Dynamic global load balancing strategy for cross stratum optimization of OpenFlow-enabled triple-M optical networks

Hui Yang (杨 辉)[1*], Yongli Zhao (赵永利)[1], Jie Zhang (张 杰)[1], Wanyi Gu (顾畹仪)[1], Shouyu Wang (王守宇)[1], Yi Lin (林 毅)[2], and Young Lee[2]

[1]*State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China.*

[2]*Huawei Technologies Co., Ltd., Shenzhen 518129 , China*

[*]*Corresponding author: yanghui@bupt.edu.cn*

Received February 4, 2013; accepted May 13, 2013; posted online July 3, 2013

With the emergence of high-bitrate applications, cross stratum optimization (CSO) attracts the interest of network operators because of its application in the joint optimization of optical networks and application stratum resources. Given the large-scale growth and high complexity of optical networks, achieving a more effective, accurate, and practical CSO becomes an important research focus. In this letter, we present a CSO-oriented, unified control architecture for OpenFlow-enabled triple-M optical networks. A novel dynamic global load balancing (DGLB) strategy with dynamic resource rating for CSO is presented based on the proposed architecture. The DGLB strategy is then compared with four other strategies by conducting experiments on a SOFT-based testbed with 1000 virtual nodes.

OCIS codes: 060.4510, 060.4250, 060.4254, 060.4256.

doi: 10.3788/COL201311.070605.

With the emergence of future internet services (e.g., high-bandwidth video streaming and large-bandwidth medical and financial data), various high-performance network-based data center (DC) applications make heavy use of optical network resources in the form of bandwidth consumption and require stringent quality of service (QoS) to control factors such as jitter and latency[1−3]. A great need for joint scheduling of network and application resources exists, in which the latter mainly refers to the storage and computational resources of various types and granularities (e.g., CPU, memory, and virtualization modules (VMs)) in DCs. Therefore, to meet the QoS requirements, cross stratum optimization (CSO) is proposed as an architecture that allows global optimization and control across optical networks and application stratum resources[4]. CSO can enhance the responsiveness to end-to-end DC demands and effectively reduce the probability of blocking[5]. Given the recent large-scale growth and high complexity of optical networks, earlier complex CSO algorithms (e.g., static load balancing (SLB) strategy and integer linear programming for CSO[5,6]) have been unable to adapt to multi-constraint, multi-domain, and multi-layer (i.e., triple-M) optical networks. Therefore, achieving a more effective, accurate, and practical CSO has become a challenge facing triple-M optical networks.

Recently, the software-defined network (SDN), a promising centralized control architecture enabled by the OpenFlow protocol, has gained popularity by supporting programmable optical network functionalities[7−9], which in turn allows applications to converse with the control software of optical network devices and resources[10]. SDN/OpenFlow aims to facilitate the rapid application of available network resources by using a centralized software control method and protocol[11,12]. Therefore, from an operator's point of view, OpenFlow-based unified control must be applied to achieve CSO, thereby allowing applications to customize network resources to enable the dynamic joint optimization of the cross-optical network and application resources at varying levels of granularity and to achieve cooperation in multi-domain scenarios. To the best of our knowledge, this concept has yet to be addressed.

In light of the above factors, this letter proposes a novel, CSO-oriented, unified control architecture in OpenFlow-enabled triple-M optical networks (SOFT), which contains hierarchical controllers that maintain cross stratum resources for CSO. A novel dynamic global load balancing (DGLB) strategy featuring a dynamic resource rating method for CSO is also introduced based on the proposed architecture. The key performance of the DGLB strategy is validated and compared with those of other resource selection strategies, namely, random-based strategy (RBS), application-based strategy (ABS), network-based strategy (NBS), and SLB strategy, by using a SOFT-based testbed with 1 000 virtual nodes.

SOFT is depicted as a scalable architecture for OpenFlow-enabled triple-M optical networks in Fig. 1(a). SOFT emphasizes the interworking between the application controller (AC) and service controller (SC) on the basis of OpenFlow to realize the CSO of application and network resources across the entire network. The centralized AC is responsible for maintaining application resources in DC, whereas SC sustains the resources of the multi-layer networks (e.g., WDM and OTN) virtualized from each local domain. The AC customizes the appropriate virtual network resources from related SCs on the basis of traffic requests by using an extended OpenFlow protocol (EOP). Additionally, DGLB, an engine used to choose the optimal destination in the AC, is

implemented based on both the application in question and the virtual network resources. The SC applies the service-aware path computation element (SA-PCE) algorithm according to the results of each related domain. Multiple SCs interact with security checks while completing collaboration among multiple domains by means of EOP. OpenFlow-enabled routers and optical nodes in SOFT are realized by extending the EOP rules[13]. The responsibilities and interactions among entities are provided in Fig. 1(b).

