Tracing Link Flooding Attacks in MobilityFirst Networks

Zhaoxu Wang*, Huachun Zhou, and Wei Quan
School of Electronic and Information Engineering, Beijing Jiaotong University, China
{zxwang, hchzhou, weiquan}@bjtu.edu.cn

Abstract

Current link flooding attack (LFA) defense strategies highly rely on static network topology and pre-collected traceroute record in current Internet, which are not fit for the MobilityFirst networks. We propose the traffic scanning defense (TSD) mechanism against LFA in MobilityFirst. TSD is cooperative, fast and stateless, and also scalable to implement in MFTP for MobilityFirst. Preliminary tests show promising efficiency of TSD.

Keywords: link flooding attack, MobilityFirst networks, traffic analysis, packet marking

1 Introduction

The MobilityFirst networks [3] has gained wider attentions and implementations in recent years. Heterogenous to the traditional Internet, MobilityFirst has the particular designed transport mechanism, MFTP [10], which provides the hop-by-hop reliable transport capability. Therefore, MobilityFirst is viewed as naturally providing dynamic caching, per-hop reliable transport and mobility support, and also promising in adapting wider dynamic network environments with changing topology, waving link parameters and unpredictable link connectivity.

However, MobilityFirst′s features make itself fragile facing the Link Flooding Attacks (LFA). LFA is a new type of Distributed Denial-of-Service (DDoS) attack emerged in recent years. Instead of flooding the victim server directly, LFA occupies the bandwidth capacity of the bottleneck links towards the victim server, thus making the victim server disconnected with most of the users on the Internet. According to MobilityFirst′s natural strategies of per-hop reliability, MFTP routers attend to cache the excessive packets rather than drop them when congestion emerges. Thus when a link is crowed by the attacker traffic in LFA, its upstream links help it cache the excessive packets and soon fall into congestion as while. Thus it is easier in MobilityFirst to cause vertical congestion upstream the target link in LFA.

What′s worse, MobilityFirst is usually implemented with effective rerouting mechanisms to deal with the congestion caused by the dynamic changing of network topology. Due to the minimum-changing strategy when rerouting in the new topology, the new routes are often triggered too nearby the congested link, which leaves the packets few choices for the alternative available paths. If the attack traffic is rerouted nearly, the rerouting mechanism will indirectly help expand the attacked area to the neighbor parallel links. In other words, it is also easier in MobilityFirst to cause horizontal congestion for LFA. In fact, a few attack flows can cause critical congestions nearby the target link in MobilityFirst.

Current defense strategies against LFA are not suitable for MobilityFirst. First, some strategies require the defender to take some reactions during an attack in order to force the attackers change their behaviors and identify them, i.e., the attacker-defender (A-D) interaction [4]. Such interactions are slow for the LFA in MobilityFirst who can cause wider and worse damage in very short time. What′s more, there is a strong assumption that the attackers do not realize the existence of such behavior monitor in

Research Briefs on Information & Communication Technology Evolution (ReBICTE), Vol. 4, Article No. 12 (October 15, 2018)
*Corresponding Author: School of Electronic and Information Engineering, Beijing Jiaotong University, Haidian, Beijing (100044), China, Tel: +86-18811774990

the network and they will act in the exact way as what the defender predicted. Second, current strategies highly rely on static network topology [6]. On the opposite, ICN highlights its adaptability to mobile and dynamic network environments, making current strategies suffer efficiency depression. Third, current strategies search the attackers based on pre-collected traceroute records [5]. However, MobilityFirst is clean-slate and the topology information are hidden behind the content service according to its information-centric nature. Without specific traceroute protocols, current strategies will find it difficult to locate the attackers. Hence the current defense strategies against LFA have high attack reaction delay or strong assumptions on topology, protocol and attackers′ behaviors, and they are not suitable in ICN.

