INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2010; **23**:1581–1599
Published online 30 March 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.1127

Optimal redundancy allocation for high availability routers

Chia-Tai Tsai, Rong-Hong Jan*, † and Kuochen Wang

Department of Computer Science, National Chiao Tung University, 1001 University Road, Hsinchu 300, Taiwan

SUMMARY

How to optimally allocate redundant routers for high availability (HA) networks is a crucial task. In this paper, a *5-tuple availability function A* $(N, M, \lambda, \mu, \delta)$ is proposed to determine the minimum required number of standby routers to meet the desired availability (ρ) of an HA router, where N and M are the numbers of active routers and standby routers, respectively, and λ, μ, and δ are a single router's failure rate, repair rate, and failure detection and recovery rate, respectively. We have derived the availability function, and analytical results show that the failure detection and recovery rate (δ) is a key parameter for reducing the minimum required number of standby routers of an HA router. Thus, we also propose a High Availability Management (HAM) middleware, which was designed based on an open architecture specification, called OpenAIS, to achieve the goal of reducing takeover delay ($1/\delta$) by *stateful backup*. We have implemented an HA Open Shortest Path First (HA-OSPF) router, which consists of two active routers and one standby router, to illustrate the proposed HA router. Experimental results show that the takeover delays of the proposed HA-OSPF router were reduced by 6, 37.3, and 98.6% compared with those of the industry standard approaches, the Cisco-ASR 1000 series router, the Juniper MX series router, and the Virtual Router Redundancy Protocol (VRRP) router, respectively. In addition, in contract to the industry routers, the proposed HA router, which was designed based on an open architecture specification, is more cost-effective, and its redundancy model can be more flexibly adjusted. Copyright © 2010 John Wiley & Sons, Ltd.

Received 1 July 2009; Revised 12 November 2009; Accepted 13 January 2010

KEY WORDS: continues time Markov chain; failure detection and recovery rate; high availability; OSPF; redundancy model

1. INTRODUCTION

With the rapid progress in Internet technologies, 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 high degree of network operational continuity during the service period. Availability problems may result from

*Correspondence to: Rong-Hong Jan, Department of Computer Science, National Chiao Tung University, 1001 University Road, Hsinchu 300, Taiwan.
†E-mail: rhjan@cs.nctu.edu.tw

Contract/grant sponsor: National Science Council of the ROC; contract/grant numbers: NSC 96-2219-E-009-023, NSC 96-2219-E009-008

various causes, including natural disasters, hardware failures, and software failures. Therefore, it is important for a service provider to build a high availability (HA) network to provide continuous services for users, whether to install spare components or dependable components. If a network cannot be accessed, it is said to be unavailable. Generally, the term downtime is used to refer to periods when a network is unavailable.

Network availability can be improved either by incremental improvements in component availability or by provision of redundant components in parallel [1, 2]. Mettas used a nonlinear programming algorithm to formulate a cost function [3], which demonstrates 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 availability of the component, the greater the cost [3, 4]. Depending on the design complexity, technological limitations, and so on, the availability of components can be very hard to improve [3]. In regard to this, adding standby routers to active routers to achieve the goal of building an HA network is a familiar design [5–9]. In general, this approach consists of a cluster of routers where some routers are active whereas the others are on standby. That is, the active routers execute the routing process, whereas standby routers are prepared to take over any active router's role immediately if the active router failed.

Generally, a large organization (e.g. a university or a company) may have several branch offices or campuses. However, it is difficult and costly to add a spare router to each router at a branch office or campus. Fortunately, for the purpose of convenient management and maintenance, the service providers or the network administrators would usually gather the routers in a single machine room and put them in a rack, and forms a router cluster. Because all the routers are placed in a single machine room, service providers can determine the appropriate number of standby routers in a router cluster to meet the requirement of carrier-grade availability easily.

In this paper, a 5-tuple availability function, $A(N, M, \lambda, \mu, \delta)$, is proposed to determine the minimum required number of standby routers in an HA router for achieving the desired availability (ρ), where N and M are the number of active routers and standby routers, respectively and λ, μ, and δ are a single router's failure rate, repair rate, and failure detection and recovery rate, respectively. The availability function can facilitate service providers or network administrators to determine a suitable redundancy model and the minimum required number of standby routers to support their HA routers.

To increase the failure detection and recovery rate, an active router needs to replicate its routing process status and link state information, to the standby routers. For this, we propose a High Availability Management (HAM) middleware, which was designed based on an open architecture specification, called OpenAIS, to achieve the goal of reducing the failure detection and recovery time (i.e. *takeover delay*, $1/\lambda$) by stateful backup [10]. The takeover delay is defined as the latency from the active router failed to the standby router taking over and recovering from the failure. In addition, we have implemented an HA Open Shortest Path First (HA-OSPF) router and evaluated the takeover delay of the proposed HA-OSPF router in the OSPF network [11].

The remainder of this paper is organized as follows. We review related work in Section 2. In Section 3, we propose a 5-tuple availability function and analyze the HA router availability under a various number of standby routers by using the continuous-time Markov chain (CTMC). Analytical results are given in Section 4. In Section 5, we describe the proposed HAM middleware design and the procedures of role assignment, routing process status and link state information backup, and failure detection and recovery. Then, in Section 6 experimental results are evaluated and discussed. Finally, we conclude this paper in Section 7.

DOI: 10.1002/dac

2. RELATED WORK

For establishing network redundancy, Virtual Router Redundancy Protocol (VRRP) [5] and Hot Standby Router Protocol (HSRP) [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 and is actually a standardized version of Cisco's HSRP. These two technologies are similar in concept, but not compatible. These two approaches belong to hardware-level redundancy and service providers or network administrators can adjust the number of active routers and standby routers flexible, but the network disconnection time is too long to achieve the goal of carrier-grade availability. One issue deserved to mention is that a lack of link state information in hardware-level redundancy. For example, in VRRP, a standby router cannot recover the routing protocol session in real time if it takes over the role of the active router. To conquer this problem, a 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 completing 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 increase router availability.

The industry routers, Cisco ASR-1000 series router [12] and Juniper MX series router [13], can provide hardware-level redundancy and support the stateful takeover. Both Cisco ASR-1000 series router and Juniper MX series router have two routers, one active and one standby. The active router replicates the link state information to the standby router to reduce the takeover delay. The standby router can take over the role of the active router immediately if the active router failed. The takeover delays for the Cisco ASR-1000 series router and Juniper MX series router are very small, about 200 ms for Cisco ASR-100 [12] and 300 ms for Juniper MX series router [13]. Although the Cisco ASR-1000 series router and Juniper MX series router have a small takeover delay, they need a specific chassis and a *midplane* to negotiate and exchange the link state information. In addition, the Cisco ASR-1000 series router has lack of ability for flexible adjustment of the redundancy model [12]. That is, it only supports one active router and one standby router. The Juniper MX series router can adjust the redundancy model's flexibility. It supports $2N$ redundancy, $N+M$ redundancy, and full mesh redundancy models.