When the OpenFlow parser (OP) receives a new flow, OP maps this flow into request parameters and forwards the flow to the server selection engine (SSE). The certified request is transmitted to the application resource virtualization module (ARVM) for application resource processing. The ARVM responds to the SSE with the suitable virtual application resource obtained from the DC network. In turn, the SSE customizes the virtual network information with desirable granularity from the SC. After completing the DGLB, the SSE chooses the optimal server or virtual machine for users, allocates application resources, and determines the location of the application or the location where virtual machines are migrated. On the basis of the result, the AC transmits the application requirements to the related SCs via OP. When the data process (DP) in the SCs receives the server/VM location and the service type, the DP translates this profile into connection and service parameters within the transport network (e.g., bandwidth, delay, and jitter) and then forwards the network resource profile to the path computing entity plus (PCE+). The extended OpenFlow module (EOM) assigns the wavelength and establishes an end-to-end lightpath on the basis of the AC request by controlling all corresponding OpenFlow-enabled nodes along the computed path by using EOP. Communication between SCs through parallel processes

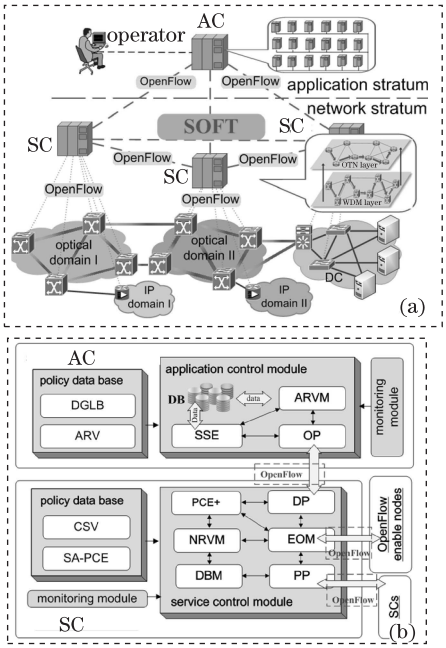

Fig. 1. (a) Architecture and (b) functional models of SOFT. DB: data base; NRVM: network resource VM; DBM: DB management; CSV: cross stratum virtualization.

(PP) meets the requirements of security, confidentiality, and virtual information interaction in the operator. Both AC and SC provide optional algorithms or strategies from a policy database to improve scalability. By monitoring the module in the AC and SC, an operator can clearly observe available network and application resources.

The CSO of the application and optical network architecture is represented as $\mathbf{G}\ (\mathbf{V}, \mathbf{L}, \mathbf{W}, \mathbf{A})$, where $\mathbf{V}$ denotes the set of wavelength switching nodes, $\mathbf{L}$ indicates the set of bi-directional fiber links between nodes in $\mathbf{V}$, $\mathbf{W}$ is the set of wavelengths on each fiber link, and $\mathbf{A}$ denotes the set of DC servers. N, L, W, and A represent the number of network nodes, links, wavelengths, and DC nodes, respectively. In each DC server, three time-varying application stratum parameters describe the service condition of the DC application resources, which consist of memory utilization ($U_{\mathrm{R}}^{(t)}$)-modeled RAM, CPU usage ($U_{\mathrm{C}^{(t)}}$), and I/O scheduling utilization ($U_{\mathrm{I}^{(t)}}$). From another perspective, the parameters of the network stratum contain the occupied bandwidth (B_{l}) and propagation delay (τ_{l}) of each link related to traffic cost and the latency of the corresponding link, as well as the hop (H_{p}) of each candidate path. Users are more concerned with the QoS experience than with knowing which server provides services. Therefore, for each application request from source node s, it needs to allocate the required network bandwidth (b), delay (τ), and application resources (ar) in the DC. We denote the$i$th traffic request as $TR_i(s, b, \tau, ar)$. TR_{i+1} arrives after the connection demand (TR_i) in sequence. Additionally, the appropriate DC server can be chosen as the destination node according to various strategies on the basis of traffic requests and resource status.

