Implementation of Highly Available OSPF Router on ATCA *

Chia-Tai Tsai, Rong-Hong Jan†, and Chien Chen
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan
{cttsai, rhjan, cchen}@cis.nctu.edu.tw

Chia-Yuan Huang
Information & Communications
Research Laboratories
Industrial Technology Research Institute
Hsinchu, Taiwan
ricehuang@itri.org.tw

Abstract

This paper proposes a Highly-Available Open Shortest Path First (HA-OSPF) router which consists of two OSPF router modules-active and standby-to support a high-availability network. Each router module runs a Linux operating system, high-availability management middleware (HAM middleware), and OSPF daemon. The HAM middleware consists of an Availability Management Framework (AMF) service, checkpoint service, interface monitor, OSPF fault manager, and fault handler; it provides a health check, state information exchange, and takeover mechanism. The experimental results are given to show the system availability of the HA-OSPF router on a PC-based prototype system. Furthermore, to build a carrier grade commercial product, we realize a HA-OSPF router on an industry standard compliant Advanced Telecom Computing Architecture (ATCA) hardware platform. From actual measurements, we show that our PC-based and ATCA-based HA-OSPF routers take only 166 and 131 ms to switch over to a standby router module when there is a software fault and 360 and 331 ms with a hardware failure respectively.

1. Introduction

With the recent progress in broadband networks, people can now access information from the Internet quickly and easily. And in addition to individuals accessing the Internet, many businesses now rely heavily on Internet applications and services. Thus, it is crucial for *Internet Service Providers* (*ISPs*) to build high-availability networks to provide the continuous service users expect and demand.

The most familiar way to improve network availability is to add redundant routers to the network. In general, this approach consists of a cluster of routers where one is active and the others are on standby. Whenever the active router fails, one of the standby routers simply takes over and becomes the active router as soon as possible. However, the takeover time depends on the restart model and the restart model depends on the amount of information available to the system at the time the failure occurred. The more information available means a faster recovery time.

The restart models can be classified into three types [1]: *hot restart*, *warm restart*, and *cold restart* models. In a hot restart model, the standby routers save all of the state information about the active router. If the active router fails, the standby router is ready to take over rapidly. Thus, the hot restart system has the fastest recovery time but is the most complex to implement. A warm restart model, in contrast, is applied to an N+1 redundant system with N active devices and one standby device. The fault management applications save state information about the current activity of the system. When one of active devices fails, the standby device is automatically designated to take over the failed device and is also configured with the necessary application and state information. The warm restart needs more time to recover but it can reduce the costs of the standby components. In a cold restart model, the standby routers do not save any state information kept in the active router. When the active router fails, the standby router begins to reconstruct the routing information from the initial state. Thus, a cold restart is the least complex to implement but requires the most recovery time. Several approaches, such as *primary backup approach* [2], *virtual router redundancy protocol* (*VRRP*) [3] [4] , and *hot standby routing protocol* (*HSRP*) [5] belong to this kind of restart model.

In addition to the restart model, fault detection methods also affect the takeover latency. There are two kinds of fault detecting methods: *active fault detection* and *passive fault detection*. The active fault detecting method detects

*This research was supported in part by the National Science Council, Taiwan, under grant NSC 96-2219-E-009-012.

†Corresponding Author; Fax: 886-3-5721490; e-mail: rhjan@cis.nctu.edu.tw

0-7695-3054-0/07 $25.00 © 2007 IEEE
DOI 10.1109/PRDC.2007.13

the problem by polling the devices: the higher the polling frequency, the faster the fault detection. In contrast, the passive fault detecting method sets a timer and waits for the heartbeats (or *Hello* messages) sent from the devices. If it does not hear a heartbeat from the monitored device within a specific time interval, it supposes the device has failed. For example, in *OSPF* (*Open Shortest Path First*) protocol [6], if a router does not hear a *Hello* message from its neighbor within a *RouterDeadInterval*, it assumes the link between them has failed. The default setting of *RouterDeadInterval* is 40 seconds (or 4 *HelloIntervals*). That is, the OSPF protocol takes at least 40 seconds to detect the link failure. Thus, the smaller RouterDeadInterval may result in shorter fault discovery time. However, the *HelloInterval* also needs to be reduced and the overhead (number of *Hello* messages) increased.

