

國立交通大學

資訊科學與工程研究所

博士論文

高可用度路由器設計與實作

Design and Implementation of High Availability Routers



研究生：蔡嘉泰

指導教授：簡榮宏 博士

中華民國 100 年 6 月

# 高可用度路由器設計與實作

Design and Implementation of High Availability Routers

研究生：蔡嘉泰

Student : Chia-Tai Tsai

指導教授：簡榮宏博士

Advisor : Dr. Rong-Hong Jan



Submitted to Department of Computer Science and Engineering  
College of Computer Science  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of  
Doctor of Philosophy  
in  
Computer Science  
June 2011  
Hsinchu, Taiwan, Republic of China

中華民國 100 年 6 月


# 高可用度路由器設計與實作

研究生：蔡嘉泰

指導教授：簡榮宏 博士

國立交通大學 資訊科學與工程研究所

## 摘要



隨著網路技術的進步，人們對於網路的依賴程度也與日俱增，對網路服務提供業者來說，如何提供一個具有高可用度的網路環境，讓使用者在進行網路存取時不會感覺有網路中斷的情形發生，是一個很重要並且亟待解決的問題。在本論文中，我們利用連續時間馬可夫鏈推導得到一個可用度方程式，根據此方程式，當路由器要達到電信服務等級時，網路服務提供業者只需要提供主要路由器個數( $M$ )、路由器錯誤率( $\lambda$ )、路由器修復率( $\mu$ )以及路由器錯誤偵測與回復率( $\delta$ )這四個參數，本方程式就可以計算並且告知需要配置的備用路由器數量( $N$ )。根據數值分析的結果，我們發現錯誤偵測與回復率是用來減少建置備用路由器數量最主要的參數，當錯誤偵測與回復率愈大，備用路由器的需求數量將會減少。

當備用路由器接手封包轉送的工作時，備用路由器會重新與鄰居路由器進行網路連結資訊交換，用以重新建立網路拓樸表，此一動作將會造成封包轉送服務

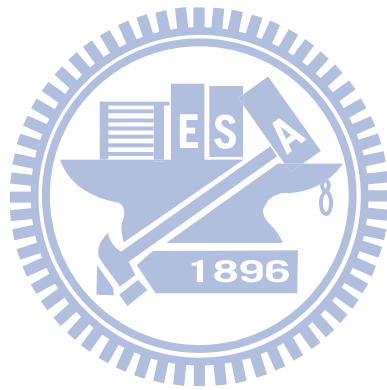
中斷。為了能夠減少網路服務中斷的時間，使得錯誤偵測與回復率能夠增加，我們利用了完整狀態回復(Stateful backup)技術，主要的技術為，當主要路由器在運作時，就會將其網路連結狀態資料庫同步至備用路由器，如此，當備用路由器進行接手封包轉送工作時，備用路由器就可以根據先前收到的網路連結資料庫立刻建立網路拓樸，並且得到路由路徑表，此時，備用路由器便可以立刻上線運作，而不需要再向其他鄰居路由器索取網路連結資料，如此將可以有效地減少備用路由器接手封包轉送的中斷時間。

為了能夠讓主要路由器同步網路連結資料庫至備用路由器，我們參考並修改 OpenAIS 系統，提出了一套高可用度管理中介軟體(HAM middleware)，此中介軟體可以有效地減少備用路由器接手封包轉送時的網路中斷時間，以達到增加錯誤偵測與回復率之目的。

我們將此高可靠度管理中介軟體安裝於個人電腦(PC)的機器上，並實際進行數值量測，以 OSPF 為例，根據實驗結果得知，當備用路由器進行換手時，其網路中斷時間將可以比 Cisco-ASR 1000、Juniper MX 系列路由器與 VRRP 路由器減少約 6%、37.3%與 98.6%。

此外，我們也將此高可用度管理中介軟體安裝於 ATCA 的機器上，ATCA 是一個可以提供工業標準模組化架構的平台，可以提供我們一個高效能、靈活調整與可靠的路由器設計。假設路由器的錯誤率與修復率分別為 7 年與 4 小時，當發生軟體類型的錯誤時，其備用路由器接手封包轉送工作的網路中斷時間為 217 ms，而當發生硬體類型的錯誤時，其中斷時間為 1066 ms。也就是說，架設於 ATCA 的高可用度路由器的可用度為 99.99999905%與 99.99999867%，皆能夠達到電信等級可用度的標準。

根據以上我們可以得知，我們所提的高可用度路由器相較於商用的路由器而言，因為我們所提的路由器是架構於一個開放式標準的規格，所以花費會較少將更具有成本效益，且其備援方式可以根據網路架設與使用狀況更靈活地與有效地進行調整。



# Design and Implementation of High Availability Routers

Student: Chia-Tai Tsai    Advisor: Rong-Hong Jan

Department of Computer Science and Engineering  
National Chiao Tung University



## Abstract

How to optimally allocate redundant routers for high availability (HA) networks is a crucial task. In this dissertation, a *5-tuple availability function*,  $A(M, N, \lambda, \mu, \delta)$ , is proposed to determine the minimum required number of standby routers to meet the desired availability ( $\rho$ ) of an HA router, where  $M$  and  $N$  are the numbers of active routers and standby routers, respectively, and  $\lambda$ ,  $\mu$ , and  $\delta$  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 ( $\delta$ ) 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 to those of the industry standard approaches, the Cisco-ASR 1000 series router, the Juniper MX series router, and the VRRP (Virtual Router Redundancy Protocol) router, 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  $1/\delta = 217 \text{ ms}$  for a software failure and  $1/\delta = 1066 \text{ ms}$  for a hardware failure, along with the router module data,  $1/\lambda = 7 \text{ years}$  and  $1/\mu = 4 \text{ hours}$ , 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 can easily meet the requirement of carrier-grade availabilities with five-nine. In addition, in contrast 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.

## 誌謝

首先感謝指導教授簡榮宏博士歷年來的諄諄教誨與耐心教導，使學生在學術研究與待人處事上受益良多，感謝之情非筆墨所能形容，浩浩師恩，心版永銘，在此獻上最高的敬意與謝忱。同時，學生也非常感謝口試委員交通大學資訊工程學系王國禎教授、張明峰教授與陳健教授、台灣科技大學電機工程學系陳俊良教授、清華大學資訊工程學系石維寬教授、中央大學資訊工程學系周立德教授、台中教育大學資訊科學系張林煌教授對本論文不吝批評與指正，使本論文更臻完善，在此深表感激。

感謝計算機網路實驗室的所有成員在我課業上的提攜與生活上的照顧，也謝謝在求學生涯中，曾經鼓勵與支持我的朋友們。最後，感謝我的家人，在我求學生涯中，無微不至的照顧與關懷，讓我毫無牽掛地專注於學業，給予我最溫暖的愛與不間斷的支持與鼓勵。在此願將完成論文的喜悅與大家分享。





# Contents

<b>Abstract (in Chinese)</b> .....	<b>I</b>
<b>Abstract (in English)</b> .....	<b>IV</b>
<b>Acknowledgements</b> .....	<b>VI</b>
<b>Contents</b> .....	<b>VII</b>
<b>List of Tables</b> .....	<b>IX</b>
<b>List of Figures</b> .....	<b>XI</b>
<b>Chapter 1 Introduction</b> .....	<b>1</b>
<b>Chapter 2 Preliminary</b> .....	<b>6</b>
2.1. Reliability Definitions.....	6
2.2. Availability Definitions.....	7
2.3. Steady-state Availability Definitions .....	8
<b>Chapter 3 HA Router Model Description and Analysis</b> .....	<b>9</b>
3.1. Continuous-Time Markov Chain for 1+N Redundancy Model .....	10

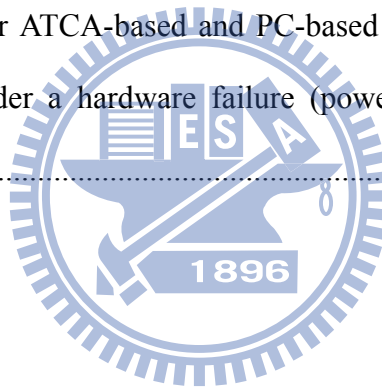


3.2. Continuous-Time Markov Chain for $M+N$ Redundancy Model.....	13
3.3. Formalizing a 5-tuple Availability Function.....	16
<b>Chapter 4 Analytical Results.....</b>	<b>18</b>
4.1. Numerical Analysis of Minimal Required Standby Routers for $1+N$ Redundancy Model .....	18
4.2. Numerical Analysis of Minimal Required Standby Routers for $M+N$ Redundancy Model .....	20
4.3. Computational Complexity .....	22
<b>Chapter 5 Proposed HA Router Design.....</b>	<b>24</b>
5.1. HAM Middleware Design.....	24
5.2. HAM Middleware Operation Procedures .....	27
<b>Chapter 6 Experiments.....</b>	<b>30</b>
6.1. Experimental Setup .....	30
6.2. Experimental Results .....	33
<b>Chapter 7 Field Trial Results.....</b>	<b>39</b>
<b>Chapter 8 Conclusion .....</b>	<b>48</b>
<b>References.....</b>	<b>50</b>