Because there is a lack of research on the integration of redundancy model, link state information backup, and failure detection and recovery, we also propose an *HA-OSPF router with HAM middleware* which consists of Availability Management Framework (AMF) service [14], Checkpoint service [14], and Failure Manager. The HAM middleware was implemented based on an open source and open architecture project, *OpenAIS* [14]. The flexible redundancy adjustment 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.

3. PROPOSED 5-TUPLE AVAILABILITY FUNCTION

With the design complexity and technology limitations, Mettas used a cost function to show that it is very difficult to improve the availability of the router, the greater the cost [3]. Thus, a

DOI: 10.1002/dac

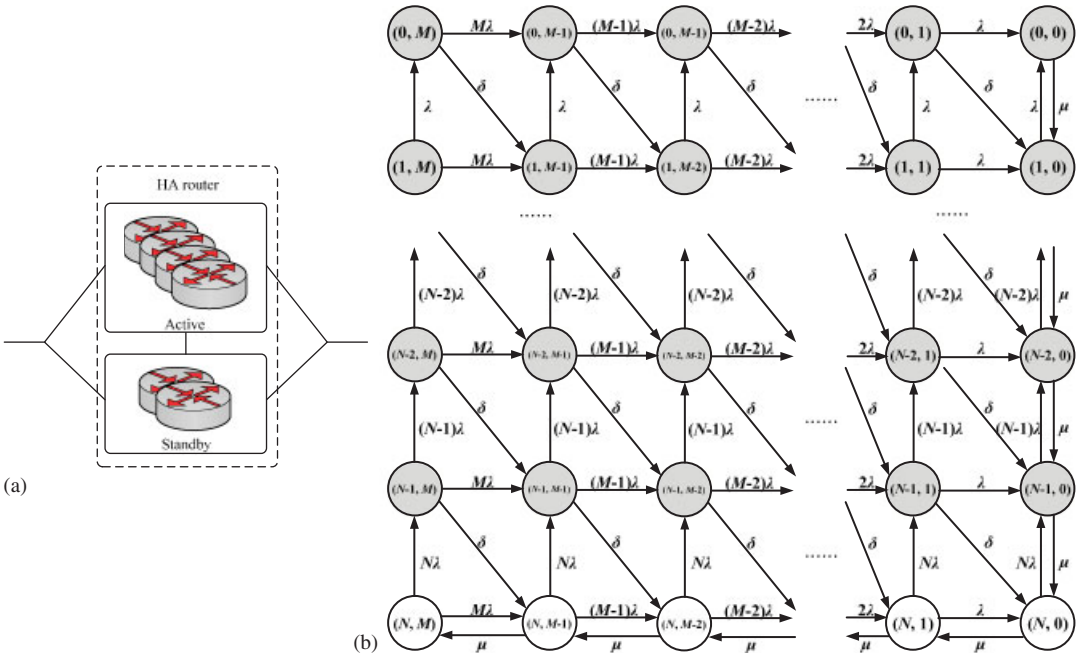

Figure 1. Logical structure and CTMC for an HA Router with $N+M$ redundancy.

feasible way to increase the router availability is to add the standby router to the HA router [5–9]. In this section, we propose a 5-tuple availability function, $A(N, M, \lambda, \mu, \delta)$, to determine the minimal number of standby routers (M) in an HA router to achieve the desired availability, under the conditions of the failure rate (λ), repair rate (μ), failure detection and recovery rate (δ), and number of active routers (N). The CTMC [15–17] is used to determine the steady-state availability of an HA router with various numbers of active routers and standby routers.

3.1. CTMC for n+m redundancy model

In this section, the CTMC of an HA router with $N+M$ redundancy (i.e. N active routers and M standby routers) is considered. Each standby router monitors the status of all active routers. If one of the active routers failed, the standby routers hold an election automatically. Then, one of the standby routers will take over the role of the active router. Figure 1(a) is the logical structure of an HA router with $N+M$ redundancy. The CTMC for an HA router with $N+M$ redundancy is depicted in Figure 1(b). The active routers work properly at state (N, p), where $0 \leqslant p \leqslant M$. If the state of an HA router moves from state (i, j) to state $(i+1, j-1)$, it represents that there is an active router that failed and the system detects and recovers the failure with rate δ, where $0 \leqslant i \leqslant N-1$ and $1 \leqslant j \leqslant M$. State $(0, 0)$ represents that all routers, including active and standby routers, of the HA router failed.

DOI: 10.1002/dac

After writing the steady-state equations and solving these equations, we obtain the following equations under the steady state:

$$((k+M)\lambda+\delta)\cdot\pi(k,M)=(k+1)\lambda\cdot\pi(k+1,M) \quad \text{where } 0\leqslant k\leqslant N-1 \tag{1}$$

$$\lambda\cdot\pi(k,0)=(k+1)\lambda\cdot\sum_{\substack{0\leqslant i,j\leqslant k+1\\ :i+j=k+1}}\pi(i,j) \quad \text{where } 0\leqslant k\leqslant N-1 \tag{2}$$

$$\mu\cdot\pi(N,k)=(N+k+1)\lambda\cdot\sum_{\substack{k+1\leqslant i\leqslant M,0\leqslant i\leqslant N\\ :i+j=N+k+1}}\pi(i,j) \quad \text{where } 0\leqslant k\leqslant M-1 \tag{3}$$

$$(k\lambda+\delta)\cdot\pi(0,k)=\lambda\cdot\pi(1,k)+(k+1)\lambda\cdot\pi(0,k+1) \quad \text{where } 1\leqslant k\leqslant M-1 \tag{4}$$

$$((i+j)\lambda+\delta)\cdot\pi(i,j)=(i+1)\lambda\cdot\pi(i+1,j)+(j+1)\lambda\cdot\pi(i,j+1)+\delta\cdot\pi(i-1,j+1)$$
$$\text{where } 1\leqslant i\leqslant N-1 \quad \text{and} \quad 1\leqslant j\leqslant M-1 \tag{5}$$

$$(M+N)\lambda\cdot\pi(N,M)=\mu\cdot\pi(N,M-1) \tag{6}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{M}\pi(i,j)=1 \tag{7}$$

