-assisted network automation. The functionality of ProML-INT is realized in the data plane that includes packet and optical layers. The packet layer consists of PDP switches, application hosts and data analyzers. The PDP switches can be either P4-based or POF-based, and they realize ProML-INT to insert realtime statistics into packets as INT fields.

Specifically, if a traffic flow is chosen for being monitored, the control plane instructs the ingress and subsequent PDP switches on its routing path to select a small portion of its packets to insert INT fields that store collected statistics. The details regarding this operation will be explained in the next section. Each INT field carries a statistic of an electrical/optical NE on the flow's routing path. For example, an INT field can contain the throughput of a PDP switch's output port or the power-level at the input port of a bandwidth-variable wavelength-selective switch (BV-WSS). Here, on each BV-WSS, we implement an OPM to collect statistics about the lightpaths going through it, *e.g.*, optical signal-to-noise ratio (OSNR), central wavelength, and power-level. If required, the PDP switch local to the BV-WSS can specify the lightpath(s) to monitor and poll related statistics from the OPM. Finally, the flow's egress PDP switch removes all the INT fields from the packets (*i.e.*, making ProML-INT be transparent to application hosts), and sends the extracted INT fields to a data analyzer for being aggregated, stored and analyzed.

The control plane of the ProML-INT system utilizes a hybrid centralized/distributed scheme for performance monitoring and troubleshooting. This scheme can avoid flooding the controller with huge volumes of telemetry data, and provide each application the flexibility to customize its performance monitoring and troubleshooting. Network applications can use ProML-INT to realize end-to-end, flow-level and distributed monitoring and troubleshooting, and they will flag alarms to the centralized controller when seeing exceptions based on their QoS requirements. This is because each application can have its own concerned network statistics. Therefore, each application-level monitor defines exceptions based on the application's QoS requirements and analyzes telemetry data distributedly, and then based on its preliminary analysis, it sends alarms to the controller to not only report exceptions but also provide suggestions on how to resolve them. Meanwhile, the controller performs coarse lightpath-level monitoring (*i.e.*, also through ProML-INT), obtains the applications' service provisioning schemes from its traffic engineering database (TED), and utilizes an AI module to analyze the applications' alarms together with its own lightpath-level monitoring results, for making timely and intelligent NC&M decisions.

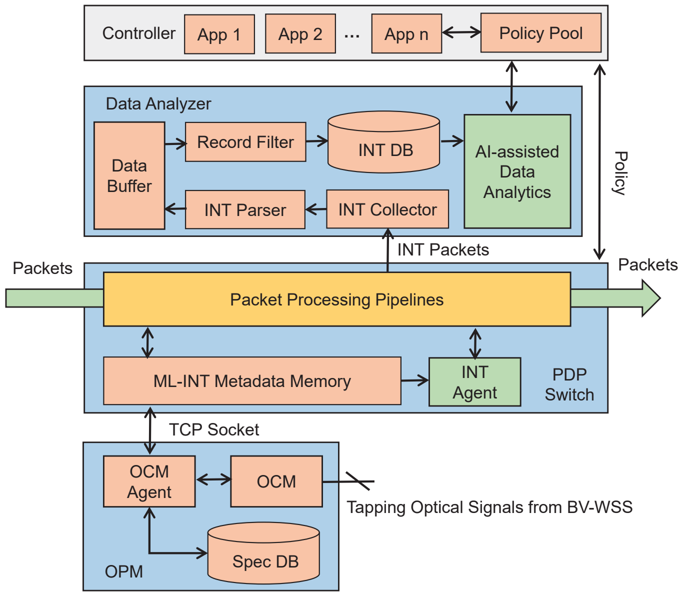

Fig. 2. System design of ProML-INT, Spec DB: Optical spectrum database, OCM: Optical channel monitor, INT DB: INT database.

