Network and Complex Systems www.iiste.org
ISSN 2224-610X (Paper) ISSN 2225-0603 (Online)
Vol.3, No.3, 2013-National Conference on Emerging Trends in Electrical, Instrumentation & Communication Engineering

Effects of OSPF Timers Configurations on Network Convergence in New Generation Routers

Himanshu Singh
Department of Computer Engineering
Institute of Technology, Banaras Hindu University, Varanasi-221005, India
Tel: +91-9450356928 E-mail: himanshu.singh.cse07@itbhu.ac.in

Abstract
QoS provided by a routing protocol is determined by several factors, one of which is network convergence time. It is the time taken by network to recover from a link failure. Hello protocol is used by Open Shortest Path First routing algorithm to detect such a failure. With default settings of the OSPF parameters, the network takes a long time in recovery process. The primary reason of this is the time required in failure detection using Hello protocol. Detection time can be reduced by reducing the value of HelloInterval. However, a small value of HelloInterval increases the network congestion causing loss of some consecutive Hellos, thus leading to false failure detection. Traditional routers could not afford such strict configurations of OSPF parameters. In this paper, we investigate the effects of such configurations of parameters in new generation routers which provide higher bandwidth and higher tolerance for congestion. We conducted experiments with Cisco ASA and Cisco IOS devices and this paper presents the results with different configurations and their effects on fast failure detection, false alarm, network congestion and failure recovery.
Keywords: OSPF, Hello protocol, Fast Hello, Network Convergence, Fast Failure Detection, Computer Networks

1. Introduction

Open Shortest Path First (OSPF) [1] is the most commonly used routing protocol in wired and wireless networks. In OSPF, each link in the topology has a cost associated with it. Each router has the knowledge of entire topology and routing is done between two nodes such that total cost in the path is least. In case of failure of a link due to software errors, hardware malfunction or power cut, the routers re-establish entire topology and re-compute the next best available path. Such a re-establishment takes time and during this process, the data being transmitted is dropped. Several service providers have to guarantee high QOS and cannot afford to drop user packets greater than a threshold. So, this motivates to reduce re-establishment time and thus reducing the number of packets dropped.

Network convergence time can be reduced by adjusting OSPF timers – hello interval, dead interval, spf hold time and spf delay. Although adjusting these timers would result in faster failure detection and recovery, it would increase the number of control packets in transit thus resulting higher congestion and increased chances of false failure detection and recovery. Older routers had limited bandwidth and processing power. We believe that with advent new generation routers like Cisco Adaptive Security Appliances [2] and Cisco IOS [3] routers, such strict adjustments can be done in some, if not all, network deployments. The new routers support 10 Gigabit Ethernet and have better processors. In this paper, we present a study of the trade-offs between failure detection time, recovery time and false failure detection and recovery with different configurations of hello interval, dead interval, spf delay and spf hold time. Section 2 of the paper presents the failure detection process in OSPF explaining the hello protocol and different timers used. Section 3 describes previous research done and section 4 highlights the significance of timers and other improvements in fast failure detection. In section 5, we present our experimental setup and results obtained with Cisco ASA and IOS routers. Section 6 concludes the paper with scope of future research in this field.

2. Process of Failure Recovery in OSPF

In OSPF, each router advertises the state of its link including link cost through Link State Advertisements (LSA). Such LSAs are flooded throughout the network and thus each router has the knowledge of entire topology. Based on link costs, router performs Shortest Path First Calculation using Dijkstra's algorithm [4] to find a path between itself and every other router. The next hops in the paths are then saved in routing table.