To solve the LFA problem in MobilityFirst, we propose TSD (Traffic Scanning Defense), a detection and defense strategy against LFA particularly in MobilityFirst. The backbone idea of TSD is monitoring traffic to detect unusual traffic increasing, scanning the historical traffic record in the upstream routers to locate the attack flows, push the reaction location upstream as remote to the target link as possible, thus enhance the rerouting mechanism′s efficiency to intercept the attack traffic. TSD owns three particular features given below:

- **Fast**. Reacting LFA in millisecond-level, no need for A-D interaction, also harder to be effected by topology changing.
- **Stateless**. No historical traceroute record or flow table is required, which is important for the scalability of MobilityFirst.
- **Easy to implement**. Taking the advantages of natural rerouting functionality in every router of MobilityFirst, providing minimum modification on the current routers.

Hence with TSD, we gain two goals:

- TSD protects the bandwidth share on the target link of the legitimate flows from upstream areas who are not infected by bots.
- TSD allocate more bandwidth share on the target link to the legitimate flows rather than the attack flows from upstream areas who are infected by bots.

By developing real codes for MFTP and TSD and implementing them on physical machines, we build the simulation environments and test TSD in real traffic. The test results show that TSD is effective defending the LFA in MobilityFirst, and also promising achieving quick attack response.

2 Background

2.1 Features of LFA

As a newly emerged DDoS pattern in recent years, LFA has some new features given below:

- **Low cost**. The attack flows of LFA act very similarly to legitimate flows which have small traffic and short living time. This is because that anonymous and misbehaved attack flows are easy to be intercepted by the firewall.
- **Small traffic**. Not like servers who have TB-level process capability per second, links in industrial usage are only limited in Gbps-level. Thus LFA does not require too big attack traffic to overflow the target link.

- **Hidden victim**. The attack flows of LFA usually do not heading the victim server directly, but to its physically neighbor servers (e.g.,Crossfire [7]). It is difficult to identify the victim server at the first time that LFA starts, let alone the attack flows themselves.
- **Obvious pre-behavior**. Attackers, as long as they have not hacked for the topology map of a network area, they have to use current sniffer protocols such as *traceroute* to learn the topology. We can record traceroute messages for LFA defense in current Internet, but not efficient in ICN without the built-in traceroute functionality.

2.2 MFTP′s per-hop reliability

The traditional TCP failed to adapt MobilityFirst deal to its end-to-end pattern and its congestion signal strategy based on packet loss [11]. To deal with this problem, MFTP, as a per-hop reliable transport mechanisms is proposed for MobilityFirst. In per-hop reliable pattern, data are transported in hops, where data chunks are acknowledged and retransmitted in every hop so that they stay integrated through the network. With the per-hop reliability, MFTP solves many performance problems which used to bother TCP a lot, yet gains promising advantages in mobility support, in-network cache and delay-tolerance.

3 Attack Model

3.1 unevenly distributed botnets

The LFA we study is under the assumption that the bots are unevenly distributed in MobilityFirst. Some areas (e.g., domains, autonomous systems) with weaker security strategies are more likely to be infected than the others, hence have more bots. Thus we make an attack model that the MobilityFirst network is constructed by the domains with high density of bots and the domains with little density of bots.

3.2 Remote rerouting strategy

TSD aims to push the rerouting location further from the target link and distinguish attack flows and legitimate flows more precisely.

The most obvious difference between the attack flows and the legitimate flows is that the attack flows insist on passing through the target link but the legitimate flows do not. In the crossfire attack pattern, if the attack flows are rerouted by the traffic engineering (TE) mechanism, the attacker will choose a group of new decoy servers and generate a new group of attack flows that insist on flooding the target link. Moreover, the attacker may have carefully picked the decoy servers to decrease the possibility of the attack flows being rerouted out of the target link. For example, the attacker can pick the only entrance link to the decoy server as a target link. Thus rerouting strategy on the attack flows is less efficient than the legitimate flows.

Moreover, in a decentralized network, the shortest routing strategy (i.e., RIP [8] and OSPF [9]) makes some links naturally more likely chosen to forward the traffic. Hence the nearer the rerouting point locates to the target link, the fewer alternative rerouting choice a flow has. Most of the traditional traffic engineering (TE) mechanisms who announce available defending against LFA react locally nearby the victim area and lack of globally cooperation to efficiently trace the remote path of the attack flows. Thus traditional TE mechanisms have fewer alternative paths to reroute the excessive traffic. In the crossfire attack, this can cause several parallel paths congested at the same time, indirectly helping the attacker to flood all the links around the victim server.

