國立交通大學

資訊科學與工程研究所

碩士論文

針對 OSPF 網路的快速偵測錯誤方法

Fast Failure Detection Methods in OSPF Networks

研 究 生：陳福文

指導教授：簡榮宏　教授

中 華 民 國　九 十 六　年 六 月

針對 OSPF 網路的快速偵測錯誤方法

Fast Failure Detection Methods in OSPF Networks

研 究 生：陳福文　　Student：Fu-Wen Chen

指導教授：簡榮宏　　Advisor：Rong-Hong Jan

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer and Information Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Science

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

針對 OSPF 網路的快速偵測錯誤方法

研究生：陳福文　　指導教授：簡榮宏 博士

國立交通大學資訊科學與工程研究所

摘　　要

在 OSPF 網路中路由器是透過 HELLO 協定去得到周遭相鄰路由器的資訊。在標準的 OSPF 設定之下，網路的路由表通常需要花費數十秒的時間才能收斂。而 *HelloInterval* 值的大小深深影響著錯誤偵測的時間長短，因此我們可以透過縮短 *HelloInterval* 的值來加速錯誤偵測的時間。而合適的 *HelloInterval* 的值會受到一些網路環境變數的影響，例如：網路的大小，網路壅塞的程度…等等。然而，用人為的方式去設定合適的 *HelloInterval* 相當沒效率，因此我們採用動態且自動的方式去調整 *HelloInterval* 並去取代原始的 OSPF 設定。透過模擬結果可以觀察出錯誤偵測的時間以及網路收斂的時間都有所改善。

Fast Failure Detection Methods in OSPF Networks

Student: Fu-Wen Chen Advisor: Dr. Rong-Hong Jan

DEPARTMENT FO COMPUTER AND INFORMATION SCIENCE
NATIONAL CHIAO TUNG UNIVERSITY

Abstract

In OSPF networks, the routers use Hello protocol to get neighbor's information. With default settings of standard OSPF protocol, the network usually takes several ten seconds to detect a failure. The value of *HelloInterval* influences the failure detection time highly; therefore, we want to shorten the *HellonInterval* to achieve faster failure detection. The suitable value of *HelloInterval* depends on several network environment parameters, such as network size, network congestion level and so on. However, it is not efficient to adjust the value of *HelloInterval* manually. We design an adaptive *HelloInterval* adjustment method which can adjust *HelloInterval* automatically to replace default setting in OSPF protocol. Simulation results show that the proposed method can accelerate the failure detection time and improve the network convergence time. And it can be easily applied to OSPF networks without any setting of *HelloInterval*.

致謝

在這兩年的研究所生活，首先感謝我的指導教授簡榮宏博士，老師悉心的教導使我得以了解 OSPF 路由協定領域的深奧，不時的討論並指點我正確的方向，使我在這些年中獲益匪淺；論文的完成另外亦得感謝老師的大力協助，使的論文能夠更完整而嚴謹。

兩年裡的日子中，實驗室裡的學長姐(鴻棋、世昌、安凱、嘉泰、蕙如)、同學(奕睿、祐慈、依璇、苑瑩)以及學弟妹們(祐笙、敬之、宇翔、俊傑、允琳)的共同努力以及共同生活的點點滴滴，不管是在學術上的討論或者言不及義的閒扯，讓我的這兩年的研究生生活變得絢麗多彩，尤其是實驗室的共同出遊更讓我留下美好的回憶。

感謝嘉泰學長百忙之中能提供我研究思考的方向，且總能在我迷惘時為我解惑，學長幽默且風趣的談吐讓我對於辛苦的研究生生活更多采多姿，也感謝同學的幫忙，恭喜我們順利走過這兩年。

最後想好好感謝我摯愛的雙親，由於他們在背後默默的支持以及鼓勵我；在我這求學期間，您們的關心讓我能專心一致地在研究領域上有所專研，此外我的女朋友心儀也陪伴著我和我一起分享在這求學研究過程的點點滴滴，這些都是讓我擁有前進的動力，沒有你們的體諒、包容，相信這兩年的生活將是很不一樣的光景。

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Routing is the one of the main components in the Internet. In wired network, all routers must require several routing protocols to help them forward their network packets. Among these routing protocols, Routing Information Protocol (RIP) [1] and Open Shortest Path First (OSPF) [2] are the two most widely used routing protocols. The OSPF protocol is a link-state routing protocol and is a common Interior Gateway Protocols (IGP) in the network today. All routers running OSPF will exchange their information and the associated weights, so each router knows the topology of the network. With this collected information, each router uses this information to determine the shortest paths to different destinations.

In OSPF networks, routers send periodic messages (HELLO) typically to check the activity of neighbor routers. After sending periodic messages, they also describe their local environment information by advertising the state of their local links and the neighbor routers with the linking costs. These advertised descriptions are called link-state advertisements (LSA), which are flooded through all routers in the routing area. The routers in the routing area collect these LSAs, and then maintain a distributed database which describes the topology information. All routers in this topology have the same view on the structure of network.

However, when there are some failures (link or router failure) or other events, the OSPF routers spend a little time detecting the failures and flood the LSAs through the

network to re-establish a consistent view of the new topology. During this transmission, the packets forwarded toward the failed device will be dropped because they do not re-calculate the newest routing path and re-establish their forwarding table. So hosts in this network will lose their data packets, and then they must retransmit the loss data. Additionally, the traffic might lead to congestion in the network.

Recently, more and more user applications require high quality of service and high availability in the network. However, we want to know how to estimate the service quality of network. One of the measurements to the service quality is the routing convergence time. During the convergence time, the packets in the network may be dropped by routers and get delayed because of routing loops or network congestion. Hence, reducing the convergence time can provide better quality in the routing networks and the user applications can get better responsibility because of less network convergence time. But there are some factors that influence the network convergence time in OSPF networks. In these factors, the HELLO interval affects the OSPF convergence time very much. The reason is that two adjacent routers in the same area periodically send Hello messages to maintain the link adjacency, if a router does not receive a Hello message from its neighbor within a dead interval, it assumes the lik between itself and the neighbor to be broken and then the router will generate a new LSA flooding through the network to reflect the changed topology. So the convergence time highly depends on HELLO interval. Hence, we want to modify the standard OSPF routing protocol's HELLO interval such that the HELLO interval can be adjusted dynamically.

