A Study on the Impact of Multiple Failures on OSPF Convergence

Dan Zhao, Xiaofeng Hu and Chunqing Wu

School of computer, National University of Defense Technology
Changsha, Hunan, China
danzhao.nudt@gmail.com,{xfhu,wuchungqing}@nudt.edu.cn

Abstract

Open Shortest Path First (OSPF) is a popular link state routing protocol widely used in Internet infrastructure. OSPF implements several timers to limit the protocol overhead. With these timers, it usually takes several tens of seconds for OSPF network to recover from a failure. The convergence time is delayed mainly by the timers of failure detection and routing calculation scheduling. In this paper we analyze OSPF convergence behavior in presence of multiple failures, where the interactions between failure detection and routing calculation scheduling could generate complicated dynamics during convergence process. We also present experimental study to understand the impact of multiple failures on convergence. The results demonstrate that multiple failures have a greater chance to delay the convergence. This suggests that operators should take it into account while configuring OSPF network.

Keywords: *OSPF, link state protocol, multiple failures, convergence*

1. Introduction

Open Shortest Path First (OSPF) [1] is a successful link state protocol that is widely used in intra-domain ISP networks. In OSPF network, every router establishes adjacency with its connected counterparts and describes the connection status using Link State Advertisement (LSA). Once the topology changes, routers employ specific mechanism, typically Hello protocol prescribed in OSPF standard, to detect the failure and generate new LSAs. After the synchronization of LSAs throughout the network by flooding, routers are capable of calculating the correct routing table for packet forwarding.

The primary philosophy of protocol design is to limit the processing/bandwidth requirements of the protocol, while the time required to recover from a failure in the network topology (speed of convergence) was of secondary importance [5]. The tradeoff between protocol overhead and efficiency is conventionally regulated by protocol timers. For instance, hello packet is periodically exchanged between neighboring routers with the frequency determined by ***HelloInterval***, which limits the number of hello packets. When routers are about to calculate the routing table using SPF algorithm, the calculation is delayed by ***spfDelay*** and ***spfHold***, expecting to acquire the most up-to-date RIB with fewer calculation.

With these timers taking effect, OSPF network normally converges in several tens of seconds because the timers are often configured in the granularity of second. Given the advent of real-time applications, significant attentions have been drawn to achieve fast convergence to accommodate uninterrupted traffic delivery. There have been proposals that reducing timers can achieve sub-second convergence [2], but it increases the processing overhead and convergence dynamics which might impact the stability. Hence the controversy of fast convergence and protocol stability requires continuous investigation.

In this paper we aim to understand OSPF convergence behavior, especially in presence of multiple failures. In recent years, there has initiated a research trend in measuring and

analyzing the impact of multiple and regional failures [6, 7]. Though the network topology change mostly involves single link failure, multiple failures do take place and may greatly change the topology [8]. Thus multiple failures probably have larger impact on network connectivity and protocol reaction behavior. Generally speaking, OSPF can definitely converge whatever the topology transitions are. However, the protocol has its intrinsic characteristics which may be enlarged by multiple failures. We intend to explore the dynamics and provide some insight to protocol design and configuration. Our analysis shows that the asynchronous detection of correlated failures can trigger multiple routing calculations, which introduce more delay to convergence.

The remaining paper is structured as follows. In Section 2 we briefly outline the convergence process and related timers of OSPF. Then we present our analysis in detail in Section 3. We also present the experimental study in Section 4, and then outline the related work in Section 5. At last we conclude in Section 6.

2. OSPF Convergence and Timers

Typical OSPF convergence includes several procedures: failure detection, LSA flooding, routing calculation and RIB/FIB update. Each procedure has corresponding timers to limit the protocol overhead. As all the timers are suggested to be configured to the granularity of second [1], they together introduce significant convergence delay. There are proposals to radically reduce these timers to sub-second range, but others claim that it may not be advisable because the side effect can damage the stability. In this section we focus on the failure detection and routing calculation related timers that are major components of convergence delay.

OSPF uses Hello protocol to detect the failure. It enables routers to periodically xchange hello packets to establish adjacency with the frequency determined by ***HelloInterval***. If one router hasn't received hello packets during ***RouterDeadInterval*** (typically 4 ***HelloIntervals***), the adjacency is declared to be down. Then corresponding LSAs are generated and flooded in the network by the router that detects the failure. The default value for ***HelloInterval*** is suggested to be 10 seconds [1]. Thus the network failure can be detected in 30 to 40 seconds after its occurrence. It's obvious that achieving faster failure detection can significantly accelerate convergence. However, reducing ***HelloInterval*** may result in false alarm because of link congestion or router CPU overload [10]. The chance of false alarm increases as ***HelloInterval*** becomes smaller. Besides, successive false alarms can cause persistent overloads on router CPUs that will ultimately result in complete meltdown of the routing function in the network. Therefore it may not be advisable to reduce ***HelloInterval*** to the millisecond range [10].