On the basis of the functional SOFT architecture, we propose a novel DGLB strategy with dynamic resource rating for CSO and compare this strategy with RBS, ABS, NBS, and SLB strategies. In the RBS strategy, the destination node of the DC server is randomly selected by the control plane when the application request arrives. In the ABS strategy, the control plane chooses the server node with the least application utilization as the destination on the basis of storage and CPU utilization to balance server load. NBS selects the node with the shortest hop from the source to the destination by using Dijkstra's algorithm. The SLB strategy chooses the server with the lowest cost calculated statically by using both application and network resources. These four strategies have been discussed in our previous study[5]. According to the DGLB strategy, the AC selects the server node and the DC location as the destination on the basis of the application status collected from the DC networks and the network condition dynamically provided by the SCs. The details and procedures of DGLB are presented and analyzed hereafter.

Receiving a new traffic request through the SC, the AC verifies this demand and maps this demand into request parameters, i.e., $TR_i(s, b, \tau, ar)$. Once the traffic demand requires DC resources and a cross-domain connection, a suitable scenario of application resource occupation obtained from the DC networks can respond to the certified demand. Simultaneously, AC customizes the virtual network information with the desired granularity from the SC. Given that parameter characteristics

change over time, a dynamic resource rating method is introduced to estimate the current importance of various parameters by considering the latest resource occupation statistics. The expectation of CPU utilization ($E_{\mathrm{C}}[t_0]$) in the last t_0 time is useful for assessing the recent average statistical CPU occupation, which is expressed as

$$E_{\mathrm{C}}[t|t=t_0]=\frac{\sum_{t=t_{\mathrm{c}}-t_0}^{t_{\mathrm{c}}} U_{\mathrm{C}}^{(t)} f_{\mathrm{C}}^{(t)}}{\sum_{t=t_{\mathrm{c}}-t_0}^{t_{\mathrm{c}}} f_{\mathrm{C}}^{(t)}}, t\in[t_{\mathrm{c}}-t_0, t_{\mathrm{c}}], \quad (1)$$

where t_{c} and $f_{\mathrm{C}^{(\mathrm{t})}}$ indicate the current time and occurred probability of CPU usage, respectively. By the same principle, the expected RAM utilization and I/O scheduling are described as $E_{\mathrm{R}}[t_0]$ and $E_{\mathrm{I}}[t_0]$, respectively.

According to the comparison of the expectations of the three application parameters, an adjustable evaluation rank rate $k_{\mathrm{C}}, k_{\mathrm{R}}$ and k_{I} for CPU, RAM utilization, and I/O scheduling are used to describe their relative proportions. To facilitate the realization of the DGLB strategy using a large-scale SOFT-based testbed, the settings of the evaluation rank rate in the strategy can be simplified appropriately when the simplification does not affect the process and effects of this strategy. We discretize the continuous rank value and assume several typical values for simplicity, i.e., R_a, R_b, and R_c. Initially, the CPU has the highest evaluation rank, with RAM ranking higher than I/O scheduling. At this point, the evaluation ranks satisfy the following expressions: $k_{\mathrm{C}}=R_a, k_R=R_b, k_{\mathrm{I}}=R_c, R_a+R_b+R_c=1, R_a\geqslant R_b\geqslant R_c$. R_a, R_b, and R_c are constants with priorities decreasing gradually, which means that higher usage corresponds to higher priority. When $E_{\mathrm{R}}[t_0]$ or $E_{\mathrm{I}}[t_0]$ exceeds $E_{\mathrm{C}}[t_0]$, for instance, $E_{\mathrm{R}}[t_0]\geqslant E_{\mathrm{C}}[t_0]\geqslant E_{\mathrm{I}}[t_0]$, the evaluation ranks are adjusted according to this change: $k_{\mathrm{C}}=R_b$, $k_{\mathrm{R}}=R_a$, $k_{\mathrm{I}}=R_c$. Thus, k_{C}, k_{R}, and k_{I} are dynamically modified based on the feedback concerning variations in utilization expectations. Therefore, the application occupation with three application stratum parameters of each current DC server is expressed as a dimensionless function:

$$f_{ac}\left[U_{\mathrm{C}}^{(t)}, U_{\mathrm{R}}^{(t)}, U_{\mathrm{I}}^{(t)}, k\right] = \frac{k_{\mathrm{C}}\times U_{\mathrm{C}}^{(t)}+k_{\mathrm{R}}\times U_{\mathrm{R}}^{(t)}+k_{\mathrm{I}}\times U_{\mathrm{I}}^{(t)}}{k_{\mathrm{C}}+k_{\mathrm{R}}+k_{\mathrm{I}}}, \quad (2)$$

in which these parameters are normalized to show a linear relationship among them.

