PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 47, NO. 1-2, PP. 89-100 (2003)

THE CONVERGENCE OF THE OSPF ROUTING PROTOCOL

Attila Rajmund NOHL and Gergely MOLNÁR

Ericsson Research
Traffic Analysis Laboratory
Ericsson Hungary Ltd.
1300 Budapest, Pf.: 107
e-mail: Attila.Rajmund.Nohl@ericsson.com,Gergely.Molnar@ericsson.com

Received: July 4, 2002

Abstract

In this paper an OSPF convergence time prediction model is introduced. It is based on examining the behaviour OSPF and on the analysis of the generated data by OSPF flooding. The model was validated and refined by tests and experiments on a test network built for this work. The resultant model can be used to predict the effect and convergence of a change in an OSPF network. This feature is very usable for pre-emptive network management and network planning.

Keywords: Internet Protocol (IP), Open Shortest Path First (OSPF), convergence, routing.

1. Introduction

Routing is a key mechanism that enables the IP networks used in today's Internet to work. During the development of the Internet, dynamic routing (implemented by routing protocols) replaced static routing in most complex networks. Many routing protocols were created, the Open Shortest Path First (OSPF) protocol [1], [2] is one of them.

The OSPF is the recommended Interior Gateway Protocol (IGP) by the Internet Engineering Task Force (IETF) [3]. It is widely deployed in the current Internet, and most router vendors support this routing protocol. One of the most important aspects of the performance of a routing protocol is its convergence. The convergence of a routing protocol is the period during the routers acclimatize themselves to the new network topology after a change in the network.

The OSPF routing protocol is a link state routing protocol. It means that each router must have the same view of the network. In the case of OSPF, each router must have the same set of Link State Advertisements (LSA) in their LSA databases. When there is a change in the network (e.g. a link goes down), the information about this change is flooded through the Autonomous System so each router can update its LSA database to reflect the new network topology. During this flooding, the above stated principle of OSPF is not true (some routers already got the new LSAs, some haven't got them yet) so there can be problems during flooding: rooting loops can arise, networks can become unreachable temporarily. These phenomena could mean degradation of service, so it is vital to know how long this convergence

period is. In this work, the convergence of OSPF was analyzed and examined by mathematical tools and by measurements on a test network.

One of the most important aims of this work was to examine the predictability of OSPF convergence from the size of the network (number of routers and networks between them). To reach this aim, a model is needed that gives proper measures about the convergence. The first step to get this model is learning how OSPF works and what properties, features and characteristics have effect on convergence. As the convergence depends on the amount of transferred data by the OSPF flooding process, the starting point was to see how much data are transferred during flooding. From these data and from the capacity of the network we can predict the convergence. Examining and analyzing the frequency of data exchange by OSPF routers and the generated data during flooding led to a deterministic model. To validate it, a test network was built and several experiments were done to see how the model works. The results showed that, on low-speed links, the deterministic model gave values which were very close to the measured values. To refine the model, a probabilistic component based on measurements was introduced, which led to a better model.

2. The Convergence of OSPF

The convergence of OSPF is achieved by its reliable flooding procedure: it ensures that each router has the same set of LSAs. The flooding is done over adjacencies: in the OSPF protocol neighbouring routers form adjacencies. On broadcast and NBMA networks (Non-Broadcast MultiAccess networks), instead of forming adjacencies between each pair of routers, the routers elect a Designated and a Backup Designated router. On these networks, the adjacencies are only formed between the Designated router and the other routers. The flooding procedure starts when a router issues a new LSA (e.g. because the status of one of its directly connected networks has changed). This router sends the new LSA to all of its adjacent routers. When a router receives a new LSA, it immediately sends the new LSA to its other adjacent routers. Of course, if a router receives an LSA it already has, it does not send the LSA further to other routers. In the flooding process, every LSA must be acknowledged. If a router doesn't receive acknowledgment for an LSA, it retries the sending of LSA until it receives acknowledgment (or the adjacency breaks up), so the flooding process is reliable. It must be noted that the basic assumption behind OSPF's correctness that the LSA databases are the same in every router, is not true during the flooding process. The exact flooding process is slightly more complex, see section 13 in [1].