# List of Tables

Table 4.1: 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. ....	20
Table 4.2: The minimum required standby routers ( $N$ ) for an HA router to achieve the goal of carrier-grade availability ( $\rho = 99.999\%$ ).....	21
Table 6.1: Default parameter values.....	33
Table 6.2: Takeover delay ( $ms$ ) of the proposed HA-OSPF router under various redundancy models.....	34
Table 6.3: Takeover delays ( $ms$ ) and failure detection and recovery rates (times/hour) for a HA-OSPF router and a VRRP-based router. ....	35
Table 6.4: Takeover delays ( $ms$ ) and failure detection and recovery rates (times/hour) due to a software failure (OSPF process down) under various polling intervals.....	36
Table 6.5: Takeover delays ( $ms$ ) and failure detection and recovery rates (times/hour) due to a hardware failure under various down check intervals.....	36
Table 6.6: The comparisons of the proposed HA-OSPF router, VRRP router, Cisco ASR-1000 series router, and Juniper MX series router.....	38

Table 7.1: Takeover delays ( <i>ms</i> ), failure detection and recovery rates (times/hour), and availabilities for ATCA-based and PC-based HA-OSPF routers.....	44
Table 7.2: CPU usages of HAM middleware and OSPF process for ATCA-based and PC-based HA-OSPF routers.....	45
Table 7.3: Takeover delays ( <i>ms</i> ), 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.....	46
Table 7.4: Takeover delays ( <i>ms</i> ), 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.....	47



# List of Figures

Figure 3.1: CTMC for an HA router with 1+N redundancy model. ....	10
Figure 3.2: Equivalent Markov chain. ....	12
Figure 3.3: Logical structure and CTMC for an HA Router with $M + N$ redundancy. ....	14
Figure 4.1: The minimum required number of standby routers for an HA router under various numbers of active routers and failure detection and recovery rates (with $p = 99.999\%$ ).....	22
Figure 5.1: The components of an HA router module. ....	26
Figure 5.2: The logical structure of an HA router with 2+1 redundancy. ....	28
Figure 5.3: Link state information backup for a protection group.....	29
Figure 6.1: Experimental environment. ....	31
Figure 7.1: An ATCA-based HA-OSPF router consisting of LC, CC and SC.....	40
Figure 7.2: ATCA-based experimental environment. ....	43

# Chapter 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]-[16]. 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].

In this dissertation, a 5-tuple availability function,  $A(M, N, \lambda, \mu, \delta)$ , is proposed to determine the minimum required number of standby routers in an HA (High Availability) router for achieving the desired availability ( $\rho$ ), where  $M$  and  $N$  are the number of active routers and standby routers, respectively, and  $\lambda$ ,  $\mu$ , and  $\delta$  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.

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.

The industry routers, Cisco ASR-1000 series router [17] and Juniper MX series router [18], 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 [17] and 300 *ms* for Juniper MX series router [18]. 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 is lack of ability for flexible adjustment of the redundancy model [17]. That is, it only supports one active router and one standby router. The Juniper MX series router can adjust the redundancy model flexibility. It supports  $2N$  redundancy,  $M+N$  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 Open Shortest Path First (HA-OSPF) router with High Availability Management

(HAM) middleware which consists of Availability Management Framework (AMF) service [19], Checkpoint service [19], and Failure Manager. The HAM middleware was implemented based on an open source and open architecture project, OpenAIS [19]. 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. In addition, we have implemented an HA-OSPF router and evaluate the takeover delay of the proposed HA-OSPF router in the OSPF network [20].

The rest of this dissertation is organized as follows. We review the preliminary 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. Analytical results are given in section 4. In section 5, we describe the proposed HAM (High Availability Management) middleware design and the procedures of role assignment, routing process status and link state information backup, and failure detection and recovery. Then, in sections 6 and 7, experimental results and field trial results are evaluated and discussed. Finally, we conclude this dissertation in section 8.

# Chapter 2

## Preliminary

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* [21].” 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  $R(t | s) = 1$ ). 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.,  $\lim_{t \rightarrow \infty} R(t) = 0$ ) [22].

Let random variable  $X$  be the lifetime (i.e., time to failure) [22] of a system then

$$R(t) = \Pr(X > t) = 1 - F(t) \quad (1)$$

where  $F(t)$  is the system lifetime CDF (cumulative distribution function) [22]. Moreover, the expected lifetime ( $E[X]$ ) or the mean time to failure of the component is given by [22] is

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

Therefore, the system MTTF can be computed from the equations (1) and (2). Suppose the system lifetime is exponentially distributed (i.e.,  $F(t) = 1 - e^{-\lambda t}$ ) [22] with failure rate  $\lambda$  then

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

and

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

Therefore, if a component obeys an exponential failure rate with parameter  $\lambda$ , then the MTTF (i.e., the expected lifetime [22]) 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, assuming that the external resources, if required, are provided* [21].” Michael et al. [23] 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.,

$$A(t) = \Pr(I(t) = 1) = E[I(t)] \quad (5)$$

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

$$A = \lim_{t \rightarrow \infty} A(t) \quad (6)$$

### 2.3. Steady-state Availability Definitions

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 [22][23][24]:

$$A = \frac{MTTF}{MTTF + MTTR} \quad (7)$$

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  $\lambda$ , and the time-to-repair distribution of the system is exponential with repair rate  $\mu$ , then equation (7) can be rewritten as [22][23][24]

$$A = \frac{\mu}{\lambda + \mu} \quad (8)$$

# Chapter 3

## HA Router Model Description and Analysis

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 [4]. Thus, a feasible way to increase the router availability is to add the standby router to the HA router [5][6][7][9][10]. In this section, we propose a 5-tuple availability function,  $A(M, N, \lambda, \mu, \delta)$ , to determine the minimal number of standby routers ( $N$ ) in an HA router to achieve the desired availability, under the conditions of the failure rate ( $\lambda$ ), repair rate ( $\mu$ ), failure detection and recovery rate ( $\delta$ ), and number of active routers ( $M$ ). The continuous-time Markov chain (CTMC) [22][25][26] is used to determine the steady-state availability of an HA router with various numbers of active routers and standby routers.

### 3.1. 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. Figure 3.1 is the state-transition diagram of a CTMC [22][25][26] modeling the failure and repair behavior of an HA router with 1+N redundancy model (i.e. one active and  $N$  standby). The failure of the active router will cause the network to recalculate routing path information. To avoid this undesirable situation, 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.

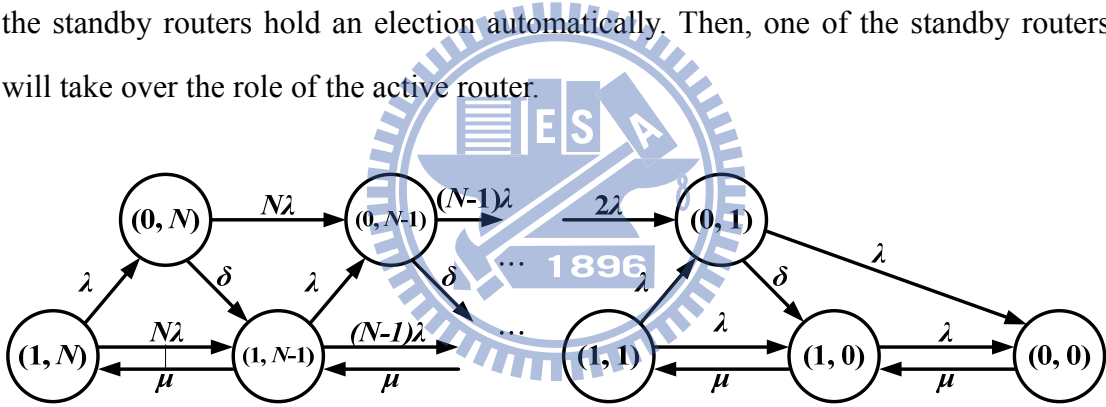


Figure 3.1: CTMC for an HA router with 1+N redundancy model.

As shown in Figure 3.1, 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.

The state diagram of the CTMC modeling the failure and repair behavior of an HA router with  $1+N$  redundancy model is depicted in Figure 3.1. The active router works properly at state  $(1, p)$ , where  $0 \leq p \leq N$ . The state  $(0, q)$  represents the active router failed and the HA router fails (i.e., cannot forward packets). The system detects and recovers the failure with rate  $\delta$  and will go to state  $(1, q-1)$ , where  $1 \leq q \leq N$ . The state  $(0, 0)$  represents that all the router modules of the HA router are failed.