IV. System Design and Implementation

A. Programmable ML-INT

The system design to realize ProML-INT is illustrated in Fig. 2. At the bottom, we have the OPM to tap optical signals from a BV-WSS to monitor lightpaths. In the OPM, the optical channel monitor (OCM) is a commercial device, and it realizes high-resolution optical spectrum analysis on the tapped optical signals. The spectrum data is sent to the OCM Agent, which further analyzes it to extract the statistics regarding lightpaths and then stores them in the optical spectrum database (Spec DB). Meanwhile, the OCM Agent can communicate with a PDP switch using a TCP socket, and report the lightpaths' statistics to the switch's ML-INT metadata memory.

In the PDP switch, the ML-INT metadata memory buffers the most updated statistics regarding the packet and optical layers, while the INT Agent organizes the statistics and prepares them for being inserted in packets by the packet processing pipelines. Here, to reduce the overhead of ProML-INT, we design the packet processing pipelines to: 1) only select a small portion of packets in a flow to insert INT fields, according to a sampling rate that is runtime programmable, and 2) ensure that each selected packet only carries a part of the required statistics regarding the electrical/optical NEs on the flow's routing path. This selective INT insertion is reasonable due to the fact that backbone networks usually have line-rates at 10 Gbps or higher, and thus sampling network statistics in a per-packet basis could be unnecessary. For instance, at 10 Gbps, the interval between two 1500-byte packets is only 1.2 μs, and the statistics of a backbone network would not change dramatically within such short time.

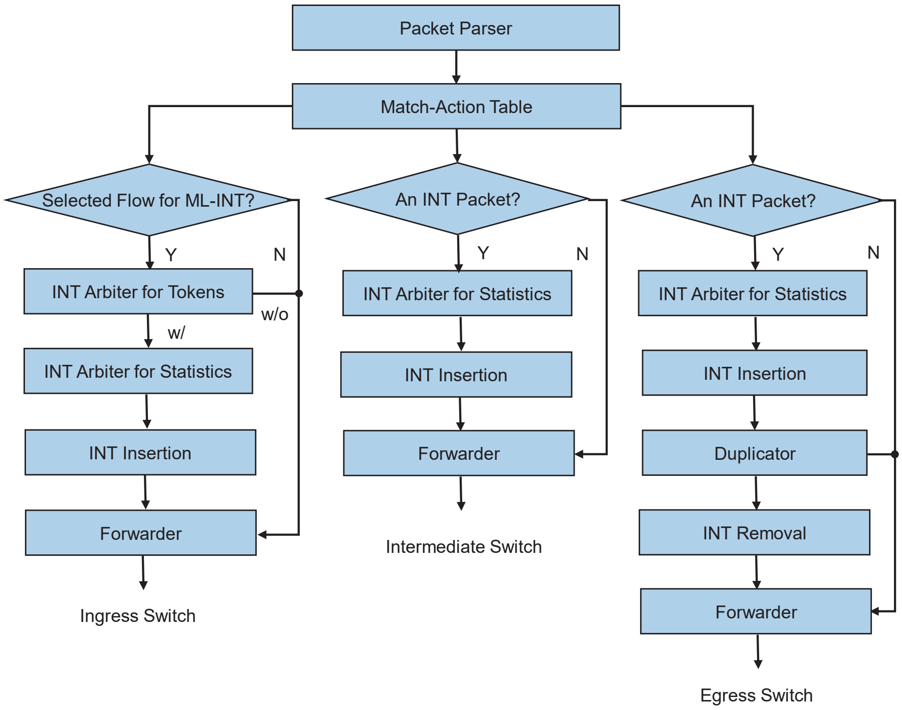

Fig. 3. Flow chart of the packet processing pipelines for ProML-INT.

We extend the INT packet format defined in [11] to that shown in Fig. 1. First of all, since ProML-INT does not encode INT fields in each packet of a flow, we leverage the *ToS* field in IP header to identify a packet with INT fields (*i.e.*, an INT packet). Specifically, we set the *ToS* field in an INT packet to a specific value. Secondly, for each INT packet, an INT header is inserted after its TCP/UDP header. The INT header includes an *INT Info* field followed by a series of *INT Fields*. The information of the whole INT header is stored in the *INT Info* field, including the version number, the number of inserted *INT Fields*, and the space left for more *INT Fields*. Each *INT Field* corresponds to a hop on the packet's routing path, and the hop refers to the corresponding lightpath in the optical layer and the lightpath's destination PDP switch in the packet layer. Again, to limit the overhead of ProML-INT, we empirically define the number of statistics to be included in an *INT Field* as two. For the two statistics, the first one is mandatorily chosen as the *Switch ID* of the PDP switch that inserts the *INT Field*, and the second one can be an arbitrary statistic supported by ProML-INT. If there are more required statistics of the hop, we distribute them in different INT packets, and the data aggregation will be done afterwards by the data analyzer.