The remainder of the thesis is organized as follows: In chapter 2, we discuss the three factors about convergence time in OSPF networks. In chapter 3, we present our

adaptive HELLO protocol to reduce convergence time. And we evaluate the failure detection time and convergence time by simulation in chapter 4. Finally, we will give the conclusion in chapter 5.

Chapter 2
Related Works

In this chapter, we will discuss the factors that influence the convergence time in OSPF networks. The convergence time [3][4] in OSPF can be divided three aspects to discuss. They are:

1. the failure detection time [5];
2. the LSA flooding time [6];
3. the shortest path first calculation and forwarding table update time [5][6].

We will discuss them in the following sections.

2.1 The Failure Detection Time

In wired networks, multiple physical links interconnect the routers and the hosts. The routing protocol will construct the routing path for each connection, and then network packets will traverse correctly through the network. Whenever a link or router fails, dynamic routing protocols, like OSPF or RIP, will re-calculate another shortest path towards the destination. For network convergence time, it is essential that the routing protocol detect the link or router failure quickly. Recently, more and more user applications require high availability network. With these applications, the network must maintain higher responsibility, so the network failure detection time and recover time must be reduced. If we have less failure detection time to discover failures, we will speed up network convergence and fewer data packets are dropped.

There are two solutions to reduce the failure detection time. First, the more convenient and faster method to detect failure can be achieved by co-operation with physical layers. In other words, we can detect failure occurrence by hardware devices [7]. For example, if we are able to detect a link breakdown by hardware, e.g. by the loss of the physical or optical signal, they can immediately notify the network layer about the failure. But this implementation has some disadvantages. We consider the case that switches are involved in the router interconnection, and then the links may fail behind such a switch, prevents the chance for the fast link level failure detection. Due to this reason, we need routing protocols, like Routing Information Protocol (RIP), Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS)[8] or Border Gateway Protocol (BGP)[9] to help us. These routing protocols typically send periodic messages to check whether their neighbor is still reachable and alive. However, these periodic check messages, such as KEEPALIVE or HELLO messages, consume network bandwidth and router's processing time.

In OSPF, the adjacent routers send their HELLO message to maintain their link adjacency. If a router does not get any HELLO messages from his neighbors during a *RouterDeadInterval* (default 40 seconds or 4 times *HelloInterval*), it will assume the link between itself and the neighbor to be down, as shown in Figure 2.1. Then this router will generate a new LSA to reflect the changed topology and flood this LSA throughout the network. All routers that receive this LSA will redo the shortest path first (SPF) calculation and update the next hop information in their forwarding table.

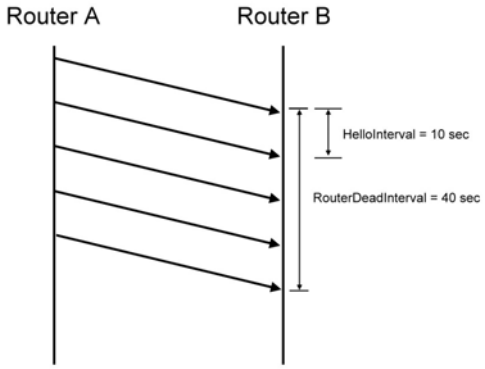


Figure 2.1: *HelloInterval* and *RouterDeadInterval*

According to standard OSPF mechanism, *HelloInterval* value is 10 seconds and RouterDeadInterval value is 40 seconds, the failure detection can take between 30 and 40 seconds. *RouterDeadInterval* is 4 times *HelloInterval*. So, we can reduce the *HelloInterval* to speed up the failure detection procedure. However, there is a limit up to which the *HelloInterval* can be safely reduced [10]. As the *HelloInterval* becomes smaller, there is an increased chance that the network congestion will occur. Because there are more HELLO packets in network traffic and these packets consume more network bandwidth, it leads to loss of several consecutive Hello messages easily and then cause false breakdown of adjacency between routers. However, the link or router does not really fail. Hence, the LSAs will be generated because of a false alarm and they will lead to new path calculations by all the routers in the network. However, the false alarm is soon corrected by successful HELLO message. Equally, all the routers will change their topology database again. Thus, false alarms cause unnecessary processing load and sometimes lead to temporary changes in the network traffic's path. It has a serious impact on QOS levels in the network. If the false alarms are too frequent, the routers will have to spend a lot of time doing unnecessary LSA processing and SPF calculations which may delay other important tasks and the router's loading is not efficient. If the false alarms are persistently in traffic, the

routers will have higher overloaded and frequent breakdown.

So we want to know how small the *HelloInterval* can be, to achieve faster detection and recovery from network failures while limiting the occurrence of false alarms.

According to the simulation results [5], we find out that the lower *HelloInterval* will lead to fast failure detection in the network while keeping the false alarm occurrence within acceptable limits. The simulation results indicate that the optimal value for *HelloInterval* for a network is strongly influenced by the congestion levels and the number of links in the topology. So it is a permanent tradeoff that must be found between delayed failure detection time and too many false alarms.

2.2 The LSA Flooding Time

When there is a change in link state database, routers use a flooding process to notify other routers in the network about the change. There are some configurable parameters which affect on the routing process in OSPF routers. Some of these parameters define a minimum time interval between two successive protocol events. The main purpose of these parameters is used to prevent a certain protocol task from overloading in the OSPF router. Frequent operations caused by smaller parameter configuration will affect the stability of OSPF network and increase the probability of failure event occurrence. Therefore, there is some minimum configurable setting about OSPF parameters. However, the interval simultaneously may slow down the OSPF routing convergence process. In this section, we discuss how the OSPF

flooding process may be adjusted in order to react to topology failures more quickly.

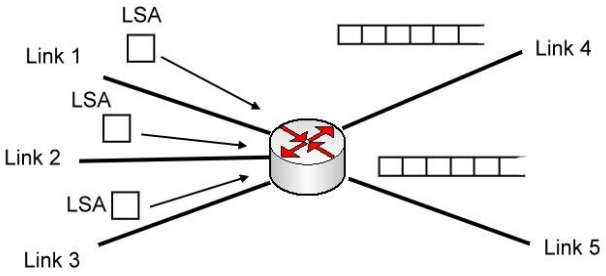


Figure 2.2: OSPF LSA flooding procedure

Figure 2.2 shows an OSPF router that has LSAs in its retransmission list. The router is attached to its neighbors with point-to-point physical links, Let us assume that the LSAs in the list have all been sent but not yet acknowledged by the receiving router. If the *PacingDelay* is adjusted to smaller interval, such as 0 second, both the arriving LSAs, from links 1, 2 and 3, lead to the retransmission of the whole list. A lot of duplicate traffic results in because of the smaller *PacingDelay*. If the *PacingDelay* would be appropriately adjusted the retransmission list would be emptied while the both arriving LSAs waited up to *PacingDelay*. When the pacing timer would fire soon after the lists are emptied the both arriving LSAs would be sent in the same LS Update packet.