On the opposite, the legitimate users are usually wide spread in the entire network and they do not care which route their flows choose. What′s more, for a legitimate flow from a remote source, its

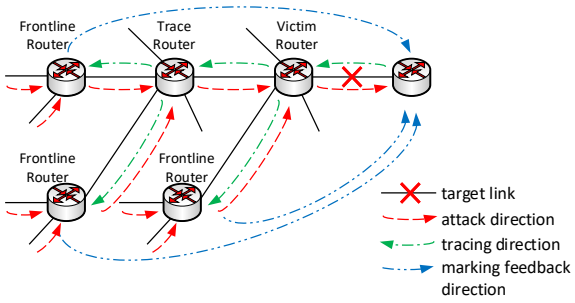

Figure 1: Overview of TSD mechanism

communication pair server may not be in the target area. Many of the legitimate flows may only pass through the target area and do not end in it. Thus we can see that compared to the attack flows, rerouting on the legitimate flows is easier and more acceptable. And rerouting them from where is further from the target area obviously brings higher possibility to avoid passing through it.

4 TSD Mechanism

In this section, we describe how TSD detect and defend against LFA in MobilityFirst network. As shown in Fig.1

4.1 Step 1: Detection

4.1.1 record

TSD is stateless that no historical flow information is required. The only historical information is the traffic record in every ingress-egress interface pair within a router, which is small in memory consumption and scalable to implement in large scale network. TSD routers record the average traffic in every ingress-egress interface pair cyclically (i.e., for a router owning n full duplex interfaces, there are n^2 traffic values recorded per cycle). Our purpose is to gain a common traffic statistics for a temporarily stable network topology, as a reference threshold to judge uncommon traffic increase as the attack signal.

The record cycle T depends on the two factors: the frequency υ_1 that the network changes its topology, and the required reaction rate υ_2 for the attack. The allowed maximum T value is given:

$$T_{\max} = \min(\upsilon_1/10, \upsilon_2/10)$$

4.1.2 alarm

When there is LFA happening, not only the target link is flooded, the traffic towards the target link will unusually increase as while. The router compare the newest traffic data to the historical record of the same ingress-egress interface pair. When an egress interface has been congested for a long time, and some of ingress interfaces have received unusually increased traffic heading to this egress interface, we can confirm that there is LFA happening.

One thing that can not be ignored is that, the changing of the network topology can also cause long-term congestion and incoming traffic increasing. If the network topology changes, the traffic record has no longer reference value for LFA detection. Hence the traffic record seems to have its own validity period, which is limited within the current topology. However, no matter attack caused or topo-changing

caused, long-term congestion need incoming traffic limitation anyway. Whoever responsible the most for the congestion, its traffic rate needs to be limited. Thus the topology changing is not an uncontrollable factor to TSD. We see long-term congestion as a hard judging point that TSD should start activating and reacting.

We see that the network congestion control mechanisms deal with normal network traffic flipping, and the TE mechanisms deal with routing unbalance caused by inappropriate topology. If the congestion lasts for an obviously long time, which is significantly beyond the reaction delay of the congestion control mechanisms and the TE mechanisms, we can infer that the congestion is caused by the sudden excessive traffic, i.e, burst service or flooding attack.

We assume that burst services are unlikely to frequently happen among normal users, especially when they happen to generate a huge quantity of traffic at the same time. Of course such collective behaviors may happen in particular activities such as general online voting, discounted online shopping or synchronous broadcast. But the network should make particular policies for these special moments, which is beyond our general focuses in this paper.

Thus we judge that there is LFA attack when the two conditions below are satisfied in a single router:

- One of the egress interfaces E_c is congested for a long time which is significantly beyond the reaction delay of the congestion control mechanisms and the TE mechanisms in the network.
- The total number of the flows from all the ingress interfaces forwarded to E_c significantly increases.