2.1 Hello Protocol

Adjacent routers in same area send Hello packets to maintain the link adjacency periodically (period being called 'HelloInterval'). If any router does not receive a Hello message from its adjacent router within a period called 'DeadInterval', it considers the neighbor router as 'dead' and assumes the link between the neighbor and itself to be down. Lesser the HelloInterval and DeadInterval, faster will be the detection of failure by router. When HelloInterval is set in milliseconds range, it is called 'fast hellos' [5]. The router has to notify other routers as

well about this failure and thus, generates a new LSA to reflect the change in topology. All such LSAs, generated due to failure, are flooded again throughout the network and cause the routers to redo the SPF calculation and update its forwarding table. Two consecutive LSAs sent down an interface are limited by pacing delay to avoid large congestion in unstable networks. Apart from this, LSA flooding times includes propagation delay. Then on receiving a new LSA, the router takes time to process the LSA and then it schedules an SPF calculation. Since Dijkstra's algorithm requires significant processing, the router cannot afford repeated SPF calculation in case it receives further LSAs informing the new changes in topology. So the router delays SPF calculation for some time (spfDelay) to let other LSAs, if any, arrive. Moreover, routers impose a limit on the frequency of such calculations meaning that it will not perform SPF within a particular time (spfHoldTime, typically 10 seconds) after its last SPF. The execution of algorithm also takes time since Dijkstra's is run several times to compute the route with every other router. Finally, updating the forwarding table with new routes also introduces minor delays, however such delays depend on processor and thus cannot be configured.

Hence, the failure recovery time consists of (i) Failure detection time (governed by hello interval and dead interval), (ii) LSA generation time (initial delay and hold value), (iii) Pacing delay – the minimum delay between two successive LSAs, (iv) LSA propagation time (governed by traffic congestion), (v) LSA processing time, (vi) SPF delay (vii) SPF Hold time – time between successive SPF calculations, (viii) SPF Calculation time and (ix) Route update time.

LSA generation time, Pacing delay, LSA propagation time, LSA processing time, SPF calculation time and route update time are usually dependent on router processing power and network conditions and hence are difficult to control. On the other hand, hello interval, dead interval, spf delay and spf hold times can be configured by network administrators through timers. In this paper, we analyze the effects of these timers on network convergence.

3. Related Works

Researchers have considered various configurations and solutions to improve convergence time earlier. Alaettinoglu et al. [6] suggested reducing the HelloInterval to millisecond range but they did not consider any adverse effects of HelloInterval reduction. Basu and Riecke [7] have also considered using sub-second HelloIntervals to achieve faster failure recovery from network failures. They also considered the tradeoff between speedy failure detection and high probability of false alarms. This research reports 275ms to be the optimal value for HelloInterval and hence matches our result. However, unlike our results, they present the simulated results without any consideration of the new routers and their processing power. [8][9] proposed to give prioritized treatment to Hello messages at router interface and processing queue since the loss or delayed processing of these messages can result in false failure detection. Since SPF calculation puts significant processing load on routers, there are delays (spfDelay and spfHoldTime) that impose a limit on such operations which ultimately result in delay in failure recovery. Alaettinoglu et al. [10] propose removing any restrictions on SPF calculations. It is argued that the frequency of SPF calculations can be reduced by careful filtering of LSAs and the processing time and load of an SPF calculation can be reduced by using new routing algorithms (such as [11][12][13]) instead of Dijkstra's algorithm. These modern algorithms are based on incremental and selective SPF calculations where algorithm computes new routes to those routers which are affected by the link failure.

4. Significance of timers in OSPF

Hello protocol is used by a router to detect the loss of adjacency with adjacent neighbor as described in section 2. A router declares its neighbor to be down if it does not receive a Hello from its neighbor for more than DeadInterval. The loss of Hello messages may be due to high congestion and this leads to false detection of failure. To avoid such a false breakdown of adjacency, the DeadInterval is configured to be four times the HelloInterval. The link failure detection time through hello protocol can be decreased by reducing the HelloInterval and DeadInterval. If HelloInterval is less than 1 second, DeadInterval cannot be reduced less than 1 second. Such a configuration is termed as 'fast hello' support. However, there is a threshold up to which this HelloInterval can be safely reduced. On decreasing the HelloInterval, number of hello messages in the network increase. Thus, there are more chances that network congestion will occur and will lead to loss of several consecutive Hello packets. The neighbor will not receive the Hellos and this will cause false detection of link failure even though the routers and the link between them are in perfect condition. The LSAs generated due to false detection will result into new SPF calculations, avoiding the supposedly down link, by all routers. Since the link is perfectly working; a successful exchange of Hello will soon take place thus informing the routers again about a topology change. New LSAs will be flooded again and fresh SPF calculation will take place. Thus, false failure detection will put unnecessary processing load on routers and affect the QOS levels in the network. If such false alarms are frequent, routers will have to spend lot of resources and time in unnecessary LSA processing and SPF calculations. This may significantly delay other important tasks such as Hello processing,

thus resulting in more false failure detection. Such a scenario may result in complete break in routing operation of the network.