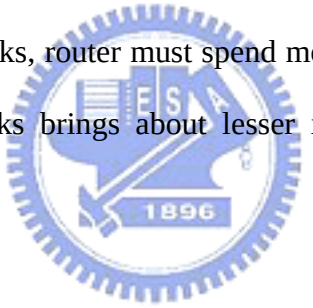
In order to speed up the flooding process, we can choose the smaller *PaccingDelay*, like 20ms, to replace the default setting (33 ms) by simulation experiment [6]. But the improvement range of flooding process is smaller than that of using shorter *HelloInterval*. The other factor, like link propagation delay, also affects the flooding time in OSPF network.

By simulation experiment, we see that the optimal value of *PacingDelay* depends on the number of LSAs simultaneously in transit. However, the number of LSAs in transit depends on network topology and failure model of network.

2.3 The Shortest Path First Calculation and Forwarding Table Update Time

In OSPF, Shortest Path First (SPF) calculation use Dijkstra's algorithm to get the shortest routing path. After this calculation, the router will update its forwarding table or routing table. This time depends on router CPU processing speed. Thus, we can reduce this time by improve router architecture design to accelerate the calculation time and forwarding table update time.

Aside from hardware architecture, this time also depends on the size of network. In large scale topology networks, router must spend more time to do SPF calculation. However, the size of networks brings about lesser influence on forwarding table update time.



According to OSPF standard, there are some parameters about SPF calculation: *spfDelay* and *spfHoldTime*. The time of *spfDelay* is the delay between the shortest path calculation and the first topology change that triggered the calculation. This delay is used to avoid frequent shortest path calculations. Usually 5 seconds. The time of *spfHoldTime* is the minimum delay between successive shortest path calculations. Usually 10 seconds. These two parameters are used to prevent router from frequent calculation, but they sometimes let network convergence time longer. We consider the case that there are two routers which fail during 5 seconds. Because the *spfDelay* is 5 seconds, the router can't re-calculate the SPF immediately. Then, the router must wait for the delay to update their routing database. It will increase the forwarding table update time. But this case is not universal in realistic network environment.

Chapter 3
Proposed Method

In this chapter, we propose a method to speed up failure detection and network convergence in OSPF networks. With this method, we divide it into two scenarios to discuss.

3.1 Adaptive *HelloInterval*

In standard OSPF protocol, the value of *HelloInterval* is set 10 seconds and *RouterDeadInterval* is four times *HelloInterval*. If the failure occurs, the routers will detect happened failure and remove the failure node or link from their database and neighbor list after *RuterDeadInterval* (40 sec.). At the same time, they will generate the newest link-state advertisement (LSA) packet and flood it through the network. The routers which receive the LSA packet will change their link state database (LSDB), re-calculate the shortest path information and update their routing table. According to newest routing table, the routers will forward the network packets correctly.

By these operations of OSPF, we can find that shorter *RouterDeadInterval* cause faster failure detection process if failure events occur. Similarly, smaller *HelloInterval* lets routers send Hello packets more frequently. If some failure events happened, the routers detect failure more quickly according to frequent Hello packets. Hence, we

can reduce the *HelloInterval* and *RouterDeadInterval* to speed up the failure detection and network convergence. The *HelloInterval* becomes smaller, however, the probability of network congestion is higher and the network congestion will lead to loss of several consecutive Hello messages. Thereby, it causes false breakdown of adjacency between routers even though the routers and the link between them are functioning well. Then, false breakdown will let routers generate LSA and re-calculate the shortest path. But the false breakdown will be soon corrected by successful Hello packet sent by neighbor routers. So probability of network variation is higher because of network congestion and the network availability is affected by these false breakdowns.

Therefore, the value of *HelloInterval* will bring some effects on OSPF network stability. At the same time, these false breakdowns also increase the unnecessary overhead in network routers and lead to temporary network changes. So we want to propose a method which adjust the value of *HelloInterval* dynamically according to network condition, accelerate the OSPF failure detection progress and reduce the network convergence time.

The main idea of adaptive *HelloInterval* is that we use the Hello packet specified in standard OSPF to bring the newest value of *HelloInterval* to the other routers. As we know, the smaller *HelloInterval* will cause a lot of Hello packet in network traffic and has an influence on network stability. By adaptive *HelloInterval* mechanism, the OSPF routers in the topology will exchange their *HelloInterval* value and adjust it by themselves without any setting configured by network manager. Hence, the network managers won’t adjust the appropriate *HelloInterval* by themselves on different network condition if they use adaptive *HelloInterval* mechanism in OSPF networks.

According to this method, we can use smaller *HelloInterval* to achieve faster failure detection and network convergence with any failure model.

3.1.1 Adaptive *HelloInterval* exchange method

In standard OSPF, the values of *HelloInterval* and *RouterDeadInterval* are attached in Hello packet header. By this way, the router can get the value of neighbor router's *HelloInterval* and *RouterDeadInterval* and we also don't need another packet format in OSPF protocol.

For instance, shown in figure 3.1, we assume the *HelloInterval* is 10 seconds and *RouterDeadInterval* is 40 seconds in RouterA. If RouterA receives the Hello packet from the neighbor RouterB and gets the *HelloInterval* is 5, *RouterDeadInterval* is 20. After checking the values of *HelloInterval* and *RouterDeadInterval* which are not the same as before, RouterA will adjust the *HelloInterval* and *RouteDeadInterval* setting by itself. By this way, all routers in this network topology can have the same view of *HelloInterval* and *RouterDeadInterval* during a period of time.