The first condition is the basic rule to distinguish common congestion and uncommon congestion. The second condition is to eliminate the possibility of some of the legitimate traffic, since there can be legitimate traffic increases when the number of flows does not increase. For example, current sleeping flows can be activated when new missions arrives, such as large file downloading who generates large traffic. Thus the increasing number of the flows is a necessary judge point to detect LFA.

In MobilityFirst, the higher dynamic the topology is, the more common the congestion is. Thereby, it is reasonable to assume that the congestion control mechanism of the dynamic network is efficient when there is no attack. Links may be congested for a while, but the congestion will soon be recovered. When there is LFA happening, the attack traffic can continue for a long time to guarantee the effectiveness of the flooding on a target link. The average traffic during the attack can be significantly higher than that when there is no attack. Thus we can monitor the time that a congestion situation lasts to judge if there is a LFA happening. What$'$s more, we can also trace the attack path according to the unusual traffic increase at the ingress interface of the target link.

There is another normal situation causing long-time congestion on a particular link. In a network whose topology changes dynamicly and the topology may become dumbbell shape at some point. The bottleneck link at the center of the dumbbell topology has to bear all the traffic between the two sides of the dumbbell without choice. However, even there is no attack, such situation can be seen as route failure and the upstream traffic need to be reduced anyway. Our mechanism can recover such failure as while.

4.2 Step 2: Trace

4.2.1 victim router

Once LFA is detected, the router owning the congested link becomes the *victim router*, and it starts to trace the incoming direction of the attack flows. Under the assumption that some of the ingress interfaces gathers most of the attack flows while the others do not, we need to find those interfaces who are responsible for the attack most, and then trace upstream.

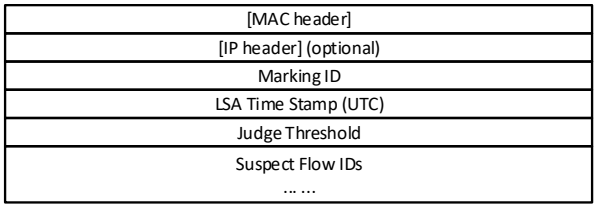

Figure 2: Format of request message

In the victim router, naming the egress interface owning the congested link as E_c, we judge and confirm n ingress interfaces (i.e.,from I_1 to I_n) on the victim router, which should be traced when the two conditions below are satisfied:

- The traffic from I_i to E_c has significantly increased compared to the historical record.
- The ratio of the traffic from I_i to E_c to the total traffic to E_c has significantly increased compared to the historical record.

The first condition is the basic rule to judge which ingress interface is mostly responsible for the congestion on E_c. The second condition is to limit the reaction strength of TSD, since not all ingress interfaces with increasing traffic have to be limited. If limiting less interfaces can prevent the congestion, limiting more interfaces will be unnecessary. Hence TSD use a dynamic ratio value as the threshold to decide the number of interfaces to be traced and limited.

Once I_i is judged as one of the most probable entrances of the attack flows, the victim router sends a request message through I_i to its adjacent upstream router. The request message is used to trigger a cooperative trace process in the adjacent upstream router, its format is shown in Fig.2. First, a detected LFA attack has a unique marking ID to represent, the marking ID can be generated by a hash sequence combining the start time (UTC) of LFA and the MAC address of E_c. Second, the start time (UTC) of LFA should be announced to the upstream routers for the scanning on their traffic record to see which upstream direction had unusually traffic increasing when LFA started. Third, the threshold value represents the reaction strength of one scanning process. It tells exact what traffic increasing should a ingress interface reach can it be traced. Finally, the request message announces all of the suspect flows judged by the downstream router.

One thing that needs to be highlighted is, as a stateless mechanism, TSD traces the attack traffic mainly according to the interfaces, rather than the flows. Although recording all the passing flows on the victim router seems easier and more specific to find the attack flows according to their misbehavior. However, recording specific information of every flow passing through needs large scale of state table on every router, causing great memory consumption and poor scalability in large scale network. The interfaces, on the other hand, are limited in number and easy to record their traffic, which is especially appropriate for the dynamic networks. When the topology of dynamic network keeps changing, the flows may be rerouted by TE mechanisms and difficult to trace in long time scale, but the interfaces stably provide real-time statistics of traffic, which is more trustworthy for LFA detection and reaction.