In the data analyzer, the INT collector captures the INT packets sent from the data path. The INT parser extracts the INT headers from the INT packets, parses the encoded statistics, and time-stamps and stores them in the data buffer temporarily. Next, the record filter will read the statistics from the data buffer, and filter out redundant records before writing them into the INT database (INT DB). The telemetry data stored in the INT DB will then be processed by the AI-assisted data analytics module, for performance monitoring and troubleshooting. We train the AI-assisted data analytics module in the offline manner. This means that the trained module will process telemetry data samples quickly to ensure realtime network monitoring and troubleshooting, because the time-consuming training phase will not happen during operation. Meanwhile, the AI-assisted data analytics module also communicates with the controller to report exceptions. The controller manages the service provisioning schemes for applications, and also, it may update the policy of ProML-INT (*e.g.*, the sampling rate for the INT insertion on a flow and the required statistics regarding an NE) in runtime if necessary.

B. *Packet Processing Pipelines for ProML-INT*

The packet processing pipelines are the key enabling components of ProML-INT, and they can be customized by the controller in runtime to define the following parameters: 1) the sampling rate for selective INT insertion, 2) the electrical/optical NEs for collecting statistics from, and 3) the required statistics regarding each concerned NE. Fig. 3 explains the operation principle of the packet processing pipelines, which are designed for the ingress, intermediate, and egress switches on the routing path of a flow.

When a packet arrives at the ingress switch to a packet-over-optical network, the PDP switch will first check whether the packet belongs to a flow that has been selected for ML-INT. If yes, the "INT arbiter for tokens" decides whether the packet should be selected for INT insertion according to a token-based mechanism that generates tokens based on the preset sampling rate. The ingress switch will only insert an INT header on the packet, when it sees a token, and then the packet is converted into an INT packet, whose processing procedure will be different from that of normal packets in the subsequent switches. Next, the "INT arbiter for statistics" determines which statistic to encode in the *INT Field* for the INT packet, while the *INT Field* is inserted by the INT insertion module. Finally, the forwarder sends the INT packet out together with other normal ones.

The operation procedure of an intermediate switch is simpler, and the switch only inserts an *INT Field* in an INT packet. Note that, before inserting the *INT Field*, the switch hypothetically checks whether the resulting packet length would be longer than the MTU of its network. If yes, the switch will not perform the INT insertion. The operation in the egress switch is similar, with the only exceptions that the switch duplicates each INT packet and sends it to a data analyzer, and removes the INT header from an INT packet before forwarding it to the destination host.

C. AI-assisted Data Analytics

Since each INT packet only contains a part of the required statistics on all the NEs over its routing path, a data analyzer first needs to aggregate the statistics in different INT packets to obtain a complete realtime view about the NEs. We design the AI-assisted data analytics module in the data analyzer based on a deep neural network (DNN), which can be utilized to detect the root-causes of exceptions after offline supervised training. Specifically, the AI module takes the time series of the statistics collected on the NEs along a flow's routing path as inputs, and outputs the root-causes (*e.g.*, low input power, degraded OSNR, packet layer congestion, and PDP switch misconfiguration) when seeing exceptions. The DNN in the AI module has one input layer, four hidden layers, and one output layer, where the neurons in them are all fully connected. The input layer consists of 5 nodes and uses them to take in multilayer telemetry data samples, each of which is 5-dimensional (*i.e.*, packet processing latency, input bandwidth, output bandwidth, input power, and OSNR). For the hidden layers, each of them consists of 128 neurons with Sigmoid or ReLU activation functions. The output layer contains 5 nodes, each of which corresponds to a root-cause of exceptions.