This paper focuses on the availability of OSPF networks with redundant routers. The OSPF, which is a link state routing protocol, is the most commonly used *Interior Gateway Protocol* (*IGP*) on the Internet. In the OSPF protocol, each router broadcasts *Link State Advertisement* (*LSA*) messages to the other routers. Thus, the router knows the network topology and it can determine the shortest path to different destinations using LSA information. As mentioned above, when a link failure occurs, the OSPF protocol may take forty to fifty seconds to detect the failure and rebuild the routing information. Such a long delay is unacceptable for an access network where many businesses rely heavily on it.

In this paper, we propose a *highly available OSPF* (*HA-OSPF*) router which consists of two OSPF router modules: *active* and *standby* [7] (see Fig. 1). A dedicated connection is used to connect two router modules. For convenience, the terms active router module (standby router module) and active router (standby router) will be used interchangeably. The active router (e.g., module A) communicates with its neighbor OSFP routers (e.g., OSFP routers S and T) by sending and receiving OSPF messages to create and modify the routing information to the nodes in the OSPF network. Each router module in the HA-OSPF router runs a Linux operating system, *high availability management* (*HAM*) middleware, and OSPF daemon. The HAM middleware includes *Availability Management Framework* (*AMF*) service [8], checkpoint service, interface monitor, OSPF fault manager, and fault handler. The AMF and checkpoint services are also known as openAIS, which is a cluster middleware defined in *Service Availability Forum* (*SAF*) *Application Interface Specification* [8]. The AMF service provides the health check (by heartbeat) and the takeover scenario. The AMF service chooses the active router from the two router modules available. The active router module will forward data packets according to information found in its routing table. In order to support the hot restart model,

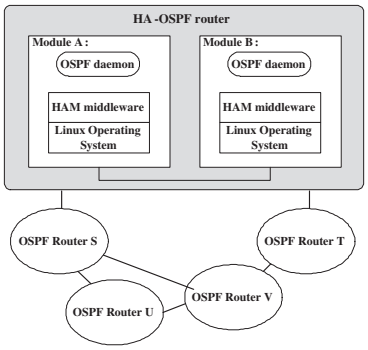

Figure 1. An HA-OSPF router.

the routing table and the link state database will be synchronized between the active and standby modules by the checkpoint service in the HA-OSPF router.

Another design issue of the HA-OSPF router is how to detect faults. Hardware faults can be the result of overheating, power failure, failed memory, or poor buggy board design. Software failures, usually more frequent than hardware failures, may be the result of an operating system crash, application crash, memory exhaustion, unhandled exceptions, deadlock, or applications going into an infinite loop. This paper illustrates fault detections for network interfaces, OSPF daemons, and AMF service. It is easy to extend the proposed fault detection methods for solving other hardware and software faults.

To implement a more realistic router, the proposed HA-OSPF router is built on the industry standard compliant *Advanced Telecom Computing Architecture* (*AdvancedTCA* or *ATCA*) hardware [9]. The *PCI Industrial Computer Manufacturers Group* (*PICMG*) [10]proposed the ATCA standard to provide an open and modular architecture that allows an economical and reliable router to be constructed. ATCA is a series of industry standard specifications for the next-generation carrier grade communication equipment. It provides a standardized architecture for carrier-grade communication applications. In this paper, we implemented HA-OSPF router modules on ATCA control cards and measured the takeover delays when the faults occurred.

Although an ATCA-based open architecture router (OAR) prototype has been developed for the first time in [11], it emphasized on the demonstration of basic forwarding function with routing protocol processing on ATCA hardware platform. However the most important reliability issue of the ATCA-based router was not discussed. [11] did not provide a mechanism to back up the *Link State Data Base* (*LSDB*) information, nor the detail measurements of takeover latency. Thus, we propose to construct HA-OSPF router modules on ATCA control cards. Two OSPF router modules-active and standby-are implemented on ATCA control cards. Each router module runs a Linux

operating system, openAIS, and OSPF daemon.