4.2.2 trace router

Once a router receives a request message, it becomes the *trace router* and it is responsible to continue tracing the coming direction of the attack traffic according to the information that the message announces.

In the trace router, naming the egress interface who receives the request message as E_r, we judge and confirm n ingress interfaces which should be traced (i.e.,from I_1 to I_n) from all the ingress interfaces when the two conditions below are satisfied:

- The traffic from I_i to E_r has significantly increased compared to the historical record during and after the time that LFA started, yet staying in a high level till now.
- At least one flow whose ID is announced as a suspect flow by the request message comes from I_i.

The first condition is the basic rule to judge which ingress interface is mostly responsible for the traffic increasing on E_r. The second condition is to prevent the traffic based tracing losing focus on the current attack. It keeps I_i combined with the suspect flows no matter how many hops TSD has traced upstream.

Once I_i is judged as one of the most probable entrances of the attack flows, the trace router also sends a request message through I_i to its adjacent upstream router, just like what the victim router does. The adjacent upstream router who receives this request message becomes anther *trace router* and repeat the tracing process above. By such hop-by-hop tracing process cooperated by several trace routers, TSD can find several remote entrances of the attack traffic witch are far from the victim router.

4.2.3 frontline router

Once a trace router can no longer figure out which ingress interface has significant traffic increase to E_r, it means the traffic of the attack flows is covered up by the background traffic in this router. Then we name such router as the *frontline router*, meaning the most remote entrance router of the attack traffic that we can trace and confirm with our best effort.

The reason why the frontline router does not simply continue to trace the announced suspect flows is that the mission of TSD is not hunting the attackers, but to prevent the attack traffic from reaching the victim router and the target link. Some of the suspect flows may be the legitimate flows, hunting their terminal senders is meaningless and with high probability of accidental injury. What the frontline router plays is the gathering point of most of the attack traffic. Holding these points means holding the frontline of the attack and raise the success probability of intercepting the attack traffic.

4.3 Step 3: Identification

4.3.1 marking

Once the frontline routers are founded, each of them start marking the passing packets. The marking strategy is given below:

- Mark the packets belonging to the announced suspect flows as red packets, which are not allowed outing through the E_rs of all the trace routers and E_c of the victim router.
- Mark the packets not belonging to the announced suspect flows, but still outing through the E_r of the frontline router as yellow packets, which are not allowed outing through the E_c of the victim router, but allowed outing through the E_rs of all the trace routers.

It is easy to understand why the flows with the reported IDs are marked as red. While marking the other flows as yellow is because that in the frontline router, flows with non-reported IDs may also be the attack flows. The attacker makes the attack flows follow the behavior of the legitimate flows. Legitimate flows spend much time sleeping and waiting for data requests and do not transport data all the time.What′s more, the attacker also keep canceling and generating the attack flows in order to avoid

being detected according to the misbehavior of the long-life flows. Hence if the frontline router does not proactively mark and limit the unannounced traffic through the frontline router, the victim router will keep busy distinguishing and announcing new attack flows endlessly with poor efficiency to defend against LFA.

The reason why yellow flows are allowed outing through the marked egress interfaces in the trace routers is that yellow flows are consist of many legitimate flows. We only need to prevent them passing through the target link. Before that, there is no need to reroute or intercept yellow traffic, affecting too many innocent flows.

4.3.2 rerouting

Red flow outing through the E_rs of all the trace routers should be rerouted to another egress interface. If there is no alternative route direction, all the packets of this flow will be dropped. In network using the shortest and distributed routing strategy, such as RIP and OSPF, red packets rerouted by an upstream trace router may come back to another downstream trace router. If so, such packets should be rerouted again and may be rerouted for several times during their trip. It is worthy for many times of rerouting on a red flow because we need to try as many paths as possible to guide the legitimate flows away from the target link, and reduce the injury to innocent civilians to the minimum ratio.

4.4 Step 4: Defense

4.4.1 intercept

Reroute all the marked packets belonging to the red and the yellow flows. If the rerouting fails, then drop them.