When new LSA reaches routers, routing calculation is scheduled. If a router execute routing calculation immediately after it receives a LSA, it may end up doing several time-consuming routing table updates in close succession because more LSAs will come to the router. This may keep the router CPU busy for a long time and prevent it from doing other important tasks such as processing hello and other protocol packets. There exists the possibility that these failures may snowball into a complete meltdown of routing functionality [9]. Hence OSPF uses a timer called ***spfDelay*** to delay the first routing calculation after a router initially receives new LSAs, hoping that the calculation is carried out based on the entire set of generated LSAs of topology change. If there are more LSAs received right after the first routing calculation, the upcoming calculations are delayed by a timer called ***spfHold*** which is dynamically adjusted according to the timing of successive calculations. Initially ***spfHold*** is set to a small value. Then the receipt of several LSAs during hold time after the last routing calculation will make ***spfHold*** quickly increase to a maximum value. If no LSA is

received during the current ***spfHold*** range, ***spfHold*** is reset to its initial value. This "***SPF Throttling***" scheme [13] is used to delay SPF calculation during network instability. Reducing these timers may gain faster convergence after single link failures, but it also may cause successive routing calculations [9] with considerable timer delay that slows the convergence and increased router load in case of multiple failures. We will illustrate this phenomenon in the following section.

3. Impact of Multiple Failures on Convergence

In this section we will illustrate the details of OSPF convergence behavior in presence of multiple failures. As multiple failures often take place where the failed components have geographical relationships such as Shared Risk Link Group (SRLG) [3] and regional faults caused by EMP attack [4] or natural disaster, we focus on concurrent regional failures in the rest of this paper.

3.1. Asynchronous Detection of Multiple Failures

As described in the previous section, neighboring routers exchange hello packets to maintain adjacency. The procedure of adjacency establishment forces a router to send a hello packet as a reply to its counterpart for adjacency negotiation. With the timer control, the operations of hello packets exchange eventually turn out to be synchronized at both end routers of single link. Therefore, both end routers can claim the adjacency down at nearly the same time if failures occur on the link.

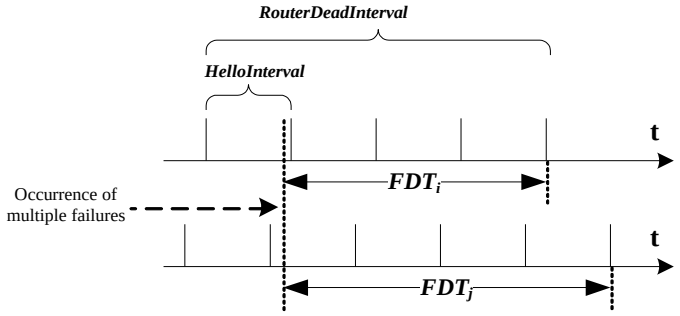


Figure 1. An Example of Asynchronous Detection of Multiple Failures

However, the adjacency maintenance is only related to single link. More precisely, it depends on the timing of hello packets exchange on specific interface of router. Multiple failures involves plenty of links, thus the failure detection corresponds to multiple neighboring routers. It can be seen that the detection of multiple link failures is asynchronous, meaning that the time it takes to detect each link failure differs from each other. Suppose that a number of routers, denoted as $\{FR\}$, fail simultaneously (shown in Figure 1). One neighboring router of FR_i that is about to expire ***HelloInterval*** right after the failure occurrence finally found that no hello packet is received during ***RouterDeadInterval***. There may be another neighboring router of FR_j that has just received a hello packet before the failure of FR_j. Therefore, the failure detection time (FDT) of the two neighboring routers differs because they connect to different failed routers and have asynchronous failure detection behavior. Given that hello packet is sent every ***HelloInterval***, we can easily

conclude that the most time variation can't exceed ***HelloInterval*** range. That is, $\Delta FDT(i, j)=FDT_j - FDT_i <$ ***HelloInterval***.