To obtain the training data set, we emulate various exceptions in a packet-over-optical network testbed, utilize the ProML-INT system to collect realtime telemetry data, and label the data with the corresponding root-causes. Then, we take 90% of the data as the training set, while treat the remaining 10% as the testing set. In the training, we define the loss as the categorical cross-entropy. The training adjusts the DNN's parameters to make the categorical cross-entropy less than a preset threshold, and then we use the testing set to check the performance of the AI module. We repeat the aforementioned training/testing operations until the classification accuracy of the AI module cannot be improved any more.

The ProML-INT system can get the fine-grained telemetry data regarding a multilayer packet-over-optical network within one millisecond. Meanwhile, after being trained in the offline manner, our AI-assisted data analytics module can process each telemetry data sample for troubleshooting within one millisecond too. Hence, the overall latency in the ProML-INT system for network monitoring and troubleshooting will be in the scale of milliseconds, which means that it can react timely to exceptions in the multilayer network and make intelligent NC&M decisions in realtime.

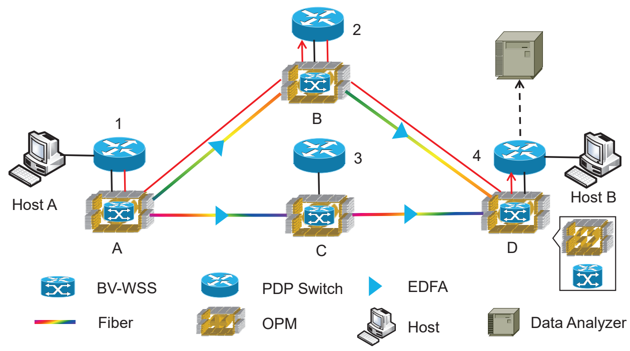

Fig. 4. Experimental setup, EDFA: Erbium-doped fiber amplifier.

V. Experimental Demonstration

A. Experiment Setup

We use the experimental setup in Fig. 4 to demonstrate a prototype of our ProML-INT system. The packet layer consists of four P4-based PDP switches, each of which equips with 10 GbE optical ports. The two hosts (*i.e.*, *Hosts* A and B) are emulated with commercial traffic generators/analyzers. The optical layer is based on EON, and includes commercial 1×9 BV-WSS', erbium-doped fiber amplifiers (EDFAs), and fiber links. Each BV-WSS operates within $[1528.43, 1566.88]$ nm with a 12.5 GHz spectrum allocation granularity. The red arrowed lines indicate the routing path of an application flow that has a throughput of 8 Gbps and needs to have ProML-INT. The packet size of the application flow is set as $1,024$ bytes, and the sampling rate for the INT insertion is 50%.

Then, for the flow, we generate various types of exceptions intentionally. For example, we attenuate the power of its lightpaths to emulate power loss, inject wide-band noise in its lightpath to generate OSNR degradation, increase the volume of the background traffic on its lightpaths to emulate packet layer congestion, and install incorrect flow tables in the PDP switches on its routing path to emulate switch misconfiguration. For each case, we monitor the receiving bandwidth of the flow at *Host* B, and tag the collected statistics as those for exceptions when the receiving bandwidth is less than 90% of that from *Host* A. Here, we assume that there is only one type of exception at a time. The telemetry data is time-stamped and labeled with the corresponding root-causes, and we use it as the training data set to let the AI module learn the correlations between the ML-INT results and various types of exceptions.

B. Experimental Results

We first verify the processing throughput of our home-made data analyzer, and the experimental results indicate that a single data analyzer can extract, parse, filter and store the *INT Fields* from packets with an arrival rate up to 2 million packets per second (Mpps). This processing throughput ensures that the data analyzer can handle the INT packets in the 8 Gbps application flow from *Host* A to *Host* B without any difficulty.