4.4.2 calibration & fire again

See if the congestion of the target link has been relieved. If so, keep the current threshold unchanged and the flow ID table not renewed. If not, see if the total traffic from all the ingress interfaces to E_c has been reduced. If reduced, keep the current threshold unchanged and keep the flow ID table renewing. If not reduced, cut down the value of the threshold and keep the flow ID table renewing.

The reason why the threshold cutting down is more than renewing the flow ID table is that renewing the flow ID table is to maintain the efficiency of TSD on the attack flows, and cutting down the threshold is to strengthen the strength of defense. When the current strength of defense is not enough, we can cut down the threshold to engage more flows from upstream, then mark, reroute and intercept them. However, more flows engaged means more legitimate flows injured. Thus as long as TSD is proved to be efficient yet not sufficient, we keep renewing the flow ID table to sustainedly drive it to cover more attack flows, and also keep the victim router sending cooperative trace request messages until the congestion is exhaustively terminated. When it is proved that the congestion truly requires stronger defense level to be reduced, we cut down the threshold to reroute or intercept more upstream flows.

5 Evaluation

5.1 Implementation

The MobilityFirst architecture we study is a basic publisher/subscriber model, consisted with some content publishers (data senders) and some content subscribers (data receivers). Other than that, routers with basic chunk-level caching functionality and servers with basic domain-level service request resolution

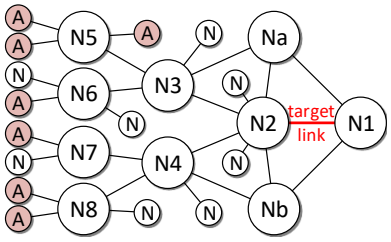

Figure 3: test topo of LFA

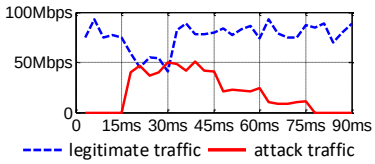

Figure 4: preliminary test on TSD efficiency

functionality are also designed. We develop this model into a prototype with 23,400 lines of C codes, and the source is open [1]. Also,we develop MFTP into a prototype with 1,500 lines of C codes, and the source is open [2].

We build testbed environment as a show case of LFA using 25 ATCA-9300 machines equipped with 8GB memory, 4 core Intel Xeon E3 1275V2 processor and 100Mbps bandwidth, 2ms delay interfaces. The topology of the testbed is shown in Fig.3.

5.2 Efficiency evaluation

In Fig.3, we set all terminal nodes marked with 'A' as attackers, all terminal nodes marked with 'N' as normal senders. Every terminal node sends a flow controlled by TCP. We manually limit the maximum rate to be 12.5Mbps, making the eight normal senders fulfill the entire bandwidth of the target link (100Mbps). On the router nodes Na and Nb, we set their static routing table as to forward all coming packets to node N1, the final receiver of all the flows. On the router nodes N3 and N4, we set their static routing table as to forward all coming packets to node N2. And in N2, we set it forward all packets to N1. That′s the basic settings for simulating the LFA attack.

Additionally for TSD, we set the rerouting strategy of N2, N3 and N4 as to forward the yellow and red packets to Na or Nb. However, in order to highlight the attacker′s target settings on the bots, we have also set that all attack flows, and the legitimate flows from normal users connected to N2, N3 and N4 are not allowed to be rerouted. Thus we have seven attack flows who insist to flood the target link, three legitimate flows that can be rerouted, and five legitimate flows that refuse to be rerouted.

As shown in Fig.4, the test runs on 8 legitimate flows started at the time 0 ms, and 7 attack flows started at the time 15 ms. When the attack flows are started, we can see seriously drop on the bandwidth occupancy of the legitimate flows. At the time 30 ms, the traffic of legitimate flows recovered, for the reason that 3 legitimate flows from N6 and N7 are rerouted and 3 attack flows from N5 has been intercepted. At the time 45 ms, 60 ms and 75 ms, 2 attack flows from N8, 1 attack flow from N7 and 1 attack flow from N6 are intercepted, respectively. Hence the attack traffic is reduced and a case of LFA is defended.