Unnecessary SPF calculations can be avoided by setting SPF delay and SPF Hold time to a higher value. If these timers are set to high values, router will wait for a larger time before starting the SPF calculation. During this wait time, chances are high that false detection would be corrected by successful reception of Hello message, thus avoiding unnecessary processing by the router. However, higher SPF delay would mean increased recovery time in case of actual failure recovery.

All the above observations are proved by our experimental results presented in next section.

5. Experiments and Results

5.1 Experimental setup

We created a topology with Cisco ASA routers and Cisco IOS routers as shown in figure 1.

The topology consists of three Cisco ASA devices and two IOS routers. Different cost is assigned to each link as shown in figure 1. The traffic is then sent from router 1 to router 2. The packets take the path with lesser cost. For example, in figure 1, if traffic is sent from router 1 to router 2, it goes through ASA 5580 and not through ASA 5520. Then we shutdown one of the links in this path for example - link between ASA 5510 and ASA 5580. The traffic flow is interrupted and routers try to establish the next better path for transmission. We observe the time taken in re-establishment and number of packets dropped in this duration. The experiments were repeated by shutting down the different links between different routers and results are presented after taking the average.

The HelloInterval, DeadInterval, SPF delay and SPF Hold Time were varied as guided in [14]. Table 1 describes different settings (A to H) used for evaluation purposes.

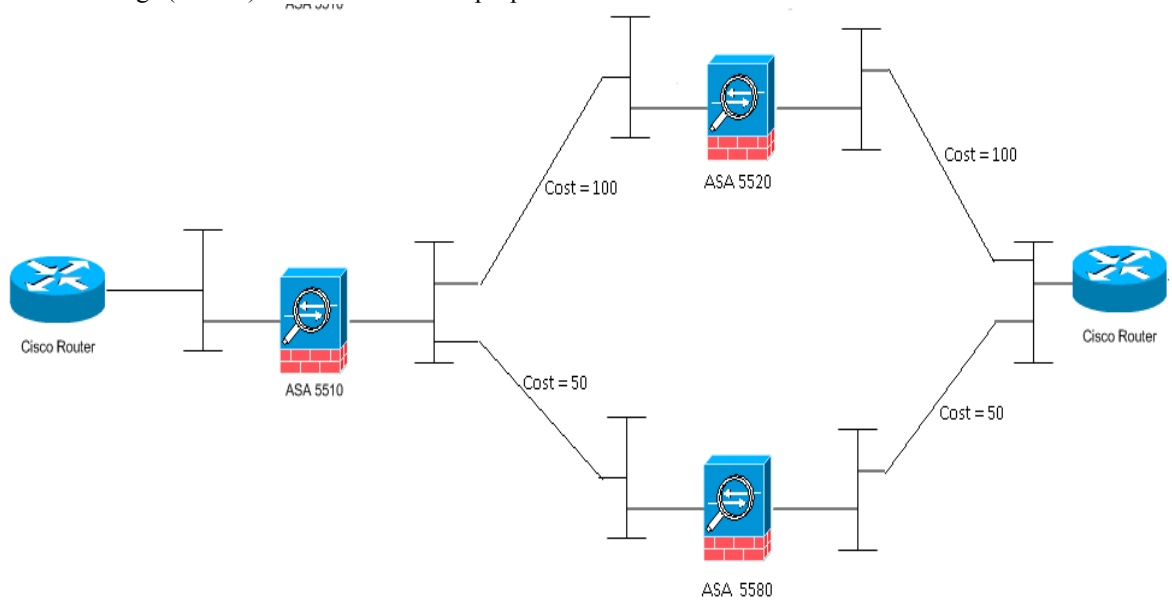

Figure 1. Topology used for testing the effects of Fast Hello on OSPF

Table 1: Different Settings of timers used in experiments.

