Mobility Support for Fog Computing: An SDN Approach

Yuanguo Bi, Guangjie Han, Chuan Lin, Qingxu Deng, Lei Guo, and Fuliang Li

Abstract

The emerging real-time and computation-intensive services driven by the Internet of Things, augmented reality, automatic driving, and so on, have tight quality of service and quality of experience requirements, which can hardly be supported by conventional cloud computing. Fog computing, which migrates the features of cloud computing to the network edge, guarantees low latency for location-aware services. However, due to the locality feature of fog computing, maintaining service continuity when mobile users travel across different access networks has become a challenging issue. In this article, we propose a novel software-defined-networking-based fog computing architecture by decoupling mobility control and data forwarding. Under the proposed architecture, we design efficient signaling operations to provide seamless and transparent mobility support to mobile users, and present an efficient route optimization algorithm by considering the performance gain in data communications and system overhead in mobile fog computing. Numerical results from extensive simulations have demonstrated that the proposed scheme can not only guarantee service continuity, but also greatly improve handover performance and achieve high data communication efficiency in mobile fog computing.

Introduction

With the widespread deployment of advanced wireless and electronic technologies, smart devices have been equipped with sensors, cameras, communication chips, and so on, enabling them to collect a huge amount of data in the smart city environment [1]. According to the report from Harald *et al.*, 50 to 100 billion smart devices will connect to the Internet by 2020 [2], which will stimulate ever more rapid growth of data traffic. For example, Cisco has predicted that smart devices will generate 507.5 ZB/year by 2019 [3]. By data mining, analysis, and decision making with the collected big data from distributed devices, a number of promising services including smart home/building, smart healthcare, intelligent transportation, and so on will greatly change the way we work, live, and play [4].

Even though smart devices are becoming more and more powerful, running resource demanding applications at terminals is still constrained by limited battery support and computation capacity [5, 6]. A feasible solution is to offload the huge processing to conventional cloud centers. However, delivering a large volume of collected data to remote centers not only induces heavy bandwidth and energy consumption that impose pressure on network infrastructure, but also prolongs the end-to-end delay of data traffic, which cannot be tolerated by real-time applications [7]. For example, when a traffic accident occurs, it is first reported to a remote server located at the transportation administration department, and alert information generated by the server has to pass through the wired network and then be broadcast to nearby vehicles by vehicle-to-infrastructure communications. As a result, the large round-trip time of the alert information may lead to more traffic accidents, since life-critical services have a tight requirement on latency.

The computation-intensive and real-time requirements of smart city services are driving service providers (SPs) to implement a more flexible network architecture, not only optimizing network operations and resource utilization but also providing satisfied quality of service (QoS) and quality of experience (QoE). Fog computing, as an alternative data processing architecture, attempts to offer cloud computing services in a real-time approach. Instead of information exchange between mobile users and cloud, fog computing migrates data centers from the core network to the network edge, which prevents data traffic from injecting the core network, and achieves a number of advantages including saving bandwidth resource, reducing energy consumption, shortening latency, improving user experience, and so on. Exploring the advantages of fog computing, location-aware services (smart healthcare, intelligent transportation, etc.) can be efficiently supported [8], since data generated in the above scenarios is likely to be processed locally.

Fog computing is typically implemented in macro/small cell base stations (BSs), WiFi access points (APs), and so on, and high user mobility tends to induce frequent handovers among small-coverage fog servers (FSs). Without efficient mobility support, mobile users start over with service discovery and service commissioning procedures whenever the attached FS changes [9, 10], which not only leads to service disruptions

The authors propose a novel SDN-based fog computing architecture by decoupling mobility control and data forwarding. Under the proposed architecture, they design efficient signaling operations to provide seamless and transparent mobility support to mobile users, and present an efficient route optimization algorithm by considering the performance gain in data communications and system overhead in mobile fog computing.

Yuanguo Bi is with Northeastern University and also with Neusoft Corporation; Chuan Lin, Qingxu Deng, Lei Guo, and Fuliang Li are with Northeastern University; Guangjie Han is with Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology. Guangjie Han is the corresponding author of this article..

Digital Object Identifier: 10.1109/MCOM.2018.1700908

In order to guarantee service continuity and provide satisfied QoE to mobile users in fog computing, the network is supposed to make centralized control of network resources and facilitate mobility management by eliminating the limitations in both CMM and DMM solutions. The emerging SDN can simplify the network management, schedule data flows, and enhance network capability by separating the data and control planes.

and degrades user experience, but also decreases network resource utilization. As a result, an efficient mobility management scheme has to be employed to guarantee service continuity when mobile users roam across different FSs.