In this dissertation, 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 Figure 3.1, when the state transfers from  $(0, p)$  to  $(1, p-1)$ ,  $0 \leq p \leq N$ , 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 ( $\delta$ ) 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. Note that in this dissertation, 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 Figure 3.1, we obtain the following equations for the steady state probabilities:

$$(N + 1)\lambda \cdot \pi(1, N) = \mu \cdot \pi(1, N - 1) \quad (9)$$

$$(N\lambda + \delta) \cdot \pi(0, N) = \lambda \cdot \pi(1, N) \quad (10)$$

$$\mu \cdot \pi(0, 0) = \lambda \cdot \pi(0, 1) + \lambda \cdot \pi(1, 0) \quad (11)$$



$$(\lambda + \mu) \cdot \pi(1,0) = \lambda \cdot \pi(1,1) + \mu \cdot \pi(0,0) + \delta \cdot \pi(0,1) \quad (12)$$

$$(K\lambda + \delta) \cdot \pi(0,K) = \lambda \cdot \pi(1,K) + (K+1)\lambda \cdot \pi(0,K+1), \text{ where } N-1 \leq K \leq 1 \quad (13)$$

$$(K\lambda + \mu) \cdot \pi(1,K) = (K+1)\lambda \cdot \pi(1,K+1) + \mu \cdot \pi(1,k-1) + \delta\pi(0,K+1), \quad (14)$$

where  $N-1 \leq K \leq 1$

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

$$\sum_{i=0}^1 \sum_{j=0}^N \pi(i,j) = 1 \quad (15)$$

The CTMC for an HA router with  $1+N$  redundancy can transit into a two-state and two-transition Markov chain [27], as shown in Figure 3.2. One state is the *Up* with the reward rate  $\lambda_{HA}$ ; the other state is the *Down* with the reward rate  $\mu_{HA}$  [27].  $\lambda_{HA}$  and  $\mu_{HA}$  are the *equivalent failure rate* and the *equivalent repair rate* of the HA router with  $1+N$  redundancy, which can be determined by applying the aggregation techniques described in [27].

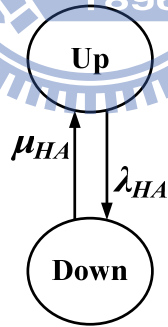


Figure 3.2: Equivalent Markov chain.

Therefore,  $\lambda_{HA}$  and  $\mu_{HA}$  of an HA router for the CTMC in Figure 3.1 can be expressed as follows:

$$\begin{aligned}
\lambda_{HA} &= \frac{\lambda \cdot \pi(1, N) + \lambda \cdot \pi(1, N-1) + \dots + \lambda \cdot \pi(1,1) + \lambda \cdot \pi(1,0)}{\pi(1, N) + \pi(1, N-1) + \dots + \pi(1,1) + \pi(1,0)} \\
&= \frac{\lambda \cdot \left( \sum_{j=0}^N \pi(1, j) \right)}{\sum_{j=0}^N \pi(1, j)} = \lambda
\end{aligned} \tag{16}$$

$$\begin{aligned}
\mu_{HA} &= \frac{\delta \cdot \pi(0, N) + \delta \cdot \pi(0, N-1) + \dots + \delta \cdot \pi(0,1) + \mu \cdot \pi(0,0)}{\pi(0, N) + \pi(0, N-1) + \dots + \pi(0,1) + \pi(0,0)} \\
&= \frac{\delta \cdot \left( \sum_{j=1}^N \pi(0, j) \right) + \mu \cdot \pi(0,0)}{\sum_{j=0}^N \pi(0, j)}
\end{aligned} \tag{17}$$

Therefore, from equation (8), the *equivalent availability* of an HA router ( $A_{HA}$ ) can be expressed as follows:

$$A_{HA} = \frac{\mu_{HA}}{\lambda_{HA} + \mu_{HA}} \tag{18}$$

Solving equations (16) and (17), we can get an equivalent availability of an HA router ( $A_{HA}$ ) based on equation (18) under failure rate ( $\lambda$ ), failure detection and recovery rate ( $\delta$ ), and repair rate ( $\mu$ ).

## 3.2. Continuous-Time Markov Chain for $M+N$ Redundancy Model

In this section, the continuous-time Markov chain (CTMC) of an HA router with  $M+N$  redundancy (i.e.,  $M$  active routers and  $N$  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 3.3 (a) is the logical structure of an HA router with  $M+N$  redundancy. The CTMC for an HA router with  $M+N$  redundancy is depicted in Figure 3.3 (b). The active routers work properly at state  $(M, p)$ , where  $0 \leq p \leq N$ . If the state of an HA router moves from state  $(i, j)$  to state  $(i+1, j-1)$ , it represents there is an active router failed and the system detects and recovers the failure with rate  $\delta$ , where  $0 \leq i \leq M-1$  and  $1 \leq j \leq N$ . State  $(0, 0)$  represents that all routers, including active and standby routers, of the HA router failed.

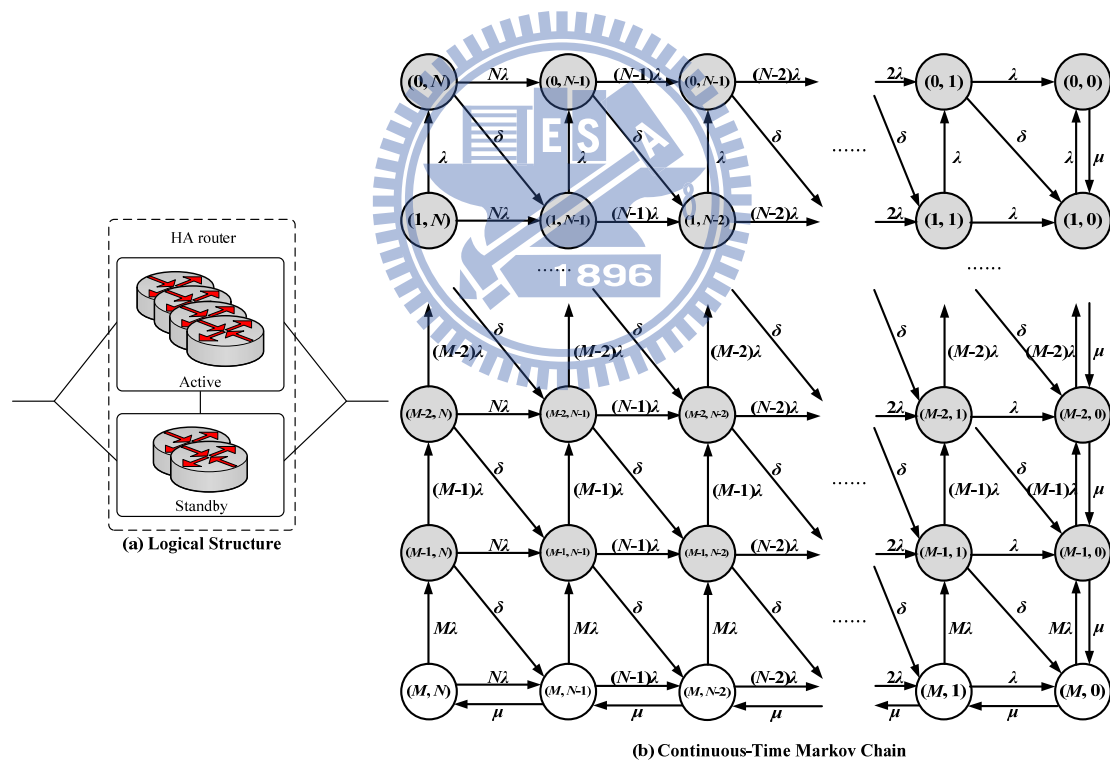


Figure 3.3: Logical structure and CTMC for an HA Router with  $M + N$  redundancy.

After writing the steady state equations and solving these equations, we obtain

the following equations under the steady state:

$$((k + N)\lambda + \delta) \cdot \pi(k, N) = (k + 1)\lambda \cdot \pi(k + 1, N), \text{ where } 0 \leq k \leq M - 1 \quad (19)$$

$$\mu \cdot \pi(k, 0) = (k + 1)\lambda \cdot \sum_{\substack{0 \leq i, j \leq k+1 \\ i+j=k+1}} \pi(i, j), \text{ where } 0 \leq k \leq M - 1 \quad (20)$$

$$\mu \cdot \pi(M, k) = (M + k + 1)\lambda \cdot \sum_{\substack{k+1 \leq i \leq N, 0 \leq j \leq M \\ i+j=M+k+1}} \pi(i, j), \text{ where } 0 \leq k \leq N - 1 \quad (21)$$

$$(k\lambda + \delta) \cdot \pi(0, k) = \lambda \cdot \pi(1, k) + (k + 1)\lambda \cdot \pi(0, k + 1), \text{ where } 1 \leq k \leq N - 1 \quad (22)$$

$$((i + j)\lambda + \delta) \cdot \pi(i, j) = (i + 1)\lambda \cdot \pi(i + 1, j) + (j + 1)\lambda \cdot \pi(i, j + 1) + \delta \cdot \pi(i - 1, j + 1), \quad (23)$$

where  $1 \leq i \leq M - 1$  and  $1 \leq j \leq N - 1$

$$(M + N)\lambda \cdot \pi(M, N) = \mu \cdot \pi(M, N - 1) \quad (24)$$

$$\sum_{i=0}^M \sum_{j=0}^N \pi(i, j) = 1 \quad (25)$$

Therefore,  $\lambda_{HA}$  and  $\mu_{HA}$  can be written as follows:

