Fault-Tolerant Reconfigurable Ethernet-Based IP Network Proxy

Jan Baranski
Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, MS 39762, USA
email: jdb2@ece.msstate.edu

Marshall Crocker
Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, MS 39762, USA
email: mac1@ece.msstate.edu

Georgios Lazarou
Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, MS 39762, USA
email: glaz@ece.msstate.edu

ABSTRACT
Using commonly available hardware and software, we present a proxy scheme for IP over Ethernet networks that provides a fault-tolerant solution without the need for modification of existing networking equipment. This type of fault-tolerant reconfigurable Ethernet-based proxy (FREP) is transparent to current applications and provides full redundancy with minimal packet loss and fast reconfiguration times. A prototype implementation yielded a reconfiguration time of 1.55 sec. Routing delays from the OSPF dynamic routing protocol, however, increased the apparent interruption of service to an average of 10.2 sec when tested in a three-subnet/three-router testbed network. An almost instantaneous recovery time of < 0.02 sec was observed in all cases. The proposed proxy scheme can be deployed in any network based on a topology that allows two connections between a subnet and the backbone. The solution relies on a dynamic routing protocol to provide backbone-level routing around the malfunctioning inter-network link.

KEY WORDS
Fault-tolerance, Ethernet, IP Network, Proxy, Reconfigurability

1 Introduction

Inexpensive and widely available hardware has made Ethernet one of the most widely used link layer protocols in today's IP networks [1]. Recently, Ethernet over fiber has also allowed longer distances than were previously possible. Fault tolerance, however, still remains a major issue in IP over Ethernet networks. IP networks built on Ethernet technology can typically be constructed using bus, star, or tree topologies. These topologies inherently allow for uninterrupted network service during the addition, removal, or failure of network nodes. They do not, however, provide a redundant connection to parent networks in a subnetted environment. This lack of fault tolerance prevents Ethernet-based IP networks from being used in situations that require a high-availability network solution.

A number of redundancy schemes for Ethernet networks have been designed and tested over the past few years [2, 3, 4]. Most of these designs, however, require modification of networking equipment (switches, hubs, NICs, etc.) or require software modification of the individual network nodes. We present an approach that focuses on a simple design mated with a specific, but flexible network topology. With this design, we have attempted to create a "plug-in" solution to typical IP over Ethernet networks that provides transparent redundancy with a minimal traffic footprint. We have focused on a design that is capable of being applied to unmodified, off-the-shelf hardware and software.

In a star topology or tree topology-based network, a parent network is accessible to its subnets via a single router. Modern standards such as the FDDI protocol address this issue; these technologies, however, often require expensive hardware and complex configurations. An Ethernet-based IP network can also be provided multiple connections to a parent network by mating the typical star topology to a central ring-shaped backbone as shown in Fig. 1. Although the central ring is not a necessary component to our redundancy scheme, it is one of the most widely used WAN backbone topologies as well as one of the most practical methods of providing two connections to an Ethernet subnet from the backbone. The central ring provides two routes to the parent network from each star-shaped subnet. This type of configuration is typically not useful to an unmodified IP over Ethernet network, however. In order to take advantage of this network architecture, our design employs a fault-tolerant reconfigurable Ethernet-based proxy (FREP), which allows traffic to be forwarded to a backup router in the event of a primary router failure. This solution provides a redundant connection to the backbone with only

*This work was supported by the Office of Naval Research (ONR).

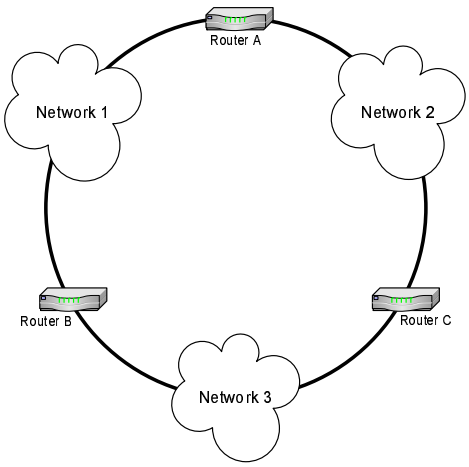

Figure 1. A hybrid star-ring network topology.

the addition of the FREP device and without any modification of individual network nodes. The need for one FREP device per network also makes this an easily scalable solution. FREPs can be added only to networks that require fault-tolerance, while other networks remain unchanged.