The experimental results show that the standby router can takeover within 131 *ms* if the OSPF daemon fails. And, it takes 1063 *ms* to takeover if a failure occurs in AMF service. The takeover latency for AMF service failures can be reduced to 331 *ms* by decreasing the down check interval. The down check interval is a period of time in which the standby router has to hear at least one heartbeat from the active router; otherwise, the standby router assumes it has failed. This paper shows that the shorter down check interval can achieve better availability yet the overhead of control messages increases only slightly.

2. System Architecture

In this section, we describe the system components, state information backup, fault detection and recovery for the proposed HA-OSPF router.

2.1. Components of HAM middleware

Fig. 2 illustrates HAM middleware which is added in the HA-OSPF router to perform high availability management. The HAM middleware consists of AMF service, checkpoint service, interface monitor, OSPF fault manager, and fault handler. The functions of these components are given in the following:

1. *AMF service*: It provides the health check and the role assignment. The AMF in the active router sends a heartbeat message on the dedicated connection periodically to report its health. If the standby does not hear a heartbeat message from the active router within a down check interval (e.g., 1 second), it will assume the active router has failed and will take over the functionality.

2. *Checkpoint service*: It provides state information exchange service for two router modules. Through checkpoint service, the active router can save the OSPF's state information to the standby router.

3. *Interface monitor*: It checks the health of the network interface cards (NICs) and informs AMF if any NIC failure occurs.

4. *OSPF fault manager*: It monitors OSPF daemon's operations, informs AMF if a fault in the OSPF daemon is detected, and collects state information for checkpoint service.

5. *Fault handler*: It has a set of callback functions. When AMF notifies it that a fault has occurred, it will execute a predefined callback function to handle the fault. Usually, it reboots the corresponding device.

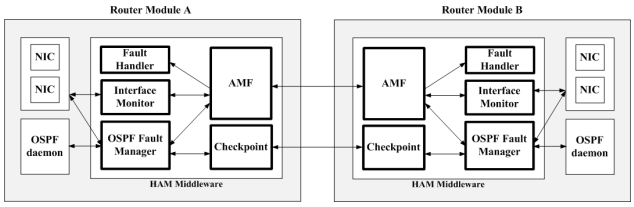

Figure 2. Architecture of HAM middleware

2.2. OSPF state information backup

When the HA-OSPF router is first started, its router modules (i.e., router modules A and B in Fig. 2) exchange messages, announcing their intentions to become the active router. The AMF uses "*smaller timestamp wins*" to break the tie. Both active and standby routers run their OSPF daemons. The active router enables its NICs to transmit and receive messages from the network while standby disables its NICs.

Then, the active router sends the *Hello* message to its neighbor periodically to test the connectivity of neighbors. After the active router finds out the state of the link to its neighbors (up or down), it broadcasts link state information to every node. Similarly, the active router can receive link state advertisements from other nodes. Thus, a *link state database* (*LSDB*), which represents the network topology, can be constructed from the active router. Based on LSDB, the active router calculates the routing path by the shortest path algorithm [12] and updates its routing table.

On the other hand, the standby router runs an OSPF daemon but is unable to create its link state information because its NICs were disabled. In order to achieve hot restart, the active router has to save state information to the standby router. For OSPF applications, the state information must include link states of active routers, LSDB, and routing tables.

Fig. 3 shows how state information flows from the active router's OSPF daemon to standby routers. As shown in Fig. 3, the OSPF daemon in the active router first passes state information to the OSPF fault manager by shared memory. The OSPF fault manager next asks the checkpoint service to send state information to the standby router. Through checkpoint service, the standby router receives state information and passes them to the OSPF fault manager. Finally, the OSPF daemon in the standby router saves the state information.

2.3. Fault detection and recovery

In the following we describe fault detection and recovery mechanisms for OSPF daemons, NICs, and router module faults. We assume that, at the start, router module A is ac-

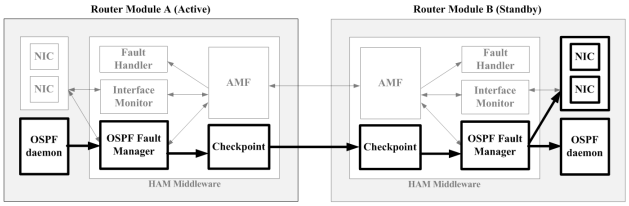

Figure 3. OSPF state information backup.

