JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 26, 2173-2198 (2010)

Design and Implementation of High Availability OSPF Router*

CHIA-TAI TSAI, RONG-HONG JAN AND KUOCHEN WANG
Department of Computer Science
National Chiao Tung University
Hsinchu, 300 Taiwan
E-mail: {tai; rhjan; kwang}@cs.nctu.edu.tw

In this paper, we propose a High Availability Open Shortest Path First (HA-OSPF) router which consists of two OSPF router modules, *active* and *standby*, to support a high availability network. First, we used the continuous-time Markov chain (CTMC) to analyze the steady-state availability of an HA-OSPF router with one active router and N standby routers ($1 + N$ redundancy model). Then, with the failure detection and recovery rate considered, from analytic results, we show that the HA-OSPF router with 1 + 1 redundancy model, one active and one standby, is the preferred model for enhancing router availability. We also show that the carrier-grade HA-OSPF router availability (*i.e.*, five-nine availability) can be achieved under an appropriate combination of the router module failure rate (λ), repair rate (μ), and the failure detection and recovery rate (δ). Since there is a lack of research on the integration of the redundancy model, link state information backup, and failure detection and recovery, we propose a high availability management middleware (HAM middleware) framework to integrate these three elements. The HAM middleware consists of Availability Management Framework (AMF) service, Checkpoint service, and Failure Manager. It supports health check, state information exchange, and failure detection and recovery. Each HA-OSPF router was designed to have a Linux operating system, HAM middleware, and OSPF process. We have implemented the HA-OSPF router on a PC-based system. Experimental results show that the failure detection and recovery times of the proposed PC-based HA-OSPF router were reduced by 98.76% and 91.45% compared to those of an industry standard approach, VRRP (Virtual Router Redundancy Protocol), for a software failure and a hardware failure, respectively. In addition, we have also implemented the HA-OSPF router on an ATCA (Advanced Telecom Computing Architecture) platform, which can provide an industrial standardized modular architecture for an efficient, flexible, and reliable router design. Based on our ATCA-based platform with $1/\delta = 217$ *ms* for a software failure and $1/\delta = 1066$ *ms* for a hardware failure, along with the router module data, $1/\lambda = 7$ years and $1/\mu = 4$ hours, obtained from Cisco, the availabilities of the proposed ATCA-based HA-OSPF router are 99.99999905% for a software failure and 99.99999867% for a hardware failure. Therefore, the experimental results have shown that both our proposed ATCA-based and PC-based HA-OSPF routers with 1 + 1 redundancy model can easily meet the requirement of carrier-grade availabilities with five-nine.

Keywords: ATCA, continues time Markov chain, failure detection and recovery mechanism, high availability, OSPF, redundancy model, router availability

Received October 29, 2008; revised December 10, 2009; accepted January 19, 2010.
Communicated by Chung-Ta King.
* The authors would like to express their appreciation to Professor Kishor S. Trivedi for his insightful comments that help improve the quality of the paper and to Dr. Chien Chen, Dr. Chia-Yuan Huang, Lo-Chuan Hu, and Ching-Chun Kao for their helpful assistance in conducting experiments. This work was supported in part by the National Science Council of Taiwan, R.O.C., under Grants No. NSC 96-2219-E-009-023 and NSC 96-2219-E009-008.

1. INTRODUCTION

With the progress in the broadband network, many people and businesses rely heavily on Internet applications and services. Critical facilities, such as data centers, communication centers, financial trading service centers and telecommunication service centers should ensure a certain degree of operational continuity during the service period. Thus, it is important for a service provider to build a high availability environment to provide continuous services for users, whether to install new components or repair existing components. If a system cannot be accessed, it is said to be unavailable. Generally, the term downtime is used to refer to periods when a network or system is unavailable.

Network availability can be improved either by incremental improvements in component availability or by provision of redundant components in parallel [1, 2]. But, it is costly to implement or use high availability components [3]. Mettas used a nonlinear programming algorithm to formulate a cost function [4], which is an exponential behavior and a monotonically increasing function of the component availability. Unfortunately, the cost function shows that the more difficult it is to improve the reliability of the router, the greater the cost [4]. Depending on the design complexity, technological limitations, and so on, the availability of certain components can be very hard to improve [4].

Therefore, adding redundant routers to a network router to achieve the goal of high availability is a familiar design [5-12]. In general, this approach consists of a cluster of routers where one is the active router and the others are on standby. That is, the active router executes the routing process, while a standby router is prepared to take over the active router's role immediately if the active router failed. For establishing network router redundancy, VRRP (Virtual Router Redundancy Protocol) [5] and HSRP (Hot Standby Router Protocol) [6] are two most familiar designs. VRRP is a non-proprietary redundancy protocol described in RFC 3768 [5] and HSRP is a Cisco proprietary redundancy protocol described in RFC 2281 [6]. VRRP is based on Cisco's proprietary HSRP concepts. These two technologies are similar in concept, but not compatible.

The increased availability of VRRP is achieved by advertising a "*virtual router*," which is an abstract object managed by VRRP that acts as a default router for hosts on a shared LAN [5]. The main purpose of the virtual router is that the hosts on the LAN are configured to forward packets to the virtual IP address, rather than to the IP address of the real interface. In VRRP, two or more physical routers are then configured to stand for the virtual router, with only one doing the actual routing at any given time. A standby router, also from the group of routers, monitors the status of the active router so that if the active router becomes inoperative, the standby router automatically begins emulating the virtual router. The host is configured to point to the virtual address so that the packets it sends out of its LAN are always directed to the virtual router which may be any router from the group of routers. If the standby router becomes inoperative or takes over for the active router's role, other routers in the group hold an election to determine which of them should take over for the standby router. In this way, the hardware availability can be improved significantly. Note that the concept of the virtual router can also be applied to a server cluster to achieve load balancing [13].

One issue deserved to mention is that a lack of link state information in VRRP, the standby router cannot recover the routing protocol session in real time if it takes over. The standby router needs to generate link state exchange messages with its neighbor

routers and to obtain the up-to-date link states of the network. Before the completion of the link state coherence, the standby router cannot take over the role of the active router. To reduce the takeover delay, stateful takeover can be used to decrease the time of link state coherence and to improve the router availability. Ho *et al.* [12] proposed a router and routing protocol redundancy model to reduce service outage or degradation for a network router and thus to increase service availability on a network due to software and hardware failures of the network router [12]. The active router generates or receives the routing protocol state change and replicates it to the standby router. Because of the replica of the routing protocol state, the standby router can recover and maintain the routing protocol sessions for network devices immediately if a failure occurs in the active router. Furthermore, the routing protocol states are maintained by the standby router in real-time to handle the dynamic changes created by routing protocols [12]. Because the standby router can reconstruct the routing information from the routing protocol states if it takes over, this model results in significantly less network disconnection time. However, the work by Ho *et al*. did not mention the takeover delay of their proposed router and the improvement of the router availability.

Wang *et al*. proposed an open programmable router [15] that is compliant to the ForCES (Forwarding and Control Element Separation) architecture [14]. However, it does not have the high availability feature. Then, Wu *et al*. proposed a pseudo-VRRP router with high availability [16]. The pseudo-VRRP router is a VRRP-based router that follows the ForCES architecture [14], which consists of several Control Elements (CEs), among which one acts as an active controller and the others as backups, and multiple Forwarding Elements (FEs). When the active CE fails, a backup CE can recover the routing states and management states, and take over the role of the active CE. However, this paper did not consider the situation of software failures.

Nevertheless, there is a lack of research on the integration of redundancy model, link state information backup, and failure detection and recovery. Thus, in this paper, we propose a high availability management (HAM) middleware which consists of Availability Management Framework (AMF) service [17], Checkpoint service [17], and Failure Manager. The redundancy model and link state information backup can be provided by the AMF service and Checkpoint service, respectively. The Failure Manager can provide procedures to achieve the goal of fast failure detection and recovery. The HAM middleware can provide a complete integration for decreasing network disconnection time and improving network availability effectively. We also implemented the HAM middleware on a network router, called High Availability (HA) router, and verified that HAM middleware can decrease the network disconnection time effectively.

In this paper, the Open Shortest Path First (OSPF) routing protocol and the broadcast network (*i.e*., Ethernet) are used to verify the correctness of the proposed HA router. The HA-OSPF router is used to represent an HA router with OSPF routing protocol. OSPF, which is a link state routing protocol [17], 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. By this way, a router learns the network topology and the cost of each link. Then, each router can determine the shortest path to each destination. When a link failure occurs, based on our measurements, the OSPF protocol may take forty to fifty seconds to detect the failure[1] and rebuild the

[1] The default setting of *Router Dead Interval* is 40 seconds (or 4 *Hello Intervals*).

routing information. Such a long delay is unacceptable for an access network that many businesses heavily rely on.

In order to maximize the availability of the HA router, we want to find the relationship between degree of redundancy and availability. The continuous-time Markov chain (CTMC for short) [21-23] was used to Analyze the steady-state availability of the HA router with a different number of standby routers. With failure detection and recovery rate considered, in our study, we used CTMC to show that the availability does not always increase with more redundancy [20, 24]. Numerical results have shown that the availability of only one standby router (*i.e.* 1 + 1 redundancy model) under an appropriate failure detection and recovery rate can provide enough availability compared to that of more than one standby router. We will show that the failure detection and recovery rate is a key factor to increase the availability in section 3.