The CTMC for an HA router with $N+M$ redundancy can transit into a two-state and two-transition Markov chain [18], as shown in Figure 2. One state is the *Up* with the reward rate λ_{HA}; the other state is the *Down* with the reward rate μ_{HA} [18]. λ_{HA} and μ_{HA} are the *equivalent failure rate* and the *equivalent repair rate* of the HA router with $N+M$ redundancy, which can be determined by applying the aggregation techniques described in [18]. Therefore, λ_{HA} and μ_{HA} can be written as follows:

$$\begin{aligned}\lambda_{HA}&=\frac{N\lambda\cdot\pi(N,M)+N\lambda\cdot\pi(N,M-1)+\cdots+N\lambda\cdot\pi(N,1)+N\lambda\cdot\pi(N,0)}{\pi(N,M)+\pi(N,M-1)+\cdots+\pi(N,1)+\pi(N,0)}\\&=\frac{N\lambda\cdot\left(\sum_{j=0}^{M}\pi(N,j)\right)}{\sum_{j=0}^{M}\pi(N,j)}=N\lambda\end{aligned} \tag{8}$$

$$\begin{aligned}\mu_{HA}&=\frac{\delta\cdot\pi(N-1,M)+\delta\cdot\pi(N-1,M-1)+\cdots+\delta\cdot\pi(N-1,1)+\mu\cdot\pi(N-1,0)}{\pi(N-1,M)+\pi(N-1,M-1)+\cdots+\pi(N-1,1)+\pi(N-1,0)}\\&=\frac{\delta\cdot\left(\sum_{j=1}^{M}\pi(N-1,j)+\pi(N-1,0)\right)+\mu\cdot\pi(N-1,0)-\delta\cdot\pi(N-1,0)}{\sum_{j=0}^{M}\pi(N-1,j)}\\&=\frac{\delta\cdot\left(\sum_{j=0}^{M}\pi(N-1,j)\right)+(\mu-\delta)\cdot\pi(N-1,0)}{\sum_{j=0}^{M}\pi(N-1,j)}\\&=\delta+\frac{(\mu-\delta)\cdot\pi(N-1,0)}{\sum_{j=0}^{M}\pi(N-1,j)}\end{aligned} \tag{9}$$

DOI: 10.1002/dac

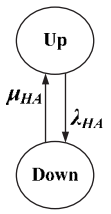

Figure 2. Equivalent Markov chain.

3.2. Steady-state availability definition

The steady-state availability is the probability of a system that is still available over a long period. The steady-state availability (A) can be expressed as [15, 19, 20]

$$A = \frac{MTTF}{MTTF + MTTR} \tag{10}$$

where *MTTF* (mean time to failure) is the arithmetic mean time between failures of a component or system and *MTTR* (mean time to repair) is the amount of time required to perform corrective maintenance and restore a component or system to operational status. *MTTR* includes total 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 Equation (10) can be rewritten as [15, 19, 20]

$$A = \frac{\mu}{\lambda + \mu} \tag{11}$$

Therefore, from Equation (11), the *equivalent availability* of an HA router (A_{HA}) can be expressed as follows:

$$A_{HA} = \frac{\mu_{HA}}{\lambda_{HA} + \mu_{HA}} \tag{12}$$

Solving Equations (8) and (9), we can get an equivalent availability of an HA router based on Equation (12) under failure rate (λ), failure detection and recovery rate (δ), and repair rate (μ).

3.3. Formalizing a 5-tuple availability function

Based on the above discussion, we propose a 5-tuple availability function, $A(N, M, \lambda, \mu, \delta)$, to determine the minimum required number of standby routers (M) needed to be allocated in an HA router to achieve the desired availability (ρ). In addition, as shown in Equation (13), the equivalent availability of an HA router (A_{HA}) is equal to the derived value of the 5-tuple availability function.

$$A_{HA} = A(N, M, \lambda, \mu, \delta) \tag{13}$$

DOI: 10.1002/dac

Therefore, problem P1 can be formally defined as follows:

Problem P1:

$$\text{Minimize} \quad M$$

$$\text{subject to}$$

$$A_{HA} = \frac{\mu_{HA}}{\lambda_{HA} + \mu_{HA}} \geqslant \rho \quad \text{where } 0 \leqslant M \leqslant N \tag{14}$$

where μ_{HA} and λ_{HA} are the equivalent repair rate and equivalent failure rate of an HA router, which can be calculated from Equations (8) and (9), respectively.

4. ANALYTICAL RESULTS

4.1. Numerical analysis of minimal required standby routers

In this section, the parameter settings of μ, λ, δ, and N are given as follows. Based on the data from Cisco, we set $\mu = 0.25$ times/h (i.e. MTTR $(1/\mu)$ is equal to 4 h). The MTTR of a router is assumed to be the time it takes to have a spare part and a knowledgeable person to arrive to repair. Three MTTFs, low MTTF ($1/\lambda = 10\,000$ h), high MTTF ($1/\lambda = 100\,000$ h), and Cisco carrier-grade router's MTTF ($1/\lambda = 61\,320$ h) are considered. The failure detection and recovery rate (δ) is set to 100, 1000, 10 000, and 100 000 times/h. In addition, three failure detection and recovery rates that were measured from the proposed HA router are also considered. Those includes $\delta = 11\,429$ times/h for hardware failures only, $\delta = 58\,065$ times/h for software failures only, and $\delta = 34\,747$ times/h for hardware and software failures (see Section 6). The number of active routers N varies from $1, 2, 4, \ldots,$ to 128. Table I shows the analytical results to determine the minimum required number of standby routers (M) for the proposed HA router under various μ, λ, δ, and N.

From the analytical results, we also found that the minimum required number of standby routers (M) can be decreased when the failure rate (λ) or the failure detection and the recovery rate (δ) of the router decreases and increases, respectively. It also shows that the failure detection and recovery rate (δ) of a router is a key parameter for reducing the minimum required number of standby routers in an HA router.

Figure 3 shows the relationship between the minimum required number of standby routers and the number of active routers for an HA router with $1/\lambda$, $1/\mu$, and ρ being set to 61 320 h, 4 h (from Cisco [21–23]), and 99.999% respectively. Based on Figure 3, service providers or network administrators can determine the appropriate number of standby routers for constructing an HA router under various numbers of active routers and the desired availability (ρ). For instance, an HA router needs only one standby router to meet the requirement of carrier-grade availability ($\rho = 99.999\%$) when the number of active routers is not greater than 47, as shown in Figure 3.