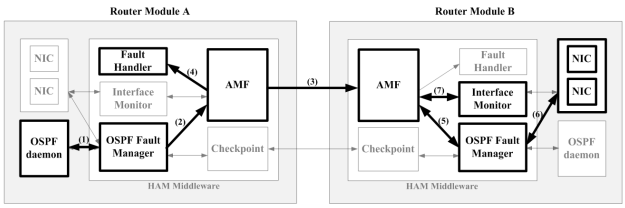

Figure 4. Fault detection and recovery for OSPF daemon, where the numbers 1-7 show the sequence of steps in the process.

tive while B is at standby. When the HA-OSPF router system starts, the OSPF fault manager and Interface monitor register themselves at the AMF and their call back functions at fault handler. So, if the OSPF fault manager or interface monitor informs AMF that a fault occurred, the AMF can ask the fault handler to perform the corresponding call back function.

Fig. 4 illustrates the fault detection and recovery for the OSPF daemon where the numbers 1-7 show the sequence of steps in the process. The stepwise description is given as follows.

1. The OSPF fault manager in router A polls the status of the OSPF daemon periodically, say, t sec. If a fault occurred in the OSPF daemon, the OSPF fault manager can detect it within t sec.

2. If a fault in the OSPF daemon is detected, the OSPF fault manager sends an error report that indicates an OSPF daemon fault to the AMF service.

3. After receiving the error report, the AMF service of router A generates an error message to router B's AMF service.

4. The AMF service of router A also executes the call back function which was defined in the fault handler to handle the fault. The call back function will reset the OSPF daemon and then the router A will become the standby router.

5. When the AMF service of router B receives the error message, it will take over and change its role to that of active router. After that, the router B's AMF service starts the OSPF fault manager.

6. Note that the OSPF network interfaces were disabled when router B was on standby. As soon as router B takes over, the OSPF fault manager of router B will enable the network interfaces and send a gratuitous ARP message to the network. The gratuitous ARP message is used to ask its neighbors to bind its MAC address to the HA-OSPF router's IP address.

7. Finally, the AMF activates the interface monitor.

Similarly, the interface monitor periodically probes the NIC's status. If a fault is found in the NIC, the interface monitor sends an error report that indicates an NIC fault to the AMF service. The remaining steps are the same as steps 3-7 above.

Next, consider the case of a fault in the router module but not in the OSPF daemon or NICs. For example, there can be a fault in HAM middleware components, such as the active router's AMF failed. This type of fault can be detected and the system can be recovered as follows: The AMF of the active router periodically announces its presence with a heartbeat message. If the standby router fails to receive the heartbeat for some period of time (i.e., down check interval), it will take over the functionality and then change its role to that of active router. Next, the router B's AMF service starts the OSPF fault manager. Finally, the AMF performs steps 6 and 7 as listed above.

2.4. ATCA-based HA-OSPF router implementation

The *Advanced Telecom Computer Architecture* (*AdvancedTCA* or *ATCA*) standard [9], which provided a common hardware platform for carrier-grade communication equipment, was introduced by the PICMG [10] Association. ATCA technology allows new communication equipment to be constructed with great attributes such as high-performance, high-availability, adaptability for adding new features, and lower cost of ownership. An open architecture solution using ATCA technology can improve time-to-market and increase service availability. Today's industries use ATCA open architecture combined with their own software solutions to quickly deploy competitive services.

There are three different types of cards that are used to build an ATCA-based router prototype shown in Fig 5. The *Link Cards* (LCs) are designed for the basic packet forwarding function. The *Switch Cards* (SCs) switch the packets between two link cards. The *Control Cards* (CCs) perform the routing protocol (e.g. OSPF) based on the received control packet. A highly available router should be composed

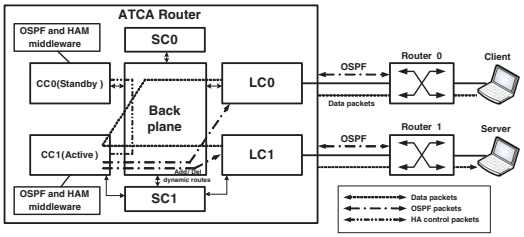

Figure 5. the ATCA router prototype.