Based on the 1 + 1 redundancy model, we have implemented the HAM middleware which will save the OSPF process status and link state database (LSDB) information of the active router and replicates them to the standby router. The standby router can take over the active router's role and recover the process' status if the active router fails. We will show that the proposed HA-OSPF router can improve the failure detection and recovery time effectively in sections 5 and 6.

Finally, experimental results show that the standby router could takeover within 166 *ms* (*i.e.*, the summation of failure detection time and failure recovery time) if a software failure occurred (*e.g.*, OSPF process failed). Moreover, it took 1240 *ms* to takeover if a hardware failure occurred (*e.g.*, power down). The failure detection and recovery times of the proposed PC-based HA-OSPF router were reduced by 98.76% and 91.45% compared to those of an industry standard approach, VRRP, for a software failure and a hardware failure, respectively. The failure detection and recovery time for a hardware failure can be further reduced to 360 *ms* by decreasing the down check interval to 200 *ms*. 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 the active router has failed. We also implemented the HA-OSPF router on the ATCA (Advanced Telecom Computing Architecture) platform [35] which can provide high performance, high availability, adaptability for adding new features, and low cost of ownership. Based on our ATCA-based platform with $1/\delta = 217$ *ms* for a software failure and $1/\delta = 1066$ *ms* for a hardware failure, which were obtained from our experimental results of the proposed ATCA-based HA-OSPF router, along with the router module data, $1/\lambda = 7$ years and $1/\mu = 4$ hours, obtained from Cisco, the availabilities of the proposed ATCA-based HA- OSPF router are 99.99999905% for a software failure and 99.99999867% for a hardware failure. Therefore, the experimental results have shown that both our proposed ATCA-based and PC-based HA-OSPF routers with 1 + 1 redundancy can have carrier-grade availabilities with five-nine.

In addition, the experimental results have also demonstrated that the failure detection and recovery time plays an important role for decreasing the router down time and thus increasing router availability. That is, the proposed HA-OSPF router has shorter failure detection and recovery time such that it can meet the requirement of carrier-grade availability with five-nine and the overhead of the CPU usage increases only slightly.

The rest of this paper is organized as follows. Section 2 describes the definition of reliability and availability. In section 3, we present the HA router availability under a dif-

ferent number of standby routers by using the continuous-time Markov chain. Then, the components and operations of the proposed HA-OSPF router will be explored in section 4. Laboratory (PC-based) test results and field trial (ATCA-based) test results are given in sections 5 and 6, respectively. Finally, we conclude this paper in section 7.

2. BACKGROUND

The International Telecommunications Union-Telecommunication Standardization Sector (ITU-T) gives the definition of the reliability and availability in the recommendation E.800. In this section, we will introduce the relationship between failure rate, repair rate, failure detection and recovery rate, and availability.

2.1 Reliability Definitions

Recommendation E.800 of the ITU-T defines reliability as the "*ability of an item to perform a required function under given conditions for a given time interval* [19]." Therefore, for any time interval $T = (s, s + t)$, the system will work properly during the interval (*i.e.*, the reliability $R(t) = 1$, where $t \in T$ and $R(t \mid s) = 1$). Generally, the system is assumed to be working properly at time $t = 0$ (*i.e.*, $R(0) = 1$), and no system can work forever without failures (*i.e.*, $\lim_{t\to\infty} R(t) = 0$) [21].

Let random variable X be the lifetime (*i.e.*, time to failure) [21] of a system then

$$R(t) = \Pr(X > t) = 1 - F(t), \tag{1}$$

where $F(t)$ is the system lifetime CDF (cumulative distribution function) [21]. Moreover, the expected lifetime ($E[X]$) or the mean time to failure of the component is given by [21] is

$$MTTF = E[X] = \int_0^\infty R(t)dt. \tag{2}$$

Therefore, the system MTTF can be computed from the Eqs. (1) and (2). Suppose the system lifetime is exponentially distributed (*i.e.*, $F(t) = 1 - e^{-\lambda t}$) [21] with failure rate λ then

$$R(t) = 1 - (1 - e^{-\lambda t}) = e^{-\lambda t} \tag{3}$$

and

$$MTTF = E[X] = \int_0^\infty e^{-\lambda t} dt = \frac{1}{\lambda}. \tag{4}$$

Therefore, if a component obeys an exponential failure rate with parameter λ, then the MTTF (*i.e.*, the expected lifetime [21]) can be determined as $1/\lambda$.

2.2 Availability Definitions

ITU-T Recommendation E.800 given the definition of availability as the "*ability of an item to be in a state to perform a required function at a given instant of time or at any*

instant of time within a given time interval, a*ssuming that the external resources*, *if required*, *are provided* [19]." Michael *et al*. [20] identified the difference between reliability and availability such is that reliability refers to failure-free operation of the system during an interval, while availability refers to failure-free operation of the system at a given instant of time.

Let random variable $I(t)$ be an indicator of a system. Then, if $I(t) = 1$, it means the component is up and 0 otherwise. Then, we suppose $A(t)$ is the instantaneous availability of the system. That is, $A(t)$ is the probability of the system which is properly working at specified time t, *i.e*.,

$$A(t) = \Pr(I(t) = 1) = E[I(t)]. \tag{5}$$

Based on the instantaneous availability, the steady state availability, A, can be defined as

$$A = \lim_{t\to\infty} A(t). \tag{6}$$

The steady-state availability means the probability of the system is still available over a long period. Under certain conditions, for instance, constant failure rate and constant repair rate, the steady-state availability can be expressed as [21]:

$$A = \frac{MTTF}{MTTF + MTTR} \tag{7}$$

where MTTR (mean time to repair) is the amount of time required to perform corrective maintenance and restore a component or system to operational status. The MTTR including any time required to detect that there is a failure, to repair it, and to place the system back into an operational status.

If the system lifetime is exponential with failure rate λ, and the time-to-repair distribution of the system is exponential with repair rate μ, then Eq. (7) can be rewritten as [21]

$$A = \frac{\mu}{\lambda + \mu}. \tag{8}$$

3. HA ROUTER MODEL DESCRIPTION AND ANALYSIS

In this section, the continuous-time Markov chain (CTMC) was used to determine the steady-state availability of the proposed HA router. We will show that the availability of the HA router does not always increase with more redundancy. We will also show that, the 1 + 1 redundancy model is the recommended framework for increasing the availability under an appropriate failure and recovery rate.

3.1 Continuous-Time Markov Chain for 1 + 1 Redundancy Model

Fig. 1 is the state-transition diagram of a CTMC [21-23] modeling the failure and repair behavior of an HA router with 1 + 1 redundancy model (*i.e*. one active and one

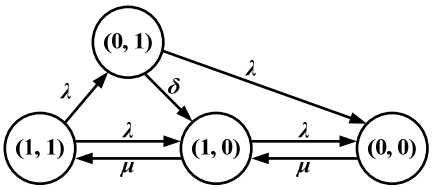


Fig. 1. CTMC for HA router with one active router and one standby router.

standby). Normally, the standby router is dedicated to take over the role of the active router if the active router failed. The failure of the active router will cause the network to recalculate routing path information. To avoid this undesirable situation, if the active router failed, the standby router takes over the role automatically.

As shown in Fig. 1, state (i, j) represents the status of the HA router, where i and j represent the status of active and standby routers, respectively. If i (or j) equal to 1 means the active (or standby) router is working and 0 otherwise. If both i and j equal to 1, it means both the active and standby routers of the HA router are working. If i equal 0 and j equal to 1, it represents the failure of the active router and if i equal to 1 and j equal to 0, it represents the failure of the standby router. Finally, if both i and j equal to 0, it means the two routers of the HA router are failed.

In this paper, the time to failure and time to repair of a router module are assumed to be exponentially distributed with mean $1/\lambda$ and $1/\mu$, respectively. In Fig. 1, when the state transfers from (0, 1) to (1, 0) which indicates that a failure has been detected and recovered, and the standby router has taken over the role of the active router. The associated failure detection and recovery rate (δ) is the multiplicative inverse of the mean time that from the active router failed to the standby router detecting that the failure had occurred and being recovered from it. During the time, if the failure of the active router is being detected, but the standby router is failed (at rate λ), the HA router in state (0, 0) is assumed to be failed. Such a fault has been called a *near-coincident fault* [21]. The HA router fails (*i.e.*, cannot forward packets) when the state is either (0, 1) or (0, 0). Note that in this paper, all failure events are assumed to be mutually independent.