Software-defined networking (SDN), as a promising and practical network paradigm, abstracts network functions by decoupling the control plane and the data plane [11]. SDN migrates routing logic to the centralized controller that makes forwarding decisions with real-time global information and configures forwarding rules at switches. In SDN, intelligent management logic is installed at the controller, which greatly simplifies network operation and management, and provides a promising solution to implement efficient mobility management in fog computing. In this article, we study the mobility support issue in the fog computing environment, and the main contributions of this work are summarized as follows.

By decoupling mobility management and data forwarding functions, we propose a novel SDN-enabled architecture that facilitates mobility management in fog computing. Under the presented framework, an efficient handover scheme is designed for providing ubiquitous mobility support to mobile users in fog computing by encapsulating mobility management and route optimization logic at the central controller.

Performance evaluations of the proposed scheme are conducted, which demonstrate that it greatly improves handover performance, achieves high data communication efficiency, and finally, guarantees service continuity in fog computing.

The remainder of this article is organized as follows. We briefly review some related works in the following section, and identify their limitations in fog computing. The details of the proposed SDN-based architecture are then illustrated. Following that, an efficient handover scheme is designed. Performance evaluations are then given, followed by concluding remarks.

Overview of Major Existing Solutions

This section provides a comprehensive overview of existing IP mobility protocols that are dedicated to guaranteeing service continuity in mobile computing. Centralized mobility management (CMM) solutions share a similar paradigm in that a mobility anchor point (MAP) involves both mobility management and data forwarding, while in distributed mobility management (DMM) solutions, MAPs are exempted from data forwarding but collaborate with points of attachment (PoAs) to perform mobility signaling operations.

Centralized Mobility Management

Mobile IPv6 (MIPv6) [12], standardized by the Internet Engineering Task Force (IETF), has been an extensively accepted proposal in supporting global mobility. In MIPv6, a mobile user registers its home address in the home network when it starts an end-to-end communication. A new care-of address is acquired whenever its attached PoA changes, and then a binding update message is delivered to the home agent to register the new care-of address. As a host-based handover solution, in MIPv6 mobile users are required to engage in a handover process. PMIPv6 [13] was also standardized by IETF for providing network-based mobility support to mobile users. In PMIPv6, the core functional entities of mobility management are the mobile access gateway (MAG) and local mobility anchor (LMA). An MAG is located at an access router and is responsible for handling the attachment of mobile users to the access network or detachment of mobile users from the access network. An LMA in PMIPv6 not only forwards data packets between the registered mobile users and correspondent nodes (CNs), but also handles mobility-related signaling operations and maintains binding information of registered mobile users. However, CMM solutions including MIPv6, PMIPv6, and their extensions face the following limitations in fog computing.

Sub-Optimal Routing: With a CMM solution, data packets from FSs to mobile users have to pass through the central MAPs, which leads to triangular routing and prolongs the end-to-end latency. The sub-optimal routing issue greatly contradicts the advantages of fog computing, which advocates guaranteeing short latency for real-time applications.

Communication Bottleneck: A central MAP is in charge of mobility support for registered users, and all the traffic has to traverse the MAP. The coupled mobility management and data forwarding functions not only induce the scalability issue, but also lead to a communication bottleneck and degrade user experience.

Reliability: A central MAP tends to become a single point of failure, which leads to service interruption in fog computing.

Distributed Mobility Management

IETF chartered the DMM working group to address the challenging issues in CMM solutions [14]. DMM separates the data plane and the control plane, and enables traffic to be offloaded from core networks by deploying distributed mobility anchors at the network edge. In DMM, data traffic is uniformly distributed among different forwarding entities, which eliminates the single point of failure at the bottleneck MAP, and accordingly achieves high bandwidth utilization efficiency. However, DMM is still confronted by several fundamental challenges.

Sub-Optimal Routing: In DMM, if a new session anchored at the serving mobility anchor and access router (MAAR), the traffic can be forwarded optimally. However, if the session experiences a handover, the data traffic of this session is tunneled between the previous MAAR and the serving MAAR for session continuity, which means end-to-end traffic has to pass through the previous MAAR, which induces sub-optimal routing.

Low Prefix Utilization: A local network prefix is allocated at each servicing MAAR, and a mobile user may occupy multiple prefixes at the same time, which degrades prefix utilization efficiency.

The Proposed SDN-Enabled Fog Computing Architecture

In order to guarantee service continuity and provide satisfactory QoE to mobile users in fog computing, the network is supposed to take centralized control of network resources and facilitate mobility management by eliminating the limitations of both the CMM and DMM solutions. The emerging SDN can simplify the network management, schedule data flows, and enhance network capability by separating the data and control planes. Implementing network management intelligence at the central SDN controller can degrade the complexity of fog computing. In this section, we propose a novel SDN-enabled fog computing architecture by decoupling mobility control and data forwarding functions as shown in Fig. 1, and each of its components is illustrated as follows.