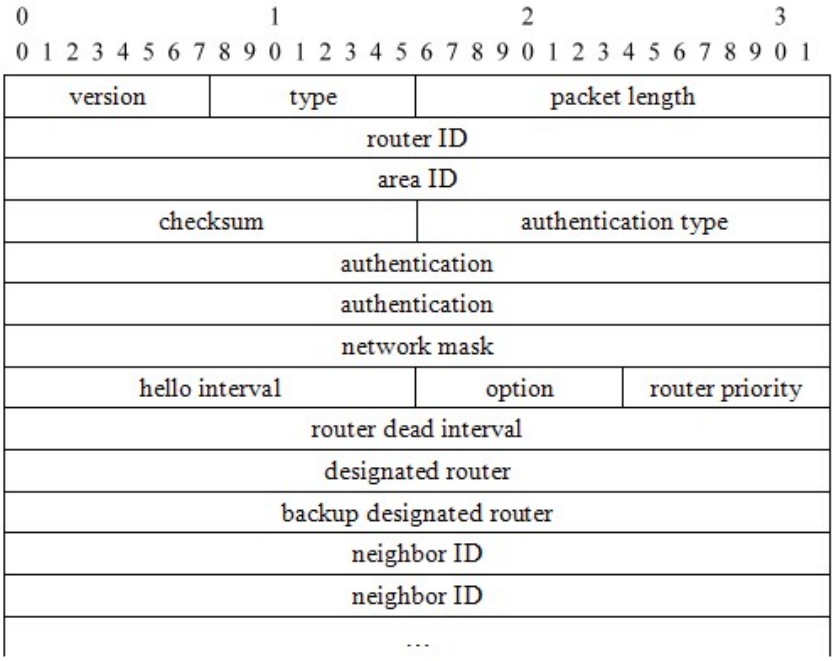


Figure 3.1 The OSPF Hello packet header

3.1.2 Adaptive *HelloInterval* change method

We can display the adaptive *HelloInterval* change method by the table as follow:

Table 3.1: Adaptive HelloInterval change method

HelloInterval range	0.1 sec. ~ 10 sec.
HelloInterval increase linearly	The network topology changes, failure event occurs or Hello packets lose.
HelloInterval decrease exponentially	The network topology doesn't change or no failure event occurs during stable time T_{stable}

T_{stable} presents a period of time that there are not any topology changes in the network. This value can be set appropriately by the network managers according to the different network environment.

In figure 3.2, it shows the flow chart of adaptive *HelloInterval*.

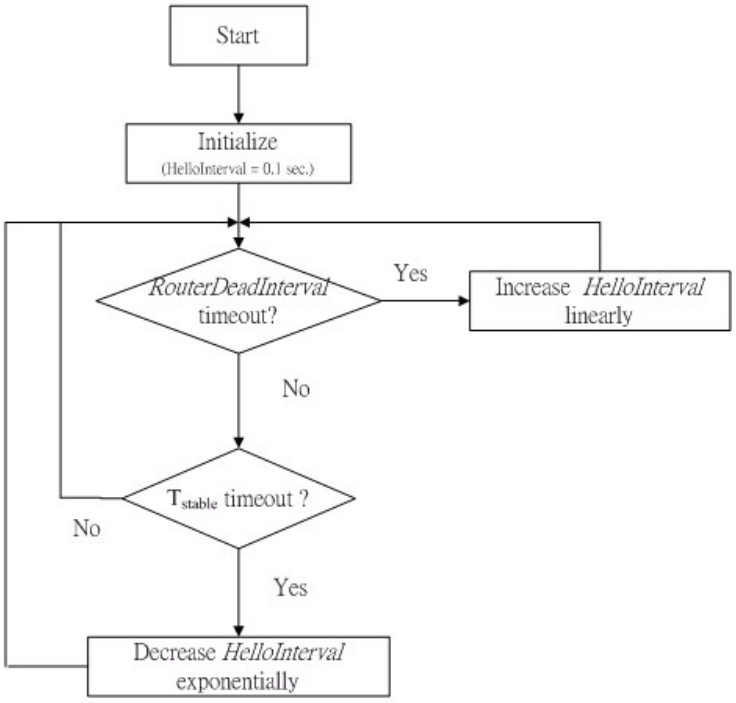


Figure 3.2: The flow chart of adaptive *HelloInterval*

Initially, we set that the default *HelloInterval* is 0.1 second and let the range of *HelloInterval* is between 0.1 second and 10 seconds. The router checks the *RouterDeadInterval* timer and T_{stable} timer. If *RouterDeadInterval* timer expired, routers will increase the *HelloInterval* linearly. If T_{stable} timer expired, routers will decrease the *HelloInterval* exponentially.

3.2 Passive Detection

In the passive detection scenario, we use the timeout mechanism in standard OSPF protocol to detect the occurrence of failure events passively. Each OSPF router use periodic Hello packet to check the neighbor router which still functions well and it decides whether delete the node from LSDB or not. Because we don’t receive any Hello packet from this node, the *RouteDeadInterval* timer timeout event will occur.

When *RouterDeadInterval* timer expired, the router will generate the LSA packet and flood it throughout the network. By this timeout mechanism, our *RouterDeadInterval* timer is binding on each neighbor router linking on the interface. We only check the Hello packets which are received from corresponding routers, and then we reset the *HelloInterval* and *RouterDeadInterval* timer if the router receives the correct Hello packet from each corresponding routers. However, the timeout event happened only that we lost four consecutive Hello packets. By this mechanism, all routers wait for *RouterDeadInterval* timer timeout event occurrence to decide when the failures happened. The failure detection time and network convergence time are highly effected by the interval length of *HelloInterval* and *RouterDeadInterval*. Therefore, we use shorter *HelloInterval* and *RouterDeadInterval* initially instead of the longer intervals to achieve faster failure detection procedure and network convergence. We use our adaptive *HelloInterval* mechanism to help us adjust appropriate interval value for any OSPF network domain.

For example, we consider the topology displayed in figure 3.3. RouterA doesn't receive four consecutive Hello packets from its neighbor RouterB because of the failure event on RouterB. When the *RouterDeadInterval* timer timeout event happened in RouterA, RouterA will modify its LSDB and generate LSA to flood throughout the network domain. At the same time, RouterA also increases its *HelloInterval* and *RouterDeadInterva* value and modify the Hello packet header. Then, RouterA send the modified Hello packets to its neighbor routers. According to this method, all OSPF routers will get the newest *HelloInterval* and *RouterDeadInterval* during a short period of time.