3.2. Scheduling Routing Calculation

After multiple failures are detected, corresponding routers generate LSAs describing the topology change and flood them out. For a particular router, LSAs may arrive at it going through different paths. However, the time of overall propagation delay ordinarily stay in the range of several hundreds of milliseconds according to physical characteristics of transmission medium nowadays. As long as the protocol timers are set in the granularity of second, the propagation delay would have negligible impact on convergence. Thus we don't consider propagation-induced dynamics in this section.

When routers have received LSAs, routing calculations are scheduled. As described in Section 2, routing calculation is delayed in order to incorporate as many LSAs as possible. It is obvious that LSAs of single link failure would have the calculation be delayed only by ***spfDelay*** because all LSAs reach routers with little time difference. However, since the detection of multiple failures behaves asynchronously, it increases the chance that both ***spfDelay*** and ***spfHold*** take effect because successive calculations are likely to be invoked. More precisely, the routing calculation delay depends mainly on the time span of multiple failure detection. We denote this variable as ΔFDT_{max} and analyze how it impact routing calculation scheduling.

Suppose failures concurrently occur at time 0. When $\Delta FDT_{max} \leqslant$ ***spfDelay***, it means all LSAs would arrive at a particular router within [0, ***spfDelay***] range (assuming propagation delay is negligible). When routing calculation is scheduled by the time that the first LSA arrives, it would be only delayed by ***spfDelay*** and the SPF algorithm is executed for only once.

However, if $\Delta FDT_{max} >$ ***spfDelay***, there must be some LSAs arrive at a router after the first routing calculation. We denote the time of last routing calculation completion as t_{spf}. Then successive routing calculations are delayed as the following situations:

(1) There exists $\Delta FDT_i < \Delta FDT_{max}$ that satisfies $\Delta FDT_i \leqslant$ ***spfDelay*** + ***spfHold***. In this situation some LSAs arrives in [t_{spf} , t_{spf} + ***spfHold***] range, and the following routing calculation is going to be delayed by current ***spfHold***. Also ***spfHold*** is increased for potential successive scheduling.

(2) If there is $\Delta FDT_i < \Delta FDT_{max}$ that satisfies $\Delta FDT_i >$ ***spfDelay*** + ***spfHold*** and the situation of (1) doesn't exist, some LSAs would arrive at some routers after t_{spf} + ***spfHold***. Then the following routing calculation is delayed by ***spfDelay***, and ***spfHold*** is set to its initial value.

Note that the situations described above can be recurrent and overlapping. Due to the asynchronous detection of multiple failures, the LSAs could arrive at a router in close succession, e.g. each is in [t_{spf} , t_{spf} + ***spfHold***] range with current ***spfHold*** value. This can quickly increase ***spfHold*** to its maximum value. Otherwise, there may be some moment that no LSA is received during [t_{spf} , t_{spf} + ***spfHold***], then the situation of (2) occurs.

It can be seen that when multiple failures show up, the convergence may be largely delayed by OSPF timers even if the failures happen concurrently. The protocol is supposed to converge as fast as possible when there are simultaneous failure events. Concurrent multiple failures, especially when the failures are regionally related, can be treated as isolated topology change and barely represent network instability. However, the interactions between failure detection and routing calculation generate complicated reactive behaviors and dynamics

which stretch convergence duration in presence of multiple failures. Considering that multiple failures greatly change the network topology that requires reconstruction of many end-to-end paths, delayed convergence definitely slows down the reconstruction and largely impact the service quality of applications.

4. Experimental Study

In this section we perform extensive experiments on emulation system with real routing platform and topology to evaluate the convergence delay in presence of multiple failures.