Fog Layer: The fog layer consists of FSs and various kinds of smart devices (e.g., wearable equipment, smartphone, intelligent vehicle). An FS resides at the network edge and runs location-aware fog services. It not only provides wireless connections to mobile users, but also serves a normal SDN switch, and its forwarding behaviors are controlled by the SDN controller. A smart device is either fixed or mobile with various mobility patterns, and it may upload collected data to FSs for further processing, or receives location-aware services from FSs. However, if it moves out of the coverage of an FS, a handover procedure will be performed for service continuity.

Network Layer: The network layer consists of SDN switches and controllers that run SDN protocols (e.g., openflow). Besides configuring forwarding rules to switches, the SDN controller installs mobility logic for handling mobility-related signaling when mobile users change their attached FSs. It also maintains a binding cache entry (BCE) table for location tracking of mobile users. An SDN switch maintains flow tables that regulate how the switch forwards data packets in the data plane.

Application Layer: The application layer consists of cloud and SDN network management components. Some resource-intensive services that can hardly be implemented at FSs or non-location-aware services reside in the cloud. On the other hand, network management components manage data transfer, including QoS guarantee, traffic monitoring, routing decision, and so on, by utilizing the service interfaces provided by SDN controllers in the control plane.

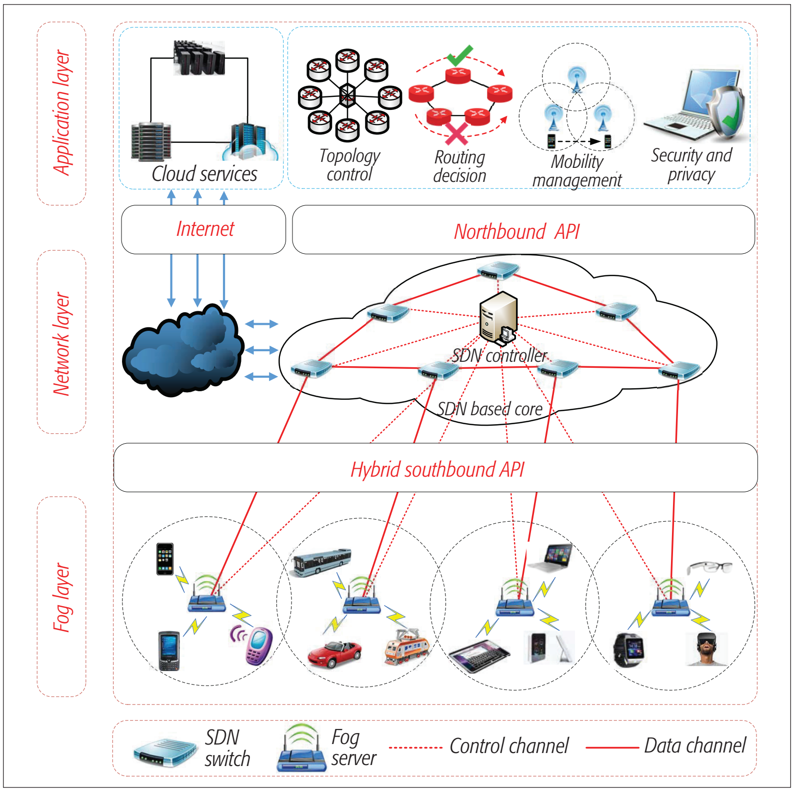

Figure 1. The SDN-enabled fog computing architecture.

The SDN-Based Mobility Management for Fog Computing

When a mobile node (MN)[1] moves into the coverage of an FS and tries to receive fog services from the FS, it performs initial registration in order to obtain an IPv6 address at the SDN controller. Then the MN can communicate with the FS, and accordingly the FS becomes the corresponding FS (C-FS) of the MN. However, for a high-speed MN (e.g., an MN in a vehicle or subway train), its residence time in the C-FS may be limited, and an active fog service may not be finished during this short interval. As a result, an efficient handover scheme has to be employed for guaranteeing fog service continuity when the MN travels across different FSs. In the proposed handover scheme, a handover process is triggered when an MN is traveling from the serving C-FS to a new FS (n-FS), and the serving C-FS becomes a previous FS (p-FS). The MN operates in either proactive or reactive mode according to the network model and configuration as follows.