$$\begin{aligned} \lambda_{HA} &= \frac{M\lambda \cdot \pi(M, N) + M\lambda \cdot \pi(M, N - 1) + \dots + M\lambda \cdot \pi(M, 1) + M\lambda \cdot \pi(M, 0)}{\pi(M, N) + \pi(M, N - 1) + \dots + \pi(M, 1) + \pi(M, 0)} \\ &= \frac{M\lambda \cdot \left( \sum_{j=0}^N \pi(M, j) \right)}{\sum_{j=0}^N \pi(M, j)} = M\lambda \end{aligned} \quad (26)$$

$$\begin{aligned}
\mu_{HA} &= \frac{\delta \cdot \pi(M-1, N) + \delta \cdot \pi(M-1, N-1) + \dots + \delta \cdot \pi(M-1, 1) + \mu \cdot \pi(M-1, 0)}{\pi(M-1, N) + \pi(M-1, N-1) + \dots + \pi(M-1, 1) + \pi(M-1, 0)} \\
&= \frac{\delta \cdot \left( \sum_{j=1}^N \pi(M-1, j) + \pi(M-1, 0) \right) + \mu \cdot \pi(M-1, 0) - \delta \cdot \pi(M-1, 0)}{\sum_{j=0}^N \pi(M-1, j)} \\
&= \frac{\delta \cdot \left( \sum_{j=0}^N \pi(M-1, j) \right) + (\mu - \delta) \cdot \pi(M-1, 0)}{\sum_{j=0}^N \pi(M-1, j)} \\
&= \delta + \frac{(\mu - \delta) \cdot \pi(M-1, 0)}{\sum_{j=0}^N \pi(M-1, j)} \tag{27}
\end{aligned}$$

Solving equations (26) and (27), we can also get an equivalent availability of an HA ( $A_{HA}$ ) router with  $M+N$  redundancy model based on equation (18) under failure rate ( $\lambda$ ), failure detection and recovery rate ( $\delta$ ), and repair rate ( $\mu$ ).

### 3.3. Formalizing a 5-tuple Availability

#### Function

Based on the above discussion, we propose a 5-tuple availability function,  $A(M, N, \lambda, \mu, \delta)$ , to determine the minimum required number of standby routers ( $N$ ) need to be allocated in an HA router to achieve the desired availability ( $\rho$ ). In addition, as shown in equation (28), the equivalent availability of an HA router ( $A_{HA}$ ) is equal to the derived value of the 5-tuple availability function.

$$A_{HA} = A(M, N, \lambda, \mu, \delta) \quad (28)$$

Therefore, problem P1 can be formally defined as follows:

Problem P1:

Minimize  $N$

subject to

$$A_{HA} = \frac{\mu_{HA}}{\lambda_{HA} + \mu_{HA}} \geq \rho, \text{ where } 0 \leq N \leq M \quad (29)$$

where  $\mu_{HA}$  and  $\lambda_{HA}$  are the equivalent repair rate and equivalent failure rate of an HA router.



# Chapter 4

## Analytical Results

In this section, we want to find the most cost-effective redundancy model for the HA router such that its availability meets the requirement of the carrier-grade availability ( $\rho = 99.999\%$ ). The parameter settings of  $\mu$ ,  $\lambda$ , and  $\delta$  are given as follows. Based on the data from Cisco, we set  $\mu = 0.25$  times/hour (i.e., MTTR ( $1/\mu$ ) is equal to 4 hours). The MTTR of a router is assumed to be the time it takes to have a spare part and a knowledgeable person arrive to repair. Three MTTFs, low MTTF ( $1/\lambda = 10000$  hours), high MTTF ( $1/\lambda = 100000$  hours) and Cisco carrier grade router's MTTF ( $1/\lambda = 61320$  hours) are considered.

### 4.1. Numerical Analysis of Minimal Required Standby Routers for $1+N$ Redundancy Model

Solving equations (16) and (17) we can get the availabilities of an HA router by using equation (18) under various failure detection and recovery rates, and a different number  $N$  of standby routers, as shown in Table 4.1. From Table 4.1, an HA router with  $1 + 1$  redundancy (i.e.,  $N = 1$ ) will meet the five-nine availability if  $\delta$  is greater

than 10 times/hour. In general,  $\delta$  is much larger than 10. For example, in Table 6.3, the  $\delta$  for the VRRP router is at least 248 times/hour and the  $\delta$  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  $\delta$  is 1800 times/hour [17]. 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 ( $\delta$ ) is a key parameter to improve the availability of an HA router. To have high availability,  $\delta$  is the larger the better. Note that, for an HA router with 1+1 redundancy, to obtain five-nine availability, the minimum  $\delta$  is 1.632 times/hour for  $1/\lambda = 7$  years and  $1/\mu = 4$  hours [28]-[30]. In Sections 5 and 6, we will show that the experimental  $\delta$ '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  $\delta$  we just mentioned. For software failures, the experimental  $\delta$ 's are even larger.

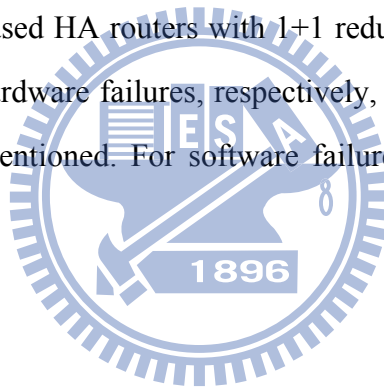




Table 4.1: The availability of an HA router ( $A_{HA}$ ) for a different number of standby routers and various failure detection and recovery rates under  $1/\lambda = 7$  years and  $1/\mu = 4$  hours [28]-[30].

	Failure detection and recovery rate ( $\delta$ ) (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%

## 4.2. Numerical Analysis of Minimal Required Standby Routers for $M+N$ Redundancy Model

The failure detection and recovery rate ( $\delta$ ) is set to 100, 1000, 10000, and 100000 times/hour. In addition, three failure detection and recovery rates which were measured from the proposed HA router, are also considered. Those includes  $\delta=11429$  times/hour for hardware failures only,  $\delta=58065$  times/hour for software failures only, and  $\delta=34747$  times/hour for hardware and software failures (see section 7). The number of active routers  $M$  varies from 1, 2, 4, ..., to 128. Table 4.2 shows the analytical results to determine the minimum required number of standby routers ( $N$ ) for the proposed HA router under various  $\mu$ ,  $\lambda$ ,  $\delta$ , and  $M$ .

Table 4.2: The minimum required standby routers ( $N$ ) for an HA router to achieve the goal of carrier-grade availability ( $\rho = 99.999\%$ ).

$\mu=0.25$ (times/hour)		$M = 1$			$M = 2$			$M = 4$			$M = 8$		
		$1/\lambda(\text{hours})$			$1/\lambda(\text{hours})$			$1/\lambda(\text{hours})$			$1/\lambda(\text{hours})$		
		10000	61320	100000	10000	61320	100000	10000	61320	100000	10000	61320	100000
$\delta$ (times/hour)	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
	100000	1	1	1	1	1	1	1	1	1	2	1	1

$\mu=0.25$ (times/hour)		$M = 16$			$M = 32$			$M = 64$			$M = 128$		
		$1/\lambda(\text{hours})$			$1/\lambda(\text{hours})$			$1/\lambda(\text{hours})$			$1/\lambda(\text{hours})$		
		10000	61320	100000	10000	61320	100000	10000	61320	100000	10000	61320	100000
$\delta$ (times/hour)	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
	100000	2	1	1	2	1	1	3	2	1	3	2	2

X: no feasible solution

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

Figure 4.1 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  $\rho$

being set to 61320 hours, 4 hours (from Cisco [28][29][30]), and 99.999% respectively. Based on Figure 4.1, 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 ( $\rho$ ). 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 4.1.