Let $\pi_{(i,j)}$ denotes the proportion of time that the system is in state (i, j). Note that in the steady state the rate at which transitions into state (i, j) must equal to the rate at which transitions out of state (i, j). Thus, from Fig. 1, we have

$$2 \cdot \lambda\pi_{(1,1)} = \mu \cdot \pi_{(1,0)}, \tag{9}$$

$$(\lambda + \delta) \cdot \pi_{(0,1)} = \lambda \cdot \pi_{(1,1)}, \tag{10}$$

$$(\lambda + \mu) \cdot \pi_{(1,0)} = \delta \cdot \pi_{(0,1)} + \lambda \cdot \pi_{(1,1)} + \mu \cdot \pi_{(0,0)}, \tag{11}$$

$$\mu \cdot \pi_{(0,0)} = \lambda \cdot \pi_{(1,0)} + \lambda \cdot \pi_{(0,1)}. \tag{12}$$

By solving the preceding set of equations, along with this equation

$$\pi_{(1,1)} + \pi_{(1,0)} + \pi_{(0,1)} + \pi_{(0,0)} = 1. \tag{13}$$

We obtain

$$\pi_{(1,1)} = \frac{\mu^2(\lambda+\delta)}{\mu^2(\lambda+\delta) + \lambda\mu^2 + 2\lambda\mu(\lambda+\delta) + \lambda^2(2(\lambda+\delta)+\mu)}, \tag{14}$$

$$\pi_{(0,1)} = \frac{\lambda\mu^2}{\mu^2(\lambda+\delta)+\lambda\mu^2+2\lambda\mu(\lambda+\delta)+\lambda^2(2(\lambda+\delta)+\mu)}, \tag{15}$$

$$\pi_{(1,0)} = \frac{2\lambda\mu(\lambda+\delta)}{\mu^2(\lambda+\delta)+\lambda\mu^2+2\lambda\mu(\lambda+\delta)+\lambda^2(2(\lambda+\delta)+\mu)}, \tag{16}$$

$$\pi_{(0,0)} = \frac{\lambda^2(2(\lambda+\delta)+\mu)}{\mu^2(\lambda+\delta)+\lambda\mu^2+2\lambda\mu(\lambda+\delta)+\lambda^2(2(\lambda+\delta)+\mu)}. \tag{17}$$

Thus, the availability (A_{HA}) of an HA router can be determined based on $\pi_{(1,1)} + \pi_{(1,0)}$. Then, the equivalent failure rate (λ_{HA}) and the equivalent repair rate (μ_{HA}) of an HA router can be determined by applying the aggregation techniques described in [24, 25]. Therefore, we obtain

$$\lambda_{HA} = \frac{\lambda\cdot\pi_{(1,1)}+\lambda\cdot\pi_{(1,0)}}{\pi_{(1,1)}+\pi_{(1,0)}} = \lambda, \tag{18}$$

$$\mu_{HA} = \frac{\delta\cdot\pi_{(0,1)}+\mu\cdot\pi_{(0,0)}}{\pi_{(0,1)}+\pi_{(0,0)}}. \tag{19}$$

Therefore, the availability of an HA router can be expressed as follows:

$$A_{HA} = \frac{\mu_{HA}}{\lambda_{HA}+\mu_{HA}}. \tag{20}$$

Table 1 shows numerical results of the equivalent repair rate (μ_{HA}) and the availability (A_{HA}) of an HA router with typical settings for the router module parameter values: $1/\lambda$ = 7 years (from Cisco), $1/\mu$ = 4 hours (from Cisco) [26-28], and δ = 3377 times/hour and 16590 times/hour (experimental values obtained from section 6) for a hardware failure and a software failure, respectively. Note that we turned off the power and killed the routing process to generate a hardware failure and a software failure, respectively. From Table 1, we found that the equivalent repair rate (μ_{HA}) does increase when the failure detection and recovery rate (δ) increases. We also found that the availability of an HA router (A_{HA}) does increase if the equivalent repair rate (μ_{HA}) increases. Thus, we conclude that the failure detection and recovery rate (δ) is a key parameter for increasing the availability of an HA router with 1 + 1 redundancy model. Remind that $\lambda_{HA} = \lambda$.

Table 1. The equivalent repair rate (μ_{HA}) and the availability (A_{HA}) of an HA router for different failure detection and recovery rates (δ) under $1/\lambda$ = 7 years and $1/\mu$ = 4 hours [26-28].

		Equivalent repair rate of an HA router (μ_{HA})	Availability of an HA router (A_{HA})
Failure detection and recovery rate (times/hour)	δ = 3377	1223 (times/hour)	99.99999866%
	δ = 16590	1718 (times/hour)	99.99999905%

3.2 Continuous-Time Markov Chain for 1 + *N* Redundancy Model

In this section, the continuous-time Markov chain (CTMC) of an HA router with 1 + *N* redundancy model (*i.e.*, one active router and *N* standby routers) is considered. Each standby router monitors the status of the active router. If a failure occurred in the active router, the standby routers hold an election automatically. Then, one of the standby routers will take over the role of the active router.

The state diagram of the CTMC modeling the failure and repair behavior of an HA router with 1 + *N* redundancy model is depicted in Fig. 2. The active router works properly at state (1, *p*), where $0 \le p \le N$. The state (0, *q*) represents the active router failed and the system detects and recovers the failure with rate δ and will go to state (1, $q - 1$), where $1 \le q \le N$. The state (0, 0) represents that all the router modules of the HA router are failed.

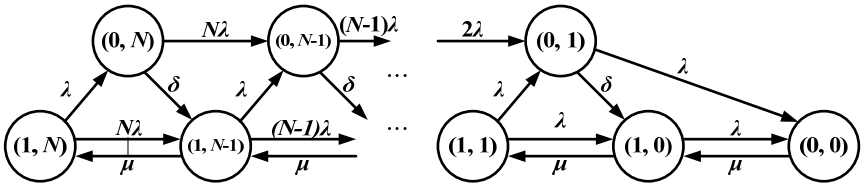


Fig. 2. CTMC for an HA router with 1 + *N* redundancy model.

After writing the steady state equations and solving these equations, we obtain the following equations for the steady state probabilities:

$$(N+1)\lambda \cdot \pi_{(1,N)} = \mu \cdot \pi_{(1,N-1)}, \tag{21}$$

$$(N\lambda + \delta) \cdot \pi_{(0,N)} = \lambda \cdot \pi_{(1,N)}, \tag{22}$$

$$\mu \cdot \pi_{(0,0)} = \lambda \cdot \pi_{(0,1)} + \lambda \cdot \pi_{(1,0)}, \tag{23}$$

$$(\lambda + \mu) \cdot \pi_{(1,0)} = \lambda \cdot \pi_{(1,1)} + \mu \cdot \pi_{(0,0)} + \delta \cdot \pi_{(0,1)}, \tag{24}$$

$$(K\lambda + \delta) \cdot \pi_{(0,K)} = \lambda \cdot \pi_{(1,K)} + (K+1)\lambda \cdot \pi_{(0,K+1)}, \text{ where } N - 1 \le K \le 1, \tag{25}$$

$$(K\lambda + \mu) \cdot \pi_{(1,K)} = (K+1)\lambda \cdot \pi_{(1,K+1)} + \mu \cdot \pi_{(1,K-1)} + \delta\pi_{(0,K+1)}, \text{ where } N - 1 \le K \le 1, \tag{26}$$

$$\sum_{i=0}^{1}\sum_{j=0}^{N} \pi_{(i,j)} = 1. \tag{27}$$

The equivalent failure rate (λ_{HA}) and the equivalent repair rate (μ_{HA}) of an HA router for the CTMC in Fig. 2 can be expressed as follows:

$$\lambda_{HA} = \frac{\lambda \cdot \pi_{(1,N)} + \lambda \cdot \pi_{(1,N-1)} + \ldots + \lambda \cdot \pi_{(1,1)} + \lambda \cdot \pi_{(1,0)}}{\pi_{(1,N)} + \pi_{(1,N-1)} + \ldots + \pi_{(1,1)} + \pi_{(1,0)}} = \frac{\lambda \cdot \left(\sum_{j=1}^{N} \pi_{(1,j)} \right)}{\sum_{j=0}^{N} \pi_{(1,j)}} = \lambda. \tag{28}$$

$$\mu_{HA} = \frac{\delta \cdot \pi_{(0,N)} + \delta \cdot \pi_{(0,N-1)} + \ldots + \delta \cdot \pi_{(0,1)} + \mu \cdot \pi_{(0,0)}}{\pi_{(0,N)} + \pi_{(0,N-1)} + \ldots + \pi_{(0,1)} + \pi_{(0,0)}} = \frac{\delta \cdot \left(\sum_{j=1}^{N} \pi_{(0,j)} \right) + \mu \cdot \pi_{(0,0)}}{\sum_{j=0}^{N} \pi_{(0,j)}}. \quad (29)$$

Solving the above two Eqs. (28) and (29), we can get the availabilities of an HA router by using Eq. (20) under various failure detection and recovery rates, and a different number N of standby routers, as shown in Table 2. Based on Cisco data, we set the parameter values for $1/\lambda$ and $1/\mu$ to 7 years and 4 hours [26-28], respectively. Note that the objective of this paper is to find the most cost-effective redundancy model for the HA router such that its availability meets the requirement of the carrier-grade availability (99.999%). From Table 2, an HA router with 1 + 1 redundancy (*i.e.*, $N = 1$) will meet the five-nine availability if δ is greater than 10 times/hour. In general, δ is much larger than 10. For example, in Table 4, the δ for the VRRP router is at least 248 times/hour and the δ for the proposed HA-OSPF router is at least 2903 times/hour. For a commercial router, such as a Cisco ASR 1000 Series router, its δ is 1800 times/hour [38]. Thus, we conclude that an HA router with 1 + 1 redundancy is preferred, which will meet the five-nine availability.