Proactive Handover

It is possible that an MN can acquire the identifier (ID) information of the n-FS before it leaves the coverage of the p-FS (e.g., by proactive signal scanning) if there is coverage overlapping between the p-FS and the n-FS. A proactive handover is initialized when a link layer (L2) detachment event occurs, and as shown in Fig. 2a the following operations are performed:

- An L2 report is delivered to the p-FS from the MN, containing ID of the MN and ID of the n-FS.
- The p-FS generates a Deregistration Proxy Binding Update (D-PBU) message and delivers it to the SDN controller, including its IPv6 address, ID of the MN, and ID of the n-FS. Then the p-FS buffers the downLink (DL) packets destined to the MN.
- After receiving the D-PBU message, the controller calculates the path from the p-FS to the n-FS, and then delivers Flow Modification (FlowMod) messages to the switches on the path, which enables buffered DL packets at the p-FS to reach the n-FS.
- The controller computes the optimal path from the C-FS to the n-FS, taking both performance gain and system overhead into account based on the optimal path selection algorithm below.
- The MN sends a Router Solicitation (RS) message to the n-FS after a successful L2 attachment. After receiving the RS message, the n-FS delivers a PBU message to the SDN controller.
- The controller finds the BCE of the MN according to the received PBU message, and then replaces the p-FS with the n-FS in the MN's BCE.

[1] Mobile node and mobile user are used interchangeably in this article.

A smart device is either fixed or mobile with various mobility patterns, and it may upload collected data to FSs for further processing, or receives location-aware services from FSs. However, if it moves out of the coverage of an FS, a handover procedure will be performed for service continuity.

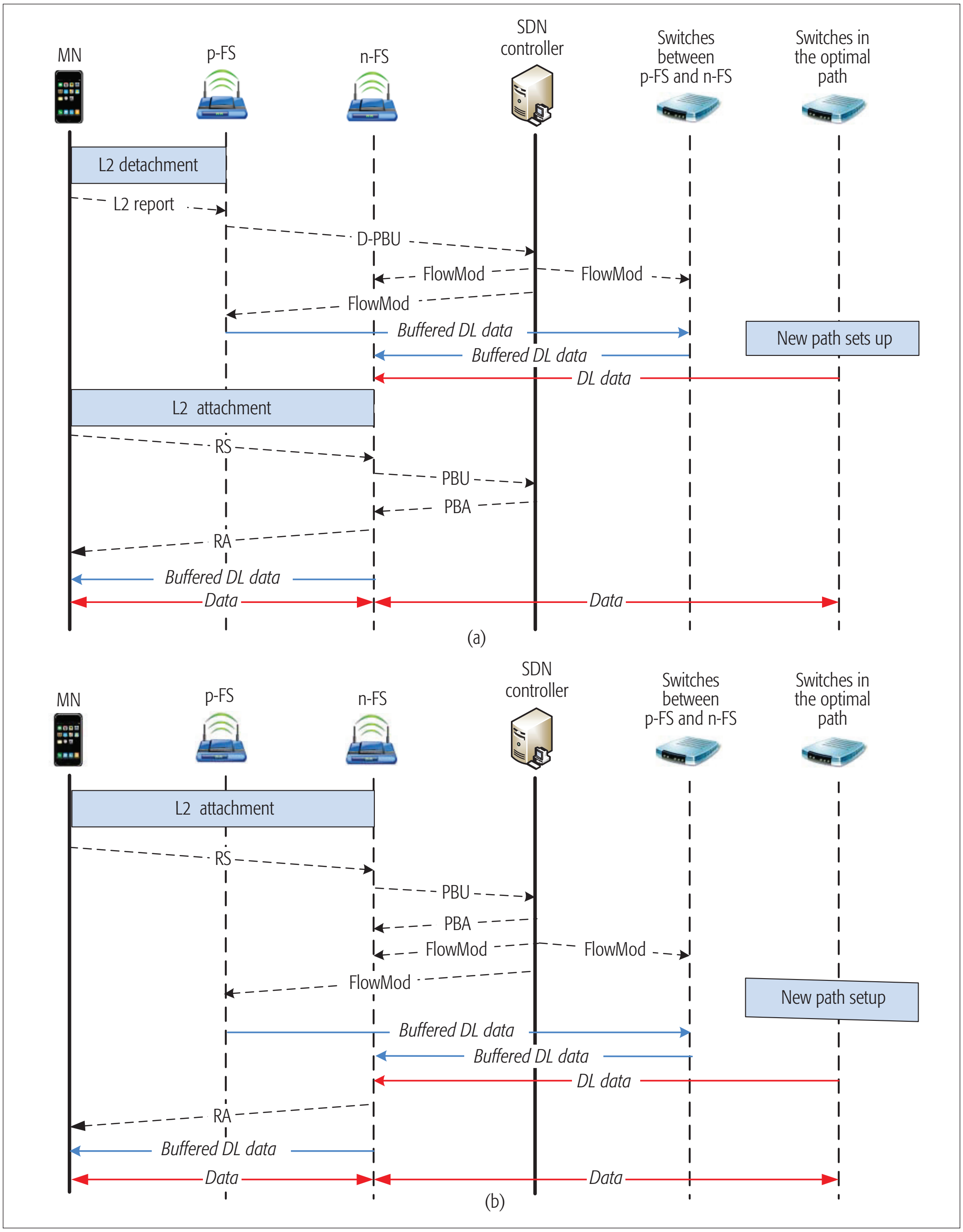

Figure 2. Signaling operations in proactive/reactive handover: a) proactive handover; b) reactive handover.