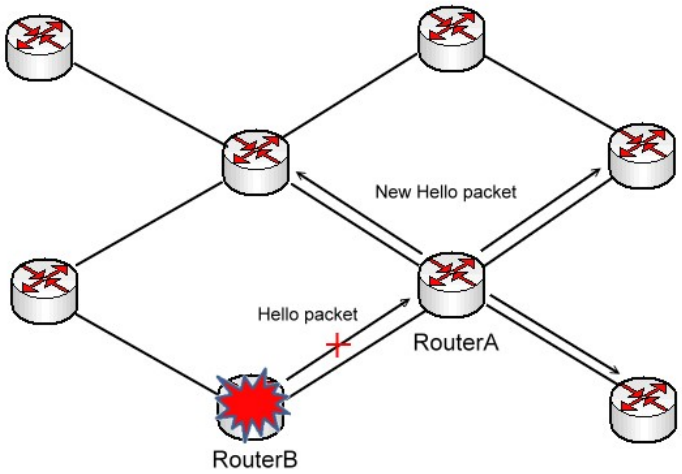


Figure 3.3: The passive detection method

3.3 Active Detection

In the active detection scenario, we design a specified program [12] to monitor the operations of OSPF routing demon in each OSPF routers. This specified program can detect the activity of OSPF routing demon by sending periodic checking messages, which the checking period can be set in sub-second range. If we want to achieve faster failure detection response, we can set smaller checking interval, such as 100ms, to get rapid and immediate response if some failure on OSPF routing demon. The specified program detects the health of OSPF routing demon; moreover, it must support that sends an announcement packet which can be read by OSPF routing demon actively to other OSPF routers. However, this announcement packet must use OSPF packet header in order to be accepted by OSPF routing demon.

When the OSPF routing demon takes something wrong or can't operate correctly, the specified monitor program will broadcast the announcement packet immediately to all neighbor nodes by all interfaces in this router. After sending this announcement

packet, the neighbor OSPF routers which receive this packet will get the failure router's IP address, and then remove this node information from LSDB and neighbor list. After removing, we re-calculate the shortest path and update the routing table by modified LSDB. Finally, we generate the new LSA which described the changed network topology and flood it through the network domain. Due to the fact that the failure happens on OSPF routing demon, which is a software problem, the hardware and network connectivity still work correctly without any failure effect. Therefore, we just let OSPF routing demon be able to recognize this announcement packet and take some operations after receiving this packet.

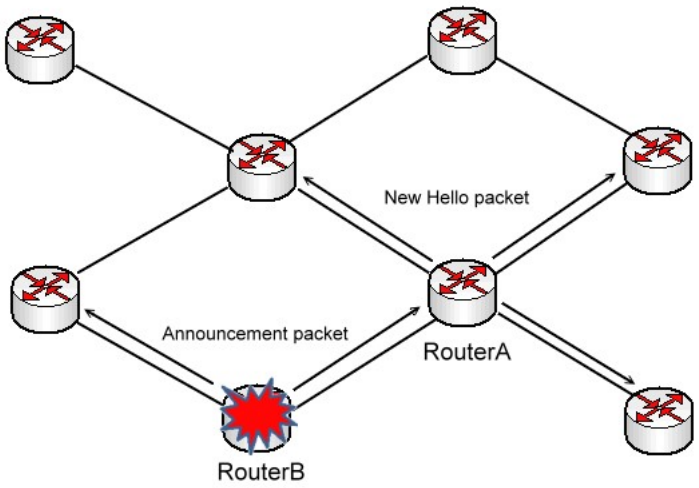


Figure 3.4: The active detection method

Figure 3.4, which shows that RouterA will send announcement packet actively after a failure event happened on RouterA's OSPF routing demon. The RouterB will generate new LSA and update routing table. At the same time, it also adjusts *HelloInterval* and *RouterDeadInterval* by itself and re-sends the Hello packets with new *HelloInterval* and *RouterDeadInterval*.

By this mechanism, OSPF router can detect failures rapidly and decrease the network convergence time without any passive waiting time. However, active

detection method must assume router's hardware and network connectivity still work correctly if some failure event happened. Active detection method is suitable for OSPF protocol failures, but it can be applied to hardware failures.

Chapter 4
Simulation Results

In this chapter, we experiment on the network convergence time of proposed passive detection method and active detection method in NCTU-ns 3.0 [13]. We simulate in 30 OSPF routers network and use 20 failure events with exponential distribution during 4000 seconds. The T_{stable} is set in 100 seconds. We also discuss the convergence time in different network size and router degree. Finally, we compare the network convergence time of proposed method with that of standard OSPF routing protocol. The network link delay is 350 ms and bandwidth is 100 Mbps.

4.1 Passive Detection

We discuss the passive detection method first. During 4000 seconds simulation, we assume that there are total 20 exponential distribution failure events which occur in the OSPF network with 30 routers randomly. We observe the convergence time about these 20 failure events.

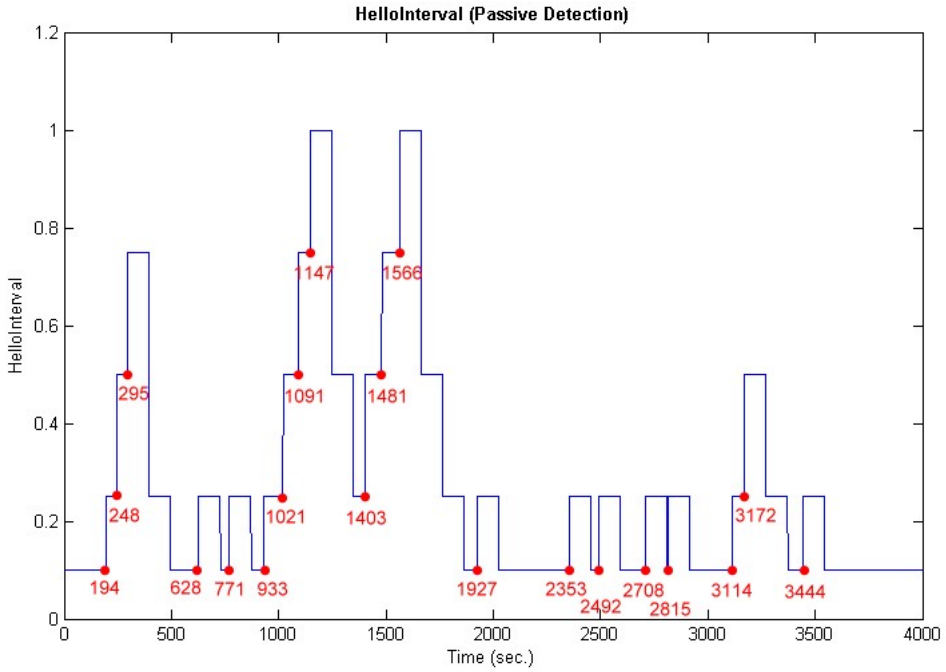


Figure 4.1: The variation of *HelloInterval* with passive detection

In figure 4.1, which shows the variation graph of *HelloInterval*, we see that the variation of *HelloInterval* ranges from 0.25 seconds to 10 seconds dynamically. In passive detection method, we also observe that the router must wait for some time to detect the failure event so there are some shorter time gaps between actual failure occurrence time and failure detection time.