The paper is organized as follows. In Section 2, we present a general overview of our design. Section 3 contains an in depth description of a prototype FREP implementation. Section 4 evaluates the performance of the FREP's reconfiguration operations, and we conclude in Section 5.

2 Design Overview

The design of the transparent redundancy scheme is based on a normally configured IP network. In the typical IP over Ethernet network, nodes on a specific subnet are unable to communicate with any "outside" nodes when the router for that particular subnet becomes unreachable. The proposed scheme uses the FREP to mimic the primary router for a particular subnet in case of a failure.

The FREP plays no role in packet routing during normal operation, although it repeatedly polls the primary router to ensure connectivity. This mode of operation is illustrated in Fig. 2. If the primary router is determined to be unreachable the FREP assumes the router's identity and transparently forwards all outbound packets to the backup router associated with that particular subnet. This "failover" mode of operation is shown in Fig. 3.

Inbound traffic must also be routed properly into the subnets during times of a primary router failure. A dynamic IP routing protocol such as OSPF[1] ensures that the routers constantly possess an accurate view of the network and that inbound packets are properly routed around the failed link. This "transparent proxy" design scheme ensures proper traffic flow into and out of individual subnets during failover operation without any modification of existing network hardware or software.

While operating in failover mode, the FREP constantly listens for the presence of the primary router. Status of the connection to the primary router is monitored at the link layer to ensure minimal overhead. Once the primary router responds and is determined to be reachable again, the FREP relinquishes control back to the router and normal routing of traffic into and out of the subnet is resumed. By using a link layer protocol, we are also able to implement a "flap detection" mechanism, which allows the FREP to maintain control of the router's identity during times of state flapping[2].

3 Prototype Implementation

The FREP was implemented using a Linux-based computer with two network interface cards and custom software to perform the failover operations. The software was written in C and is responsible for the control of the FREP. It is a user-space program that runs as a daemon. The following information is collected by the software upon initial execution:

- IP address to assign to each network interface (*eth0* and *eth1*)
- The local subnet mask
- IP address and hardware MAC address of the primary router
- IP address of the backup router
- A number of parameter values that affect the frequency of various actions performed by the FREP

A complete listing of all configuration options is available in Appendix I. Once the values have been properly input, a configuration file is written for future use. The FREP then proceeds to perform connectivity checks to the primary router at specified intervals. If the primary router is determined to be unreachable, the FREP enters a failover mode in which it forwards all traffic destined for the primary router to a backup router. While in failover mode, the FREP also monitors the status of the primary router and relinquishes control once it becomes reachable again. All of the operations performed by the FREP are illustrated in Fig. 4.

Testing and implementation of the FREP took place in a testbed network consisting of three subnet/router pairs. The network was designed using a hybrid star-ring topology as described in section 1. Netspec [5] software was

[1] OSPF is suggested because of its fast convergence rate. Other dynamic routing protocols, however, such as RIP and EIGRP can be used as well.

[2] State flapping refers to the repeated changing of a state in a particular system. In this case, state flapping refers to the repeated changing state of the connection to a primary router. Thus, flap-detection allows us to maintain the primary router's identity and continue forwarding traffic to a backup router until the primary router's connection has stabilized.