4.2. Computational complexity

To solve Problem P1, we can apply binary search method on $M (0 \leqslant M \leqslant N)$. For a given M, we evaluate $A(N, M, \lambda, \mu, \delta)$ and check to see if $A(N, M, \lambda, \mu, \delta) \geqslant \rho$ or not. By this way, the minimum value of M such that $A(N, M, \lambda, \mu, \delta) \geqslant \rho$ can be found. In each iteration, we have to solve

DOI: 10.1002/dac

Table I. The minimum required standby routers (M) for an HA router to achieve the goal of carrier-grade availability ($\rho = 99.999\%$).

		$N=1$			$N=2$			$N=4$			$N=8$		
		$1/\lambda$(h)			$1/\lambda$ (h)			$1/\lambda$ (h)			$1/\lambda$ (h)		
$\mu=0.25$ times/h		10 000	61 320	100 000	10 000	61 320	1 00000	10 000	61 320	100 000	10 000	61 320	100 000
δ (times/h)	100	1	1	1	1	1	1	1	1	1	2	1	1
	1000	1	1	1	1	1	1	1	1	1	2	1	1
	10000	1	1	1	1	1	1	1	1	1	2	1	1
	11429	1	1	1	1	1	1	1	1	1	2	1	1
	34747	1	1	1	1	1	1	1	1	1	2	1	1
	58065	1	1	1	1	1	1	1	1	1	2	1	1
	1000000	1	1	1	1	1	1	1	1	1	2	1	1
		$N=1$			$N=2$			$N=4$			$N=8$		
		$1/\lambda$ (h)			$1/\lambda$ (h)			$1/\lambda$ (h)			$1/\lambda$ (h)		
$\mu=0.25$ times/h		10 000	61 320	100 000	10 000	61 320	100 000	10 000	61 320	100 000	10 000	61 320	100 000
δ (times/h)	100	X	1	1	X	1	1	X	X	2	X	X	X
	1000	2	1	1	2	1	1	3	2	1	X	2	2
	10000	2	1	1	2	1	1	3	2	1	3	2	2
	11429	2	1	1	2	1	1	3	2	1	3	2	2
	34747	2	1	1	2	1	1	3	2	1	3	2	2
	58065	2	1	1	2	1	1	3	2	1	3	2	2
	1000000	2	1	1	2	1	1	3	2	1	3	2	2

X: no feasible solution.

DOI: 10.1002/dac

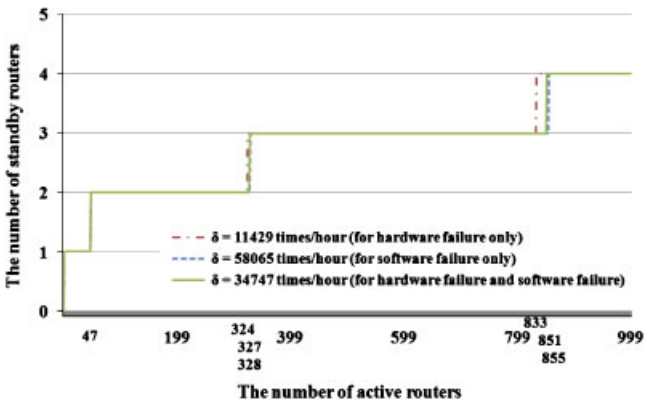

Figure 3. The minimum required number of standby routers for an HA router under various numbers of active routers and failure detection and recovery rates (with $\rho = 99.999\%$).

Equations (1)–(7) for evaluating $A(N, M, \lambda, \mu, \delta)$. Note that Equations (1)–(7) can be rewritten as a system $Ax = b$ of linear equations where A is $n \times n$ matrix. The system $Ax = b$ can be solved by Gaussian elimination with time complexity $O(n^3)$. Thus, we can apply Gaussian elimination to the Equations (1)–(7) with $n = (N+1)(M+1)$. That is, it takes $O([(M+1)(N+1)]^3) = O((MN)^3)$ time to evaluate $A(N, M, \lambda, \mu, \delta)$ in each iteration. The number of iterations needed for the binary search is $O(\log N)$. Therefore, the total time for solving Problem P1 is $O(M^3 N^3 \log N)$.

5. PROPOSED HA ROUTER DESIGN

The proposed 5-tuple availability function shows that the failure detection and recovery rate (δ) is a key parameter to increase the availability of an HA router. In order to increase the failure detection and recovery rate, an HAM middleware was designed, which can decrease the takeover delay ($1/\lambda$) and meet the requirement of carrier-grade availability with five–nine. In this section, we are going to discuss the function of each component in the proposed HAM middleware design.

5.1. HAM middleware design

As shown in Figure 4, the HAM middleware (within the two-dot chain square) includes two different entities, OpenAIS middleware and Failure Manager. The OpenAIS middleware is a cluster middleware defined in the *Service Availability Forum* (*SAF*) *Application Interface Specification* [14]. In this paper, two services, AMF service and Checkpoint service, were used to construct the HA-OSPF router. The processes in the router can communicate with AMF service and Checkpoint service through the *interface*, which is a set of Application Programming Interface (APIs) and callback functions, of OpenAIS middleware. The functions of AMF service and Checkpoint service are described as follows:

- *AMF service*: It provides role assignment and health check. The AMF service can provide three kinds of redundancy model, $2N$ redundancy, $N+M$ redundancy, and N-way redundancy. When a router first starts, the AMF service will assign a role, *active* or *standby*, to the router.

DOI: 10.1002/dac

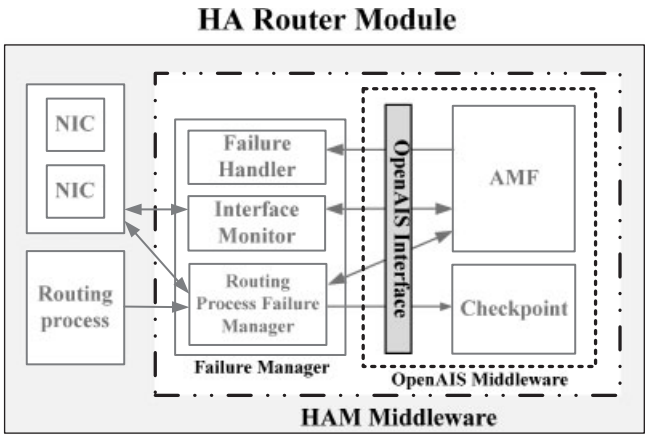

Figure 4. The components of an HA router module.

The AMF service of the active router sends a *heartbeat* message to the standby router(s) periodically to report its health status. If the standby router does not hear the heartbeat message from the active router within a *down check interval* (e.g. 1 s, which is a default value), it will assume that the active router has failed and the AMF service will find a router from the standby router(s) to take over the role of the active router.