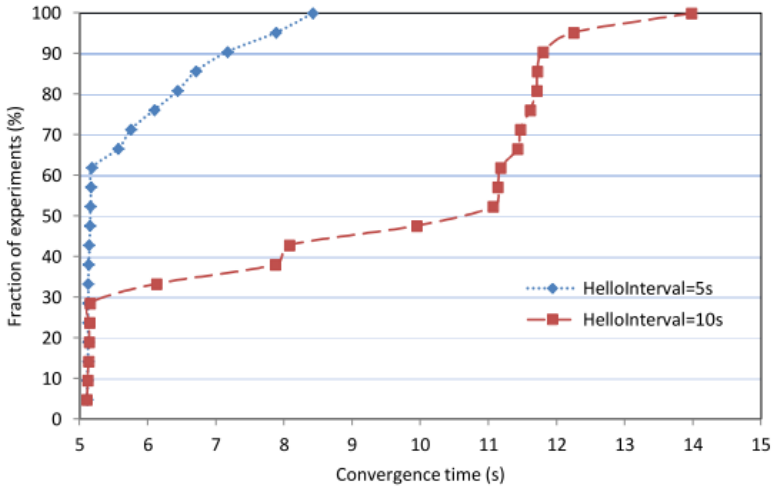


Figure 2. Convergence time in presence of single router failures

4.1. Methodology

We perform experiments on an emulation system named *CORE* [11] that uses the Linux virtualization provided by *OpenVZ* [14] to build non-modified operating systems for running real applications. Each virtualized node in CORE runs *Quagga* [15] which implements full set of OSPF functionality by *ospfd* daemon. The ***spfDelay*** is set to default value of 5s, and the initial and maximum value of ***spfHold*** is 1s and 10s respectively. If successive routing calculations are scheduled, *Quagga* increases ***spfHold*** in a linearly way. Hence it will execute routing calculation for 10 times until ***spfHold*** reaches its maximum value.

The experiments are conducted using a real ISP backbone topology, AS3967, reported from Rocketfuel [12]. There are 79 routers and 147 links in this topology, and all the link latency is set to 10ms which brings in trivial propagation delay. To invoke the convergence process, we intentionally inject failure by kill *ospfd* process inside the virtualized operating system such that the router disables OSPF functionality. The failure scenarios include single and multiple router failures. For single router failure, it means that all its connections are broken down. We extract each router's geographical information to divide the routers into 21 groups, so that all routers in a group are brought down simultaneously. We monitor the convergence time with different ***HelloInterval*** to see how convergence is delayed in presence of multiple failures.

4.2. Results

In order to clarify how the convergence is delayed by timers related to failure detection and routing calculations, we refer to convergence time here as the duration from the moment that the failure is initially detected to the time the last router has updated its routing and forwarding table. We set ***HelloInterval*** to 5s and 10s, and the results of convergence time are shown by Figure 2 and Figure 3 respectively.

We can clearly observe from Figure 2 that the convergence delay is largely increased as ***HelloInterval*** becomes longer. When ***HelloInterval*** is set to 5s, the detection time variation would not exceed ***spfDelay*** according to our previous analysis. Considering that routing calculation only takes a few hundreds of milliseconds, the convergence time is expected to be less than 6s. However, there are some network branches that only connect to the failed router. They are isolated from the topology and cannot receive LSAs from other partitions when the failure occurs. Unfortunately, the failure detection by the isolated partition is slower than other partitions, thus the convergence time of entire network is delayed. In our experiments the delay is more than 3s at most, so the convergence time reaches more than 8s in some cases. When ***HelloInterval*** comes to 10s, it increases the chance that both ***spfDelay*** and ***spfHold*** may delay successive routing calculations, as well as slower failure detection exists in isolated network partitions. Therefore the convergence time takes at most about more than 13s in our experiments.

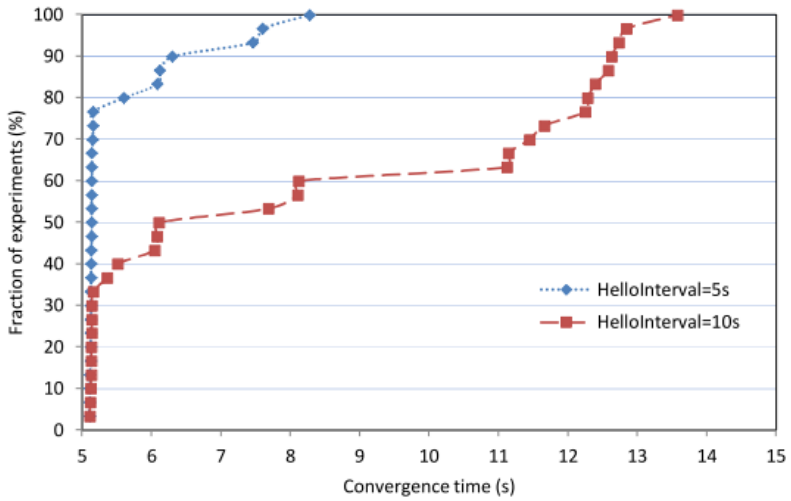


Figure 3. Convergence time in presence of multiple router failures

When multiple failures happen, more delay is introduced into convergence process. This is partly because multiple failures have a greater chance to partition the network into several isolated parts. The partition where the failure is lastly detected contributes to the most convergence time. On the other hand, the effect of asynchronous detection of multiple failures is more obvious. When ***HelloInterval*** is set to 10s, only less than 30% of experiments converge less than 6s where there is only single run of routing calculation delayed by ***spfDelay***. For those convergence takes time in [6, 7] range, a considerable number of routers experience 2 routing calculations delayed by ***spfDelay*** and ***spfHold*** which is 1s at the beginning. We can observe from figure 3 that convergence time of some scenarios locate in [8, 9] range. This is because routing calculation is scheduled for 3 times, and the last delay of ***spfHold*** has increased to 2s. When convergence time exceeds 11s, there are two situations. In