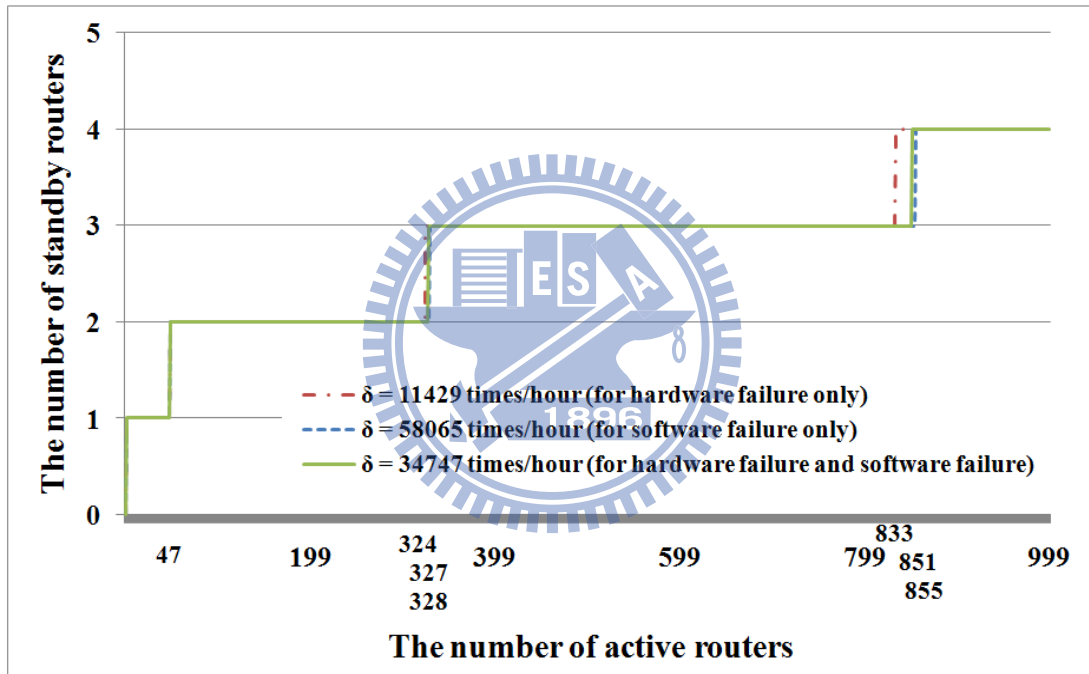
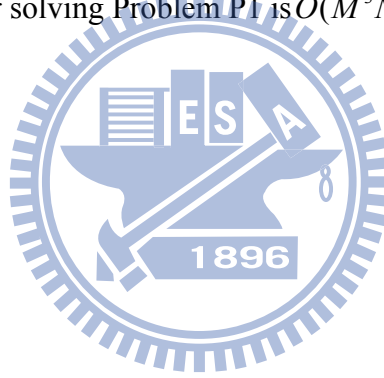


Figure 4.1: The minimum required number of standby routers for an HA router under various numbers of active routers and failure detection and recovery rates (with  $\rho = 99.999\%$ ).

### 4.3. Computational Complexity

To solve Problem P1, we can apply binary search method on  $N$  ( $0 \leq N \leq M$ ).

For a given  $N$ , we evaluate  $A(M, N, \lambda, \mu, \delta)$  and check to see if  $A(M, N, \lambda, \mu, \delta) \geq \rho$  or not. By this way, the minimum value of  $N$  such that  $A(M, N, \lambda, \mu, \delta) \geq \rho$  can be found. In each iteration, we have to solve the equations (19) ~ (25) for evaluating  $A(M, N, \lambda, \mu, \delta)$ . Note that the equations (19) ~ (25) 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 (19) ~ (25) with  $n = (M+1)(N+1)$ . That is, it takes  $O([(M+1)(N+1)]^3) = O((MN)^3)$  time to evaluate  $A(M, N, \lambda, \mu, \delta)$  in each iteration. The number of iterations needed for the binary search is  $O(\log M)$ . Therefore, the total time for solving Problem P1 is  $O(M^3 N^3 \log M)$ .



# Chapter 5

## Proposed HA Router Design

The proposed 5-tuple availability function shows that the failure detection and recovery rate ( $\delta$ ) is a key parameter to increase the availability of an HA router. In order to increase the failure detection and recovery rate, a High Availability Management (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 5.1, 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* [19]. In this dissertation, 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 APIs (Application Programming Interface) 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,  $M+N$  redundancy, and  $N$ -way redundancy. When a router first starts, the AMF service will assign a role, *active* or *standby*, to the router. 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 second, which is a default value), it will assume 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 reduce the takeover delay and improve the availability when it takes over.

## HA Router Module

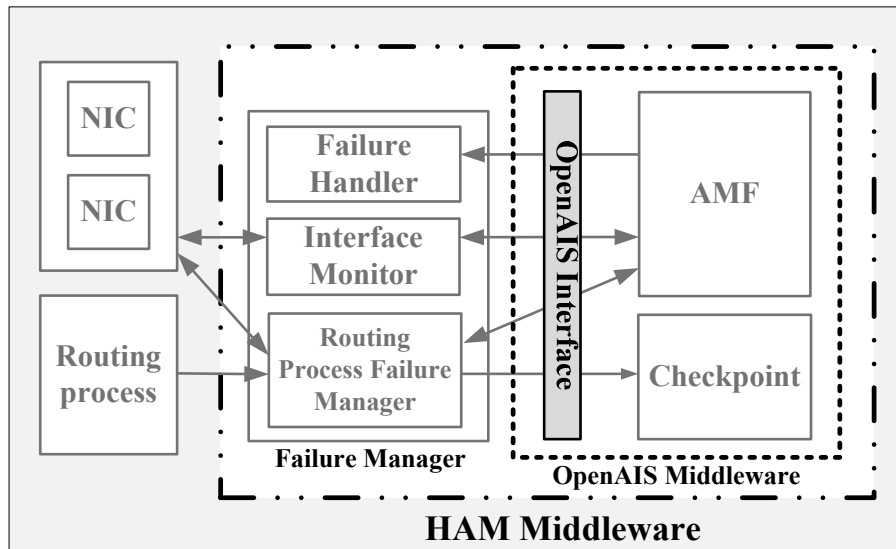


Figure 5.1: The components of an HA router module.

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.

## 5.2. HAM Middleware Operation Procedures

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

- *Role assignment:* We use  $M = 2$  and  $N = 1$  as an example to illustrate an HA router with  $M+N$  redundancy and it can be easily extended to the general case. As shown in Figure 5.2, 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* [19] 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, firstly. 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 [31] 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.

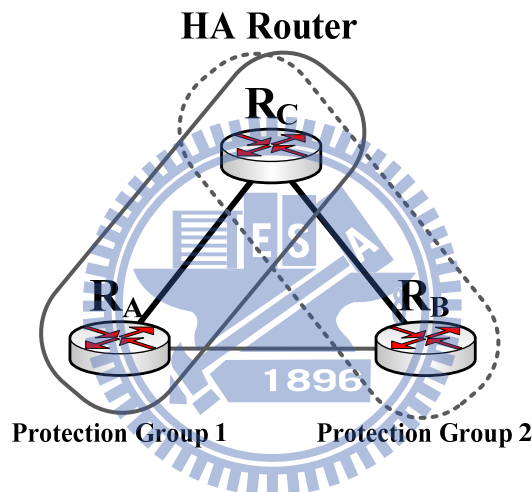


Figure 5.2: The logical structure of an HA router with 2+1 redundancy.

- *Routing process status and link state information backup*: Figure 5.3 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.

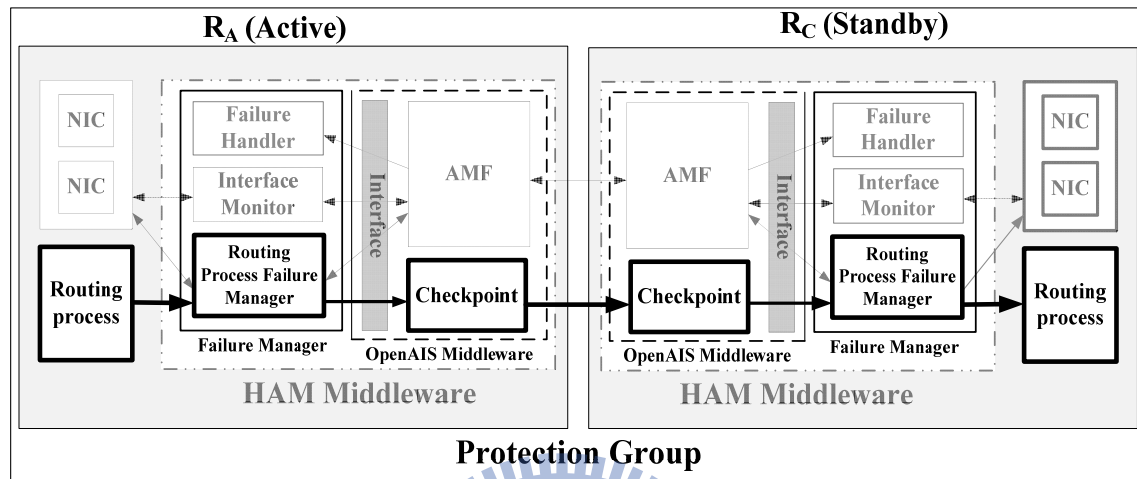
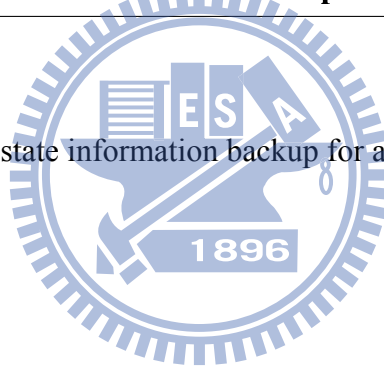


Figure 5.3: Link state information backup for a protection group.



