Proactive Performance Monitoring in Software Defined Networking

Lian-Kuan Chen, and Hon-Tung Luk
The Chinese University of Hong Kong, Hong Kong
lkchen@ie.cuhk.edu.hk

Abstract: To facilitate fast performance monitoring, proactive monitoring in Software Defined Networking is proposed and different monitoring strategies are discussed. The reduction of the acquisition time and the additional blocking probability induced are investigated.

OCIS codes: (060.4250) Networks; (060.4257) Networks, network survivability; (060.2300) Fiber measurements

1. Introduction

Efficient and agile control-plane design is crucial for emerging flexible, on-demand optical networks [1]. The Software Defined Networking (SDN) is a promising network architecture that provides highly flexible network control and enables programmable network functions. SDN is a networking approach that allows network administration of services by abstracting the lower layer functions [2-3]. The network is decoupled into two layers, the control plane (for routing and resource allocation control) and data plane (for data forwarding). The abstraction of the control function from data forwarding function makes the decision of routing and resource allocation programmable. To define the communication between the control plane and data plane, controlling protocols such as OpenFlow is used. The OpenFlow architecture is an open and standardized controlling protocol that enables access to the forwarding plane of the network switch [4-6]. By manipulating the forwarding plane of the network switch, the routing path of data packets can be controlled.

In today's networks, routing and flow control are mainly performed independently by individual routers [7-8], and thus it leads to inefficient use of network resources, non-optimized routing arrangements and difficulty in handling dynamic traffic demands. In order to provide customized, dynamic and optimized network control, a centralized controlling system is suggested to coordinate the operations across different routers by utilizing the SDN architecture [9-10]. With the help of programmable routing tables and centralized controls, complex network routing functions such as impairment aware routing can be enabled [11]. This calls for optical performance monitoring (OPM) to provide necessary network information for proper and timely SDN operations. The information obtained from OPM is beneficial for the optimization, coordination and routing arrangement performed by the centralized controlling system of SDN. Also OPM facilitates fault management and supports the operations of network protection and fast restoration.

On the other hand, OPM can be performed more efficiently with the assistance of SDN. Using the same centralized controlling system in SDN, the monitoring process can be customized, optimized and adjusted responsively, in accordance with dynamic network needs. The accuracy and monitoring speed can be improved and the cost can be reduced. Accurate, fast and low cost OPM is the key for the establishment of efficient SDN controlling system.

2. Passive monitoring and proactive monitoring

Network monitoring can be divided into two groups, passive monitoring and proactive monitoring. If the monitoring information is extracted directly from in-service data signal when it is received, this monitoring process is termed as passive monitoring in this paper. On the other hand, if monitoring signal is deliberately generated and injected into the optical path for monitoring purpose, the monitoring process belongs to proactive monitoring. For example, flow-based monitoring, such as NetFlow, that samples the network flow at the router to obtain the statistical information of the flow belongs to passive monitoring [12]. Since passive monitoring can only extract monitoring information from the existing data signals directly, the obtained information is very limited. By utilizing customized monitoring signals, such as a pilot tone, for the desired parameters to be monitored, proactive monitoring exhibits higher flexibility and accuracy in measurement. Note that some links may have a lower chance to be traversed by data channels. The monitoring time required for obtaining sufficient flow samples from the data channels in those links may be too long to meet the requirement of fast OPM in dynamic reconfigurable networks. For such networks, the acquisition time for monitoring information needs to be short to assure the acquired monitoring information is not outdated. By creating monitoring channels for desired monitoring paths, proactive monitoring can substantially reduce the monitoring time.

To establish a monitoring system for SDN, proactive monitoring utilizing existing SDN infrastructure is very desirable. SDN has the advantages of programmability, agility and centralized management [13]. Programmability provides direct control to the paths of data signals and monitoring signals so that they can be manipulated easily.

Agility shortens the time required for changing routing paths and enables fast monitoring of dynamic network status. Centralized management guarantees that the monitoring schemes are globally optimized and coordinated.

3. Optimization strategies and monitoring schemes

Accuracy of the monitoring results can be improved by using optimized monitoring setups, monitoring signals and monitoring paths. By utilizing the network management functions provided by the control layer, the unoccupied transceivers can be utilized to set up an extra channel for injecting a proactive monitoring signal for desired paths [3]. Proactive monitoring can be performed using available wavelength channel in optical links.