In addition, we also found that the failure detection and recovery rate (δ) is a key parameter to improve the availability of an HA router. To have high availability, δ is the larger the better. Note that, for an HA router with 1 + 1 redundancy, to obtain five-nine availability, the minimum δ is 1.632 times/hour for $1/\lambda = 7$ years and $1/\mu = 4$ hours [26-28]. In sections 5 and 6, we will show that the experimental δ's for a PC-based and an ATCA-based HA routers with 1 + 1 redundancy are 2903 times/hour and 3377 times/hour for hardware failures, respectively, which are much higher than the minimum δ we just mentioned. For software failures, the experimental δ's are even larger.

Table 2. The availability of an HA router (A_{HA}) for a different number of standby routers and various failure detection and recovery rates under $1/\lambda$ = 7 years and $1/\mu$ = 4 hours [26-28].

	Failure detection and recovery rate (δ) (times/hour)			
	$\delta = 1$	$\delta = 10$	$\delta = 100$	$\delta = 1000$
$N = 0$	99.99347727%	99.99347727%	99.99347727%	99.99347727%
$N = 1$	99.99836852%	99.99983608%	99.99998284%	99.99999752%
$N = 2$	99.99836921%	99.99983692%	99.99998369%	99.99999837%
$N = 4$	99.99836921%	99.99983692%	99.99998369%	99.99999837%
$N = 8$	99.99836921%	99.99983692%	99.99998369%	99.99999837%

4. PROPOSED HIGH AVAILABILITY MANAGEMENT (HAM) MIDDLEWARE

In order to decrease the network disconnection time, we propose a High Availability Management (HAM) Middleware to achieve the goal of high availability. In this section, we are going to discuss the software architecture and procedures of the proposed scheme.

4.1 Software Architecture

Fig. 3 shows the software architecture for each router module in the HA-OSPF router. Each router module runs a Linux operating system, HAM middleware, and OSPF process. The HAM middleware consists of a Failure Manager and OpenAIS middleware [29]. We have developed the Failure Manager on top of the open-source OpenAIS middleware to realize the HA-OSPF router. Thus, the HAM middleware is an enhancement to the OpenAIS middleware. It provides an OSPF process with high availability.

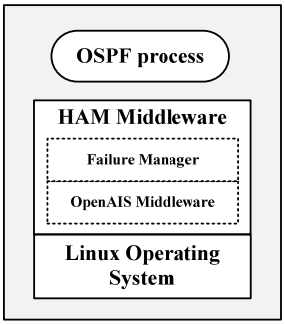


Fig. 3. Software architecture for each router module in the HA-OSPF router.

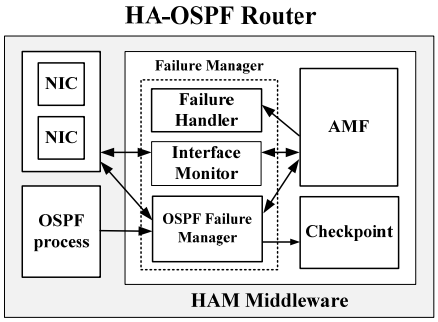


Fig. 4. The components of the HA-OSPF router.

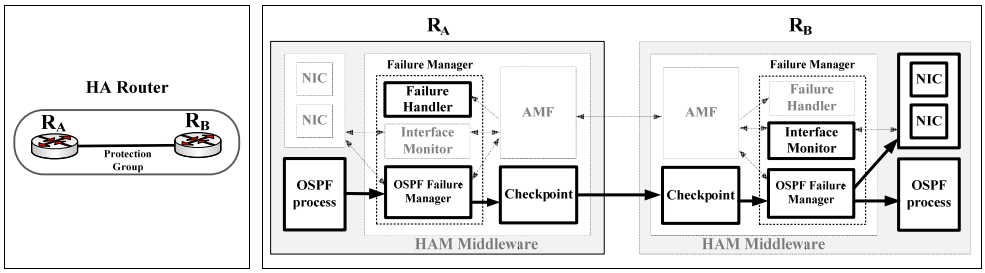
The HAM middleware consists of an AMF service, Checkpoint service, and Failure Manager, as shown in Fig. 4. The AMF service is part of the OpenAIS middleware. We implemented our software (*i.e.*, the Failure Manager) on top of the OpenAIS middleware. The reason why we choose the OpenAIS middleware is that it is an open source based on the Service Availability (SA) Forum Application Interface Specification [39]. It provides a set of APIs for high availability applications. The AMF service can provide the health check and role assignment services. Using the OpenAIS middleware, it is easy to implement the failure handler, interface monitor, and OSPF failure manager. The functions of these components are described in the following:

- AMF service: The AMF service of the active router sends a heartbeat message to the standby router(s) periodically to report its health. If the standby router does not hear the heartbeat message from the active router within a down check interval (*e.g.*, 1 second), it will assume the active router has failed and the AMF service will find a router from the standby router to take over the role of the failed active router.
- Checkpoint service: It provides state information exchange service between active and standby routers. Through the Checkpoint service, the active router can save its OSPF's process status and link state database (LSDB) information to the standby router. The state information can help the standby router reduce the recovery latency and improve the availability.
- Failure Manager: It includes OSPF Failure Manager, Interface Monitor, and Failure Handler, also shown in Fig. 4. The OSPF Failure Manager takes care of the OSPF process operations, informs the AMF service if a failure in the OSPF process is detected,

and collects the OSPF process status and LSDB information for the checkpoint service. The Interface Monitor checks the health of the network interface cards (NICs) and informs the AMF service if any NIC failure occurs. The Failure Handler has a set of callback functions. When the AMF service notifies the Failure Handler that a failure has occurred it will execute a predefined callback function to handle the failure. The callback function will reinitialize the failed process or device if the failure can be determined by the Failure Manager (*e.g.*, the OSPF process or NIC failed). But, if the failure (*e.g.*, AMF service failed or HA-OSPF router halted) cannot be determined by the Failure Manager, the callback function will reboot the failed router.

4.2 OSPF State Information Backup

When a router is first started, it executes the role assignment operation and receives a role (active or standby). After all routers of the HA-OSPF router with 1 + 1 redundancy determine their roles, the HA-OSPF router will form a protection group as shown in Fig. 5 (a), where the protection group includes an active router and a standby router that run their OSPF processes. The active router enables its NICs to transmit and receive messages from the network while the standby router disables its NICs.



(a) Logical structure. (b) Protection group.

Fig. 5. OSPF state information backup for a protection group.

Then, the active router sends a *Hello* message to its neighbor routers on the network periodically to test the connectivity of neighbors. After the active router finds out the state of a link to its neighbor (up or down), it broadcasts the link state information to every router. Similarly, the active router can receive link state advertisements from other routers. Thus, a LSDB, which represents the network topology, can be constructed by the active router. Based on the LSDB, the active router calculates the routing path by the shortest path algorithm [30] and updates its routing table.

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

Fig. 5 (b) shows how state information flows from the active router's OSPF process to the standby router's OSPF process in the protection group. As shown in Fig. 5 (b), the OSPF process in the active router first passes state information to the OSPF Failure Man-

ager by the shared memory. The OSPF Failure Manager next asks the Checkpoint service to send state information to the standby router. Through the Checkpoint service, the standby router receives the state information and passes it to the OSPF Failure Manager. Finally, the OSPF process in the standby router saves the received state information.

4.3 Failure Detection and Recovery

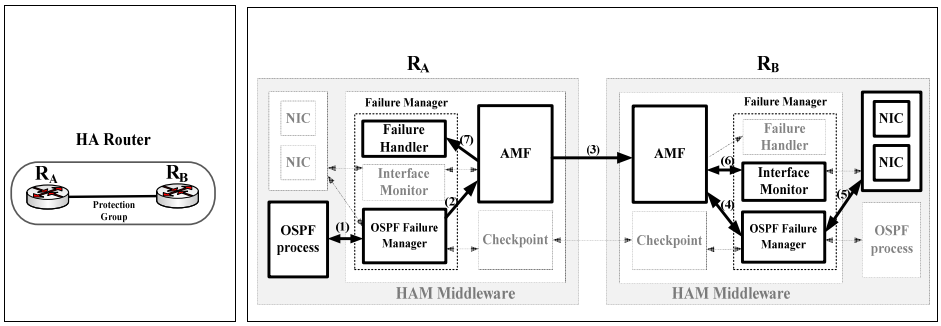
In the following we describe the failure detection and recovery mechanisms for OSPF processes, NICs, and router module failures for a protection group. These failure detection and recovery mechanisms are the same for each protection group. When an HA-OSPF router starts, the OSPF Failure Manager and Interface Monitor in each router register themselves at the AMF and register their callback functions at the Failure Handler. So, if the OSPF Failure Manager or Interface Monitor informs the AMF that a failure occurred, the AMF can ask the Failure Handler to perform the corresponding callback function.