- *Checkpoint service*: It provides routing process status and link state information exchange service between active and standby routers. Through this service, the active router can replicate its routing process status and link state information to the standby router(s). The information can help a standby router to reduce the takeover delay and improve the availability when it takes over.

Moreover, the proposed Failure Manager is designed to monitor the status of NICs and routing process and to backup the routing process status and link state information. The Failure Manager will register itself to the OpenAIS middleware and get the permission for using the AMF service and Checkpoint service. The Failure Manager consists of following three modules:

- The *Routing Process Failure Manager* takes care of the routing process operations, informs the AMF service if a failure in the routing process is detected, and replicates the routing process status and link state information to the Checkpoint service.
- The *Interface Monitor* checks the health status 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. For instances, the callback function will reinitialize the failed process or device if the failure can be determined by the Failure Manager (e.g. the routing process or an NIC failed). However, if the failure (e.g. AMF service failed or HA router failed) cannot be determined by the Failure Manager, the failed router will be restarted by the callback function after a *down check interval* and the standby router will send a report to the network administrator.

DOI: 10.1002/dac

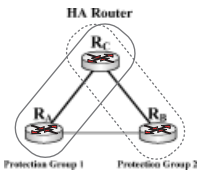

Figure 5. The logical structure of an HA router with $2+1$ redundancy.

5.2. HAM middleware procedures

The operation procedures of the HAM middleware can be divided into three parts:

- *Role assignment*: We use $N=2$ and $M=1$ as an example to illustrate an HA router with $N+M$ redundancy and it can be easily extended to the general case. As shown in Figure 5, there exist two protection groups (e.g. protection groups (R_A, R_C) and (R_B, R_C)) in an HA router. A *protection group* [14] is defined as a pair of routers, one active and one standby. When the router in an HA router is started, it will get the role, active or standby, first. The standby router monitors the active router's health status in each protection group. If an active router fails, the standby router will take over the role of the active router. Note that at this moment all protection groups are lost. After a failed router having been repaired, it will re-initiate and execute the role assignment operation to form a protection group again. Like VRRP, the active router and the standby router in the same protection group use the private IP addresses to communicate with each other. Moreover, the active router uses the real IP address to communicate with its adjacent routers. As soon as the standby router takes over, the standby router changes its IP addresses to the real IP addresses. For a broadcast network (e.g. Ethernet), the standby router will send a gratuitous ARP [24] message to the network. The gratuitous ARP message is used to ask its neighbors to bind the MAC address of the standby router to the real IP address. Thus, the standby router can receive and forward the packets continuously when it takes over.
- *Routing process status and link state information backup*: Figure 6 shows how routing process status and link state information flow from the active router to standby router. The Routing Process Failure Manager of active router gets the routing process status and link state information and replicates those to the standby router through the Checkpoint service. Then, the standby router receives and saves the routing process status and the link state information. When the standby router takes over, the information can help the standby router to decrease the takeover delay and improve the availability of the HA router.
- *Failure detection and recovery*: As shown in Figure 7 when an HA router starts, the Routing Process Failure Manager and Interface Monitor in each router register themselves at the AMF and register their callback functions at the Failure Handler. If the Routing Process 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 and generate an error message to the AMF service of the standby router. After receiving the error message, the standby router takes over and changes its role as active router. Then the Routing Process Failure Manager of standby router changes its IP addresses to the real IP addresses and sends a *gratuitous ARP* [24] message to the network to ask its neighbors to bind its MAC address to the real IP address. Thus, the packets can be forwarded continuously.

DOI: 10.1002/dac

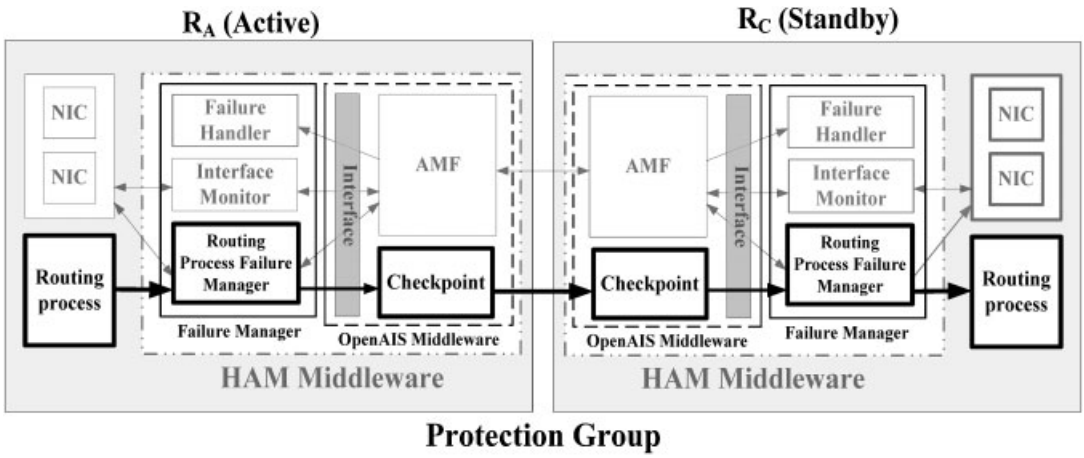

Figure 6. Link state information backup for a protection group.

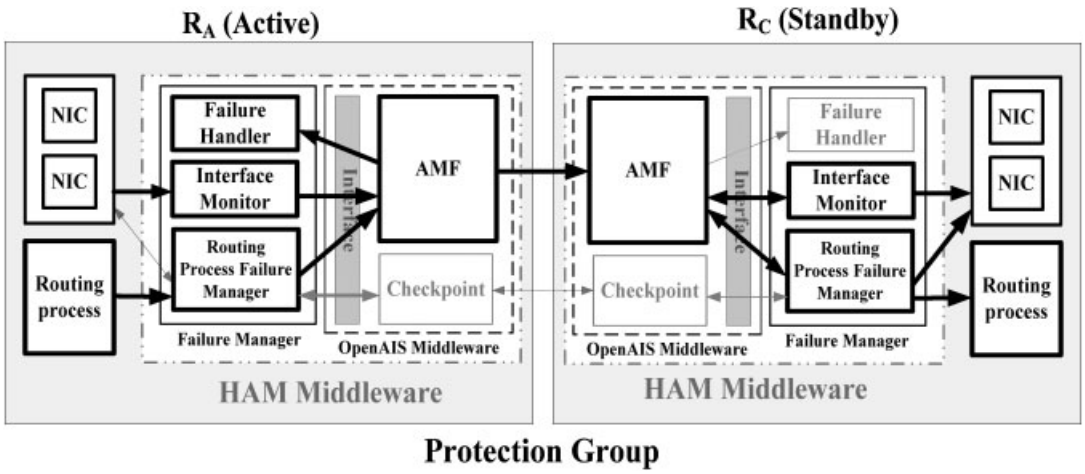

Figure 7. Failure detection and recovery procedure for the routing process in the protection group.