# Chapter 6

## Experiments

In Figure 4.1, we have shown that an HA router with  $M+1$  redundancy (for  $M \leq 47$ ) is the recommended scheme to meet the carrier-grade ( $\rho = 99.999\%$ ) availability under an appropriate failure rate ( $\lambda$ ), failure detection and recovery rate ( $\delta$ ), and repair rate ( $\mu$ ). In this section, we will actually measure the failure detection and recovery rate ( $\delta$ ) of the proposed HA-OSPF router with  $M+1$  redundancy on an OSPF network ( $M = 2$  in our experiments for illustration). We will show 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 [17], and a VRRP router [5]. The takeover delay (the multiplicative inverse of the failure detection and recovery rate) 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.

### 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 6.1. 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 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 which run IMUNES (Integrated Multiprotocol Network Emulator Simulator) [33], 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 6.1.

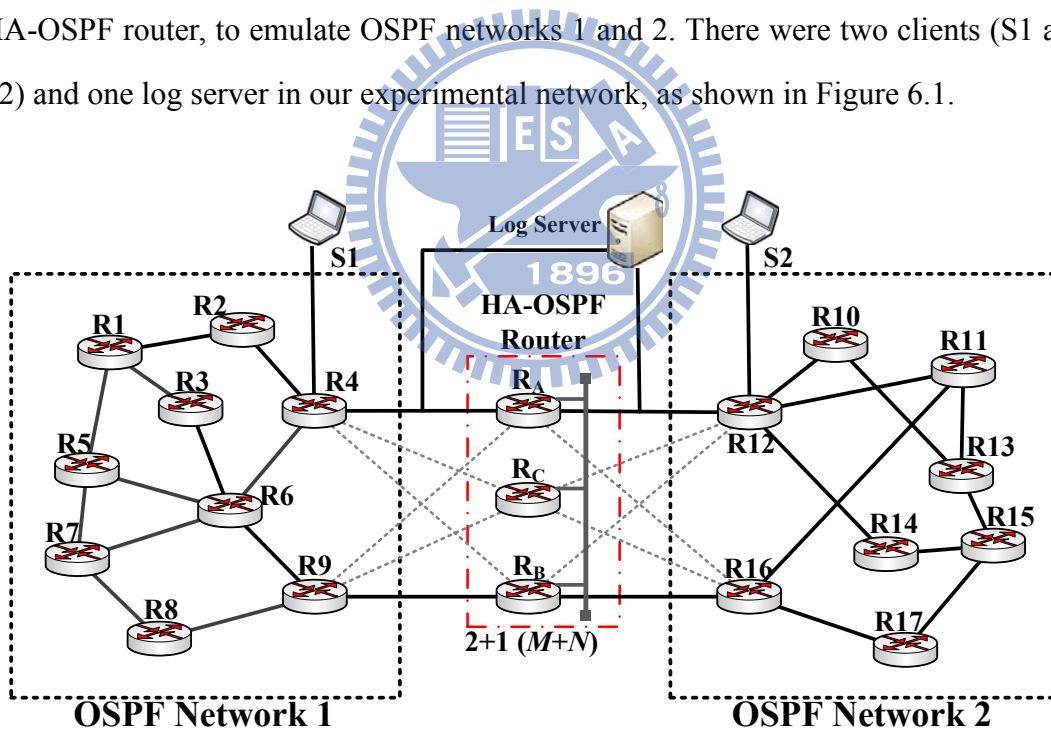


Figure 6.1: Experimental environment.

In the experiment, S1 sent UDP data packets with specific sequence numbers to S2 to examine the network connectivity (see Figure 6.1). 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 6.1 [19][20][28][29][30]. The *Hello* interval is the number of seconds this router waits before sending out the next *Hello* packet [19][20]. 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 seconds (four times the default *Hello* interval) [19][20]. Based on Cisco data, the MTTF ( $1/\lambda$ ) and MTTR ( $1/\mu$ ) of a commercial router need at least 7 years (i.e., 61320 hours) and not exceed 4 hours, respectively [28][29][30]. The default values for the down check interval of AMF service and polling interval of the Failure Manager are 1000 *ms* and 100 *ms*, respectively [34]. The down check interval is a period of time in which the standby router has to hear at least one heartbeat from the active router; otherwise, the standby router assumes it has failed. 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.

Table 6.1: Default parameter values [19][20][28][29][30][34].

Router dead interval of OSPF	40 <i>sec</i>
<i>Hello</i> interval	10 <i>sec</i>
Down check interval of AMF service	1000 <i>ms</i>
Polling interval of Failure Manager	100 <i>ms</i>
MTTF ( $1/\lambda$ )	7 years (61320 hours)
MTTR ( $1/\mu$ )	4 hours

## 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 6.2 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 6.2, 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  $565 \pm 3$  *ms*,  $569 \pm 3$  *ms*, and  $576 \pm 4$  *ms* ( $110 \pm 2$  *ms*,  $112 \pm 3$  *ms*, and  $118 \pm 4$  *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 a more cost-effective configuration, was used to measure takeover delays of the proposed HA-OSPF router in the subsequent experiments.

Table 6.2: Takeover delay (*ms*) of the proposed HA-OSPF router under various redundancy models.

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

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 [17], 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* [17]. 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.
- *Proposed HA-OSPF router with 2+1 redundancy*: Each active router backs up its full state information, including its link states, LSDB (link state database), 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 6.1, UDP packets traveled along path S1, R4, R<sub>A</sub>, 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<sub>C</sub>, 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 6.3. The takeover delay for a hardware failure (a software failure) of the VRRP-based router and the proposed HA-OSPF router were  $14511 \pm 36$  ms and  $569 \pm 3$  ms ( $13383 \pm 3$  ms and  $112 \pm 3$  ms), respectively. Experimental results show that the takeover delays of the proposed HA-OSPF router were reduced by 96.08% and 99.16% compared to 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.

Table 6.3: Takeover delays (*ms*) and failure detection and recovery rates (times/hour) for a HA-OSPF router and a VRRP-based router.

		Emulation Scenario	
		VRRP	HA-OSPF router
Hardware failure	Takeover delay (ms)	$14511 \pm 36$	$569 \pm 3$
	Failure detection and recovery rate (times/hour)	248	6327
Software failure	Takeover delay (ms)	$13383 \pm 3$	$112 \pm 3$
	Failure detection and recovery rate (times/hour)	269	32143

Next, we measured the takeover delay for the PC-based HA-OSPF router due to a software failure under various polling intervals. Table 6.4 shows that the takeover delays (failure detection and recovery rates) due to a software failure were,  $62 \pm 1$  ms ( $\delta = 58065$  times/hour),  $112 \pm 3$  ms ( $\delta = 32143$  times/hour), and  $170 \pm 2$  ms ( $\delta = 21176$  times/hour) for three polling intervals, 50 ms, 100 ms, 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.

Table 6.4: Takeover delays (*ms*) and failure detection and recovery rates (times/hour) due to a software failure (OSPF process down) under various polling intervals.

	Polling interval		
	50 <i>ms</i>	100 <i>ms</i>	200 <i>ms</i>
Takeover delay (ms)	62 ± 1	112 ± 3	170 ± 2
Failure detection and recovery rate (times/hour)	58065	32143	21176

We then investigated the takeover delay of the proposed HA-OSPF router due to a hardware failure under different down check intervals. In Table 6.5, the takeover delays (failure detection and recovery rates) due to a hardware failure under down check intervals of 500 *ms*, 1000 *ms*, and 2000 *ms* were 315 ± 2 *ms*, 569 ± 3 *ms*, and 1087 ± 9 *ms* (11429 times/hour, 6327 times/hour, and 3312 times/hour), respectively. That is, the smaller down check intervals result in the shorter takeover delays.

Table 6.5: Takeover delays (*ms*) and failure detection and recovery rates (times/hour) due to a hardware failure under various down check intervals.

	Down check interval		
	500 <i>ms</i>	1000 <i>ms</i>	2000 <i>ms</i>
Takeover delay (ms)	315 ± 2	569 ± 3	1087 ± 9
Failure detection and recovery rate (times/hour)	11429	6327	3312

Table 6.6 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 which 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 6.6, 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 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 delay of the proposed HA-OSPF router were reduced 6%, 37.3%, and 98.6% compared to 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 VRRP-based router, Cisco ASR-1000 series router, and Juniper MX series router to construct a high availability network.

Table 6.6: The comparisons of the proposed HA-OSPF router, VRRP router, Cisco ASR-1000 series router, and Juniper MX series router.