Again, Fig. 6 (a) shows an HA-OSPF router with 1 + 1 redundancy model. Fig. 6 (b) illustrates the failure detection and recovery procedure for the OSPF process of the protection group where numbers 1-7 show the sequence of steps in the procedure. The operation of each step is described as follows:

(1) The OSPF Failure Manager in router R_A polls the status of the OSPF process periodically, *e.g.*, *t* sec. If a failure occurred in the OSPF process, the OSPF Failure Manager can detect it within *t* sec.
(2) If a failure in the OSPF process is detected, the OSPF Failure Manager sends an error report that indicates an OSPF process failure to the AMF service.
(3) After receiving the error report, the AMF service of router R_A generates an error message to router R_B's AMF service.
(4) When the AMF service of router R_B receives the error message, it will take over and change its role as the active router. After that, router R_B's AMF service starts the OSPF Failure Manager.
(5) Note that the OSPF NICs were disabled when router R_B was on standby. As soon as router R_B takes over, the OSPF Failure Manager of router R_B will enable the NICs. For a broadcast network (*e.g.*, Ethernet), R_B will send a *gratuitous ARP* [31] message to the network. The *gratuitous ARP* message is used to ask its neighbors to bind its MAC address to the active router's IP address.
(6) The AMF of router R_B activates the Interface Monitor.
(7) On the other side, the AMF service of router R_A executes a callback function which was defined in the Failure Handler to handle the failure. The callback function will reinitialize the OSPF process and then router R_A will become the standby router if it functions correctly.

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

Next, consider the case of a failure in the router but not in the OSPF process or NICs. For example, there can be a failure in HAM middleware components, such as the active



(a) Logical structure. (b) Protection group.

Fig. 6. Failure detection and recovery procedure for the OSPF process in the protection group, where numbers 1-7 show the sequence of steps in the procedure.

router's AMF service failed. This type of failures can be detected and the system can be recovered as follows. The AMF service of the active router periodically announces its presence with a heartbeat message. If the standby router fails to receive the heartbeat message for some period of time (*i.e.*, *down check interval*), it will take over the functionality and then change its role as the active router. Next, router R_A's AMF service starts the OSPF Failure Manager. Finally, the AMF performs steps 5 and 6 as described above.

5. EXPERIMENTAL RESULTS

In order to evaluate the performance of the HA-OSPF router with 1 + 1 redundancy, an important parameter to measure is the failure detection and recovery time (or called takeover delay). The failure detection and recovery time is defined as the latency from the active router of the HA-OSPF router failed to the standby router of the HA-OSPF router taking over and recovering from the failure. In the following experiment, we want to show that the proposed HA-OSPF router can meet the requirement of carrier-grade availability with five-nine under an appropriate failure detection and recovery time.

To implement the HA-OSPF router with 1 + 1 redundancy, two desktop PCs with Intel Pentium 4 3.0 GHz processors and 512 MB memories connected via Ethernet are used to emulate a PC-based HA-OSPF router. That is, the PC-based HA-OSPF router consists of routers R2 and R3, as shown in Fig. 7. A Linux operating system and GNU Zebra [32] were selected as the developing platform for the PC-based HA-OSPF router. The GNU Zebra is a well-known and free software that manages the TCP/IP based routing protocol. 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. A heartbeat message is exchanged between interface eth2s of R2 and R3, periodically. We assign IP addresses 10.0.0.1 and 10.0.0.2 to interface eth2s of R2 and R3, respectively. There are two additional OSPF routers (R1 and R4), two clients (S1 and S2), and one log server 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 connectivity (see Fig. 7). A log server was constructed to record the sequence number and timestamp of each packet that

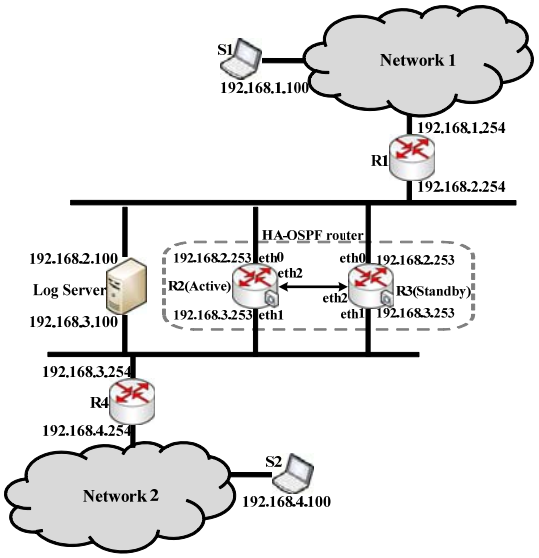


Fig. 7. Experimental environment (PC-based).

Table 3. Default parameter values [17, 18].

Router dead interval of OSPF	40 *sec.*
Hello interval	10 *sec.*
Down check interval of AMF service	1000 *ms*
Polling interval of Fault Manager	100 *ms*

it received. If S1 sent a packet to S2, it had to also send a copy of the packet to the log server. That is, S2 forwarded the packet that it 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 packets from S2. By this way, the network disconnection time can be determined. The default parameter values of the OSPF routing protocol and HAM middleware are listed in Table 3 [17, 18].

First, we investigate how the failure detection and recovery time (*i.e.*, takeover delay) is affected by the information backup of the standby router. Two cases were implemented and evaluated as follows:

- *VRRP-based router with* 1 + 1 *hardware redundancy*: the active router does not save any state information in the standby router.
- *Proposed HA-OSPF router with* 1 + 1 *redundancy*: 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 failures were considered. One is when R2 halts by an unexpected power down (referred as a hardware failure), and the other is when an OSPF process failed (referred to as a software failure). First, UDP packets traveled along path S2, R4, R2, R1, S1 until the active router failed. After R4, R3, and R1 reestablished their

Table 4. Takeover delays (*ms*), failure detection and recovery rates (times/hour), and availabilities (%) for PC-based HA-OSPF router and VRRP-based router.

		Emulation Scenario	
		VRRP	PC-based HA-OSPF router
Hardware failure	Takeover delay (ms)	14511 ± 36	1240 ± 12
	Failure detection and recovery rate (times/hour)	248	2903
	Availability	99.99999257%	99.99999859%
Software failure	Takeover delay (ms)	13383 ± 3	166 ± 9
	Failure detection and recovery rate (times/hour)	269	21687
	Availability	99.99999309%	99.99999907%

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 [33, 34] to estimate the takeover delays for these two cases.

The average takeover delays for the two cases are shown in Table 4. From Table 4, the takeover delay for a hardware failure (a software failure) of the VRRP-based router and the PC-based HA-OSPF router are 14511 ± 36 *ms* and 1240 ± 12 *ms* (13383 ± 3 *ms* and 166 ± 9 *ms*), respectively. Experimental results show that the takeover delays (*i.e.*, failure detection and recovery times) of the proposed PC-based HA-OSPF router were reduced by 98.76% and 91.45% compared to those of VRRP for a software failure and a hardware failure, respectively. The proposed PC-based HA-OSPF router with full state information backup shows its benefits. In the following, we will show that the network with the proposed PC-based HA-OSPF router will have higher availability than the network with the VRRP-based router.

Table 4 also shows numerical results of availabilities (A_{HA}) for the proposed PC-based HA-OSPF router with 1 + 1 redundancy and VRRP-based router with 1 + 1 redundancy under $1/\lambda$ = 7 years and $1/\mu$ = 4 hours [26-28]. The availabilities (A_{HA}) of the proposed PC-based HA-OSPF router with 1 + 1 redundancy are 99.9999859% and 99.9999907% for a hardware failure and a software failure, respectively, which are higher than those of the VRRP-based router with 1 + 1 redundancy. Since the failure detection and recovery rates (δ) of the PC-based HA-OSPF router with 1 + 1 redundancy are 2903 times/hour for a hardware failure and 21687 times/hour for a software failure, which are much higher than 1.632 times/hour, the minimum required δ to obtain five-nine availability (refer to section 3), the proposed PC-based HA-OSPF router with 1 + 1 redundancy can easily meet the requirement of carrier-grade availability with five-nine.

On the other hand, we learned from Table 5 that the CPU usages for the VRRP-base and the PC-based HA-OSPOF router were 0.17 ± 0.03% and 4.47 ± 0.73%. Note that the proposed PC-based HA-OSPF router did not use a large percent of CPU time.

Next, we measured the takeover delay for the PC-based HA-OSPF router due to a software failure under various polling intervals. As addressed above, the takeover delay consists of failure detection time and failure recovery time. Experimental results show that the failure detection time depends on the polling intervals. On average, the failure detection time was half that of the polling intervals. Table 6 shows the takeover delays (failure detection and recovery rates) due to a software failure, 121 ± 5 *ms* (δ = 29752 times/hour),

Table 5. CPU usages of HAM middleware and OSPF process for PC-based HA-OSPF router and VRRP-based router.

	Emulation Scenario	
	VRRP	PC-based HA-OSPF router
CPU Usage	0.17 ± 0.03 %	4.47 ± 0.73 %

Table 6. Takeover delays (*ms*), failure detection and recovery rates (times/hour), and availabilities due to a software failure (OSPF process down) under various polling intervals.