Following the above method, the adjustable evaluation rank rate k_{B}, k_τ between bandwidth and latency can be dynamically adjusted in the network stratum. Thus, the network function with the parameters of each current node is expressed as a dimensionless function:

$$f_{bc}[B_l, \tau_l, H_p]=k_B\sum_{l=1}^{H_p}\frac{B_l}{H_pB}+k_\tau\sum_{l=1}^{H_p}\frac{\tau_l}{H_p\tau}, \quad (3)$$

where B and B_l denote the total bandwidth and occupied bandwidth of the link, respectively, and τ_l, τ, and H_p denote the propagation delay of the link and the latency and hop of the path, respectively.

Among the DC nodes, the candidate servers with the first K minimum of application functions are selected by the AC and expressed as the set $\mathbf{F}_a=\{f_{a1}, f_{a2}, \cdots, f_{ak}\}$. The candidate path between the source and each candidate server can then be calculated by using the minimum network function and denoted as $\mathbf{F}_b=\{f_{b1}, f_{b2}, \cdots, f_{bk}\}$. From a vector graphics standpoint, $\mathbf{F}_a$ can be seen as a K-elements-sized vector space of K application occupation vectors $f_{a1}, f_{a2}, \cdots, f_{ak}$, with the mean vector $\overline{f_a}$ of vector space $\mathbf{F}_a$ expressing the center. The distance between vector f_a and mean vector $\overline{f}_a$ is described by $\parallel f_a-\overline{f}_a\parallel_2$. Among these vectors, vectors f_{ai} and f_{aj} are farthest from and nearest to the mean vector $\overline{f_a}$, respectively, and are chosen by

$$\begin{aligned}\parallel f_{ai}-\overline{f}_a\parallel_2 &= \max_{\forall a}\left\{\parallel f_a-\overline{f}_a\parallel_2\right\}, \parallel f_{aj}-\overline{f}_a\parallel_2\\ &= \min_{\forall a}\left\{\parallel f_a-\overline{f}_a\parallel_2\right\}. \end{aligned}\quad (4)$$

The correlation coefficient of vectors f_{ai} and f_{aj} is calculated as β, is shown as

$$\begin{aligned}\beta &= \frac{cov(f_{ai}, f_{aj})}{D(f_{ai})\cdot D(f_{aj})}\\ &= \frac{E(f_{ai}\cdot f_{aj})-E(f_{ai})\cdot E(f_{aj})}{\sqrt{E(f_{ai}^2)-[E(f_{ai})]^2}\cdot\sqrt{E\left(f_{aj}^2\right)-[E(f_{aj})]^2}}. \end{aligned}\quad (5)$$

The correlation coefficient is related with the degree of DC load balancing. A larger coefficient indicates better balancing in the DCs because the correlation coefficient of the application occupation on different DC servers represents their degree of correlation. A larger correlation coefficient indicates greater interdependence among the server loads. Therefore, a larger coefficient indicates that the server loads and DC servers can be more balanced. In this study, we define α as the joint optimization factor used to assess global resource utilization in the application and network stratums, whereas the dynamic weight between the network and application parameters is described as β. According to Eq. (5), the application utilization weight β changes dynamically based on the degree of load balancing feedback. Thus, the joint optimization factor α fulfills

$$\alpha=\frac{\beta f_{ac}}{\max\{f_{a1}, f_{a2}\cdots f_{ak}\}}+\frac{(1-\beta)f_{bc}}{\max\{f_{b1}, f_{b2}\cdots f_{bk}\}}. \quad (6)$$

With regard to application and network utilization, the node with a minimumαvalue on the basis of the joint optimization factor is selected among the K candidates as the destination node. Upon receiving the traffic request and the pairs of source and destination nodes from AC, SC completes the end-to-end path computation with SA-PCE in the connection and service parameter constraints, as well as performs wavelength assignment for the computed path and lightpath provisioning by EOP. The flowchart of the DGLB strategy with dynamic resource rating is shown in Fig. 2.