6. EXPERIMENTS

In Figure 3, we have shown that an HA router with $N+1$ redundancy (for $N \leqslant 47$) is the recommended scheme to meet the carrier-grade ($\rho = 99.999\%$) availability under an appropriate failure rate (λ), failure detection and recovery rate (δ), and repair rate (μ) In this section, we will actually measure the failure detection and recovery rate (δ) of the proposed HA-OSPF router with $N+1$ redundancy on an OSPF network ($N=2$ in our experiments for illustration). We will show that the takeover delay of the proposed HA-OSPF router with HAM middleware is smaller than those of an industry standard approach, Cisco ASR-1000 router [12] and a VRRP router [5]. The takeover delay (the multiplicative inverse of the failure detection and recovery rate) is defined as the latency

DOI: 10.1002/dac

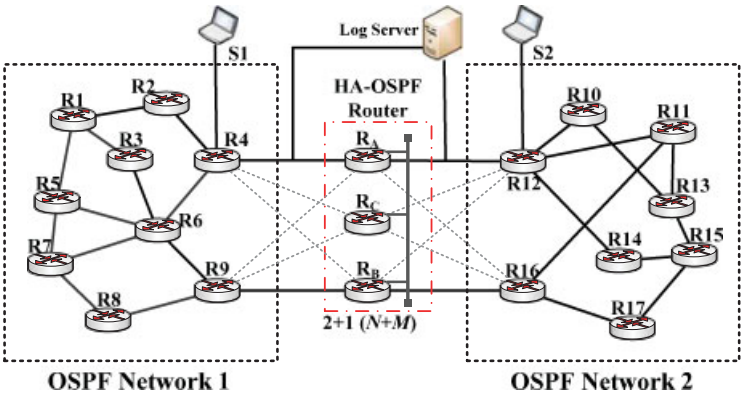

Figure 8. Experimental environment.

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.

6.1. Experimental setup

We have implemented an HA-OSPF router on a PC-based environment. We used the 2+1 redundancy model as an example to construct the HA router to verify the correctness of the proposed HA-OSPF router. To implement the HA-OSPF router with 2+1 redundancy, three desktop PCs with Intel Pentium 4 3.0 GHz processors and 512 MB memories connected via Ethernet were used to emulate an HA-OSPF router. That is, the HA-OSPF router consists of three routers R_A, R_B, and R_C, as shown in Figure 8. A Linux operating system and GNU Zebra [25] were selected as the developing platform for the PC-based HA-OSPF router. The GNU Zebra is a well-known open source software that manages the TCP/IP-based routing protocol. Suppose that R_A and R_B are active routers and R_C is a standby router when the HA-OSPF router is first started. Then, we used two PCs that run Integrated Multiprotocol Network Emulator Simulator (IMUNES) [26], which could send OSPF control messages to the HA-OSPF router, to emulate OSPF networks 1 and 2. There were two clients (S1 and S2) and one log server in our experimental network, as shown in Figure 8.

In the experiment, S1 sent UDP data packets with specific sequence numbers to S2 to examine the network connectivity (see Figure 8). The log server was used to record the sequence number and timestamp of each packet that it received. If S1 sends a packet to S2, it also has to send a copy of the packet to the log server. Then, S2 will forward the packet it received from S1 to the log server. During the takeover period, the network will be disrupted. The log server will not receive any packets transferred from S2. After the standby router takes over the role of the active router, the log server will continue to receive packets from S2. In this way, the takeover delay can be determined. The default parameter values for the OSPF routing protocol and HAM middleware are listed in Table II [11, 14, 21–23]. The *Hello* interval is the number of seconds this router waits before sending out the next *Hello* packet [11, 14]. If a router does not receive a *Hello* packet from a neighbor router within a fixed amount of time, the router modifies its topological database to indicate that the neighbor router is not operational. The time that the router waits is called the router dead interval. By default, this interval is 40 s (four times the default *Hello* interval) [11, 14].

DOI: 10.1002/dac

Table II. Default parameter values [11, 14, 21–23, 27].

Router dead interval of OSPF	40 s
Hello interval	10 s
Down check interval of AMF service	1000 ms
Polling interval of Failure Manager	100 ms
MTTF ($1/\lambda$)	7 years (61 320 h)
MTTR ($1/\lambda$)	4 h

Table III. Takeover delay (ms) of the proposed HA-OSPF router under various redundancy models ($\rho = 99.999\%$).

	Redundancy model		
	1+1	2+1	2+2
Hardware failure	565 ± 3	569 ± 3	576 ± 4
Software failure	110 ± 2	112 ± 3	118 ± 4

Based on Cisco data, the MTTF ($1/\lambda$) and MTTR ($1/\mu$) of a commercial router need at least 7 years (i.e. 61 320 h) and not exceeding 4 h, respectively [21–23]. The default values for the down check interval of AMF service and polling interval of the Failure Manager are 1000 and 100 ms, respectively [27]. 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 that it has failed. The polling interval is a period of time in which the Routing Process Failure Manager and the Interface Monitor check the status of routing process and the NICs, respectively.

6.2. *Experimental results*

First, we will show that the failure detection and recovery time (i.e. takeover delay) is not affected too much by the redundancy model used in the HA router. The takeover delays for the proposed HA-OSPF router under various redundancy models are shown in Table III with the down check interval of 1000 ms and the polling interval of 100 ms for a hardware failure and software failure, respectively. From Table III, the takeover delay for a hardware failure (a software failure) of the proposed HA-OSPF router with 1+1, 2+1, and 2+2 redundancy are 565 ± 3, 569 ± 3, and 576 ± 4 ms (110 ± 2, 112 ± 3, and 118 ± 4 ms), respectively. The experimental results show that the redundancy model of the HA-OSPF router does not affect too much the takeover delay. Therefore, the 2+1 redundancy model, which is a more cost-effective configuration, was used to measure takeover delays of the proposed HA-OSPF router in the subsequent experiments.

Then, we investigate how the takeover delay is affected by the state information backup of the standby router. We did not measure the takeover delay of Cisco ASR-1000 series router due to lack of facilities. However, in [12], it describes that if an active router of Cisco ASR-1000 series router experiences a hardware or software failure that makes it unable to forward traffic and a standby router of Cisco ASR-1000 series router is configured, the standby router becomes the active router within 200 ms [12]. Therefore, only the following two cases were implemented and evaluated as follows:

- *VRRP-based router with* $2+1$ *redundancy*: The active routers do not save any state information in the standby router.