($1/\lambda$ = 7 years, $1/\mu$ = 4 hours)	Polling interval		
	50 *ms*	100 *ms*	200 *ms*
Takeover delay (ms)	121 ± 5	166 ± 9	222 ± 23
Failure detection and recovery rate (times/hour)	29752	21687	16216
Availability (A_{HA})	99.99999909%	99.99999907%	99.99999905%

Table 7. Takeover delays (*ms*), failure detection and recovery rates (times/hour), and availabilities due to a hardware failure under various down check intervals.

($1/\lambda$ = 7 years, $1/\mu$ = 4 hours)	Down check interval		
	1000 *ms*	500 *ms*	200 *ms*
Takeover delay (ms)	1240 ± 12	740 ± 15	360 ± 6
Failure detection and recovery rate (times/hour)	2903	4865	10000
Availability (A_{HA})	99.99999859%	99.99999881%	99.99999913%

166 ± 9 *ms* (δ = 21687 times/hour), and 222 ± 23 *ms* (δ = 16216 times/hour) for three polling intervals, 50 *ms*, 100 *ms*, and 200 *ms*, respectively. We conclude that the shorter the polling interval, the faster the failure detection and recovery time is.

We then investigated the takeover delay for the PC-based HA-OSPF router due to a hardware failure under different down check intervals. Again, we used the *t* distribution with 9 degrees of freedom and a 95% confidence interval [33, 34] to estimate the takeover delay of the hardware failure. The takeover delays are shown in Table 7. From Table 7, note that the takeover delays (failure detection and recovery rates) of hardware failure for down check intervals 1000 *ms*, 500 *ms*, and 200 *ms* were 1240 ± 12 *ms*, 740 ± 15 *ms*, and 360 ± 6 *ms* (2903 times/hour, 4865 times/hour, and 10000 times/hour). That is, the smaller down check intervals result in the shorter takeover delays (*i.e.*, the HA-OSPF router has higher availability).

From Tables 6 and 7, we learnt that the takeover delays due to software failures and hardware failures depend on the chosen polling interval and down check interval, respectively. However, the takeover delay for software failures is always less than that for hardware failures if the polling interval and the down check interval are set the same. For example, in Tables 6 and 7, if we set both the polling interval and the down check interval to 200 *ms*, the takeover delays for software failures and hardware failures are 222 *ms* and 360 *ms*, respectively. This is because, on average, the time it takes to detect software fail-

ures by the standby router is $t_1/2$ *ms*, where t_1 is the polling interval. Moreover, the time it takes to detect hardware failures by the standby router is at least t_2 *ms*, where t_2 is the down check interval. Therefore, the takeover delay for software failures is always less than that for hardware failures when the polling interval and down check interval are the same (*i.e.*, $t_1 = t_2$).

With the failure detection and recovery rate and the repair rate available, the MTTF (or $1/\lambda$) of an HA-OSPF router can be determined. Assume that the repair rate is 1/4 (times/hour) (*i.e.*, $1/\mu = 4$ hours) [26-28]. Fig. 8 shows the minimum required MTTF of the HA-OSPF router when the required router availability is 99.999% (five-nine) under various failure detection and recovery rates. As addressed above, the failure detection and recovery rate is a key parameter to improve the availability of the HA-OSPF router, which is demonstrated in this figure. From Fig. 8, we found the minimum required MTTF of the HA-OSPF router for achieving the five-nine availability does decrease when the failure detection and recovery rate increases. That is, the availability (A_{HA}) of the HA-OSPF router does increase when the failure detection and recovery rate (δ) increases. Remind that the experimental results in Table 4 show that the failure detection and recovery rate due to a hardware failure and a software failure for the PC-based HA-OSPF router are at least 2903 times/hour and 21687 times/hour, respectively. To have five-nine availability (carrier-grade availability), the MTTF of the proposed PC-based HA-OSPF router with δ = 2903 times/hour must be at least 1820 hours, which is not hard to achieve based on the current router technology. With $1/\lambda$ = 1820 hours and δ = 248 times/hour, the availability of the VRRP-based router is 99.9988171%, which does not meet the requirement of carrier-grade availability with five-nine.

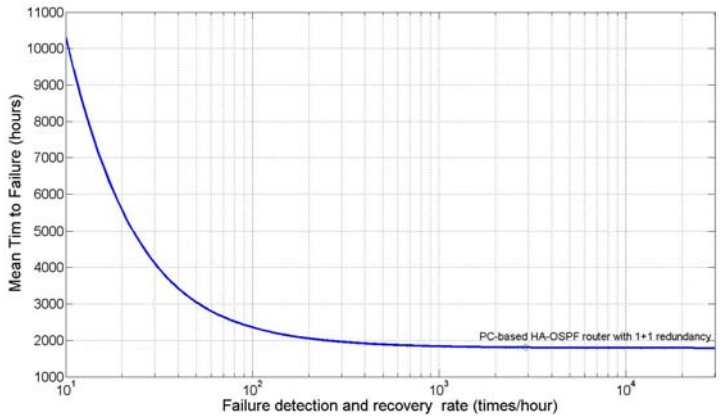


Fig. 8. Mean time to failure (MTTF) when the required availability is 99.999% under various failure detection and recovery rates.

6. FIELD TRIAL RESULTS

This section describes how to implement the HA-OSPF router on an ATCA (Advanced Telecom Computing Architecture) platform and experimental results of the field trial is given. ATCA technology [35] allows new communication equipment to be con-

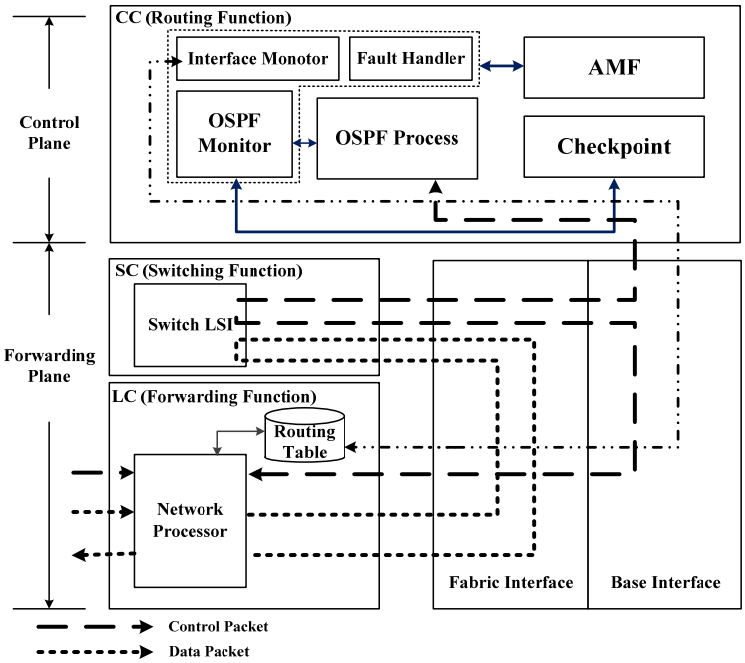


Fig. 9. An ATCA-based HA-OSPF router consisting of LC, CC and SC [35].

structed with great attributes such as high performance, high availability, adaptability for adding new features, and lower cost of ownership. An open architecture solution using the ATCA technology can improve service availability. Thus, industries often use ATCA open architecture combined with their own software solutions to quickly deploy competitive services.

Three types of ATCA cards (*i.e.*, line card, control card, and switch card) were used to build an ATCA-based HA-OSPF router, as shown in Fig. 9 [35]. Based on the operating function of ATCA cards and the concepts of ForCES (Forwarding and Control Element Separation) [14-16], the router can be separated into two parts: control plane and forwarding plane. The control plane service was designed to send control messages and to manage routing information. The forwarding plane service is to decide the outgoing interface for each incoming packet. In general, the forwarding plan looks up the destination address of an incoming packet, refers to a routing table (or forwarding table), finds an outgoing interface for the incoming packet, and then sends the incoming packet through the outgoing interface.

The details of each ATCA card are described as below [35]:

- Line Card (LC): The LC belongs to the forwarding plane and was designed for the basic packet forwarding function. When the LC receives OSPF control packets from its neighbor router, the LC will forward the packets to the control card. Then, if the LC receives the packets, it will forward the packets to correct destinations according to the routing table.
- Control Card (CC): It belongs to the control plane. The CC performs the OSPF routing

protocol based on a received OSPF control packet. When the CC receives an OSPF control packet from its neighbors, the CC resets the waiting timer (*e.g.*, the *Hello* message timer of its neighbor). If the network topology has changed, the CC recalculates the routing table. After that, the CC updates the LC's forwarding table. In addition to the OSPF process, the HAM middleware and OpenAIS middleware have been installed in the CC to perform the state information backup and failure detection and recovery functions.

- Switch Card (SC): The SC belongs to the forwarding plane. It switches packets to a correct card (LC or CC) through the backplane. For example, as shown in Fig. 9, control packets received by the LC will be forwarded to the SC via the base interface and then the SC switches the packets to the CC. Data packets received by the LC will be forwarded to the SC via a fabric interface and then the SC switches these packets to the LC.