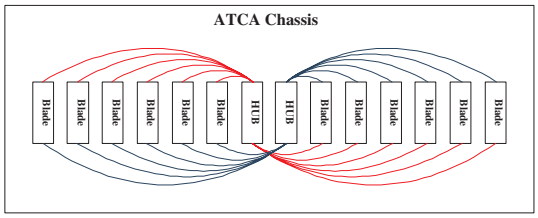

Figure 6. Dual Star topology.

with two SCs and two CCs. When the ATCA-based router system is turned on, a control card is elected as an active control card to perform the routing functionality. The control cards send and receive the heartbeat and checkpoint information through the backplane. When the link cards receive the OSPF control packets, they forward the packets to the active control card. When the active control card fails, the redundancy control card should be able to take over the routing protocol processing.

The ATCA cards communicate to the other cards through the backplane. In order to increase the availability, ATCA backplane supports five types of communication model [9], *Star, Dual Star, Dual-Dual Star, Replicate Mesh*, and *Full Mesh*. In this paper, the Dual Star Topology Backplane, which had a redundant channel, was chosen. Dedicated and redundant slots, called hub slots, were installed on the backplane for the two switch cards. Fig. 6 indicates that each hub slot has a channel connected to each node slot through the backplane. Each node slot can host either a LC or a CC. Both hub slots are also connected to each other in the backplane.

In [11], the ATCA-based open architecture router prototype did not support the seamless takeover of OSPF protocol when the active control card failed. The standby needs to re-initiate with its neighbor routers and the packets cannot be forwarded during the re-initialization stage. In this paper, we propose an ATCA-based HA-OSPF router by implementing the HAM middleware on the ATCA control cards. The proposed ATCA-based HA-OSPF router can support seamless takeover and dynamic routing. Our experiment results showed that the ATCA-based router had a better takeover performance than an ordinary router when a fault occurred.

In our system, two AdvancedTCA compliant processor cards (control cards), aTCA-6890, constructed by the *ADLINK Technology Inc.* [13] were used to build the ATCA-based HA-OSPF router. The aTCA-6890 is available in a dual processor configuration with the Low Voltage Intel 3.2 GHz Xeon processor with 800MHz System Bus. The aTCA-6890 also features the Intel E7520 chipset and 4 GB DDR2-400 memories. Peripherals include six Gigabit Ethernet ports and two 10/100/1000Mbps Ethernet maintenance ports. The configuration of the ATCA-based HA-OSPF router and the experimental environment are discussed in the next section.

3. Experimental Results

In order to assess the performance of the HA-OSPF router, the most relevant measurement is the takeover latency. In this paper, the takeover latency is defined as the delay from a link disconnected to the system being recovered. We first evaluate the takeover delay in a PC-based environment. Then the takeover delay is measured in an ATCA-based system. Finally, the two results are shown and compared.

3.1. PC-based system

In this experiment, two PCs with Intel Pentium 4 3.0 GHz processors connected via Ethernet are used to emulate an HA-OSFP router. That is, the HA-OSPF router consists of PCs R2 and R3 as shown in Fig. 7. The PCs for R2 and R3 are the desktop computers with Intel Pentium 4 3.0 GHz and 512 MB memories. A Linux operating system and *GNU Zebra* [14] were selected as the developing platform for the HA-OSPF router. GNU Zebra is the well-known and free routing software that performs well. Both R2 and R3 have three Ethernet interfaces: eth0, eth1, and eth2. Interface eth0 connects to the 192.168.2.0 network while interface eth1 goes to the 192.168.3.0 network. The IP address 192.168.2.253 (192.168.3.253) is assigned to both interface eth0s (eth1s) of R2 and R3. The interface eth2 of R2 is connected to the interface eth2 of R3 via the Ethernet. The heartbeat is exchanged between interface eth2s of R2 and R3. We assign IP addresses 10.0.0.1 and 10.0.0.2 to interface eth2s of R2 and R3, respectively. There are two OSPF routers, R1 and R4, in our experimental network.