Then, with the data analyzer, we leverage the procedure mentioned in the previous subsection to collect around $18,000$ INT data samples regarding the flow's transmission over *Link* B-D. All the data samples are tagged, and we divide them into training and testing sets, which includes 90% and 10% samples, respectively. For instance, Fig. 5 plots partial training samples regarding the optical power and OSNR of the lightpath that goes into *BV-WSS* D. There are three types of samples in Fig. 5, *i.e.*, normal, low input power, and degraded OSNR, and they indicate that it would be difficult to pick out the exception cases with a simple threshold-based scheme. This explains the necessity of introducing AI-assisted data analytics in the performance monitoring and troubleshooting.

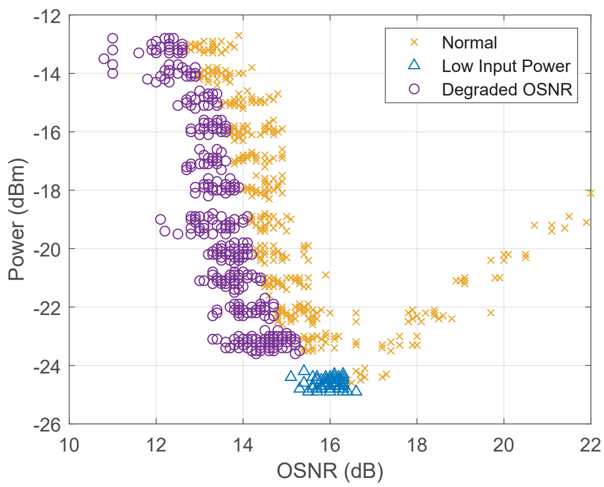

Fig. 5. Training samples regarding optical power and OSNR.

After being trained, the AI-assisted data analytics module performs well on the testing set. Specifically, it can identify packet layer congestion and PDP switch misconfiguration with 100% accuracy, find the exceptions due to low input power and degraded OSNR with accuracies of 96.35% and 98.64%, respectively. Moreover, if we mix the exception cases together, its overall classification accuracy on the testing set is 96.99%.

Next, we conduct an experiment with the trained AI-assisted data analytics module to demonstrate that our ProML-INT system can achieve fine-grained and realtime performance monitoring and troubleshooting for packet-over-optical networks. This time, we still send the 8 Gbps application flow from *Host* A to *Host* B, and run the flow for an hour. During this period, we generate the exceptions of low input power to *BV-WSS* B, degraded OSNR to *BV-WSS* B, and packet layer congestion in *PDP Switch* 2. Meanwhile, the data analyzer extracts and aggregates the statistics carried by INT packets, time-stamps them, and then leverages the AI-assisted data analytics module to process them, all in the realtime manner. The experimental results for realtime performance monitoring and troubleshooting are illustrated in Fig. 6, which shows the INT data samples labeled by the data analyzer. Note that, to show the results clearly, we omit a few similar samples here.

The experimental results in Fig. 6 further confirm that our ProML-INT system can monitor the packet and optical layers in the realtime manner, preserve the temporal correlations among the monitoring results, and identify the root-causes of exceptions correctly and timely. Therefore, the network operator can leverage it to abstract knowledge regarding NC&M and realize knowledge-defined network automation.

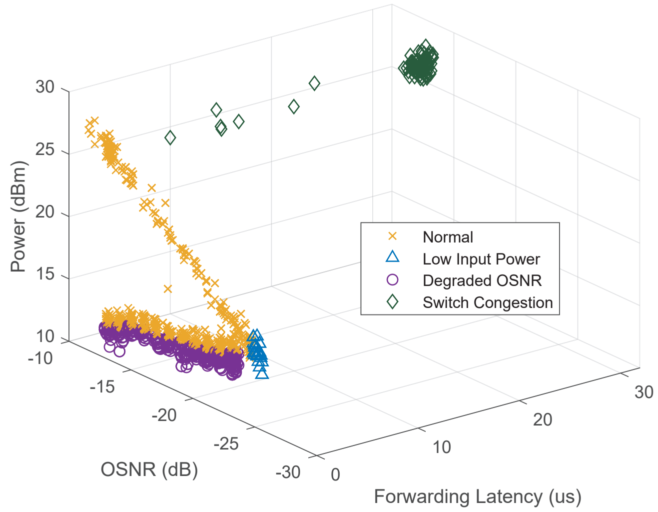

Fig. 6. Results from ProML-INT for realtime performance monitoring and troubleshooting.