Many passive OPM techniques have been proposed to monitor channel impairments during data transmission [14-16].With the assistance of proactive monitoring, the required monitoring time can be greatly reduced and the coverage of monitoring schemes can be improved [17-18]. Simultaneously probing multiple links is also possible and can reduce monitoring time when proactive monitoring is used. However, proactive monitoring generates additional traffic and may increase blocking probability, thus the number of wavelength channels used by proactive monitoring signals should be limited to avoid jamming the network.

The cost for monitoring can be divided into two categories, Capital Expenditures (CapEx) and Operational Expenditures (OpEx). The CapEx includes monitoring devices and additional equipment to inject and tap monitoring signals. The OpEx includes the computation power for monitoring channels optimization, the additional channel usage for monitoring signals and the energy used for generating monitoring signals [19-20]. To minimize CapEx, the number of monitoring transmitters and detectors should be minimized. The investigation of Ho gives the lower bound of the number of monitoring transmitters and detectors for a proactive monitoring scheme to operate properly [21]. In order to minimize OpEx, it is desirable to minimize the total number of hops traversed by the proactive monitoring signals, so that the number of wavelength channels used for proactive monitoring and the energy consumed can be reduced. In [22], monitoring schemes aiming at reducing the cost for monitoring linearly-accumulated impairments have been investigated. It can be shown that the number of hops used for proactive monitoring can be further reduced when some of the links in the desired paths are overlapped.

4. Simulation results

New monitoring schemes have been proposed for reducing the acquisition time by using proactive monitoring to assist passive monitoring [23]. Since proactive monitoring may generate additional traffic, efforts have been made to attain short monitoring time and maintain reasonable blocking probability. Based on the arrangement of the number of proactive monitoring channels to be generated in different monitoring sessions and the priority of the links to be monitored by proactive monitoring, nine different monitoring schemes are proposed in [23]. The duration of each monitoring session is the time required for one proactive monitoring. In this paper, we investigate the performance of the nine different schemes applied on three different networks, namely, NSFNET, Bellcore, and ARPA2 network. In the following, we use A and I to represent the schemes using the same number and increasing number for the number of proactive monitoring channels to be generated in sequential monitoring sessions, and C for constant summation for the data and proactive monitoring channels in two adjacent monitoring sessions, respectively. We use R, B, and U to represent the schemes using randomly selection, probing busy edges first and probing unpopular edges first as the probing priority of the proactive monitoring channels. 10,000 iterations are established.

First, we study the monitoring time required (T_{total}) to achieve a given percentage of coverage for the three networks, as shown in Table 1. The parameter values used in the simulation are as follows. The average waiting time, $1/\lambda$, of arrival and departure of data channels is set to be 0.5 time units. The time for a monitoring session, T, is also set to be 0.5 time units. The total number of proactive monitoring channels, P_{total}, is 10 in the given monitoring time T_{total}. The last column in Table 1 is the case without proactive monitoring channels for reference. The maximum number of concurrent probing channels allowed, is P_{max}, is equal to 3. Simulation results show that the schemes with increasing probing number (I) outperform other schemes on monitoring time reduction. In NSFNET, the scheme with increasing probing number and probing unpopular edge first (IU) takes 6 time units to achieve coverage of 98% of network links compared to 10.5 time units for the AU case and 24 time units for the case without proactive monitoring. Similar results can be obtained for the other two networks. With the assistance of proactive monitoring, it is shown that the monitoring time can be reduced by around 70%-77% of that without proactive monitoring for the three networks.

Next we study the additional blocking probability induced by proactive monitoring when the number of proactive monitoring channels (P_{total}) is fixed (Table 2). The parameter values used are the same as in the previous simulation except that P_{total} is set to 5 and T_{total} is set to 10. Simulation results show that probing unpopular edge first (U) has better performance on blocking probability reduction. In Bellcore network, the additional blocking probability induced by probing unpopular edge first (U) is substantially lower than other schemes, with less than 0.8% blocking probability induced. For NSFNET and ARPA2, the blocking probability induced by probing unpopular edge first (U) is less than 3.5% and 4%, respectively. The Bellcore network has higher tolerance to additional proactive monitoring channels because it has more edges (28) compared to the other two networks (21 for NSFNET and 25 for ARPA2). The same conclusion can be drawn for the three networks, i.e., the combination of increasing probing number (I) and probing unpopular edge first (U), namely the IU scheme, has the best overall performance. Note that in this study we limit the P_{total} in the whole monitoring cycle, T_{total}. On the other hand, we can also just limit P_{max} in each monitoring session and remove the P_{total} constraint, if we only consider the availability of transceiver in each monitoring session. Then the monitoring time can be reduced, at the expense of larger blocking probability induced.

Table 1. The monitoring time required (T_{total}) to achieve a given percentage of coverage.