In the experiment, a notebook PC S1 sent UDP data packets with specific sequence numbers to a notebook PC S2 to examine the network's connectivity (see Fig. 7). The Log Server was constructed to record the sequence number and timestamp of each packet that it received. If S1 sent a packet to S2, it had to also send a copy of the packet to the Log Server. Then, the S2 forwarded the packet that it

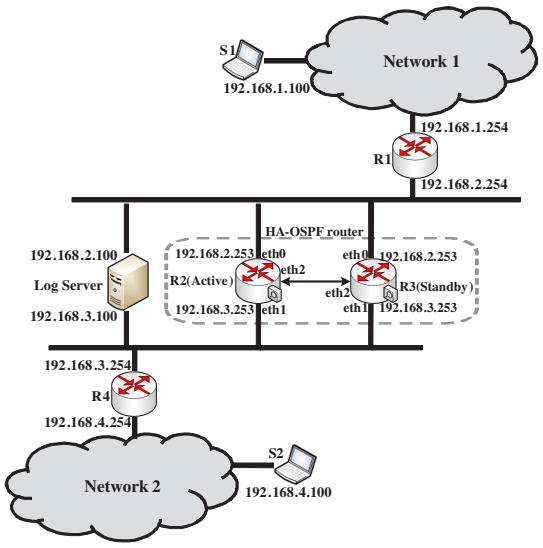

Figure 7. Experimental Environment.

received from S1 to the Log Server. During the takeover period, the network was disconnected. The Log Server did not receive any packets transferred from S2. After the HA-OSPF router was recovered, the Log Server continued to receive the transferred packets from S2. By this way, the time that the network was disconnected can be found.

At first, we investigated how the takeover delay was affected by the amount of information backup in the standby router. Three cases are considered as follows.

- Case 1: The active router does not store any state information in the standby router.
- Case 2: The active router only backs up its routing table to the standby router.
- Case 3: The active router backs up full state information, including its link states, LSDB, and routing table to the standby router.

In addition, two types of faults are considered. One is when the R2 is shut down (referred to as a hardware failure), and the other is when an OSPF daemon crashes (referred to as a software failure). Firstly, the UDP packets traveled along path S2, T4, R2, R1, S1 until the active router was failed. After R4, R3, and R1 reestablished their routing information, the UDP packets can go through the path S2, R4, R3, R1, S1. We used the t distribution with 9 degrees of freedom and a 95% confidence interval to estimate the takeover delays for these three cases.

The average takeover delays are shown in Table 1. From Table 1, note that the total recovery delays of hardware failure (software failure) for Cases 1 and 2 were 14511 ± 36 *ms* and 11179 ± 2 *ms* (13383 ± 3 *ms* and 10066 ± 2 *ms*), while

Table 1. Takeover delay (*ms*) for three cases.

	Case 1	Case 2	Case 3
Power Down	14511 ± 36	11179 ± 2	1240 ± 12
OSPF Daemon Down	13383 ± 3	10066 ± 2	166 ± 9

Table 2. CPU usages of HA-routers.

	Case 1	Case 2	Case 3
CPU Usage	$0.17 \pm 0.03\%$	$0.35 \pm 0.07\%$	$4.47 \pm 0.43\%$

for Case 3 it was only 1240 ± 12 *ms* (166 ± 9 *ms*). Our HA-OSPF router with full state information backup (Case 3) shows its benefits. Remarkably, it only takes 166 ± 9 *ms* to recover a software fault. This recovery is most seamless. This recovery is most seamless. On the other hand, we learned from Table 2 that the CPU usages of HAM middleware and forwarding process for Cases 1, 2, and 3 were $0.17 \pm 0.03\%$, $0.35 \pm 0.07\%$, and $4.47 \pm 0.73\%$. Note that Case 3 only required a small percent of CPU time.

Next, we measured the takeover delay for the case of NIC failure. The takeover delay consists of failure detection time and recovery time. The failure detection time depends on the polling intervals. On average, the failure detection time was half that of the polling intervals. Table 3 shows the takeover delays, 121 ± 5 *ms*, 166 ± 9 *ms*, and 222 ± 23 *ms* for three polling intervals, 50 *ms*, 100 *ms*, and 200 *ms*. It is easy to see that the shorter the polling interval, the faster the fault detection and recovery is.