some circumstances, there are 4 successive running of routing calculation and ***spfHold*** is adjusted to 3s, which adds up the delay to 11s. Another possibility is that routing calculation is scheduled twice, but the second is scheduled beyond ***spfHold*** after the fist calculation. Therefore the second calculation is delayed for 5s as well, and the additional time comes from the detection time variation. In our experiments, there are more than 50% of convergence experiments that take more than 10s to stabilize the network after multiple failures. The results demonstrate that the convergence can be delayed by timers because of protocol reaction to multiple failures.

5. Related work

Improving OSPF convergence is always a hot topic in network research area. The network is expected to converge as soon as possible when the topology changes. Considering the delayed response of OSPF timers [1], researchers have proposed algorithms and schemes to avoid convergence process. The IETF IPFRR framework [16] proposes to use precomputed backup paths to reroute around failures in IP networks. MRC [17] is to use the network graph and the associated link weights to produce a small set of backup network configurations. The goal of FCP [18] is eliminating the convergence process completely that allowing routers to find a working path. However, these techniques resemble the patches that need to be added to protocol and require complex configuration.

Compared with the patch-like schemes, the intuitive way is to reduce protocol timers to accelerate the reaction of OSPF. P. Francois et al. proposed that sub-second convergence can be achieved by setting timers to millisecond order of magnitude [2]. However, others have investigated the impact of timer regulation on failure detection [10] and routing calculation scheduling [9]. Their results show that small timer values can cause complex dynamics as well as prolonged convergence delay. Besides, the setting for timer values depends on various parameters such as the topology size, the density of connection and the expected congestion level. Therefore it requires careful investigations to decide the optimal value for a given network, and it may not be suitable to set the timers in millisecond range.

Most existing researches about OSPF convergence are mainly based on the assumption that there is single failure in the network. It's publicly known that network failures are mostly in the category of single link failure [8], but multiple failures do occur sometimes. Recently, researchers begin to notice the great impact of multiple failures caused by EMP [4] attacks and dragging anchors [19], as well as natural disasters such as earthquakes, hurricanes and floods. Since a considerable number of routers and links fail at the same time can cause great transition to topology, network routing is definitely impacted by multiple failures. However, most existing researches concern about assessing the vulnerability of topology in presence of various multiple or regional failure model [6, 7, 20]. Their contributions are to provide some guidelines to topology design and maintenance. To our knowledge, we are the first to analyze the impact of multiple failures on convergence dynamics. The purpose of this paper is to present fundamental insight into protocol design and configuration that can help improving network convergence.

6. Conclusion

In this paper we formally analyze OSPF convergence dynamics in presence of multiple failures. According to our analysis, the convergence is mainly and largely delayed by protocol timers. To our point of view, the cause of such dynamics mainly lies in that the detection of multiple failures is asynchronous. The results of our experimental study also demonstrate that

convergence is greatly delayed in presence of multiple failures. This suggests that network operator should carefully configure OSPF protocol taking into account their dependency and possible network failure scenarios, aiming that network could gain faster convergence while keeping the processing overhead in considerable level.

According to our research, it seems that decrease the timer value may help alleviating the impact on convergence delay. However, smaller timers may overreact to subtle network change and amplify network instability. Tuning timers requires extensive investigation on specific network and much experience about network management. Furthermore, we only study the impact of concurrent multiple failures in this paper. We believe that the failures that have cascading characteristics can result in tremendous impact than normal multiple failures, and this is the issue that we hope to address in the future.

Acknowledgements

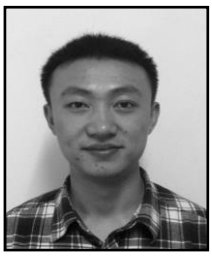
The work described in this paper is supported by the NSFC under Grant No.61103189 and No.61070199, Program for Changjiang Scholars and Innovative Research Team in University (No.IRT 1012), Program for Science and Technology Innovative Research Team in Higher Educational Institutions of Hunan Province: "network technology", and Hunan Province Natural Science Foundation of China (11JJ7003).