6 Discussion

Since TSD requires the cooperation of all the adjacent upstream routers, there may be a problem for the scalability for TSD in the current Internet. It may also be the main reason why TSD is now proposed only in MobilityFirst. MobilityFirst is implemented in evolving stacks and limited scale, making the routers in dynamic networks easy to modify or replace, but not on the current Internet. How to implement novel adjacent cooperative strategies like TSD on the Internet may be a useful topic for network innovations.

7 Conclusion & Future work

We propose a defense mechanism for the link flooding attack on the MobilityFirst networks, TSD. By simply analyzing the traffic record to trace the attack path, TSD achieves good efficiency in quick response for the LFA. We will continue using TSD to defend LFA in other dynamic network environments like the Delay Tolerant Network (DTN).

Acknowledgement

This paper is supported by NSAF under Grant No. U1530118, NSFC of China under Grant No. 61271202 and No. 61232017.

References

[1] Icn prototype source code with basic publisher/subscriber model. `https://github.com/bloodykapok/ICNprototype/` [Online; accessed on August 20, 2018].

[2] Mftp source code with basic per-hop reliability model. `https://github.com/bloodykapok/R2T/` [Online; accessed on August 20, 2018].

[3] Nsf mobility first project. `http://mobilityfirst.winlab.rutgers.edu/` [Online; accessed on August 20, 2018].

[4] D. Gkounis, V. Kotronis, C. Liaskos, and X. Dimitropoulos. On the interplay of link-flooding attacks and traffic engineering. *ACM SIGCOMM Computer Communication Review*, 46(2):1024–1049, April 2016.

[5] T. Hirayama, K. Toyoda, and I. Sasase. Fast target link flooding attack detection scheme by analyzing traceroute packets flow. In *Proc. of the 2015 IEEE International Workshop on Information Forensics and Security (WIFS'15), Rome, Italy*, pages 1–6. IEEE, November 2015.

[6] M. Kang and V. Gligor. Routing bottlenecks in the internet: Causes, exploits, and countermeasures. In *Proc. of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS'14), Scottsdale, Arizona, USA*, pages 321–333. ACM, September 2014.

[7] M. Kang, S. Lee, and V. Gligor. The crossfire attack. In *Proc. of the 2013 IEEE Symposium on Security and Privacy (S&P), San Francisco, California, USA*, pages 127–141. IEEE, May 2013.

[8] G. Malkin. Rip version 2. IETF RFC 2453, November 1998. `https://tools.ietf.org/html/rfc2453`, [Online; accessed on August 20, 2018].

[9] J. Moy. Ospf version 2. IETF RFC 2328, April 1998. `https://tools.ietf.org/html/rfc2328`, [Online; accessed on August 20, 2018].

[10] K. Su, F. Bronzino, K. K. Ramakrishnan, and D. Raychaudhuri. Mftp: A clean-slate transport protocol for the information centric mobilityfirst network. In *Proc. of the 2nd ACM Conference on Information-Centric Networking (ACM-ICN'15), San Francisco, USA*, pages 127–136. ACM, September 2015.

[11] K. Su, K. K. Ramakrishnan, and D. Raychaudhuri. Scalable, network-assisted congestion control for the mobilityfirst future internet architecture. In *Proc. of the 22nd IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN'16), Rome, Italy*, pages 1–2. IEEE, June 2016.

Author Biography

Zhaoxu Wang received the B.S degree in University of Electronic Science and Technology of China (UESTC) in 2013. He is currently pursuing the Ph.D. degree in the School of Electronic and Information Engineering, Beijing Jiaotong University (BJTU). He has participated in the National Basic Research Programs of China (973 Program), as the main developer of the network prototype. His recently research interests are in Internet architecture and reliable transport.

Huachun Zhou received his B.S. degree from People′s Police Officer University of China in 1986. He received his M.S. and Ph.D. degrees from Beijing Jiaotong University in 1989 and 2009, respectively. He is recently Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests include mobility management, mobile and secure computing, routing protocols and network management technologies.

Wei Quan received the B.S., M.S. and Ph.D. degrees from Beijing Jiaotong University. His research interests are in the wide areas of network technologies including routing, Internet architecture, and network security.