DOI: 10.1002/dac

Table IV. Takeover delays (ms) and failure detection and recovery rates (times/h) for an HA-OSPF router and a VRRP-based router ($\rho = 99.999\%$).

		Emulation scenario	
		VRRP	HA-OSPF router
Hardware failure	Takeover delay (ms)	14511 ± 36	569 ± 3
	Failure detection and recovery rate (times/h)	248	6327
Software failure	Takeover delay (ms)	13383 ± 3	112 ± 3
	Failure detection and recovery rate (times/h)	269	32 143

Table V. Takeover delays (ms) and failure detection and recovery rates (times/h) due to a software failure (OSPF process down) under various polling intervals ($\rho = 99.999\%$).

	Polling interval		
	50 ms	100 ms	200 ms
Takeover delay (ms)	62 ± 1	112 ± 3	170 ± 2
Failure detection and recovery rate (times/h)	58 065	32 143	21 176

- *Proposed HA-OSPF router with* $2+1$ *redundancy*: Each active router backs up its full state information, including its link states, link state database (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, in Figure 8, UDP packets traveled along path S1, R4, R_A, R12, S2 until the active router failed. After R12 and R4 reestablished their routing tables, the UDP packets could go through the path S1, R4, R_C, R12, S2.

The takeover delays for the proposed HA-OSPF router with $2+1$ redundancy and VRRP-based router with $2+1$ redundancy are shown in Table IV. The takeover delays for a hardware failure (a software failure) of the VRRP-based router and the proposed HA-OSPF router were 14511 ± 36 and 569 ± 3 ms (13383 ± 3 and 112 ± 3 ms), respectively. Experimental results show that the takeover delays of the proposed HA-OSPF router were reduced by 96.08 and 99.16% compared with those of VRRP for a hardware failure and a software failure, respectively. The proposed HA-OSPF router with full state information backup demonstrates its benefits.

Next, we measured the takeover delay for the PC-based HA-OSPF router due to a software failure under various polling intervals. Table V shows that the takeover delays (failure detection and recovery rates) due to a software failure were 62 ± 1 ($\delta = 58065$ times/h), 112 ± 3 ($\delta = 32143$ times/h), and 170 ± 2 ms ($\delta = 21176$ times/h) for three polling intervals, 50, 100, and 200 ms, respectively. Experimental results show that the takeover delay depends on the polling interval. We found that the shorter the polling interval, the faster the takeover delay (i.e. failure detection and recovery time) is.

We then investigated the takeover delay of the proposed HA-OSPF router due to a hardware failure under different down check intervals. In Table VI, the takeover delays (failure detection and recovery rates) due to a hardware failure under down check intervals of 500, 1000, and 200 ms were

DOI: 10.1002/dac

Table VI. Takeover delays (ms) and failure detection and recovery rates (times/h) due to a hardware failure under various down check intervals ($\rho = 99.999\%$).

	Down check interval		
	500 ms	1000 ms	2000 ms
Takeover delay (ms)	315 ± 2	569 ± 3	1087 ± 9
Failure detection and recovery rate (times/h)	11 429	6327	3312

315 ± 2, 569 ± 3, and 1087 ± 9 ms (11429 times/h, 6327 times/h, and 3312 times/h), respectively. That is, the smaller down check intervals result in the shorter takeover delays.

Table VII summarized the comparisons of the proposed HA-OSPF router, VRRP router, Cisco ASR-1000 series router, and Juniper MX series router in terms of cost, takeover delay, implementation flexibility, flexible redundancy model, stateful backup, open specification and open source, storage overhead, and bandwidth overhead. The router that supports stateful backup needs the additional bandwidth and storage to transfer and save the routing process status and link state information, respectively. As shown in Table VII, the bandwidth overhead is the amount of bandwidth (in bps) used by the active router transmitting the heartbeat and replicating its routing process status and the link state information to the standby router. The storage overhead is the number of bytes used by standby router saving the routing process status and link state information of active router. Moreover, since the proposed HA-OSPF router is constructed based an open source and open architecture specification, OpenAIS, and it does not need the specific chassis and hardware to achieve the goal of carrier-grade availability, the cost and the implementation difficulty for constructing the proposed HA-OSPF router are less than those of the Cisco ASR-1000 series router and Juniper MX series router. Furthermore, from experimental results, we found that the takeover delays of the proposed HA-OSPF router were reduced 6, 37.3, and 98.6% compared with those of the Cisco-ASR 1000 series router, the Juniper MX series router, and the VRRP router, respectively. Therefore, we concluded that the proposed HA-OSPF router is more feasible than the VRRP-based router, Cisco ASR-1000 series router, and Juniper MX series router to construct an HA network.

7. CONCLUSION

We have presented a 5-tuple availability function, $A(N, M, \lambda, \mu, \delta)$, to relate to the desired availability (ρ), where N, M, λ, μ, and δ are number of active routes, number of standby routers, failure rate, repair rate, and failure detection and recovery rate, respectively. By applying this 5-tuple availability function, service providers can determine the minimum required number of standby routers for constructing an HA router to meet the requirement of the carrier-grade availability ($\rho = 99.999\%$). The CTMC has been used to estimate the steady-state availability of an HA router with a different combination of numbers of active and standby routers. The analytical results have shown that the failure detection and recovery rate (δ) is a key parameter for reducing the minimum required number of standby routers. In order to increase the failure detection and recovery rate, the active router needs to replicate its routing process status and link state information to the standby routers. The HAM middleware, which includes AMF service, Checkpoint service, Failure Manager, has also been proposed. It has been integrated to the proposed HA router to achieve

DOI: 10.1002/dac

Table VII. The comparisons of the proposed HA-OSPF router, VRRP router, Cisco ASR-1000 series router, and Juniper MX series router.