- Furthermore, the controller responds a Proxy Binding Acknowledgment (PBA) message to the n-FS. After receiving the PBA message, the n-FS returns a Router Acknowledgment (RA) message to the MN. Thereafter, the MN can resume end-to-end communications with the C-FS via the optimal path.

Reactive Handover

It is possible that the MN maintains communications with the p-FS until it loses the wireless connection to the p-FS. However, when the MN travels into the coverage of the n-FS, a reactive handover is performed after a successful L2 attachment for guaranteeing service continuity. As a result, the following signaling operations are performed, as shown in Fig. 2b:

- The MN transmits an RS message to the n-FS containing its ID.
- The n-FS initializes a PBU message containing the ID of the MN and the address of the n-FS after receiving the RS message, and then delivers the message to the controller.
- The controller can feasibly find the BCE of the MN based on the received PBU message, and replaces the p-FS with the n-FS in the BCE of the MN. At the same time, the controller returns a PBA message to the n-FS.
- In addition, the controller computes the path from the p-FS to the n-FS, and sets up the path by delivering FlowMod messages to the switches on the path, which guarantees that buffered DL packets at the p-FS can be deliv-

ered to the n-FS. At the same time, the controller computes the optimal path from the C-FS to the n-FS, and performs the optimal path selection algorithm below.

- The n-FS delivers an RA message to the MN after receiving the PBA message from the controller.

By the above operations, a reactive handover process completes, and the MN can restart communications with the C-FS by the optimal path.

Optimal Path Selection

After an MN moves into the n-FS, the previous path from the C-FS to the MN passing through the p-FS may not be the optimal one. For example, as shown in Fig. 3, in order to eliminate packet loss, the path between the p-FS and the n-FS is established for buffered DL packets delivery after the MN attaches to the n-FS. Meanwhile, the previous path passing the p-FS is longer than the new path directly connecting the n-FS and the C-FS. As a result, in order to reduce latency and provide improved QoE for fog services, the controller needs to compute the optimal end-to-end path. However, since it is resource consuming to establish a new path, the controller should make a trade-off between the performance gain by using the optimal path and the overhead due to the establishment of the optimal path. For example, it may not need to establish a new path if the MN's residence time in the n-FS is short, since the performance gain in data communications during the short interval is quite limited.

In the proposed scheme, the SDN controller performs the procedures as shown in Algorithm 1 in deciding the optimal path, where H_p is the hop distance of the previous path that passes through the p-FS, K is the amount of paths from the C-FS to the n-FS, H_k is hop distance of the kth path ($k \in [1, K]$). If the optimal path is the previous path, there is no FlowMod message exchange for new path establishment. Otherwise, the SDN controller needs to send FlowMod messages to the switches in the new path for path establishment. As a result, the upcoming packets can be delivered along this new path.

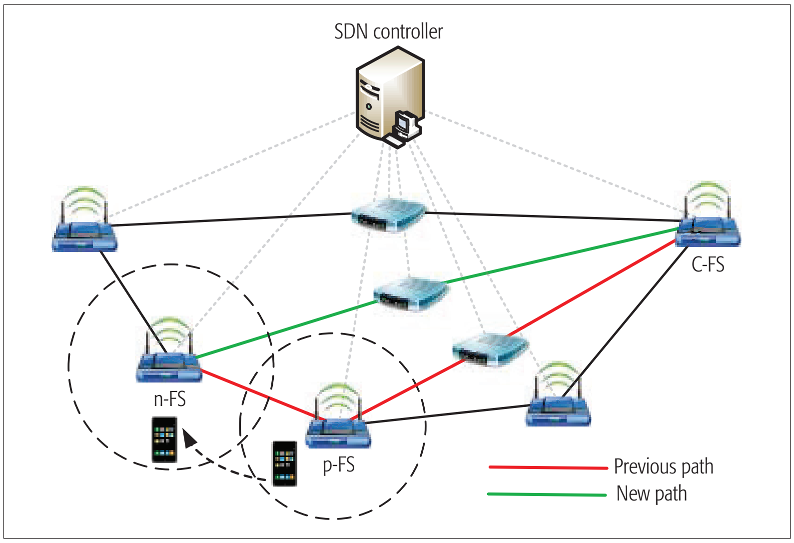

Figure 3. The optimal path selection.