VI. Discussion and Conclusions

In this paper, we introduced our proposal of the ProML-INT system, which can visualize a packet-over-optical network in realtime, and enable customized performance monitoring and troubleshooting. We discussed its system design in detail, and explained how to control the overhead of ML-INT by inserting INT fields in packets selectively. The implementation of the ProML-INT system was also described, and then the whole system was experimentally demonstrated in a small-scale but real packet-over-optical network testbed. The experimental results suggested that our ProML-INT system can monitor packet and optical layers jointly in realtime, and the AI-assisted data analytics module in it can identify the root-causes of exceptions correctly and timely. Hence, our ProML-INT system can be considered as a potential enabler for AI-assisted network automation in packet-over-optical networks.

Meanwhile, our ProML-INT system could still be improved in the following aspects. Firstly, the control plane can be further enhanced to close the loop of performance monitoring and troubleshooting. More specifically, after determining the root-causes of exceptions, the control plane should apply proper strategies to reconfigure the packet-over-optical network timely for failure recovery. Secondly, the AI-assisted data analytics module should be further studied to ensure good scalability and cover exceptions in a more comprehensive manner. For instance, the combined effects of multiple simultaneous exceptions should be investigated with it.

Acknowledgments

This work was supported in part by the NSFC projects 61871357 and 61701472, and CAS Key Project (QYZDY-SSW-JSC003).

REFERENCES

[1] P. Lu *et al.*, “Highly-efficient data migration and backup for Big Data applications in elastic optical inter-datacenter networks,” *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.
[2] A. Mestres *et al.*, “Knowledge-defined networking,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, pp. 2–10, Jul. 2017.
[3] B. Li, W. Lu, S. Liu, and Z. Zhu, “Deep-learning-assisted network orchestration for on-demand and cost-effective vNF service chaining in inter-DC elastic optical networks,” *J. Opt. Commun. Netw.*, vol. 10, pp. D29–D41, Oct. 2018.
[4] S. Liu *et al.*, “DL-assisted cross-layer orchestration in software-defined IP-over-EONs: From algorithm design to system prototype,” *J. Lightw. Technol.*, vol. 37, pp. 4426–4438, Sept. 2019.
[5] Y. Kim, D. Suh, and S. Pack, “Selective in-band network telemetry for overhead reduction,” in *Proc. of CloudNet 2018*, pp. 1–3, Oct. 2018.
[6] 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. 82 413–82 423, 2019.
[7] F. Cugini *et al.*, “P4 in-band telemetry (INT) for latency-aware VNF in metro networks,” in *Proc. of OFC 2019*, pp. 1–3, Mar. 2019.
[8] S. Yan *et al.*, “Multilayer network analytics with SDN-based monitoring framework,” *J. Opt. Commun. Netw.*, vol. 9, pp. A271–A279, Feb. 2017.
[9] P. Bosshart *et al.*, “P4: Programming protocol-independent packet processors,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87–95, Jul. 2014.
[10] S. Li *et al.*, “Protocol oblivious forwarding (POF): Software-defined networking with enhanced programmability,” *IEEE Netw.*, vol. 31, pp. 12–20, Mar./Apr. 2017.
[11] C. Kim *et al.*, “In-band network telemetry (INT),” *Tech. Spec.*, Jun. 2016. (Accessed on Mar. 26, 2019). [Online]. Available: https://p4.org/assets/INT-current-spec.pdf
[12] L. Gong and Z. Zhu, “Virtual optical network embedding (VONE) over elastic optical networks,” *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.
[13] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, “Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing,” *J. Lightw. Technol.*, vol. 31, pp. 15–22, Jan. 2013.
[14] L. Gong *et al.*, “Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks,” *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.
[15] Y. Yin *et al.*, “Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks,” *J. Opt. Commun. Netw.*, vol. 5, pp. A100–A106, Oct. 2013.