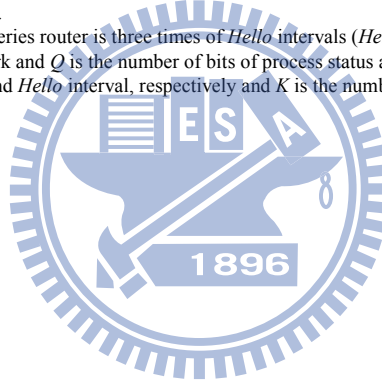
Scheme	HA-OSPF router (proposed)	VRRP router	Cisco ASR-1000 series router	Juniper MX series router
Cost	Medium	Low	Very High	Very High
Takeover delay	189 ms <sup>*1</sup>	13383 ms	about 200 ms	300 ms <sup>*2</sup>
Implementation flexibility	Easy	Easy	Hard (Cisco IOS)	Hard (Juniper JUNOS)
Flexible redundancy 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 <sup>*3</sup>	$((NM) \times P \times Q) / 8$ bytes	No	$(P \times Q) / 8$ bytes	$((NM) \times P \times Q) / 8$ bytes
Bandwidth overhead <sup>*4</sup>	$((NM) \times P \times Q / T_C) + (K / T_H)$ 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

<sup>\*1</sup>  $189 \text{ ms} = (62 \text{ ms} + 315 \text{ ms}) / 2$ , 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.

<sup>\*2</sup> The takeover delay of the Juniper MX series router is three times of *Hello* intervals (*Hello* interval is 100 ms ~ 65535 ms).

<sup>\*3</sup>  $P$  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.

<sup>\*4</sup>  $T_C$  and  $T_H$  are the checkpoint interval and *Hello* interval, respectively and  $K$  is the number of bits of heartbeat for each router.



# Chapter 7

## 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 [34][35] allows new communication equipment to be constructed with great attributes such as high performance, high availability, adaptability for adding new features, and lower cost of ownership. An open architecture solution using 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 Figure 7.1 [34][35]. Based on the operating function of ATCA cards and the concepts of ForCES (Forwarding and Control Element Separation) [36][37][38], 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.

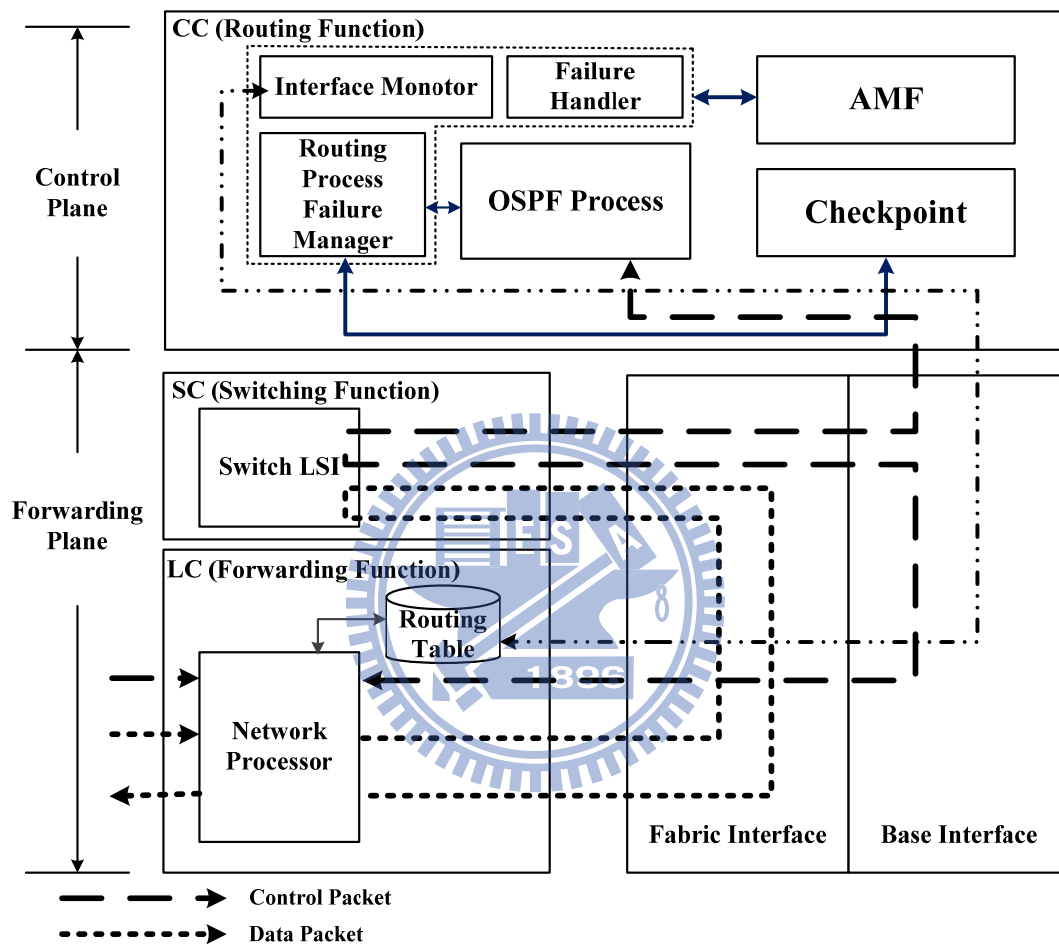


Figure 7.1: An ATCA-based HA-OSPF router consisting of LC, CC and SC [34][35].

The details of each ATCA card are described as below [34][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

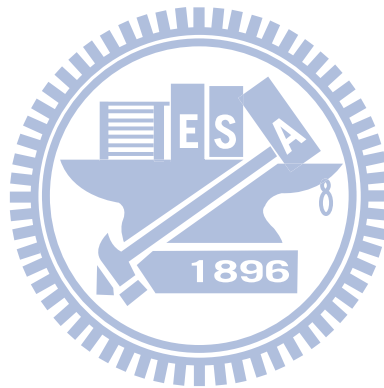
packets from its neighbor router, the LC will forward the packets to the control card. Then, if the LC receives the packets, it will forward the packets to correct destinations according to the routing table.

- 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 Figure 7.1, 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 [35], 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.

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 [34][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 Figure 7.2). The configuration of control cards on the ATCA is the same as that on the PC-based system.



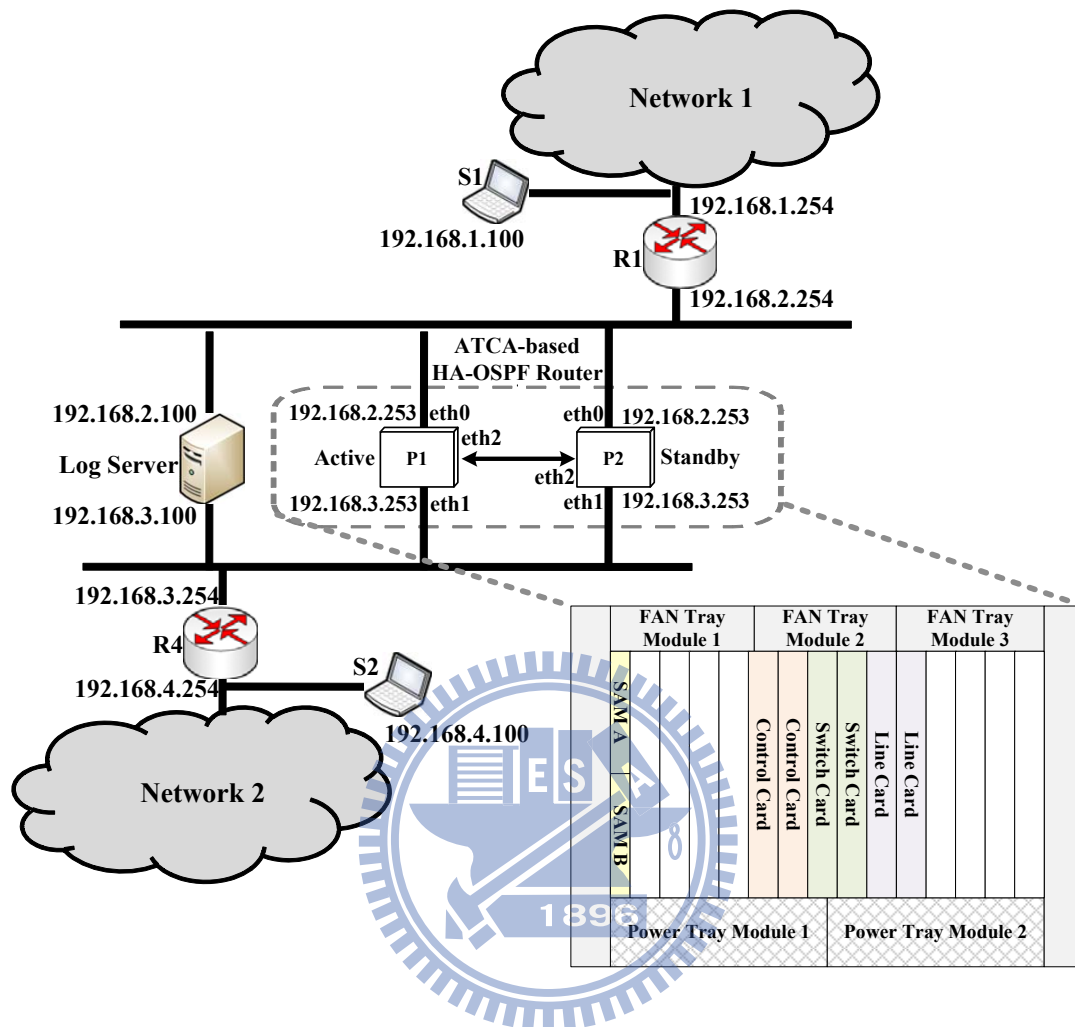


Figure 7.2: ATCA-based experimental environment.