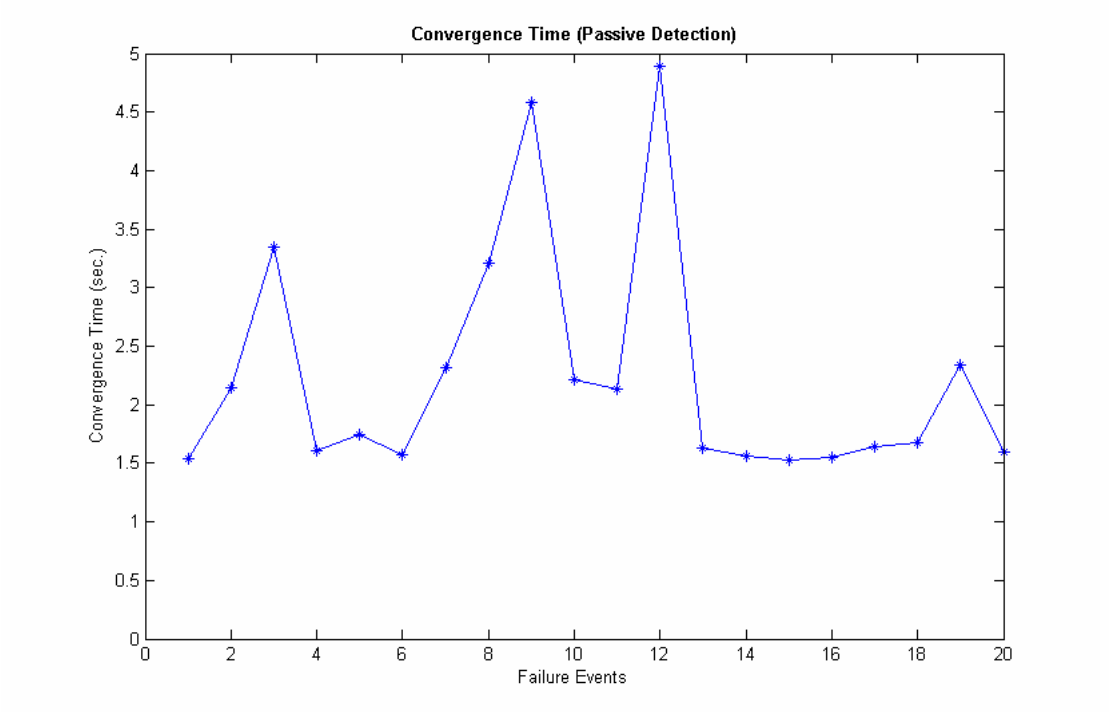


Figure 4.2: The network convergence time with passive detection

In figure 4.2, we observe that the convergence time that ranges from X seconds to X seconds is different in 20 failure events. The reason is that network convergence time of 20 failure events are affected by different value of *HelloInterval*. Because the *RouterDeadInterval* four times the *HelloInterval* and the routers must passively wait for *RouterDeadInterval* timer timeout event triggering, the network convergence time is fluctuant.

4.2 Active Detection

In active detection method, the simulation environment we use is as same as the passive detection method. Similarly, we discuss the variation of *HelloInterval* and network convergence time as follow.

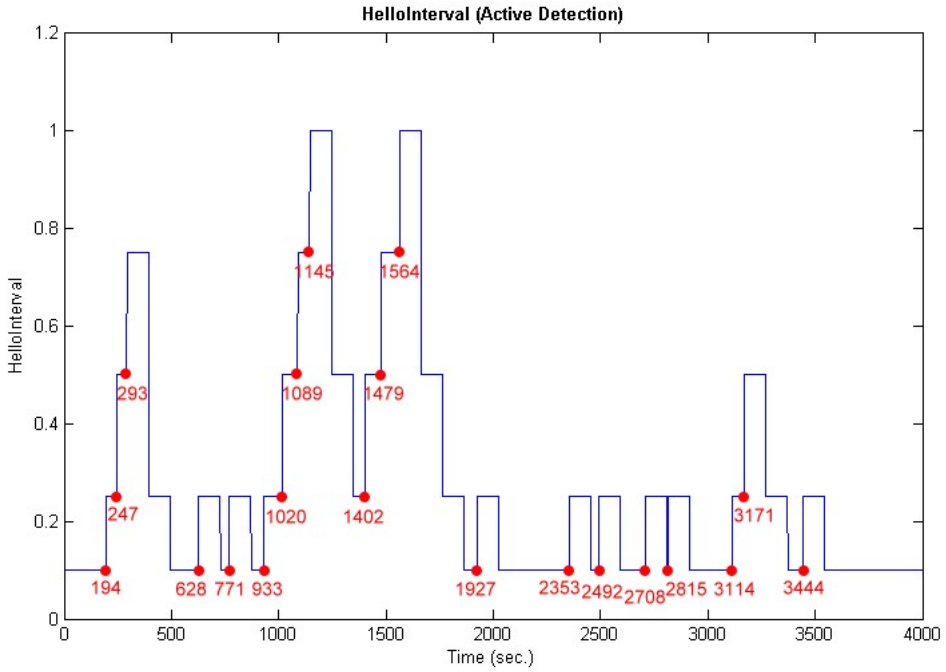


Figure 4.3: The variation of *HelloInterval* with active detection

Due to the same simulation environment as passive detection method, the variation of *HelloInterval* is also similar. However, which the failure detection time is earlier than passive detection method is the different point. The active detection method uses another specified program running on the OSPF router to detect the health of OSPF routing demon periodically. Therefore, it can get immediate response if some failure events are detected. Thus, the failure detection time is always earlier than passive detection method. By sending announcement to neighbor nodes actively, the network convergence time is shorter than passive detection method or standard OSPF protocol, in Figure 4.4. We also observe that the network convergence time is independent of the value of *HelloInterval*. Because the routers don't need to wait for *RouterDeadInterval* timer timeout, the network convergence time can be reduced a lot by active detection method. And we see that the network convergence time is almost the same as LSA flooding time and routing table update time.