Configuration	Hello Interval (seconds)	Dead Interval (seconds)	SPF delay (seconds)	SPF Hold Time (seconds)
A	10	40	5	10
B	1	4	5	10
C	0.25	1	5	10
D	0.2	1	5	10
E	0.1	1	5	10
F	0.05	1	5	10
G	1	4	1	2
H	0.25	1	1	2

Configuration 'A' uses default setting used in old routers. B has the minimum HelloInterval and DeadInterval (DeadInterval has to be four times HelloInterval) that can be set without using Fast Hello support. C introduces fast hello support with default SPF delay and SPF Hold time. D, E and F further reduce the HelloInterval to observe the effects on congestion. G is the setting which does not use fast hellos but reduces SPF delay and SPF Hold time to minimum. H uses the fast hellos as well as reduces the SPF delay and SPF Hold times.
Results showing the failure recovery time and number of packets in the network (a measure of congestion) are presented in next sub-section.

5.2 RESULTS and DISCUSSIONS

Table 2 shows the total time recovery time and number of packets with different settings. Figure 2 shows the same results in the form of a histogram.

Table 2: Failure recovery time and number of packets in network with different configurations

Configuration	Failure Recovery Time (seconds)	Number of Packets in network
A	53	5
B	11	10
C	8	29
D	8	30
E	9	55
F	9	111
G	5	10
H	2	25

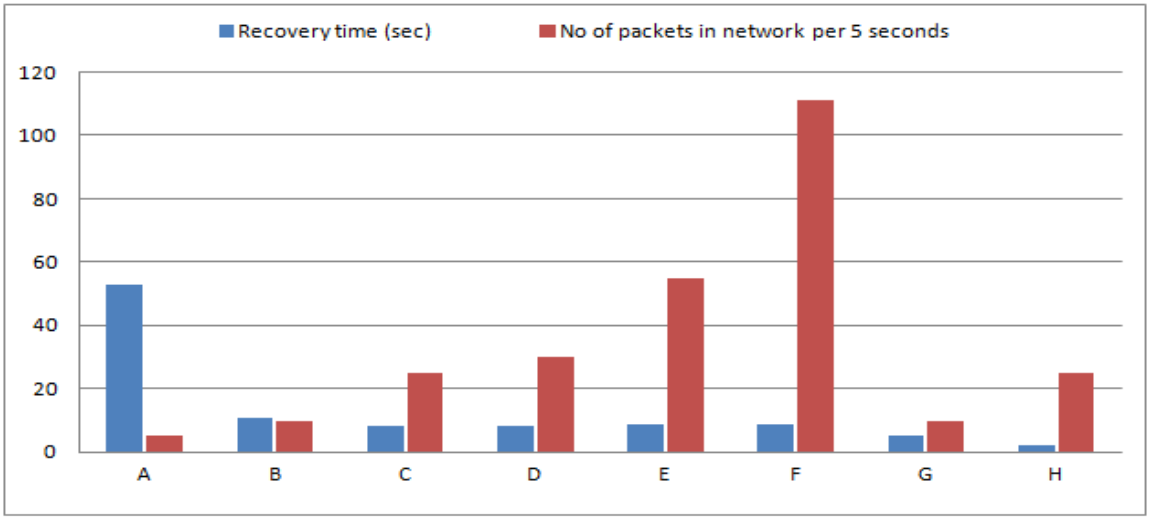

Figure 2: Histogram of recovery time and number of packets with different configurations

Discussions:

i) Default Settings (configuration A) take a long time in recovering from failure (53 seconds) due to large HelloInterval and DeadInterval.

ii) With best configuration without fast hellos, minimum recovery time is 11 seconds. (Configuration B)

iii) Fast hellos decrease the recovery time to 8 seconds, an improvement of 3 seconds. (Configuration C)

iv) Recovery time does not improve on decreasing the HelloInterval and keeping DeadInterval fixed as seen in configuration D, E and F. This is because failure detection is done only after no Hello message is received for the DeadInterval. So detection time cannot be reduced beyond a limit.

v) Decreasing Hello to 0.05 seconds in F, increases recovery time to 9 seconds. Number of packets increase highly leading to congested network. In high congestion, chances of dropping of Hello messages are high. This increases the chances of false failure detection. It should be noted however, that false SPF calculations are less likely because SPF delay and SPF Hold time is high.

vi).With best configuration without fast hellos and with decreased SPF timers, recovery time is decreased to 5 seconds. In case of false failure detection, SPF calculations will be high due to reduced SPF delay and SPF Hold time.

vii) With Fast hellos and decreased SPF timers, recovery time becomes 2 seconds with acceptable congestion in network.