In our system, an *AdvancedTCA* compliant processor card, named as aTCA-6890 [36], was used as a control card to build a router. The aTCA-6890 is available in a dual processor configuration with the Low Voltage Intel 3.2 GHz Xeon processor and 800 MHz System Bus. The aTCA-6890 also features the Intel E7520 chipset and 4 GB DDR-400 memories. Peripherals include six Gigabit Ethernet ports and two 10/100/1000 Mbps Ethernet maintenance ports.

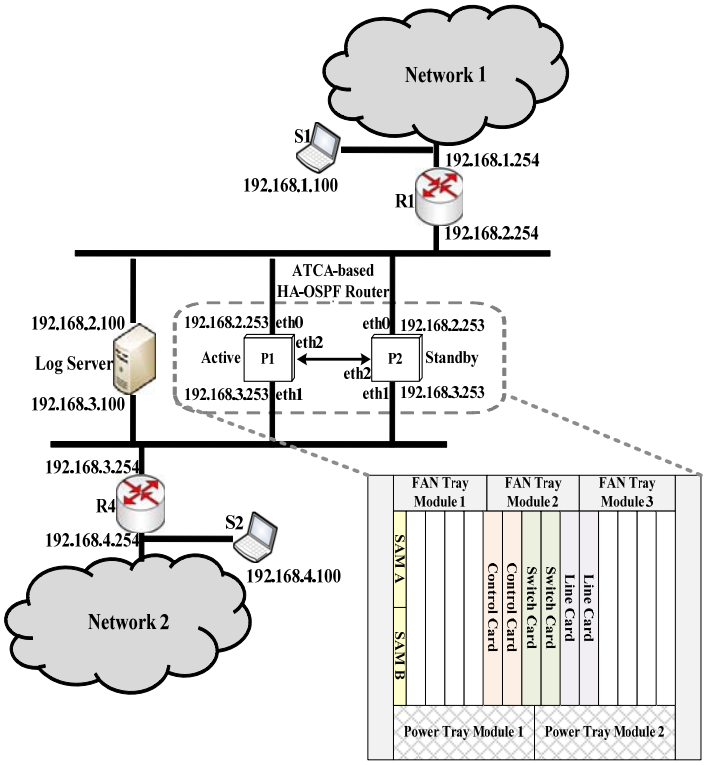


Fig. 10. ATCA-based experimental environment.

Remind that we used two PCs connected via the Ethernet to emulate a PC-based HA-OSPF router in the previous experiment; our HA-OSPF router can be easily implemented on an ATCA platform. We employed the OSPF process and HAM middleware on the ATCA control card and then integrated it on an ATCA chassis to build an ATCA-based HA-OSPF router. In the ATCA, both control cards have two Ethernet interfaces connected to the backplane [35]. Therefore, heartbeat and checkpoint messages can be exchanged between control cards by the backplane. In this experiment, the PCs R2 and R3 were replaced by control cards P1 and P2 (see Fig. 10). The configuration of control cards on the ATCA is the same as that on the PC-based system.

Based on the default parameter values in Table 3, we measured takeover delays of the ATCA-based HA-OSPF router with 1 + 1 redundancy, and experimental results are shown in Table 8. The takeover delays of the PC-based HA-OSPF router from Table 4 are also included in Table 8 for easy reference. The takeover delays (failure detection and recovery rates) of the ATCA-based HA-OSPF router with 1 + 1 redundancy due to a hardware failure and a software failure are 1066 ± 54 *ms* (δ = 3377 times/hour) and 217 ± 17 *ms* (δ = 16590 times/hour), respectively. The takeover delay of the ATCA-based HA-OSPF router due to a hardware failure was reduced by 14% compared to that of the PC-based HA-OSPF router. The availabilities (A_{HA}) of the proposed ATCA-base HA-OSPF router with 1 + 1 redundancy are 9.99999867% and 99.99999905% due to a hardware failure and a software failure, respectively, under $1/\lambda$ = 7 years and $1/\mu$ = 4 hours [26, 28]. That is, the proposed ATCA-based HA-OSPF router with 1 + 1 redundancy can easily meet the requirement of carrier-grade availability with five-nine.

According to Table 9, we found that the CPU usage of the ATCA-based HA-OSPF router is much less than that of the PC-based routers (0.11% vs. 4.47%). This means that the processing capability of an ATCA control card is much more powerful than that of an ordinary PC.

Table 8. Takeover delays (*ms*), failure detection and recovery rates (times/hour), and availabilities for ATCA-based and PC-based HA-OSPF routers.

($1/\lambda$ = 7 years, $1/\mu$ = 4 hours)		Emulation Scenario	
		ATCA-based HA-OSPF router	PC-based HA-OSPF router
Hardware failure	Takeover delay (ms)	1066 ± 54	1240 ± 12
	Failure detection and recovery rate (times/hour)	3377	2903
	Availability (A_{HA})	99.99999867%	99.99999859%
Software failure	Takeover delay (ms)	217 ± 17	166 ± 9
	Failure detection and recovery rate (times/hour)	16590	21687
	Availability (A_{HA})	99.99999905%	99.99999907%

Table 9. CPU usages of HAM middleware and OSPF process for ATCA-based and PC-based HA-OSPF routers.

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

Table 10. Takeover delays (*ms*), failure detection and recovery rates (times/hour), and availabilities for ATCA-based and PC-based HA-OSPF routers with 1 + 1 redundancy under a software failure (OSPF process failed) and various polling intervals.

($1/\lambda$ = 7 years, $1/\mu$ = 4 hours)		Polling interval (*ms*)		
		50 *ms*	100 *ms*	200 *ms*
ATCA-based	Takeover delay (ms)	188 ± 9	217 ± 17	242 ± 26
	Failure detection and recovery rate (times/hour)	19149	16590	14876
	Availability (A_{HA})	99.99999906%	99.99999905%	99.99999904%
PC-based	Takeover delay (ms)	121 ± 5	166 ± 9	223 ± 23
	Failure detection and recovery rate (times/hour)	29752	21687	16216
	Availability (A_{HA})	99.99999909%	99.99999907%	99.99999905%

Table 11. Takeover delays (*ms*), failure detection and recovery rates (times/hour), and availabilities for ATCA-based and PC-based HA-OSPF routers with 1 + 1 redundancy under a hardware failure (power down) and various down check intervals.

($1/\lambda$ = 7 years, $1/\mu$ = 4 hours)		Down check interval		
		1000 *ms*	500 *ms*	200 *ms*
ATCA-based	Takeover delay (ms)	1066 ± 54	743 ± 36	331 ± 28
	Failure detection and recovery rate (times/hour)	3377	4845	10876
	Availability (A_{HA})	99.99999867%	99.99999881%	99.99999900%
PC-based	Takeover delay (ms)	1240 ± 12	740 ± 15	360 ± 6
	Failure detection and recovery rate (times/hour)	2903	4865	10000
	Availability (A_{HA})	99.99999859%	99.99999881%	99.99999899%

Table 10 shows the takeover delays under various polling intervals when a software failure occurred. The takeover delays (failure detection and recovery rates) of the ATCA-based HA-OSPF router with 1 + 1 redundancy were 188 ± 9 *ms* (δ = 19149 times/hour), 217 ± 17 *ms* (δ = 16590 times/hour), and 242 ± 26 *ms* (δ = 14876 times/hour) for three different polling intervals. Because the control card of the standby router needs several seconds to recover the routing information and sends the up-to-date routing table information to the line card [37], the average failure recovery time of the ATCA-based HA-OSPF router (about 150 *ms*) is greater than that of the PC-based HA-OSPF router (about 100 *ms*). However, the difference in takeover delays between the PC-based HA-OSPF router and the ATCA-based HA-OSPF router decreases when the polling interval increases.

Table 11 shows the takeover delays and failure detection and recovery rates due to a hardware failure of power down for different down check intervals. The takeover delay of the ATCA-based HA-OSPF router was reduced by 14% compared to that of the PC-based HA-OSPF router when the down check interval is 1000 *ms*. Experimental results show

that the ATCA-based HA-OSPF router performed better than the PC-based HA-OSPF router under a hardware failure.

From Tables 10 and 11, the experimental results show that the failure detection and recovery rates (δ's) for the ATCA-based HA-OSPF router with 1 + 1 redundancy are at least 3377 times/hour and 14876 times/hour due to a hardware failure and a software failure, respectively. From Fig. 8, to have five-nine availability (carrier-grade availability), the minimum required MTTF ($1/\lambda$) of the proposed ATCA-based HA-OSPF router with 1 + 1 redundancy and $1/\mu = 4$ hours must be at least 1800 hours, which is not hard to achieve based on the current ATCA technology (for example, $1/\lambda = 7$ years from Cisco). The experimental results also show that the failure detection and recovery rates of the proposed ATCA-based HA-OSPF router with 1 + 1 redundancy is much higher than 1.632 times/hour, the minimum required δ to obtain five-nine availability. Therefore, we conclude that the proposed ATCA-based HA-OSPF router with 1 + 1 redundancy can easily achieve the goal of carrier-grade availability with five-nine.