Based on the default parameter values in Table 6.1, we measured takeover delays of the ATCA-based HA-OSPF router with 1+1 redundancy, and experimental results are shown in Table 7.1. The takeover delays of the PC-based HA-OSPF router from Table 6.3 are also included in Table 7.1 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  $1066 \pm 54 \text{ ms}$  ( $\delta =$



3377 times/hour) and  $217 \pm 17$  ms ( $\delta = 16590$  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 [28][29][30]. 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.

Table 7.1: Takeover delays (ms), failure detection and recovery rates (times/hour), and availabilities for ATCA-based and PC-based HA-OSPF routers.

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

According to Table 7.2, 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 7.2: 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	$0.11 \pm 0.01 \%$	$4.47 \pm 0.73 \%$

Table 7.3 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  $188 \pm 9 \text{ ms}$  ( $\delta = 19149$  times/hour),  $217 \pm 17 \text{ ms}$  ( $\delta = 16590$  times/hour), and  $242 \pm 26 \text{ ms}$  ( $\delta = 14876$  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 [39], the average failure recovery time of the ATCA-based HA-OSPF router (about  $150 \text{ ms}$ ) is greater than that of the PC-based HA-OSPF router (about  $100 \text{ ms}$ ). 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 7.3: 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.

		Polling interval ( <i>ms</i> )		
		50 <i>ms</i>	100 <i>ms</i>	200 <i>ms</i>
( $1/\lambda=7$ years, $1/\mu = 4$ hours)				
ATCA-based	Takeover delay ( <i>ms</i> )	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 ( <i>ms</i> )	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 7.4 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.

Table 7.4: 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/λ=7 years, 1/μ = 4 hours)		Down check interval		
		1000 <i>ms</i>	500 <i>ms</i>	200 <i>ms</i>
ATCA-based	Takeover delay ( <i>ms</i> )	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 ( <i>ms</i> )	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%

From Table 7.3 and Table 7.4, the experimental results show that the failure detection and recovery rates ( $\delta$ ) 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. 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  $\delta$  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.

# Chapter 8

## Conclusion

We have presented a 5-tuple availability function,  $A(M, N, \lambda, \mu, \delta)$ , to relate to the desired availability ( $\rho$ ), where  $M$ ,  $N$ ,  $\lambda$ ,  $\mu$ , and  $\delta$  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 continuous-time Markov chain 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 ( $\delta$ ) 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 replicate its routing process status and link state information to the standby routers. The HAM (High Availability Management) middleware, which includes AMF (Availability Management Framework) service, Checkpoint service, Failure Manager, has also been proposed. It has been integrated to the proposed HA router to achieve 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 *ms* 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. In addition, we have also implemented the HA-OSPF router on an ATCA 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 can easily achieve the goal of carrier-grade availability with five-nine. 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.

# References

- [1] N. Budhiraja, K. Marzullo, F. B. Schneider, and S. Toueg, Distributed Systems, 2<sup>nd</sup> Edition, 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,” *The 9th Asia-Pacific Conference on Communications*, Vol. 3, Sept. 2003, pp. 1157 – 1162.
- [4] A. Mettas “Reliability Allocation and Optimization for Complex Systems,” *Proceedings of the Annual Reliability and Maintainability Symposium*, Jan. 2000, pp. 216 – 221.
- [5] R. Hinden, “Virtual Router Redundancy Protocol (VRRP),” RFC 3768, Internet Engineering Task Force (IETF), Apr. 2004.
- [6] T. Li, B. Cole, P. Morton, and D. Li, “Cisco Hot Standby Router Protocol (HSRP),” RFC 2281, Internet Engineering Task Force (IETF), Mar. 1998.
- [7] J. Li, and B. Cole, “Standby Router Protocol,” 5473599, United State Patent, Dec. 1995.
- [8] N. Dennis, H. Michael, D. Peter, and M. John, “Method and System for Router Redundancy in a Wide Area Network,” 7554903, United State Patent, Jun. 2009.
- [9] J. Ranta, “Router Redundancy and Scalability Using Clustering,” *Seminar on Internetworking*, Spring 2004.
- [10] S. Bommarreddy, 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,” 6779039, United State Patent, Aug. 2004.
- [11] C.T. Tsai, R.H. Jan, C. Chen, and C.Y. Huang, “Implementation of Highly Available OSPF router on ATCA,” *The 13<sup>th</sup> IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'07)*, Dec. 2007

- [12] C.F. Ho, A.Gupta, M. Grandhi, and A. Bachmutsky, "Router and Routing Protocol Redundancy," 6910148, United State Patent, June 2005.
- [13] T. Bourke, Server Load Balancing, 1st Edition, O'Reilly Media, Aug. 2001.
- [14] N. Milanovic and B. Milic, "Automatic Generation of Service Availability Models," *IEEE Transactions on Services Computing*, vol. 4, no. 1, Jan 2011, pp. 56 – 29.
- [15] E.A.P Alchieri, A.N. Bessani, J.D.S. Fraga, "A Dependable Infrastructure for Cooperative Web Services Coordination," *IEEE International Conference on Web Services*, Sept. 2008.
- [16] V. Ermagan, I. Kruger, M. Menarini, "A Fault Tolerance Approach for Enterprise Applications," *IEEE International Conference on Services Computing*, July 2008.
- [17] Cisco ASR 1000 Series Aggregation Services Router High Availability: Delivering Carrier-Class Services to Midrange Router, Cisco, <http://www.cisco.com/>.
- [18] Juniper Networks, <http://www.juniper.com/>.
- [19] Open Specifications for Service Availability, <http://www.saforum.org/home/>.
- [20] J. Moy, "Open Shortest Path Protocol (OSPF)," RFC 2328, Internet Engineering Task Force (IETF), Apr. 1998.
- [21] ITU-T (1993), "Terms and Definitions Related to Quality of Service and Network Performance Including Dependability," *ITU-T Rec. E. 800*, Aug. 1994.
- [22] K.S. Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, 2<sup>nd</sup> Edition, John Wiley and Sons, Inc, New York, 2002, pp. 405 – 504.
- [23] G. Michael, S. Hairong, F.M. Ricardo, and K.S. Trivedi, "Ten Fallacies of Availability and Reliability Analysis," *The 5<sup>th</sup> International Service Availability Symposium (ISAS 2008)*, May 2008.
- [24] K.S. Trivedi, A. Sathaye, O. Ibe, and R. Howe, "Should I Add a Processor?" *Proceeding Twenty-third Hawaii International Conference on System Science*, 1990.
- [25] W. Stewart, "Introduction to the Numerical Solution of Markov Chains," Princeton Univ. Press, 1994.
- [26] S. Gokhale and K.S. Trivedi, "Analytical Models for Architecture-Based Software Reliability Prediction: A Unification Framework," *IEEE Transactions on Reliability*, Dec. 2006, pp. 578 – 590.
- [27] M. Lanus, Y. Lin, and K.S. Trivedi, "Hierarchical Composition and Aggregation of State-Based Availability and Performability Models," *IEEE Transactions on Reliability*, 52(1), 2003, pp. 44 – 52.



- [28] Telcordia Technologies, Local Access and Transport Area Switching Systems Generic Requirements (LSSGR): Reliability Section 12, Issue 2, Telcordia, Jan. 1998.
- [29] F. Hawley and Telcordia Technologies, "Network Reliability Definitions and Requirements," *IWPC Last Mile Workshop*, July. 2002.
- [30] C. Oggerino, High Availability Network Fundamentals: a Practical Guide to Predicting Network Availability, Cisco Press, 2001.
- [31] C. David, "An Ethernet Address Resolution Protocol," RFC 826, Internet Engineering Task Force (IETF), Nov. 1982.
- [32] GNU Zebra, <http://www.zebra.org/>.
- [33] An Integrated Multiprotocol Network Emulator Simulator (IMNES), <http://www.tel.fer.hr/imunes/>.
- [34] OpenAIS Standard based Cluster Framework, <http://www.openais.org/>.
- [35] AdvancedTCA Specifications for Next Generation Telecommunications Equipment, <http://www.picmg.org/v2internal/newinitiative.htm>.
- [36] IETF ForCES (Forwarding and Control Element Separation), <http://www.ietf.org/>.
- [37] 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," *The Sixth International Conference on Networking (ICN)*, Apr. 2007.
- [38] X. Wu and L. Dong, "Research and Design of the Pseudo-VRRP based High Availability Mechanism in the ForCES Router," *The Eighth International Conference on Networking (ICN)*, Mar. 2009.
- [39] M. Aoki, K. Habara, T. Hamano, K. Ogawa, and S. Chaki, "ATCA-based Open-Architecture Router Prototype," *IEICE Transactions on Communications*, E89-B(5), 2006, pp. 1685–1687.