The time of the flooding procedure is the convergence time of OSPF. We introduce the following notation to describe the time components of the flooding procedure (R_0 is the router that detects a change, R_i is another router in the network that receives the new LSA):

- t_{00}: the time, when R_0 detects a change

- t_{01}: the time, when R_0 produces the new LSA
- t_{i0}: the new LSA has arrived on the network to R_i $(i = 1..n)$
- t_{i1}: R_i has read and processed the new LSA
- t_{i2}: R_i has put the new LSA on the network $(i = 0..n)$
- t_{i3}: R_i has finished its SPF calculation
- t_{i4}: R_i has updated its routing table

In this case, the time of the convergence is

$$\max_{i=0}^{n}(t_{i4} - t_{00}) \tag{1}$$

Fig. 1 shows the flooding of one LSA from R_0 to R_2 with the above introduced notations.

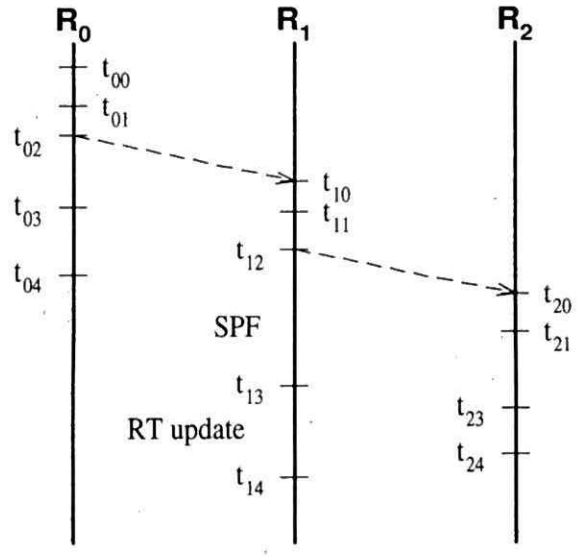

Fig. 1. Flooding of an LSA

In the process depicted in *Fig. 1* R_0 detects a change in the network at t_{00}, produces the new LSA at t_{01} and puts it on the network between R_0 and R_1 at t_{02}. The dashed arrow represents the LSA's travel through a network (between two adjacent routers). R_1 receives the new LSA at t_{10}, reads and processes it at t_{11} and puts it on the network at t_{12}. Similar events happen also on R_2. Of course, after sending the new LSA, each router has to run an SPF calculation on the changed network topology and they must update their routing table as well. In the example above, the time of convergence is $t_{14} - t_{00}$, because the calculations are slower on R_1.

3. The Deterministic Model

Our deterministic model is based on only the OSPF specification. This means that it only examines the time the packets spend travelling on the network (this is the time that can be calculated from the specification), we took the other components (packet processing time, SPF calculation, etc.) as 0. Also, this model only examines the flooding inside an OSPF area. However, it is quite simple to extend this model to cover inter-area flooding. To make the problem manageable, we made some more simplifying assumptions. These assumptions are mostly technical, directed to make easy to talk about the problem:

- The change in the network happened at R_0.
- R_n is the farthest router from R_0 (i.e. R_n receives the new LSA last).
- The routers between R_0 and R_n are R_1, R_2, ... R_{n-1} in this order.
- The routers' hardware is the same.

In this case, the time of convergence is:

$$t_{02} - t_{00} + \text{min\_time}\,(R_0, R_n) + t_{n4} - t_{n0} \tag{2}$$

The min_time (R_0, R_n) function is defined in the next section.

3.1. *The* min_time

The min_time (R_0, R_n) is the time the new LSA takes to reach R_n from R_0. It is the period of the time the LSA spends on the network between the routers and in the routers:

$$\text{min\_time}\,(R_0, R_n) = \sum_{i=1}^{n} \text{link\_time}\,(R_{i-1}, R_i) + (n-1)\cdot(t_{x2} - t_{x0}), \tag{3}$$

where link_time(R_x, R_y) is the time the new LSA takes to go through the network between the adjacent R_x and R_y routers. The link_time depends on the size of the LSA, the speed of the network between R_x and R_y, the traffic on that network and the type of the network. In the deterministic model, we left out of consideration the background traffic. We took also $t_{02} - t_{00}$, $t_{n4} - t_{n0}$ and $t_{x2} - t_{x0}$ as 0, because they depend on the router hardware rather than on the routing protocol.