7. CONCLUSION

We have presented an HA-OSPF router with 1 + 1 redundancy which consists of two router modules, the active and standby, to support high availability networks. The continuous-time Markov chain was used to estimate the steady-state availability of the HA router. With the failure detection and recovery rate considered, we have shown that the HA-OSPF router with only one standby router is the preferred model for meeting the requirement of carrier-grade availability. Since there is a lack of research on the integration of redundancy model, link state information backup, and failure detection and recovery, we have proposed an HAM (High Availability Management) middleware, which includes AMF (Availability Management Framework) service, Checkpoint service, Interface Monitor, OSPF Failure Manager, and Failure Handler. The HAM middleware have been designed and added to the HA-OSPF router to support health check, state information exchange, and failure detection and recovery. We have implemented the HA-OSPF router on a PC-based platform. Based on the 1 + 1 redundancy model, experimental results have shown that the failure detection and recovery times of the proposed PC-based HA-OSPF router were reduced by 98.76% and 91.45% compared to those of an industry standard approach, VRRP (Virtual Router Redundancy Protocol), due to a software failure and a hardware failure, respectively. In addition, we have also implemented the HA-OSPF router on an ATCA (Advanced Telecom Computing Architecture) platform, which can provide an industrial standardized modular architecture for an efficient, flexible, and reliable router design. The availabilities of the proposed ATCA-based HA-OSPF router with 1 + 1 redundancy are 99.99999905% due to a software failure and 99.99999867% due to a hardware failure under the failure detection and recovery rates $\delta = 16590$ (times/hour) and 3377 (times/hour), respectively, along with the router module data, $1/\lambda = 7$ years and $1/\mu = 4$ hours, obtained from Cisco. The experimental results have shown that both our proposed ATCA-based and PC-based HA-OSPF routers with 1 + 1 redundancy model can easily achieve the goal of carrier-grade availability with five-nine. For the future work, we will extend the $1 + N$ redundancy model to the $M + N$ redundancy model, where M is the number of active routers and N is the number of standby routers. That is, given M

active routers, we want to find the minimum number of standby routers (N) for an HA router to meet the requirement of carrier-grade availability.

REFERENCES

1. N. Budhiraja, K. Marzullo, F. B. Schneider, and S. Toueg, *Distributed Systems*, 2nd ed., ACM Press/Addison-Wesley Publishing Co., New York, 1993, pp. 199-216.
2. W. Kuo and R. Wan, "Recent advances in optimal reliability allocation," *Studies in Computational Intelligence*, Vol. 39, 2007, pp. 1-36.
3. S. Srivastava, "Redundancy management for network devices," in *Proceedings of the 9th Asia-Pacific Conference on Communications*, Vol. 3, 2003, pp. 1157-1162.
4. A. Mettas "Reliability allocation and optimization for complex systems," in *Proceedings of the Annual Reliability and Maintainability Symposium*, 2000, pp. 216-221.
5. R. Hinden, "Virtual router redundancy protocol (VRRP)," RFC 3768, Internet Engineering Task Force (IETF), 2004.
6. T. Li, B. Cole, P. Morton, and D. Li, "Cisco hot standby router protocol (HSRP)," RFC 2281, Internet Engineering Task Force (IETF), 1998.
7. J. Li and B. Cole, "Standby router protocol," 5473599, United State Patent, 1995.
8. N. Dennis, H. Michael, D. Peter, and M. John, "Method and system for router redundancy in a wide area network," United State Patent 7554903, 2009.
9. J. Ranta, "Router redundancy and scalability using clustering," *Seminar on Internetworking*, 2004, http://www.tml.tkk.fi/Studies/T-110.551/2004/.
10. S. Bommareddy, M. Kale, and S. Chaganty, "System and method for routing message traffic using a cluster of routers sharing a single logical IP address distinct from unique IP addresses of the routers," United State Patent 6779039, 2004.
11. C. T. Tsai, R. H. Jan, C. Chen, and C. Y. Huang, "Implementation of highly available OSPF router on ATCA," in *Proceedings of the 13th IEEE Pacific Rim International Symposium on Dependable Computing*, 2007, pp. 147-154.
12. C. F. Ho, A. Gupta, M. Grandhi, and A. Bachmutsky, "Router and routing protocol redundancy," United State Patent 6910148, 2005.
13. T. Bourke, *Server Load Balancing*, 1st ed., O'Reilly Media, Sebastopol, California, 2001.
14. Forwarding and Control Element Separation (ForCES), IETF, http://www.ietf.org/.
15. W. Wang, L. Dong, B. Zhuge, M. Gao, F. Jia, R. Jin, and X. Wu, "Design and implementation of an open programmable router compliant to IETF ForCES specifications," in *Proceedings of the 6th International Conference on Networking*, 2007, pp. 82-87.
16. X. Wu and L. Dong, "Research and design of the pseudo-VRRP based high availability mechanism in the ForCES router," in *Proceedings of the 8th International Conference on Networking*, 2009, pp. 440-444.
17. Open Specifications for Service Availability, http://www.saforum.org/home/.
18. J. Moy, "Open shortest path protocol (OSPF)," RFC 2328, Internet Engineering Task Force (IETF), 1998.
19. ITU-T (1993), "Terms and definitions related to quality of service and network performance including dependability," ITU-T Recommendation E.800, 1994.
20. G. Michael, S. Hairong, F. M. Ricardo, and K. S. Trivedi, "Ten fallacies of availability

and reliability analysis," in *Proceedings of the 5th International Service Availability Symposium*, 2008, pp. 187-206.
21. K. S. Trivedi, *Probability and Statistics with Reliability*, *Queuing and Computer Science Applications*, 2nd ed., John Wiley and Sons, Inc., New York, 2002, pp. 405-504.
22. W. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, New Jersey, 1994.
23. S. Gokhale and K. S. Trivedi, "Analytical models for architecture-based software reliability prediction: a unification framework," *IEEE Transactions on Reliability*, 2006, pp. 578-590.
24. K. S. Trivedi, A. Sathave, O. Ibe, and R. Howe, "Should I add a processor?" in *Proceeding of the 23rd Hawaii International Conference on System Science*, Vol. 1, 1990, pp. 214-221.
25. M. Lanus, Y. Lin, and K. S. Trivedi, "Hierarchical composition and aggregation of state-based availability and performability models," *IEEE Transactions on Reliability*, Vol. 52, 2003, pp. 44-52.
26. Telcordia Technologies, Local Access and Transport Area Switching Systems Generic Requirements (LSSGR): Reliability Section 12, Issue 2, Telcordia, 1998.
27. F. Hawley and Telcordia Technologies, "Network reliability definitions and requirements," in *Proceedings of International Wireless Packaging Consortium Last Mile Workshop*, 2002, http://www.iwpc.org/library/Home.asp.
28. C. Oggerino, *High Availability Network Fundamentals: A Practical Guide to Predicting Network Availability*, Cisco Press, Indianapolis, Indiana, 2001.
29. OpenAIS Standard based Cluster Framework, http://www.openais.org/.
30. E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, Vol. 1, 1959, pp. 269-271.
31. C. David, "An Ethernet address resolution protocol," RFC 826, Internet Engineering Task Force (IETF), 1982.
32. GNU Zebra, http://www.zebra.org/.
33. W. Gosset, "The problem error of a mean," *Biometrika*, Vol. 6, 1908, pp. 1-25.
34. M. G. Kendall and D. G. Stuart, *The Advanced Theory of Statistics*, Vol. 2, Inference and Relationship, Griffin, London, 1973.
35. AdvancedTCA Specifications for Next Generation Telecommunications Equipment, http://www.picmg.org/v2internal/newinitiative.htm.
36. ADLINK Technology Incorporation, http:// www.adlink.com.tw/.
37. M. Aoki, K. Habara, T. Hamano, K. Ogawa, and S. Chaki, "ATCA-based open-architecture router prototype," *IEICE Transactions on Communications*, Vol. E89-B, 2006, pp. 1685-1687.
38. Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide, Cisco, http://www.cisco.com/.
39. Service Availability Forum, http://www.saforum.org/.

Chia-Tai Tsai (蔡嘉泰) received the B.S. degree in Computer Science from Tamkang University and the M.S. degree in Computer Information and Science from National Chiao Tung University in 2002 and 2004, respectively. He is currently pursuing the Ph.D. degree in the Department of Computer Science at National Chiao Tung University, Taiwan, R.O.C. His research interests include computer networks, network reliability, wireless networks, and operations research.

Rong-Hong Jan (簡榮宏) received the B.S. and M.S. degrees in Industrial Engineering and the Ph.D. degree in Computer Science from National Tsing Hua University, Taiwan, in 1979, 1983, and 1987, respectively. He joined the Department of Computer and Information Science, National Chiao Tung University, in 1987, where he is currently a Professor. From 1991-1992, he was a visiting Associate Professor in the Department of Computer Science, University of Maryland, College Park, Maryland. His research interests include wireless networks, mobile computing, distributed systems, network reliability, and operations research.

Kuochen Wang (王國禎) received the B.S. degree in Control Engineering from National Chiao Tung University, Taiwan, in 1978, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Arizona in 1986 and 1991, respectively. He is currently a Professor in the Department of Computer Science, National Chiao Tung University. From 1980 to 1984, he was a Senior Engineer at the Directorate General of Telecommunications in Taiwan. He served in the army as a second lieutenant communication platoon leader from 1978 to 1980. His research interests include wireless (ad hoc/sensor) networks, mobile computing, and power management for multimedia portable devices.