	AR	AB	AU	IR	IB	IU	CR	CB	CU	P_{total} =0
NSFNET;98% coverage	>15	>15	10.5	6	6	6	9	9.5	7.5	24
Bellcore; 88% coverage*	>15	>15	9.5	7.5	8	7	12	>15	8.5	30
ARPA2; 98% coverage	13.5	>15	10.5	6.5	7	6.5	8.5	11.5	8	21.5

*: Note that 88% coverage is used for Bellcore network because a majority of those monitoring schemes cannot achieve 98% coverage within 15 time units for the given settings.

Table 2. The additional blocking probability induced by proactive monitoring with P_{total}=5.

	AR	AB	AU	IR	IB	IU	CR	CB	CU
NSFNET	6.13%	6.64%	3.3%	4.43%	4.87%	3.47%	6.01%	6.4%	3.43%
Bellcore	4.66%	7.94%	0.67%	2.96%	4.7%	0.79%	4.26%	7.48%	0.69%
ARPA2	5.01%	9.15%	3.93%	3.73%	5.39%	3.69%	4.96%	8.86%	3.95%

5. Conclusions

OPM is an essential component for network management to pursue reliable and flexible SDN. On the other hand, SDN also facilitates fast and efficient OPM with its capability of programmability, agility and centralized management. Proactive monitoring through SDN demonstrates great potentials of reducing acquisition time and increasing accuracy for OPM. We have investigated the reduction of the acquisition time for dynamic reconfigurable network with the assistance of proactive monitoring. We have also compared nine different proactive monitoring schemes on the monitoring time and the blocking probability of three different networks, NSFNET, Bellcore, and ARPA2 network. The simulation results show that the monitoring scheme IU (increase probing number and probe unpopular edge first) has the best overall performance under different network conditions for all three different networks. With the assistance of proactive monitoring, the monitoring time can be reduced by around 70%-77% of that without proactive monitoring for the three networks. This project is supported in part by HKSAR RGC grant (GRF 14200914).

References

[1] D. C. Kilper et al., J. Lightw. Technol., vol. 22, no.1 Jan. 2004.
[2] K. Sato, S. Okamoto, and H. Hadama., IEEE J. Sel. Area Comm.,vol. 12, no. 1, Jan. 1994.
[3] Ralph Ballart and Yau-Chau Ching, IEEE Comm. Magazine, vol. 27, no. 3, pp. 8-15, Mar. 1989.
[4] F. Paolussi et al., ECOC 2012, Paper Tu.1.D.5, 2012.
[5] N. McKeown et al., ACM SIGCOMM, vol. 39, no. 2, Apr. 2008.
[6] M. Channegowda et al., ECOC 2012, Paper Tu.1.D.2, 2012.
[7] Y. Minlan, L. Jose, and R. Miao, NSDI 2013, vol. 13. 2013.
[8] Subramanyan et al., Proceedings ACM/IEEE conference on Supercomputing, 2000.
[9] S. Sezer et al., IEEE Comm. Magazine, vol. 51, no. 7, pp. 36-43, Jul. 2013.
[10] D. Levin et al., HotSDN 12, pp. 1-6, 2012.
[11] S. Azodolmolky et al., J. Lightw. Technol., vol. 29, no. 4, Feb. 2011.
[12] L. Deri, Proceedings of Terena TNC 2003, 2003.
[13] http://en.wikipedia.org/wiki/Software-defined_networking
[14] W. Shieh et al., Opt. Express, vol. 15, no. 2, pp. 350-356, 2007.
[15] S. K. Shin, K. J. Park, and Y. C. Chung, OFC 2000, 182–184, Mar. 2000.
[16] G. Lu, M. Cheung, L. Chen, and C. Chan, IEEE Photon. Technol. Lett., vol. 17, no. 12, Dec. 2005.
[17] G. J. Pendock et al., IEEE Photon. Technol. Lett., vol. 20, no. 10, May 2008.
[18] B. Auguie et al., IEEE Photon. Technol. Lett., vol. 18, no. 17, Sep. 2006.
[19] Verbrugge et al., ONDM 2005, Milan, Italy. 2005.
[20] Channegowda et al., J. Opt. Commun. Netw.,vol.5, no. 10, A274-A282, 2013.
[21] S. T. Ho and L. K. Chen, J. Opt. Commun. Netw., vol. 1, No. 1, June 2009.
[22] H. T. Luk, and L. K. Chen, OECC 2012, Paper 5A4-2, Jul. 2012.
[23] H. T. Luk, and L. K. Chen, OFC 2015, Paper W4J.2, Mar. 2015.