We then investigated takeover delays for different down check intervals. Two types of failures, power down and AMF failures, are considered. We used the t distribution with 9 degrees of freedom and a 95% confidence interval to estimate the takeover delays. The average takeover delays are shown in Table 4. From Table 4, note that the total recovery delays of power down failure (AMF failed) for down check intervals 1000 *ms*, 500 *ms*, and 200 *ms* were 1240 ± 12 *ms*, 740 ± 15 *ms*, and 360 ± 6 *ms* (1204 ± 9 *ms*, 707 ± 13 *ms*, and 328 ± 4 *ms*). That is, smaller down check intervals result in shorter takeover delays.

Table 3. Takeover delay for different polling intervals (*ms*).

Polling Interval	50 *ms*	100 *ms*	200 *ms*
Interface Failed	121 ± 5	166 ± 9	222 ± 23

Table 4. Takeover delay for different polling intervals (*ms*).

Down check interval	1000 *ms*	500 *ms*	200 *ms*
Power Down	1240 ± 12	740 ± 15	360 ± 6
AMF Failed	1204 ± 9	707 ± 13	328 ± 4

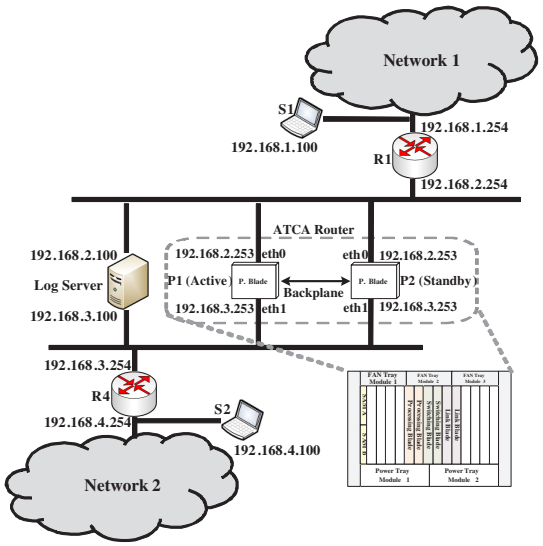

Figure 8. Experimental Environment (ATCA).

3.2. ATCA-based system

Even though we used two PCs connected via Ethernet to emulate an HA-OSPF router in the previous experiment, our HA-OSPF router can be easily implemented on an ATCA platform. We employed HA-OSPF software modules and openAIS middleware on ATCA control cards and then integrated them on an ATCA chassis to build an ATCA-based HA-OSPF router. In the ATCA, both of the control cards have two Ethernet interfaces connected to the backplane. Therefore, the heartbeat and checkpoint can be exchanged between cards by the backplane. In this experiment, the PC R2 and R3 were replaced by the control cards P1 and P2 (see Fig. 8). The configurations of the control cards on the ATCA are the same as on the PC-based system. We evaluated the takeover delay of ATCA-based HA-OSPF router under the three different cases as defined in section 3.1.

We measured the average takeover delays of the PC-based and ATCA-based routers, and the results are shown in Table 5. The total recovery delays of hardware failure and software failure are 1063 ± 54 *ms* and 131 ± 8 *ms* for ATCA-based routers. The takeover delays of the ATCA-based HA-OSPF router were less than the PC-based router.

Table 5. Takeover delays *(ms)* (case 3) on PC and ATCA.

	PC-based	ATCA-based
Power down	1240 ± 12	1063 ± 54
OSPF daemon down	166 ± 9	131 ± 8

Table 6. CPU usages of HAM middleware and forwarding process on ATCA.

	PC-based	ATCA-based
CPU Usage	4.47 ± 0.73 %	0.11 ± 0.01 %

According to Table 6, we can find out that the CPU usage of ATCA-based HA-OSPF router is much less than the PC-based routers (0.11% vs. 4.47%). This means that the processing resources of ATCA control card are much more powerful than the ordinary PC. As a result, the ATCA-based routers have a shorter takeover time.

Table 7 shows the takeover latencies for the different polling intervals when the NIC failed. The takeover delays of ATCA-based HA-OSPF were 188 ± 9 *ms*, 217 ± 17 *ms*, and 242 ± 26 *ms* for three different polling intervals. The takeover delays of an ATCA-based router were greater than that of a PC-based router. It might be the NICs' driver of the control card that cannot detect the disconnection of NICs immediately. Table 8 shows the total recovery delays of power down and AMF failures for different down check intervals. The results show that ATCA-based routers performed better than the PC-based router.