To evaluate the performance of the proposed strategy, we set up the multi-domain optical networks with DCs comprising both control and data planes on a SOFT-based testbed (Fig. 3). In the data plane, four OpenFlow-enabled ROADM nodes supporting 40 wavelengths and multi-granularity client-side interfaces are equipped, and the DCs and other nodes are achieved on an array of virtual machines created by VMware software

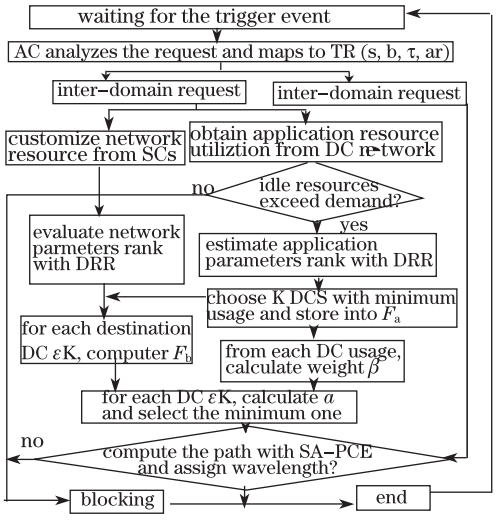

Fig. 2. Flowchart of the DGLB strategy. DRR: dynamic resource rating.

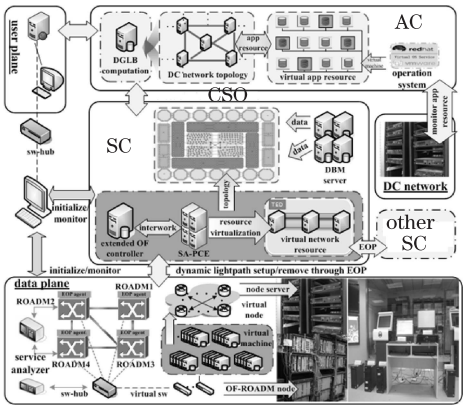

Fig. 3. SOFT-based testbed.

running at servers. Given that each virtual machine has an operation system (OS) and its own independent IP address, CPU, and memory resource, each virtual machine can be considered a real node. The virtual OS technology facilitates the easy setup of the experiment topology on the basis of real edge, metro, and core network configurations. The testbed consists of 1 000 nodes and is divided into 20 domains by either a ring or a mesh[14]. The SCs are assigned to support the SOFT architecture and are deployed in four servers for SA-PCE computation and extended OpenFlow control. The database servers are responsible for maintaining the traffic engineering database, connection status, and configuration of the optical network resources. The AC server is used for DGLB and receives application and network virtual resources from the DCs and SC of each domain through another OpenFlow controller.

On the basis of the established testbed, we have experimentally designed and implemented the DGLB strategy and compared this strategy with RBS, ABS, NBS, and SLB strategies on a SOFT-based testbed. Given the large-scale network feature, we increase network traffic from 100 to 500 Erlang in the service information generator of the testbed and assign the DC nodes to the core side of each domain. The service application usage is randomly selected from 1% to 0.1% for each application demand. The network bandwidth required for each application is assumed equivalent to one wavelength. Each node supports 40 wavelengths without wavelength conversion capability. For simplicity, we set the values of R_a, R_b, R_c and t_0 in the DGLB strategy as 50%, 30%, 20%, and 10 ms, respectively, according to the experience of the experiments. The experimental results are shown in Fig. 4.

Figure 4(a) shows the comparison between the performance of DGLB and other strategies in terms of load balancing degree, which is defined as the correlation of application usage in each DC server. When the load balancing degree increases, the effect of load balancing worsens. Figure 4(a) shows that DGLB leads to significantly lower load balancing degree compared with RBS, NBS, and SLB strategies. The load balancing degree of DGLB is close to that of the ABS. ABS computes only the node considered in the application, and the path may not be set up without sufficient wavelength resource. Figure 4(b) shows that the average hop of DGLB is clearly less than that of the RBS and ABS strategies. Another phenomenon occurs when the offered load increases, i.e., the curve of DGLB becomes closer to that of SLB. This phenomenon occurs because the DGLB strategy computes the lightpath by dynamically considering both network and application resources. By contrast, the RBS strategy randomly chooses the destination node, whereas the ABS strategy selects the destination node with the minimum application occupation. The RBS and ABS strategies cannot consider the network factor in the path computation, resulting in their average hops being higher than that of the DGLB strategy. The NBS strategy considers only network resource and calculates the path with the minimum hop. The SLB focuses on selecting the node with fixed proportion of network and application parameters. Therefore, more network resources are saved, and the current network obtains fixed application resources and sufficient network resources. Owing to immutable parameters, SLB consumes an amount of application resources in exchange for the slight improvement in the