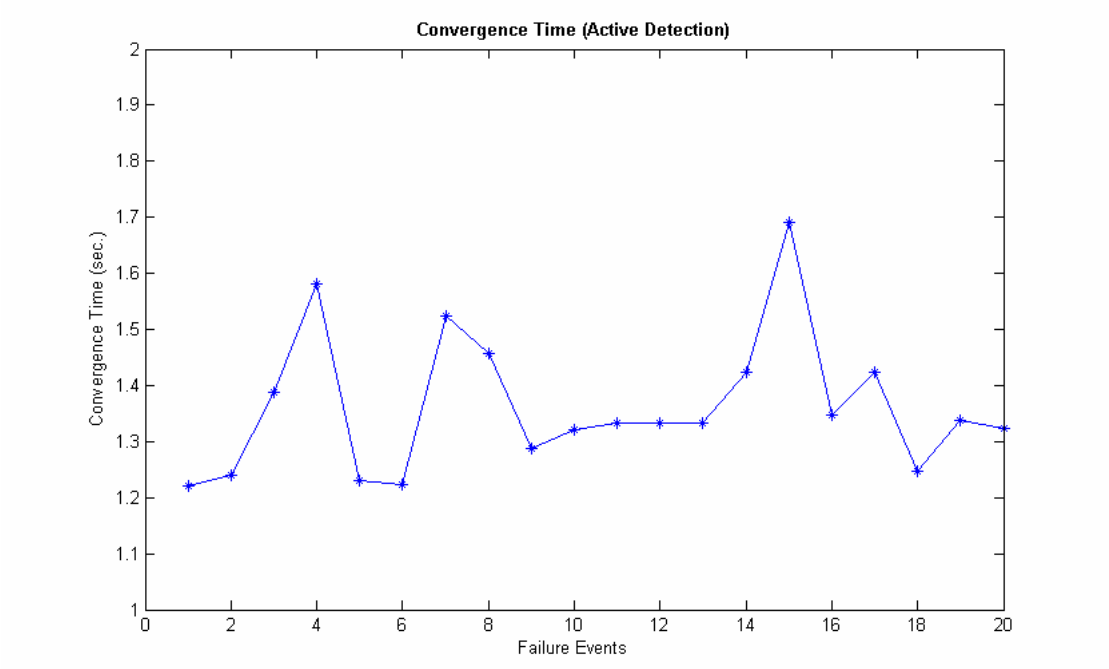


Figure 4.4: The network convergence time with active detection

In the passive and active detection scenarios, the time of the newest *HelloInterval* which adjusted by routers also affects the network convergence time in OSPF routing. So we show the total delay of propagating adjusted *HelloInterval* value in table 4.1.

Table 4.1: *HelloInterval* flooding delay

Scenario	Propagating delay
Passive	0.261 ± 0.017
Active	0.265 ± 0.016

Finally, we shows the convergence time with proposed method and standard OSPF protocol in table 4.1.

Table 4.2: Network Convergence Time

	HelloInterval	Convergence Time
Standard OSPF	10	42.77 ± 0.35
	0.25	2.24 ± 0.13
Passive Detection	0.25025	2.17 ± 0.37
Active Detection	0.2506	1.35 ± 0.11

`

4.3 Convergence Time with Network Size and Router Degree

In this section, we compare our proposed method with standard OSPF routing protocol in different network size and router degree.

First, we consider the network size issue. The figure 4.5 shows that the network convergence time highly depends on the number of router in network domain. In this simulation, the *HelloInterval* of standard OSPF is set 0.25 second. We see that the network convergence time grows quickly with more routers and it nearly grows exponentially. The reason is that the network propagation delay and LSA flooding procedure time is longer in larger network size.

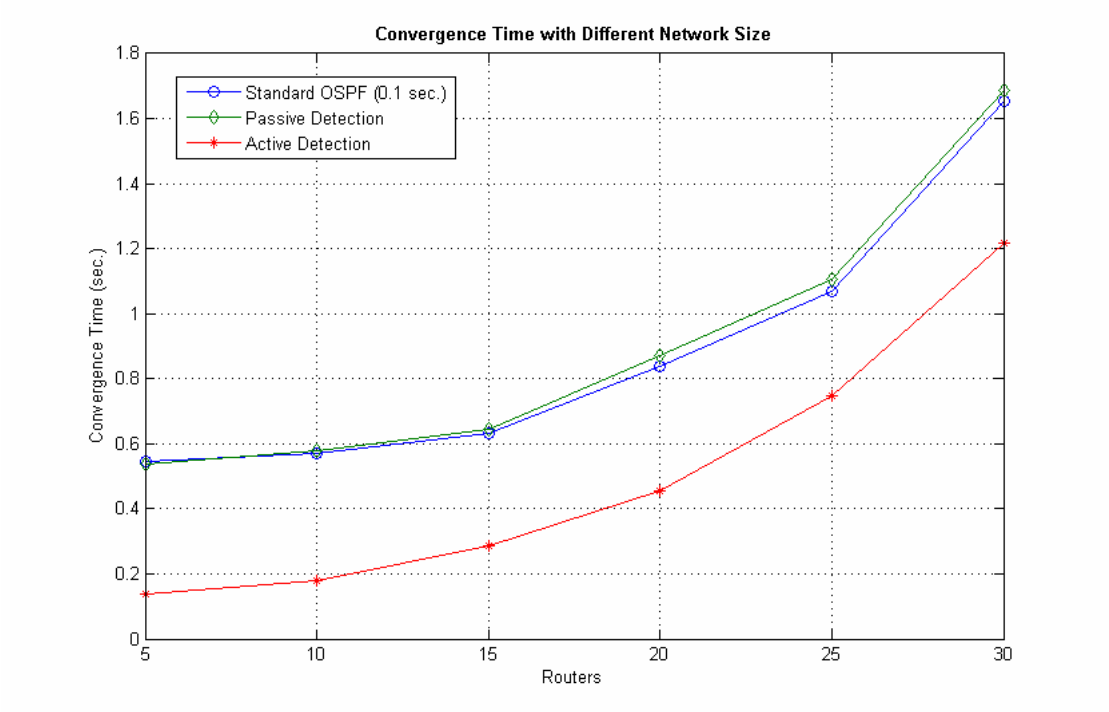


Figure 4.5: The convergence time with different network size

Second, we consider the router degree with different router degree. In figure 4.6, which we see that the network convergence time grows very quickly if the router degree increases. The *HelloInterval* of standard OSPF protocol is still set 0.25 second. We observe that the network convergence time grows nearly exponentially, too. Because of LSA flooding procedure, we know that the total links in the network domain affect the LSA flooding processing time. In larger router degree networks, routers may receive a lot of duplicated LSA packets, and these duplicated packets will slow down the processing speed of the network routers. They must check for these packets and their LSDB, so LSA flooding time will be longer.