Estimate the MN's residence time T in the n-FS.
Compute N_T as the amount of transmitted packets during T.
Let $k \leftarrow 1$.
while $k < K$ **do**
    if $H_k < H_p$ **then**
        Compute the performance gain D_k in data communications with path k based on N_T.
        Compute the signaling cost C_k in establishing path k.
        Obtain performance gain $G_k = D_k - C_k$.
    else
        $k \leftarrow k + 1$.
    end if
end while
if $\forall k \in [1, K], G_k \leq 0$ **then**
    The optimal path is the previous path.
else
    The optimal path is path k with the maximum G_k.
end if

Algorithm 1. Optimal path selection algorithm.

Simulation Results

In this section, we implement the proposed handover scheme in Mininet by updating the Mininet-WiFi package to support wireless communications under the simulated fog computing architecture. We evaluate the performance of the proposed handover scheme in terms of handover latency and system cost. Handover latency is denoted as the time interval from the MN losing wireless connection to the p-FS to the MN starting data communications in the n-FS. System cost includes:

- The cost for signaling message transmissions, which is the product of the signaling messages during a handover process and their corresponding hop distance
- The cost for data packet transmissions, which is the product of the total delivered packets during the residence time in the n-FS and the hop distance of the optimal path

Furthermore, we make performance comparisons between the proposed handover scheme and SDN-DMM [15]. In the simulated scenario, an MN sets up a flow with its C-FS, and then roams across different FSs and conducts handovers in proactive/reactive mode based on the network settings. In the following subsections, we evaluate each performance metric of our proposed handover scheme.

Handover Latency

Handover latency is a key performance metric for guaranteeing fog service continuity. For example, an MN moves out of the C-FS and induces a large handover latency, which will greatly degrade the QoE of fog services. Figure 4a demonstrates the performance comparisons of handover latency between SDN-DMM and the proposed scheme in proactive/reactive mode, where H_{p-c} is the hop distance from the p-FS to the SDN controller, while H_{n-c} is the hop distance from the n-FS to the SDN controller. From this figure, we can observe that the handover latency in proactive handover is much shorter than those in reactive handover and SDN-DMM. This is because before the MN loses wireless connection to the p-FS, a D-PBU message is delivered to the SDN controller including the n-FS information. As a result, the controller is able pre-establish a path between the p-FS and the

The system cost includes the cost for signaling message transmissions and the cost for data packet transmissions, which indicates the resource utilization efficiency. An efficient mobility management scheme in fog computing not only needs to guarantee QoE, but also improves the efficiency of bandwidth resources.

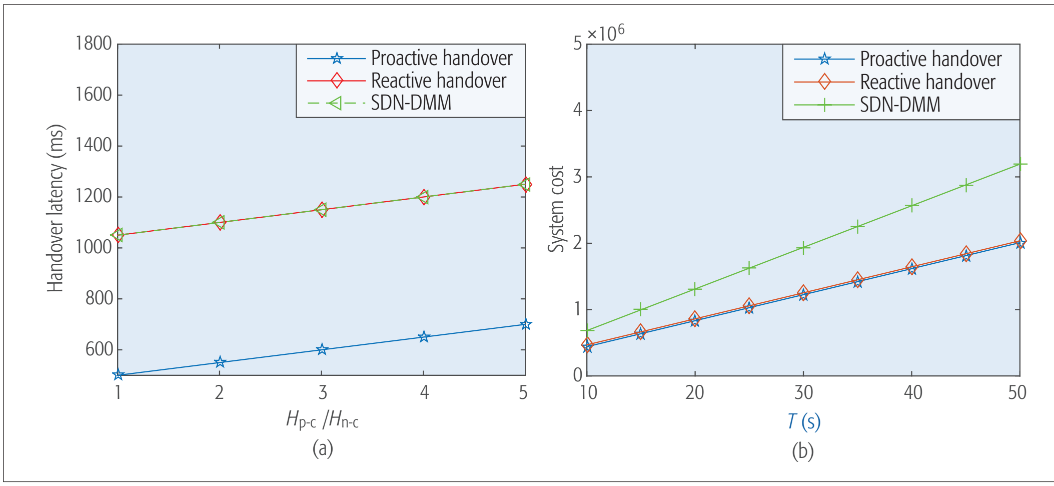

Figure 4. Performance comparisons between SDN-DMM and the proposed scheme: a) handover latency; b) system cost.

n-FS, and DL packets can be delivered to the n-FS from the p-FS in advance. If the MN successfully attaches to the n-FS, it can immediately resume data communications with the n-FS, which greatly shortens the handover delay. However, in reactive handover and SDN-MM, the controller has to establish the path between the p-FS and the n-FS after the MN attaches to the n-FS, and the signaling operations in path establishment prolongs the handover delay. In addition, we can observe that SDN-MM achieves a similar handover latency performance with reactive handover, and that is because they employ similar signaling operations in a handover process.