6. Conclusion

In this paper, we explained the network convergence process and highlighted the importance of configuration timers. We presented the trade-offs in recovery time with different configurations of HelloInterval, DeadInterval, SPF delay and SPF Hold time. It is shown that with current settings of OSPF timers, the network takes a large time to converge. Convergence time can be decreased by decreasing the HelloInterval, DeadInterval, SPF delay and SPF Hold time. However, there are overheads of reducing these timers and thus they cannot be reduced limitlessly. In this paper, we try to find the optimal value of these timers so that false detections do not occur. However, such an optimal value would be dependent on the network congestion and number of interfaces on a router (denser topology would result in higher number of Hello messages in the network). Different routers have different bandwidth and processing power. Hence, network administrators should take into account quality of routers, the expected user traffic and the density of topology while configuring these timers. So, the next logical step in this work is to analyze the effects of topology (links per node) and variable network traffic on network convergence.

References

Inetdaemon tutorial, Open Shortest Path First. [Online] Available: http://www.cisco.com/en/US/docs/security/asa/asa82/configuration/guide/route_ospf.html Accessed on 15th February, 2012.

Cisco Systems, Cisco ASA Adaptive Security Appliances. [Online] Available: http://www.cisco.com/en/US/prod/collateral/vpndevc/ps6032/ps6094/ps6120/product_data_sheet0900aecd802930c5.html Accessed on 15th February, 2012.

Wikipedia, Cisco IOS. [Online] Available: http://www.cisco.com/en/US/prod/collateral/vpndevc/ps6032/ps6094/ps6120/product_data_sheet0900aecd802930c5.html Accessed on 15th February, 2012.

E. Dijkstra. (1959). "A note on two problems in connection with graphs," *Numerische mathematik,* 1:269-271.

Cisco Systems, OSPF Support for Fast Hellos. [Online] Available: http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/fasthelo.html Accessed on 15th February, 2012.

C. Alaettinoglu, V. Jacobson, and H. Yu, (2000). "Toward millisecond IGP convergence," *NANOG 20.*

A. Basu, and J. Riecke, (2001). "Stability issues in OSPF routing," *Proc. ACM ,SIGCOMM.*

G. Choudhury, V. Sapozhnikova, A. Maunder, V. Manral, (2002). "Explicit marking and proritized treatment of specific IGP packets for faster IGP convergence and improved network scalability and stability," *IETF Internet Draft draft-ietf-ospf-scalability-01.*

J. Ash, G. Choudhury, V. Sapozhnikova, M. Sherif, V. Manral, and A. Maunder, "Congestion avoidance and control for OSPF networks," *IETF Internet Draft draft-ash-manral-ospf-congestion-control-00.txt*,

C. Alaettinoglu, and S. Casner, "Detailed analysis of IS-IS routing protocol on the Qwest backbone," *NANOG 24,* February 2002.

P. Fraciosa, D. Frigioni, and M. Giaccio, "Semi-dynamic shortest paths and breadth-first search in digraphs," *Proc. 14th Symp. Theoratical Aspects of Computer Science,* 113-124, 1997.

D. Frigioni, M. Ioffreda, U. Nanni, and G. Pasqualone, "Experimental analysis of dynamic algorithms for the single-source shortest path problem," *ACM Journal of Experimental Algorithmics,* 3:5, 1998.

G. Ramalingam, and T. Reps, "An incremental algorithm for a generalization of the shortest-path problem," *Journal of Algorithms,* 21(2):267-305, 1996.

Cisco Systems, Configuring OSPF. [Online] Available: http://www.cisco.com/en/US/docs/security/asa/asa82/configuration/guide/route_ospf.html (Accessed: 15th February, 2012)