4. Conclusions

This paper presents a highly available OSPF router which consists of two OSPF router modules to support high availability networks. The HAM middleware, which in-

Table 7. Takeover delay of ATCA and PC-based routers for different polling intervals.

	Polling interval (*ms*)		
	50 *ms*	100 *ms*	200 *ms*
PC-based	121 ± 5	166 ± 9	222 ± 23
ATCA-based	188 ± 9	217 ± 17	242 ± 26

Table 8. Takeover delay of ATCA and PC-based routers for different down check timeout(*ms*).

	Down check interval					
	1000 *ms*		500 *ms*		200 *ms*	
	PC-based	ATCA-based	PC-based	ATCA-based	PC-based	ATCA-based
Power down	1240 ± 12	1066 ± 54	740 ± 15	743 ± 36	360 ± 6	331 ± 28
AMF failed	1204 ± 9	1027 ± 70	707 ± 13	707 ± 45	328 ± 4	295 ± 36

cludes AMF service, checkpoint service, interface monitor, OSPF fault manager, and fault handler, is designed and added in the HA-OSPF router to provide a health check, state information exchange, and takeover mechanism. The experimental results show that the HAM middleware is highly effective for failure detection and system recovery. We also implemented the HA-OSPF modules on the ATCA control cards. The ATCA provides an industrial standardization modular architecture for an efficient, flexible, and reliable router design. Finally, we measure the average takeover delay of our ATCA HA-OSPF router is about 131 *ms* for a software failure and 331 *ms* for a hardware failure.

5 Acknowledgements

The authors would like to thank Lo-Chuan Hu and Ching-Chun Kao, Information and Communications Research Laboratories, Industrial Technology Research Institute (ITRI), Taiwan, for their helpful assistance in conducting the experimental results.

References

[1] S. Srivastava, "Redundancy Management for Network Devices," *The 9th Asia-Pacific Conference on Communications*, vol. 3, Sept, 2003, pp. 1157-1162.

[2] N. Budhiraja, K. Marzullo, F. B. Schneider, and S. Toueg, *Distributed Systems*, 2nd Edition, ACM Press/Addison-Wesley Publishing Co., 1993, pp. 199-216.

[3] S. Knight, D. Weaver, D. Whipple, R. Hinden, D. Mitzel, P. Hunt, P. Higginson, M. Shand, and A. Lindem, "Virtual Router Redundancy Protocol (VRRP)," RFC 2338, Internet Engineering Task Force (IETF), Apr, 1998.

[4] J. Ranta, "Router Redundancy and Scalability Using Clustering," *Seminar on Internetworking*, 2004.

[5] T. Li, B. Cole, P. Morton, and D. Li, "Cisco Hot Standby Router Protocol (HSRP)," RFC 2281, Internet Engineering Task Force (IETF), Mar 1998.

[6] J. Moy, "Open Shortest Path Protocol (OSPF)," RFC 2328, Internet Engineering Task Force (IETF), Apr 1998.

[7] M. Goyal, K.K. Ramakrishnan, W.C. Feng, "Achieving faster failure detection in OSPF networks," *Proceeding of IEEE International Conference on Communications*, 2003, vol. 1, May, 2004, pp. 296-300.

[8] Open Specifications for Service Availability, http://www.saforum.org/home/

[9] AdvancedTCA Specifications for Next Generation Telecommunications Equipment, http://www.picmg.org/v2internal/newinitiative.htm.

[10] PICMG (PCI Industrial Computer Manufacturers Group), http://www.picmg.org/

[11] M. Aoki, K. Habara, T. Hamano, K. Ogawa, and S. Chaki, "ATCA-Based Open-Architecture Router Prototype," *IEIEC Transactions on Communication*, May, 2006, pp. 16851687

[12] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, 1959, pp. 269271.

[13] ADLINK Technology Incorporation, http://www.adlink.com.tw/

[14] GNU Zebra, http://www.zebra.org/.

[15] An Integrated Multiprotocol Network Emulator /Simulator(IMNES), http://www.tel.fer.hr/imunes/