3.2. *The* link_time

From the OSPF specification, we can compute link_time(R_x, R_y). The results are in *Table I* (assuming that the LSA fits into one IP packet and there are m routers on the network if it's not a Point to Point network).

Table 1. The link_time

Type of network	link_time (R_x, R_y)
(R_x, R_y) is PtP network	packet_time
(R_x, R_y) is broadcast network	
R_x is not designated router	$2 \cdot$ packet_time $+ (t_{x2} - t_{x0})$
R_x is the Bck Designated router	$2 \cdot$ packet_time $+ (t_{x2} - t_{x0})$
R_x is the Designated router	packet_time
(R_x, R_y) is NBMA network	
R_x is not designated router	$m \cdot$ packet_time $+ (t_{x2} - t_{x0})$
R_x is the Bck Designated router	$2 \cdot (m-1) \cdot$ packet_time $+ (t_{x2} - t_{x0})$
R_x is the Designated router	$(m-1) \cdot$ packet_time $+ (t_{x2} - t_{x0})$
(R_x, R_y) is PtMP network	$m \cdot$ packet_time

In *Table 1* packet_time is the time one Link State Update packet takes to go through the network. The results in *Table 1* are based on calculations and considerations in [4], here we show only one calculation in detail: on NBMA network, if the router, which has got the new LSA (R_x) is not a designated router, first sends the new LSA to both the Designated and Bck (Backup) Designated routers ($2 \cdot$ packet_time). Then the Designated router has to process the new LSA ($t_{x2} - t_{x0}$) and sends it to every non-designated router ($(m-2) \cdot$ packet_time). It means that, in the worst case, when R_y is a non-designated router and it receives the new LSA last from the Designated router, it takes $m \cdot$ packet_time $+ (t_{x2} - t_{x0})$ time to the LSA to travel from R_x to R_y.

3.3. *The* packet_time

The packet_time depends on the size of the LSA, the speed of the network and the overhead of the link layer protocol underneath IP. From the OSPF specification we get the following for packet_time:

$$\text{packet\_time} = \frac{(\text{lsa\_size} + 48 + \text{ll\_header}) \cdot b_p b}{\text{network speed}} \tag{4}$$

where lsa_size is the size of the new LSA in bytes, ll_header is the size of the link layer header in byte, $b_p b$ is the number of bits needed to transfer one byte through the network and *network speed* is the speed of the network in bps. The 48 byte is the size of the OSPF and IP header. It must be noted that some link level technologies use 9 or 10 bits to transfer one byte.

3.4. The Transferred Data

To wrap up the deterministic model, we describe the amount of transferred data during the flooding process. The complete amount of transferred data depends on the topology of the network and on the exact time of receiving the new LSA in various routers. However, we can compute the amount of transferred data between the R_0, R_1, ..., R_n routers:

$$\text{transferred\_data} = \sum_{i=1}^{n} \text{link\_data}\,(R_{i-1}, R_i), \tag{5}$$

where link_data (R_x, R_y) is the transferred data between R_x and R_y. The link_data (R_x, R_y) depends on the type of network between R_x and R_y and on the size of the LSA. From the OSPF specification we can compute link_data (R_x, R_y). The results are in *Table 2*. These equations are very similar to the ones in *Table 1*, but in this case, the acknowledgment packets had to be counted as well.