References

[1] J. Moy, OSPF version 2, Internet Engineering Task Force, Request For Comments (Standards Track) RFC 2328, **(1998)** April.
[2] P. Francois, C. Filsfils, J. Evans and O. Bonaventure, "Achieving sub-second IGP convergence in large IP networks", Computer Commun. Rev., vol. 35, no. 3, **(2005)**, pp. 35-47.
[3] J. Strand, A. Chiu and R. Tkach, Issues for routing in the optical layer, IEEE Communication Magazine, vol. 39, **(2001)**, pp. 81-87.
[4] J. S. Foster Jr., "Report of the commission to assess the threat to the United States from electromagnetic pulse (EMP) attack", vol. I, Executive report, **(2004)**.
[5] M. Goyal, "Improving Convergence Speed and Scalability in OSPF: A Survey", IEEE Commun. Surveys & Tutorials, vol. 14, no. 2, **(2012)**, pp. 443-463.
[6] S. Neumayer and E. Modiano, "Network Reliability With Geographically Correlated Failures", in Proc. of IEEE INFOCOM, **(2010)**.
[7] S. Neumayer, G. Zussman, R. Cohen and E. Modiano, "Assessing the Impact of Geographically Correlated Network Failures", in Proc. of IEEE MILCOM, **(2008)**.
[8] A. Markopoulou, G. Iannaccone, S. Bhattacharaya, C. Chuah and C. Diot, "Characterization of failures in an ip backbone", in Proc. of IEEE INFOCOM, **(2004)**.
[9] M. Goyal, W. Xie, M. Soperi, H. Hosseini and K. Vairavan, "Scheduling routing table calculations to achieve fast convergence in OSPF protocol", in Proc. IEEE Broadnets, **(2007)** September.
[10] M. Goyal, K. Ramakrishnan and W. Feng, "Achieving faster failure detection in OSPF networks", in Proc. IEEE International Conference on Communications (ICC2003), **(2003)**, pp. 296-300.
[11] J. Ahrenholz, "Comparison of CORE Network Emulation Platforms", in Proc. of IEEE MILCOM Conference, **(2010)**, pp. 864-869.
[12] N. Spring, R. Mahajan and D. Wetheral, "Measuring ISP topologies with RocketFuel", In Proc. of ACM SIGCOMM, **(2002)**, pp. 133-145.
[13] Cisco, OSPF Shortest Path First Throttling, http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fs_spftrl.html.
[14] OpenVZ Linux Containers, http://wiki.openvz.org/.
[15] Quagga Routing Suite. http://www.quagga.net.
[16] M. Shand and S. Bryant, "IP Fast Reroute Framework, Internet Engineering Task Force", Request For Comments (Standards Track) RFC 5714, **(2010)**.
[17] A. Kvalbein, A. F. Hansen, T. Čičić, S. Gjessing and O. Lysne, "Fast IP network recovery using multiple routing configurations", in Proc. of IEEE INFOCOM, **(2006)** April.

[18] K. Lakshminarayanan, M. Caesar, M. Rangan and T. Anderson, "Achieving Convergence-Free Routing using Failure-Carrying Packets", in Proc. of ACM SIGCOMM, **(2007)** August.
[19] J. Borland, "Analyzing the Internet collapse", MIT Technology Review, **(2008)** February, http://www.technologyreview.com/Infotech/20152/?a=f.
[20] S. Banerjee, S. Shirazipourazad and A. Sen, "Design and Analysis of Networks with Large Components in Presence of Region-Based Faults", in Proc. of IEEE International Conference on Communications (ICC2011), **(2011)**, pp. 1-6.

Authors



Dan Zhao received the B.S. degree and M.S. degree from the National University of Defense Technology (NUDT) in 2006 and 2008, respectively, all in school of computer. He is a PhD candidate in Institute of Network and Information Security, National University of Defense Technology (NUDT) since March 2009. His current research interests are in network architecture, routing and protocols.

Xiaofeng Hu received his Ph.D. degree in computer science from National University of Defense Technology, China, in 2004. He is currently an associated professor in the School of Computer at the same university. His research interest includes Internet architecture, routing protocol, and high performance router design.

Chunqing Wu received the PhD. degree from National University of Defense Technology, China. She is now a professor in School of Computer, National University of Defense Technology. Her current research interest includes high performance routers, Internet routing and space network.