System Cost

The system cost includes the cost for signaling message transmissions and the cost for data packet transmissions, which indicates the resource utilization efficiency. An efficient mobility management scheme in fog computing not only needs to guarantee QoE, but also improves the efficiency of bandwidth resources. Figure 4b demonstrates the performance comparisons of system cost between SDN-DMM and the proposed scheme in proactive/reactive mode, where T is the MN's residence time in the n-FS. From this figure, we can observe that the system cost in the proposed scheme is much less than that in SDN-DMM. In our proposed scheme, the SDN controller sets up a path between the p-FS and the n-FS, and thus the buffered DL packet at the p-FS can be delivered to the n-FS, which eliminates packet loss during a handover process. In addition, the SDN controller computes the optimal path from the C-FS to the n-FS during a handover process by estimating the performance gain, and it can greatly improve bandwidth utilization efficiency. Furthermore, utilizing the optimal path can also reduce end-to-end latency, which improves QoS for real-time fog services. In SDN-DMM, when an MN travels into a new MAAR, the path between the new MAAR and the previous MAAR is established after a handover. However, when the MN moves into the new MAAR, the end-to-end path that passes through the previous MAAR may not be the optimal one, and it may increase the packet delivery cost. As a result, the proposed scheme achieves a lower system cost than SDN-DMM.

In both proactive and reactive modes, packets are delivered through the optimal path between the n-FS and the C-FS after a handover process, and they achieve the same cost for data packet transmissions. In addition, there are similar signaling message exchanges between proactive mode and reactive mode, as shown in Figs. 2a and 2b. As a result, they have a similar system cost that includes the cost for signaling messages transmissions and the cost for data packet transmissions.

Conclusions

In this article, we study the mobility support issue in fog computing for guaranteeing service continuity. We first propose a novel SDN enabled architecture that is able to facilitate mobility management in fog computing by decoupling mobility management and data forwarding functions. Then, under the proposed framework, an efficient handover scheme is designed by migrating mobility management and route optimization logic to the SDN controller. In addition, by employing link layer information, the SDN controller can pre-compute the optimal path by estimating the performance gain of each path. Finally, performance evaluations of the proposed scheme demonstrate that the proposed framework is able to greatly improve handover performance, achieve high data communication efficiency, and finally, guarantee service continuity in fog computing. Possible future research directions derived from this work can be summarized as follows.

Route Optimization: After the change of an MN's attachment, the controller is responsible for computing the optimal path from the C-FS to the n-FS for the MN. As an extensively adopted approach, the shortest path scheme only considers hop distance in deciding the optimal path between the C-FS and the n-FS. However, a shorter path does not mean lower end-to-end delay, which is mainly determined by the traffic load in the path. In order to satisfy the QoS requirements for location-aware fog services, a more intelligent routing logic should be employed at the SDN controller in determining the optimal path during a handover process.

Virtual Machine Migration: When an MN detaches from the p-FS and attaches to the n-FS, the hop distance between the MN and the C-FS probably increases, which results in longer delay and eliminates the advantages of fog computing. A feasible solution to guarantee service continuity and QoS requirements is to migrate a virtual machine (VM). There is performance gain by VM migration since the MN achieves to experience lower latency to access the local FS where the migrated VM resides. However, migrating the VM to the n-FS introduces extra cost, including the time for VM migration, bandwidth resource consumption, and so on, and the SDN controller needs to make a trade-off in determining VM migration.

Security and Privacy: Fog computing may integrate various access technologies, including WiFi, 5G, and so on, to provide ubiquitous fog services to mobile users. Each communication technology employs an individual security protocol, and inevitably forms a separate trust domain. As a result, due to user mobility, it is challenging to negotiate session keys across various trust domains to guarantee privacy and data integrity. In addition, since SDN is employed to simplify mobility management in fog computing, the SDN controller or the control channel may suffer from attacks since it is software by nature.

Each communication technology employs an individual security protocol, and inevitably forms a separate trust domain. As a result, due to user mobility, it is challenging to negotiate session keys across various trust domains to guarantee privacy and data integrity.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (61671142 and 61373159) , Program for Liaoning Excellent Talents in University (No. LR2017009) and the Fundamental Research Funds for the Central Universities (No. N171602004 and No. DUT17RC(3)094).

References

[1] A. Al-Fuqaha *et al.*, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 4, 2015, pp. 2347–76.

[2] H. Sundmaeker *et al.*, "Vision and Challenges for Realising the Internet of Things," *Cluster of European Research Projects on the Internet of Things*, European Commision, vol. 3, no. 3, 2010, pp. 34–36.