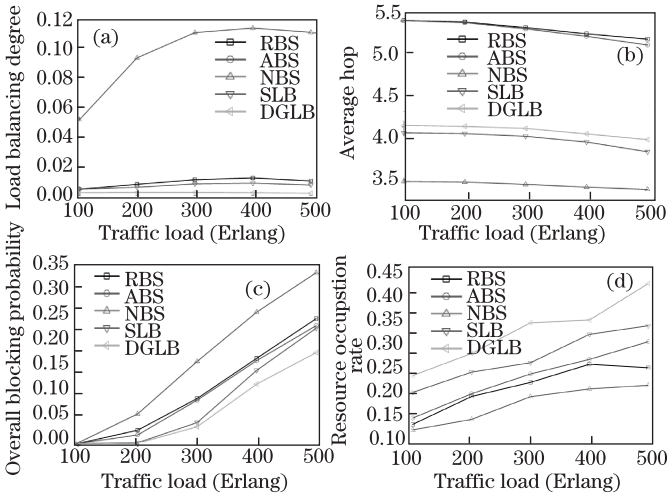

Fig. 4. (a) Load balancing degrees, (b) average hop, (c) overall blocking probability, and (d) resource occupation rate of five strategies.

network stratum. Figure 4(c) shows that the overall blocking probability determines the blocking situation of both network and application, which is measured by CPU and memory overflow. The DGLB strategy has significantly lower overall blocking probability compared with the other strategies, particularly when the network is heavily loaded, because the DGLB generally considers both application and network resources and dynamically adjusts resource rank rate on the basis of the statistical feedback of the current resource status. Figure 4(d) compares the performance of the five strategies in terms of resource occupation rate, which represents the joint usage of application and network resources. Results show that the DGLB strategy significantly outperforms the other strategies in terms of resource occupation rate because the DGLB strategy enhances the utilization of network and application resources by dynamically considering the scarcity degree of various resources. For example, the DGLB strategy focuses on scarce resources and facilitates a dynamic weight adjustment of the application and network resources.

With regard to the specified request, we have verified the protocol implementation for lightpath provisioning by using the DGLB strategy in the SOFT-testbed through Wireshark. The specified request is set from the source node in the domain governed by SC1. On the basis of the DGLB result in AC, two domains governed by SC1 and SC2 are related to the computed path, and the destination node is located in the SC2 domain. Figure 5 shows the Wireshark captures deployed in the AC. As shown in Fig. 5, the computation latency of the DGLB strategy for the dynamic traffic is approximately 15.3 ms between receiving the traffic request and sending the results to the corresponding SC.

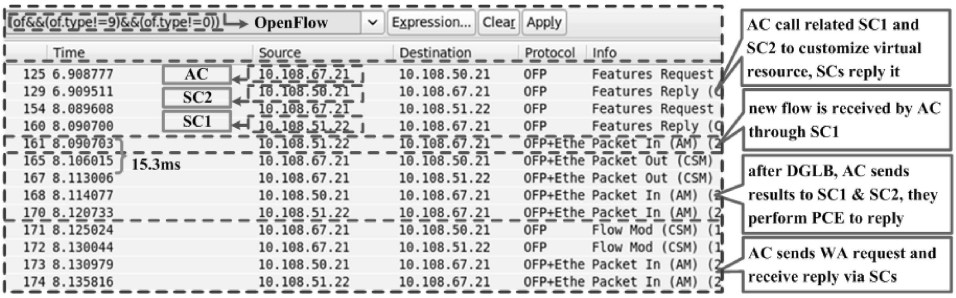

Fig. 5. Wireshark capture of the message sequence for SOFT in AC. WA: wavelength assignment.

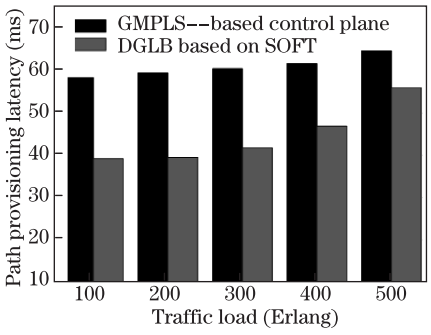

Fig. 6. Comparison of DGLB based on SOFT and GMPLS for path provisioning latency.