Table 2. The link_data

Type of network	link_data (R_x, R_y)
(R_x, R_y) is PtP link	$lsa\_size + 48 + 20 + 44$
(R_x, R_y) is broadcast link	
R_x is not designated router	$2 \cdot (\text{lsa\_size} + 48) + (m - 1) \cdot (20 + 44)$
R_x is the Bck Designated router	$2 \cdot \text{lsa\_size} + (m - 1) \cdot 20 + 2 \cdot (48 + (m - 2) \cdot 44)$
R_x is the Designated router	$\text{lsa\_size} + (m - 1) \cdot (20 + 44) + 48$
(R_x, R_y) is NBMA link	
R_x is not designated router	$m \cdot (\text{lsa\_size} + 20 + 48 + 44) - 44$
R_x is the Bck Designated router	$2 \cdot (m - 1) \cdot (\text{lsa\_size} + 20 + 48 + 44)$
R_x is the Designated router	$(m - 1) \cdot (\text{lsa\_size} + 20 + 48 + 44)$
(R_x, R_y) is PtMP link	$m \cdot (\text{lsa\_size} + 20 + 48 + 44)$

The numbers in *Table 2* contain the IP and OSPF headers, but the link level headers must not be forgotten for a precise result. Again, see [4] for the exact calculations and considerations behind the results above, here we show only one calculation in detail: if the network between R_x and R_y is a broadcast network and R_x is the Designated router, R_x first sends one IP packet containing the new LSA to the AllSPFRouters multicast address (it is $lsa\_size + 48$ bytes long), so every routers on the broadcast network receive the new LSA, and then they acknowledge it: the $m - 1$ routers send LSA Acknowledgments ($20 + 44$ bytes long) to the Designated router.

4. The Probabilistic Model

The deterministic model does not consider some factors in the real network. We model them with the following random variables:

- $p_1(x) = t_{02} - t_{00}$: the time to produce the new LSA and send it into the network.
- $p_2(x) = t_{i2} - t_{i0}$: the time to process and send the LSA into the network $(i = 1..n-1)$.
- $p_3(x) = t_{n4} - t_{n0}$: the time to process the LSA, run the SPF calculation and to update the routing table.

$p_1(x)$, $p_2(x)$, $p_3(x)$ are random variables with normal distribution. In this case, the equation in (2) becomes the following:

$$p_1(x) + \sum_{i=1}^{n} \text{link\_time}\,(R_{i-1}, R_i) + (n-1)\cdot_2(x) + p_3(x) \tag{6}$$

Of course, the values of $p_1(x)$, $p_2(x)$ and $p_3(x)$ are dependent on the router hardware and software. We've built a test network of PCs running the GNU zebra routing software on RedHat Linux operating system. We've made experiments and measurements on this network to get estimates for these random variables. We got the following values:

- $p_1(x)$: expected value: 11.228 ms, variance: 0.008 ms
- $p_2(x)$: expected value: 0.463 ms, variance: 0.027 ms
- $p_3(x)$: expected value: 7.868 ms, variance: 3.602 ms

For the details of these experiments, see Section A. Similar estimates for Cisco routers can be found in [5].

4.1. Experiments

We created three experiments to verify our models. All three experiments used the same topology. *Fig. 2* shows this topology.

In the different experiments, the routers were connected with different kinds of network: 9600 bps PPP (Point to Point Protocol) over null-modem cable, 57600 bps PPP over null-modem cable, 100M bps FastEthernet on cross-linked cable. In each case, the networks were configured as Point-to-Point links (even the FastEthernet network). The PPP links used 10 bits to transfer 1 byte, while the FastEthernet used only 8 bits.

In the experiments, the IP address of the interface of R_0 marked with '*' was changed and we measured the time, when the routing table was updated on R_3. In each experiment, the size of the LSA was 108 bytes. *Table 3* summarizes the results. See B for the details of the computations.

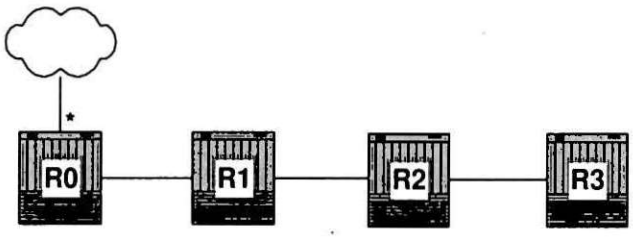

Fig. 2. The topology of the experiments

Table 3. The result of the experiments

Experiment	Deterministic	Probabilistic	Measured
100M bps FastEthernet	0.075 ms	19.634 ms	20.128 ms
57600 bps PPP	88.541 ms	108.1 ms	135.968 ms
9600 bps PPP	531.250 ms	551.272 ms	576.891 ms

5. Conclusion

It is important to know the properties of a routing protocol's convergence. It is not only useful when the network operator plans his network, but also, when he has to decide between two routing protocols. Our work shows that the convergence of OSPF is really fast. Although our experiments used only four routers, the convergence was fast in larger networks too if they used FastEthernet or some other similarly fast technologies. It must be noted that in the backbone of networks, a 100 M bps FastEthernet is not considered high speed nowadays.