[3] C. V. Networking, "Ciscoglobal Cloud Index: Forecast and Methodology, 2015-2020; White Paper"; http://www.cisco.com/c/ dam/en/us/solutions/collateral/service-provider/global-cloud-index-gci/ white-paper-c11-738085.pdf, accessed Mar. 8, 2017.

[4] X. Sun and N. Ansari, "Edgeiot: Mobile Edge Computing for the Internet of Things," *IEEE Commun. Mag.*, vol. 54, no. 12, Dec. 2016, pp. 22–29.

[5] G. Jia *et al.*, "Dynamic Adaptive Replacement Policy in Shared Last-Level Cache of DRAM/PCM Hybrid Memory for Big Data Storage," *IEEE Trans. Industrial Informatics*, vol. 13, no. 4, 2017, pp. 1951–60.

[6] G. Han *et al.*, "HySense: A Hybrid Mobile CrowdSensing Framework for Sensing Opportunities Compensation under Dynamic Coverage Constraint," *IEEE Commun. Mag.*, vol. 55, no. 3, Mar. 2017, pp. 93–99.

[7] G. Han *et al.*, "Green Routing Protocols for Wireless Multimedia Sensor Networks," *IEEE Wireless Commun.*, vol. 23, no. 6, Dec. 2016, pp. 140–46.

[8] P. Yang *et al.*, "Catalyzing Cloud- Fog Interoperation in 5G Wireless Networks: An SDN Approach," *IEEE Network*, vol. 31, no. 5, Sept./Oct. 2017, pp. 14–20.

[9] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How Can Edge Computing Benefit from Software-Defined Networking: A Survey, Use Cases & Future Directions," *IEEE Commun. Surveys & Tutorials*, vol. 19, no. 4, 2017, pp. 2359–91.

[10] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Commun. Surveys & Tutorials*, vol. 19, no. 3, 2017, pp. 1628–56.

[11] M. Dabbagh *et al.*, "Software- Defined Networking Security: Pros and Cons," *IEEE Commun. Mag.*, vol. 53, no. 6, June 2015, pp. 73–79.

[12] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," IETF RFC 3775, June 2004.

[13] H. Yokota *et al.*, "Fast Handovers for Proxy Mobile IPv6," IETF RFC 5949, Sept. 2010.

[14] D. Liu and P. Seite, "Distributed Mobility Management: Current Practices and Gap Analysis," IETF Draft, draft-ietf-dmm-best-practices-gapanalysis- 07, Sept. 2014.

[15] H. Yang and Y. Kim, "SDN-Based Distributed Mobility Management," *Proc. IEEE 2016 Int'l. Conf. Info. Networking*, 2016, pp. 337–42.

Biographies

Yuanguo Bi (biyuanguo@mail.neu.edu.cn) received his Ph.D. degree from Northeastern University, Shenyang, China, in 2010. He joined the School of Computer Science and Engineering, Northeastern University as an associate professor in 2010. His current research interests focus on fog computing, software-defined networking, QoS routing, multihop broadcast, mobility management, and vehicular networks.

Guangjie Han (hanguangjie@gmail.com) is currently a professor with the School of Software at Dalian University of Technology, China. He finished his work as a postdoctoral researcher with the Department of Computer Science at Chonnam National University, Korea, in 2008. From October 2010 to 2011, he was a visiting research scholar at Osaka University, Suita, Japan. His current research interests include sensor networks, green computing, cloud computing, and mobile computing.

Chuan Lin (chuanlin1988@gmail.com) is currently a Ph.D. candidate with the School of Computer Science and Engineering, Northeastern University. He received his B.S. degree in computer science and technology from Liaoning University, Shenyang, China, in 2011 and his M.S. degree in computer science and technology from Northeastern University in 2013. His research interests include network performance analysis, QoS routing, software defined networking, and network modeling.

Qingxu Deng (dengqingxu@cse.neu.edu.cn) received his Ph.D. degree from Northeastern University in 1997. He is currently a full professor with the School of Computer Science and Engineering, Northeastern University. His current research interests include the Internet of Things, multiprocessor real-time scheduling, and formal methods in real-time system analysis.

Lei Guo [M] (guolei@cse.neu.edu.cn) received his Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, in 2006. He is currently a professor with the School of Computer Science and Engineering, Northeastern University. His research interests include optical networks and network security. He has published over 300 technical papers in the above areas. He is a Senior Member of China Institute of Communications.

Fuliang Li [M] (lifuliang@cse.neu.edu.cn) received his B.S. degree in computer science from Northeastern University in 2009 and his Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2015. He is currently an assistant professor with the School of Computer Science and Engineering, Northeastern University. His research interests include software-defined networking and mobile computing.