We also compare the performance of DGLB based on SOFT with that of the GMPLS-based control plane in terms of path provisioning latency. The results related to GMPLS are measured on another testbed[14]. Figure 6 shows that the DGLB based on SOFT outperforms the GMPLS-based control plane in terms of average lightpath setup latency. This result can be attributed to the capability of the DGLB strategy in the centralized model to facilitate parallel computing, resource assignment and reservation, and simultaneous control of all nodes, which relatively reduces the calculation time compared with the distributed method. As the traffic load increases, the difference decreases because more requests require queuing in the AC, which in turn increases the calculation time of the DGLB strategy as well as the delay time.

In conclusion, we propose a CSO-oriented unified control architecture SOFT in OpenFlow-enabled triple-M optical networks by introducing the DGLB strategy. We experimentally demonstrate a SOFT-based testbed with 1000 virtual nodes to achieve the DGLB strategy for CSO and compare the DGLB strategy with other strategies. Numerical results show that DGLB based on SOFT can effectively utilize cross-optical network and application stratum resources, resulting in lower overall blocking probability and in enhanced end-to-end responsiveness of DC services.

This work was supported by the National "863" Program of China (No. 2012AA011301), the National "973" Program of China (No. 2010CB328204), the National Natural Science Foundation of China (Nos. 61271189, 61201154, and 60932004), the Research Fund for the Doctoral Program of Higher Education of China (No. 20120005120019), the Fundamental Research Funds for the Central Universities (No. 2013RC1201), and the Fund of State Key Laborotory of Information Photonics and Optical Communications (BUPT).

References

1. M. Channegowda, P. Kostecki, N. Efstathiou, S. Azodolmolky, R. Nejabati, P. Kaczmarek, A. Autenrieth, J. P. Elbers, and D. Simeonidou, in *Proceedings of ECOC 2012* Tu.1.D.2 (2012).
2. W. Sun, P. Li, C. Li, and W. Hu, Chin. Opt. Lett. **11,** 010601 (2013).
3. Z. Du, Y. Lu, and Y. Ji, Chin. Opt. Lett. **10,** 020604

(2012).

4. Y. Lee, G. Bernstein, N. So, T. Y. Kim, K. Shiomoto, and O. G. Dios, "draft-lee-cross-stratum-optimization-datacenter-00," IETF draft (Mar. 3, 2011).
5. H. Yang, Y. Zhao, J. Zhang, S. Wang, W. Gu, Yi Lin, and Young Lee, in *Proceedings of OFC/NFOEC 2012* JTh2A.38 (2012).
6. W. Huang, M. Tacca, N. So, M. Razo, and A. Fumagalli, in *Proceedings of ISPA* 523 (2012).
7. J. Rubio-Loyola, A. Galis, A. Astorga, J. Serrat, L. Lefevre, A. Fischer, A. Paler, and H. Meer, IEEE Commun. Mag. **49,** 84 (2011).
8. L. Liu, T. Tsuritani, I. Morita, H. Guo, and J. Wu, Opt. Express **19,** 26578 (2011).
9. S. Das, G. Parulkar, and N. McKeown, in *Proceedings of OFC/NFOEC* OTuG1 (2010).
10. H. Yang, Y. Zhao, J. Zhang, S. Wang, W. Gu, J. Han, Y. Lin, and Y. Lee, in *Proceedings of OFC/NFOEC* NTu3F.7 (2013).
11. L. Liu, R. Casellas, T. Tsuritani, I. Morita, R. Martínez, and R. Muñoz, in *Proceedings of OFC/NFOEC* OM3G.2 (2012).
12. J. Zhang, Y. Zhao, H. Yang, Y. Ji, H. Li, Y. Lin, G. Li, J. Han, Y. Lee, and T. Ma, in *Proceedings of OFC/NFOEC* PDP5B.1 (2013).
13. J. Zhang, J. Zhang, Y. Zhao, H. Yang, X. Yu, L. Wang, and X. Fu, Opt. Express **21,** 1364 (2013).
14. J. Zhang, Y. Zhao, X. Chen, Y. Ji, M. Zhang, H. Wang, Y. Zhao, Y. Tu, Z. Wang, and H. Li, Opt. Express **19,** B746 (2011).