From the experiments we can also see that our probabilistic model is better than the deterministic model, and it is accurate enough to be used during network planning. On a high speed network, the convergence time is dominated by the packet processing, that's the reason why our deterministic model (from which we left out the packet processing time) is quite inaccurate, but on a very low speed network, even the deterministic model might be useful.

Unfortunately, there are other factors that can change the time of convergence dramatically. For example, a router might not detect immediately the change in the network, e.g. if a link level technology does not notify the router about the loss of connection, the router must wait for RouterDeadInterval seconds to discover the loss of connection. The default value of RouterDeadInterval is 40 seconds, which is not in the same magnitude as our results. On really large network, the SPF calculation can take quite a long time (measurable in seconds). Also some routers can be configured to delay their SPF calculation as long as 60 seconds, because one change in the network might trigger the issue of more than one new LSAs, so the router tries to wait for them. For the discussion of these issues, see [6].

A. Getting the Estimations

We made experiments to get estimations for the $p_1(x)$, $p_2(x)$ and $p_3(x)$ random variables introduced in 4. These experiments were conducted on the same test network depicted in *Fig 2.*

A.1. Measuring $p_1(x)$

The $p_1(x)$ random variable represents the time to produce the new LSA and to send it into the network. We measured it as the time between the change of the interface's address and the time of the Link State Update packet's appearance on the network. The time of changing the interface's address was simply measured by printing the actual date before and after issuing the *ifconfig* command and we took the average time. The *ifconfig* command is used to change the IP address of an interface on Linux. The time of the appearance of the new LSA on the network was measured with running the *tcpdump* utility on the interface where the routers sent the new LSA. The *tcpdump* utility monitors the traffic of an interface: it prints out every packet's header and the time when the packet goes through the interface. The difference between the two times is the value we were looking for. We made hundreds of these experiments, and then applied a maximum likelihood estimation on the result to get the values stated in Section 4.

A.2. Measuring $p_2(x)$

The $p_2(x)$ random variable represents the time to process and send the LSA into the network. We measured it as the time between receiving and sending the new LSA. The time of receiving the new LSA was measured with running the *tcpdump* utility on the interface, where the new LSA came. The time of sending the new LSA was measured with running the *tcpdump* utility on the interface, where the router sent the new LSA. The difference between the two times is the value we were looking for. We made hundreds of these experiments, and then applied a maximum likelihood estimation on the result to get the values stated in Section 4.

A.3. Measuring $p_3(x)$

The $p_3(x)$ random variable represents the time to process the LSA, run the SPF calculation and to update the routing table. We measured it as the time between receiving the new LSA and updating the routing table. The time of receiving the new LSA was measured with running the *tcpdump* utility on the interface where the LSA came. The time of updating the routing table was measured with running the *rtmon* utility, and later examining its result with the *ip monitor* command. The *rtmon*

utility is used to monitor the changes in the routing table of the Linux operating system. It generates a log file, that can examined with the *ip monitor* command. The difference between the two times is the value we were looking for. We made hundreds of these experiments, and then applied a maximum likelihood estimation on the result to get the values stated in Section 4.