Scheme	HA-OSPF router (proposed)	VRRP router [5]	Cisco ASR-1000 series router [12]	Juniper MX series router [13]
Cost	Medium	Low	Very high	Very high
Takeover delay	189 ms*	13383 ms	About 200 ms	300 ms†
Implementation flexibility	Easy	Easy	Hard (Cisco IOS)	Hard (Juniper JUNOS)
Flexible redundancy model	Yes	Yes	No	Yes
Stateful backup	Yes	No	Yes	Yes
Open specification/source	Yes	Yes	No, proprietary (Cisco IOS)	No, proprietary (Juniper JUNOS)
Storage overhead‡	$((NM)\times P\times Q)/8$ bytes	No	$(P\times Q)/8$ bytes	$((NM)\times P\times Q)/8$ bytes
Bandwidth overhead§	$((NM)\times P\times Q/T_C)+(K/T_H)$ bps	(K/T_H) bps	$(P\times Q/T_C)+(K/T_H)$ bps	$((NM)\times P\times Q/T_C)+(K/T_H)$ bps

*189 ms = (62 ms + 315 ms)/2, where 62 ms is for a software failure (*Hello* interval is 50 ms) and 315 ms is for a hardware failure (*Hello* interval is 500 ms), see Section 6.

†The takeover delay of the Juniper MX series router is three times of *Hello* intervals (*Hello* interval is 100-65535 ms [13]).

‡P is the number of routers in the network and Q is the number of bits of process status and link state information for each router.

§T_C and T_H are the checkpoint interval and *Hello* interval, respectively, and K is the number of bits of heartbeat for each router.

DOI: 10.1002/dac

the goal of reducing the takeover delay by stateful backup. In addition, we have implemented the proposed HA-OSPF router on a PC-based platform based on the $N+1$ redundancy model ($N=2$ in our experiments). Experimental results have shown that the takeover delay of the proposed PC-based HA-OSPF router is slightly better than that of Cisco ASR-1000 series router under the same redundancy model (189 vs 200 ms for 2+1 redundancy). However, unlike Cisco ASR-1000 series router, our HA-OSPF router does not need a specific hardware and the redundancy model of the proposed HA router can be adjusted flexibly. From the analytical results and experimental results, we conclude that the proposed 5-tuple availability function can be used to determine the minimum required number of standby routers and the HAM middleware can decrease the takeover delay while meeting the carrier-grade availability and achieving cost-effectiveness.

ACKNOWLEDGEMENTS

This paper was supported in part by the National Science Council of the ROC, under Grants NSC 96-2219-E-009-023 and NSC 96-2219-E009-008.

REFERENCES

1. Budhiraja N, Marzullo K, Schneider FB, Toueg S. *Distributed Systems* (2nd edn). ACM Press, Addison-Wesley: New York, 1993; 199–216.
2. Kuo W, Wan R. Recent advances in optimal reliability allocation. *Studies in Computational Intelligence* 2007; **39**:1–36.
3. Mettas A. Reliability allocation and optimization for complex systems. *Proceedings of the Annual Reliability and Maintainability Symposium*, Los Angeles, CA, U.S.A., January 2000; 216–221.
4. Srivastava S. Redundancy management for network devices. *Proceedings of the Ninth Asia–Pacific Conference on Communications*, Penang, Malaysia, vol. 3, September 2003; 1157–1162.
5. Hinden R. Virtual Router Redundancy Protocol (VRRP). *RFC 3768*, Internet Engineering Task Force (IETF), April 2004.
6. Li T, Cole B, Morton P, Li D. Cisco Hot Standby Router Protocol (HSRP). *RFC 2281*, Internet Engineering Task Force (IETF), March 1998.
7. Li J, Cole B. Standby Router Protocol. *5473599*, U.S. Patent, December 1995.
8. Ranta J. Router redundancy and scalability using clustering. *Seminar on Internetworking*, Helsinki, Finland, Spring 2004.
9. Bommareddy S, Kale M, Chaganty S. 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. *6779039*, U.S. Patent, August 2004.
10. Ho CF, Gupta A, Grandhi M, Bachmutsky A. Router and routing protocol redundancy. *6910148*, U.S. Patent, June 2005.
11. Moy J, Open Shortest Path Protocol (OSPF). *RFC 2328*, Internet Engineering Task Force (IETF), April 1998.
12. Cisco ASR 1000 Series Aggregation Services Router High Availability: Delivering Carrier-Class Services to Midrange Router, Cisco. Available at: http://www.cisco.com/.
13. Juniper Networks. Available at: http://www.juniper.com/.
14. Open Specifications for Service Availability. Available at: http://www.saforum.org/home/.
15. Trivedi KS. *Probability and Statistics with Reliability, Queuing and Computer Science Applications* (2nd edn). Wiley: New York, 2002; 405–504.
16. Stewart W. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press: Princeton, 1994.
17. Gokhale S, Trivedi KS, Analytical models for architecture-based software reliability prediction: a unification framework. *IEEE Transactions on Reliability* 2006; **55**(4):578–590.
18. Lanus M, Lin Y, Trivedi KS. Hierarchical composition and aggregation of state-based availability and performability models. *IEEE Transactions on Reliability* 2003; **52**(1):44–52.
19. Michael G, Hairong S, Ricardo FM, Trivedi KS. Ten fallacies of availability and reliability analysis. *Proceeding of the Fifth International Service Availability Symposium* (*ISAS 2008*), Tokyo, Japan, May 2008.

DOI: 10.1002/dac

20. Trivedi KS, Sathaye A, Ibe O, Howe R. Should I add a processor? *Proceeding of Twenty-third Hawaii International Conference on System Science*, Kailua-Kona, HI, U.S.A., January 1990.
21. 7600 Series Routers, Cisco. Available at: http://www.cisco.com/.
22. Catalyst 6500 Series Switches, Cisco. Available at: http://www.cisco.com/.
23. Oggerino C. *High Availability Network Fundamentals*: *A Practical Guide to Predicting Network Availability*. Cisco Press: Indianapolis, IN, U.S.A., 2001.
24. David C. An Ethernet Address Resolution Protocol. *RFC 826*, Internet Engineering Task Force (IETF), November 1982.
25. GNU Zebra. Available at: http://www.zebra.org/.
26. An Integrated Multiprotocol Network Emulator Simulator (IMNES). Available at: http://www.tel.fer.hr/imunes/.
27. OpenAIS Standard based Cluster Framework. Available at: http://www.openais.org/.

AUTHORS' BIOGRAPHIES

Chia-Tai Tsai received the BS degree in computer science from the Tamkang University and the MS degree in computer information and science from the National Chiao Tung University in 2002 and 2004, respectively. He is currently pursuing the PhD degree in the Department of Computer Science at the National Chiao-Tung University, Taiwan. His research interests include computer networks, network reliability, wireless networks, and operations research.

Rong-Hong Jan received the BS and MS degrees in industrial engineering and the PhD degree in computer science from the 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, MD. His research interests include wireless networks, mobile computing, distributed systems, network reliability, and operations research.

Kuochen Wang received the BS degree in Control Engineering from the National Chiao Tung University, Taiwan, in 1978, and the MS and PhD 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.

DOI: 10.1002/dac