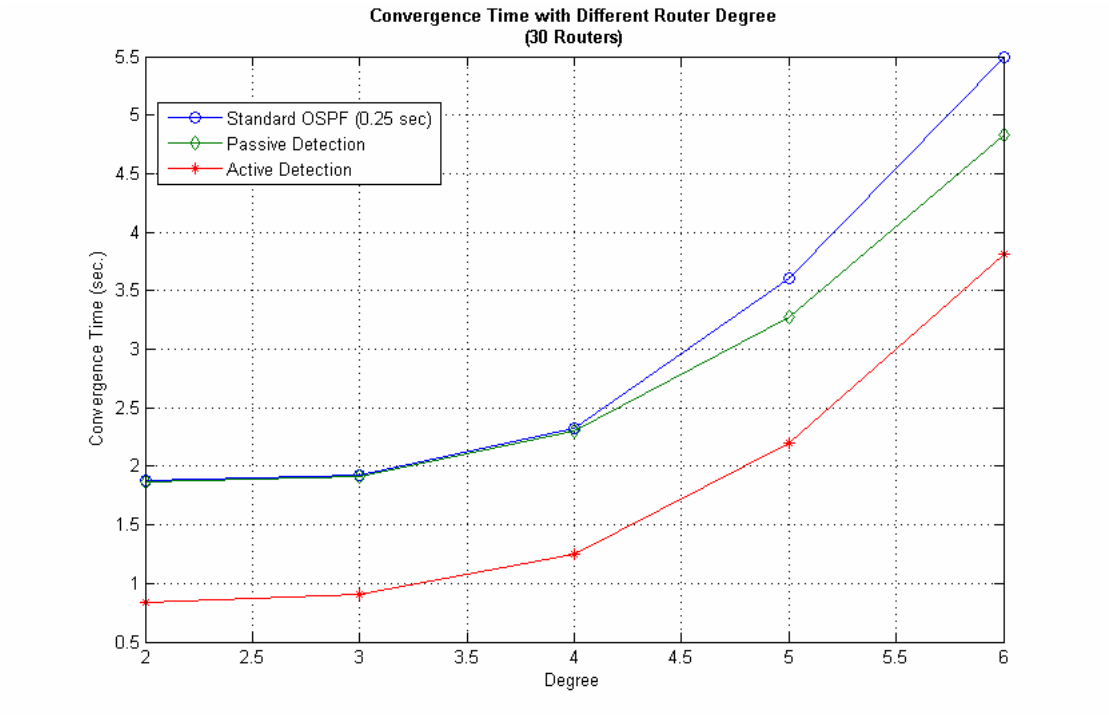


Figure 4.6: The convergence time with different router degree

Chapter 5
Conclusion

In this chapter, we propose a modified OSPF routing protocol to speed up the failure detection time and network convergence time. We use adaptive *HelloInterval* mechanism to replace the standard OSPF 10 seconds *HelloInterval*. According to adaptive *HelloInterval* mechanism, the OSPF routers can adjust the appropriate *HelloInterval* value by themselves without any configuration set by network manager. The simulation result shows that our proposed method can get smaller failure detection time and network convergence time than standard OSPF protocol.

Bibliography

[1] G. Malkin, “RIP version 2”, *IETF Request for Comments 2453*, November 1998

[2] J. Moy, “OSPF version 2”, *IETF Request for Comments 2328*, April 1998

[3] G. Lichtwald, U. Walter, and M. Zitterbart, “Improving Convergence Time of Routing Protocols”, In *Proceedings of the 3rd International Conference on Networks (IEEE ICN 2004*), Guadeloupe, French Caribbean, February 29. - March 04, 2004

[4] A. Basu, and J. G. Riecke, “Stability Issues in OSPF Routing”, In *Proceedings of the ACM Special Interest Group on Data Communications (ACM SIGCOMM 2001)*, Mandeville auditorium, UC San Diego, CA, USA, pp. 225-236, August 27-31, 2001

[5] M. Goyal, K. K. Ramakrishnan, and W. C. Feng, “Achieving Faster Failure Detection in OSPF Networks”, In *Proceedings of the IEEE International Conference on Communications (ICC 2003)*, Alaska, vol.1, pp. 296-300, May 11-15, 2003

[6] M. Rtkanen, and M. Luoma, “OSPF Flooding Process Optimization”, In *Proceedings of the High Performance Switching and Routing (HPSR 2005)*, Hong Kong, R. P. China, pp. 448- 52, May 12-14, 2005

[7] R. Mahajan, D. Wetherall, and T. Anderson, Eds, “Understanding BGP Misconfiguration”, In *Proceedings of the ACM Special Interest Group on Data Communications (ACM SIGCOMM 2002)*, Pittsburgh, PA,USA, pp. 3-16, August 19-23, 2002

[8] D. Oran, “OSI IS-IS intra-domain routing protocol”, *IETF Request for Comments 1142*, February 1990

[9] Y. Rekhter and T. Li, “A Border Gateway Protocol 4 (BGP-4)”, *IETF Request for Comments 1771*, March 1995

[10] A. Shaikh, L. Kalampoukas, R. Dube, and A. Varma, “Routing Stability in Congested Networks: Experimentation and Analysis”, In *Proceedings of the ACM Special Interest Group on Data Communications (ACM SIGCOMM 2000)*, Stockholm, Sweden, pp. 163-174, August 28 – September 1, 2000

[11] S. Vutukury, J.J. Garcia-Luna-Aceves , “A Traffic Engineering Approach based on Minimum-Delay Routing”, In *Proceedings of IEEE International Conference on Computer Communications and Networks (ICCN 2000)*, Las Vegas, Nevada, USA, pp 42-47, November 16-18, 2000

[12] E. Baccelli, and R. Rajan, “Monitoring OSPF Routing”, In *Proceedings of the IEEE/IFIP International Symposium on Integrated Network Management (IM 20001)*, Seattle, USA, no.1, pp. 825-838, May 14-18, 2001

[13] NCTUns 3.0. Network Simulator and Emulator. [Online]. Available: http://nsl10.csie.nctu.edu.tw/