B. Computing the results

In the second experiment, the prediction of the deterministic model is

$$\begin{aligned} t_{02} - t_{00} + \text{min\_time}(R_0, R_3) + t_{34} - t_{30} &= \text{min\_time}(R_0, R_3) = \\ &= \text{link\_time}\,(R_0, R_1) + \text{link\_time}\,(R_1, R_2) + \text{link\_time}\,(R_2, R_3) = \\ &= 3 \cdot \text{link\_time}\,(R_0, R_1) \end{aligned} \tag{7}$$

The last equation is valid because the networks between the routers were the same. If we compute link_time, we get

$$\text{link\_time}\,(R_0, R_1) = \text{packet\_time} = \frac{(\text{lsa\_size} + 48 + \text{ll\_header}) \cdot b_p b}{\text{network speed}} \tag{8}$$

Because lsa_size was 108 bytes, the link level header was 14 bytes, $b_p b$ was $10\frac{\text{bits}}{\text{bytes}}$ and the network speed was 57600 bps, we got 88.541 ms as the convergence time.

In the third experiment, the prediction of the probabilistic model is

$$\begin{aligned} p_1(x) + \sum_{i=1}^{n} \text{link\_time}\,(R_{i-1}, R_i) + 2\cdot_2(x) + p_3(x) \\ = p_1(x) + 3 \cdot \text{link\_time}\,(R_0, R_1) + 2\cdot_2(x) + p_3(x). \end{aligned} \tag{9}$$

The last equation is valid because the networks between the routers were the same. If we compute link_time, we get

$$\text{link\_time}\,(R_0, R_1) = \text{packet\_time} = \frac{(\text{lsa\_size} + 48 + \text{ll\_header}) \cdot b_p b}{\text{network speed}} \tag{10}$$

Because lsa_size was 108 bytes, the link level header was 14 bytes, $b_p b$ was $10\frac{\text{bits}}{\text{bytes}}$ and the network speed was 9600 bps, we got the following as convergence time:

$$11.228 + 531.25 + 2 \cdot 0.463 + 7.868 = 551.272 \text{ ms} \tag{11}$$

C. Glossary

- OSPF – Open Shortest Path First: an Interior Gateway Protocol based on link state technology.
- LSA – Link State Advertisement: the basic data unit of OSPF, it describes one router or network. The routers store the LSAs describing the whole network in their LSA database.
- SPF calculation: the process of calculating the shortest path tree to every routers in the network. The input for this calculation is the set of LSAs in the routers LSA database.
- flooding: the mechanism of OSPF that distributes the topology information of the network through the network.
- PtP – Point to Point network: one of the four network types distinguished by OSPF. There can be only two routers connected to a PtP network.
- Broadcast network: the second network type distinguished by OSPF. More than one router can be connected to a broadcast network and they can directly communicate with each other. The link layer technology provides a multicast mechanism: a router can send one single packet to more than one router on the same network.
- NBMA – Non-Broadcast Multi Access network: the third network type distinguished by OSPF. It is very similar to the broadcast network, but the link layer technology does not provide a multicast mechanism, the routers cannot send one single packet to more than one other router.
- PtMP – Point to Multipoint network: the fourth network type distinguished by OSPF. It is similar to NBMA, but some routers might not be able to communicate directly with each other, although they are connected to the same network.
- PPP – Point to Point Protocol: a widely used link layer technology on physical links that connect two computers.
- RFC – Request for Comments: the standards of the Internet are documented in the various RFCs.

References

[1] MOY, J., OSPF Version 2, RFC 2328 http://www.ietf.org/rfc/rfc2328.txt
[2] BAKER, F. – COLTUN, R., OSPF Version 2 Management Information Base, RFC 1850 http://www.ietf.org/rfc/rfc1850.txt
[3] GROSS, P. Choosing a 'Common IGP' for the IP Internet, RFC 1371 http://www.ietf.org/rfc/rfc1371.txt
[4] NOHL, A. R., Az OSPF routing protokoll teljesítményanalízise (in Hungarian) http://people.inf.elte.hu/nar/diplomamunka/
[5] SHAIKH, A. – GREENBERG, A., Experience in Black-box OSPF Measurement ACM SIGCOMM Internet Measurements Workshop 2001, http://www.research.att.com/^albert/papers/sigcom-imw01/mes-ws01-paper.pdf
[6] ALAETTINOGLU, C. – JACOBSON, V. – YU, H., Toward Milli-Second IGP Convergence Internet Draft, http://www.packetdesign.com/Documents/convergence.pdf