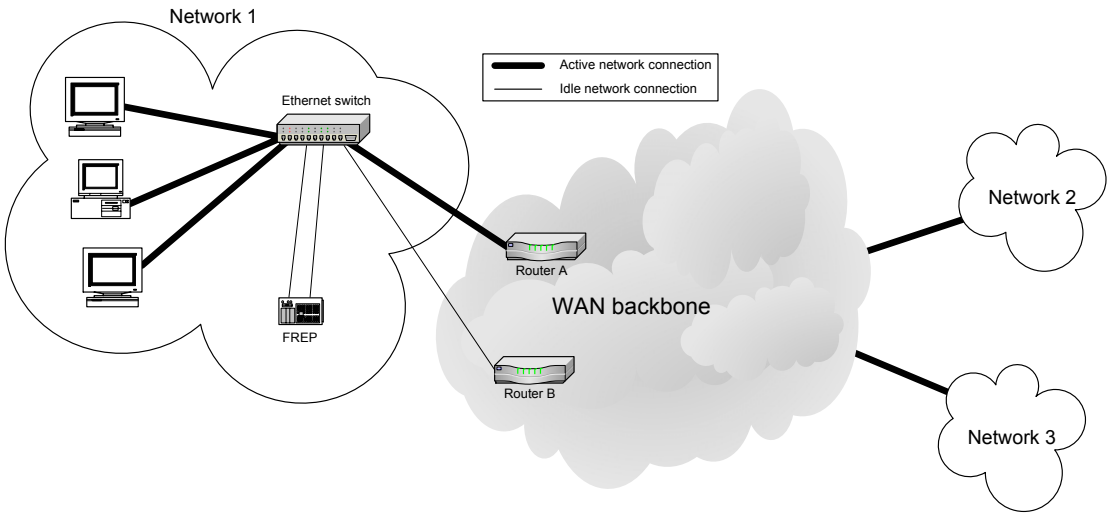

Figure 2. A subnet equipped with a FREP during typical operation. All traffic to/from the subnet is sent directly to the primary router, which is constantly polled by the FREP to determine connectivity status.

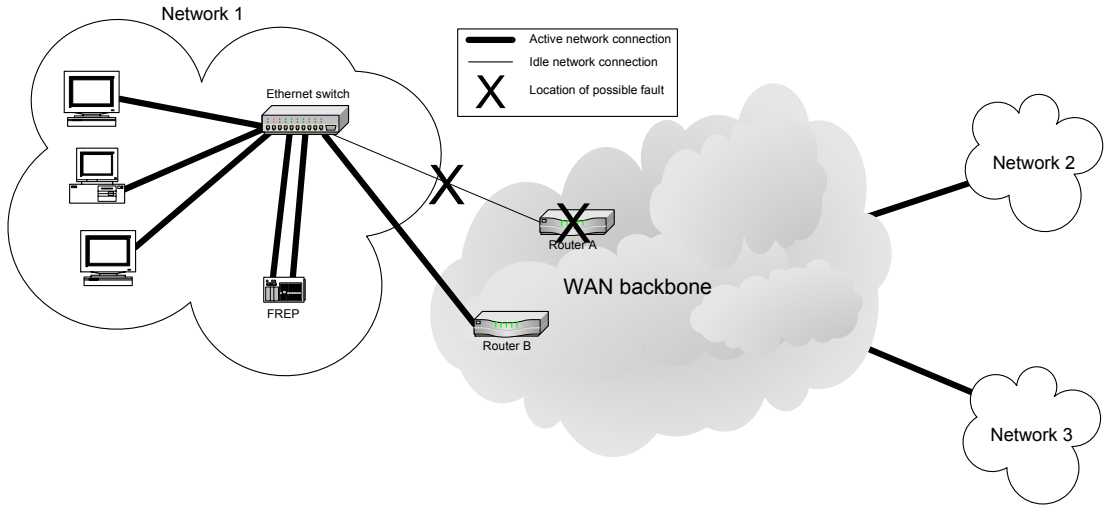

Figure 3. A subnet equipped with a FREP during failover operation. The primary router has become unreachable due to failure and the FREP has taken over its identity. All traffic sent from the nodes to the primary router is transparently forwarded to the backup router via the FREP. All inbound traffic enters the subnet through the backup router after reconfiguration in the backbone occurs via the dynamic routing protocol.

used to place a load on the network during testing to more closely simulate a real-world network. Testing of the FREP is discussed in detail in Section 4.

3.1 Connectivity Checks

The connectivity checks performed by the FREP consist of a `connect()` system call with a stream socket that points to the echo port[3] of the router. The `connect()` call is set to time out after a specified interval. The timeout operation is achieved via synchronous I/O multiplexing by way of a `select()` system call. This method of attempting to establish a connection to the echo port ensures that the primary router is reachable via TCP traffic[4]. Most hardware and software routers available today are configured with the echo service disabled. This type of configuration yields a "Connection refused" error upon a connection attempt and can be used to determine whether the router is reachable.

[3] Port 7 is the IANA assigned port number of the echo service [6].

[4] The FREP software can be modified to perform other types of checks such as ICMP or UDP pings. The TCP protocol was chosen to most closely match the traffic used in the connectivity check with the type of traffic that would typically be handled by a router.

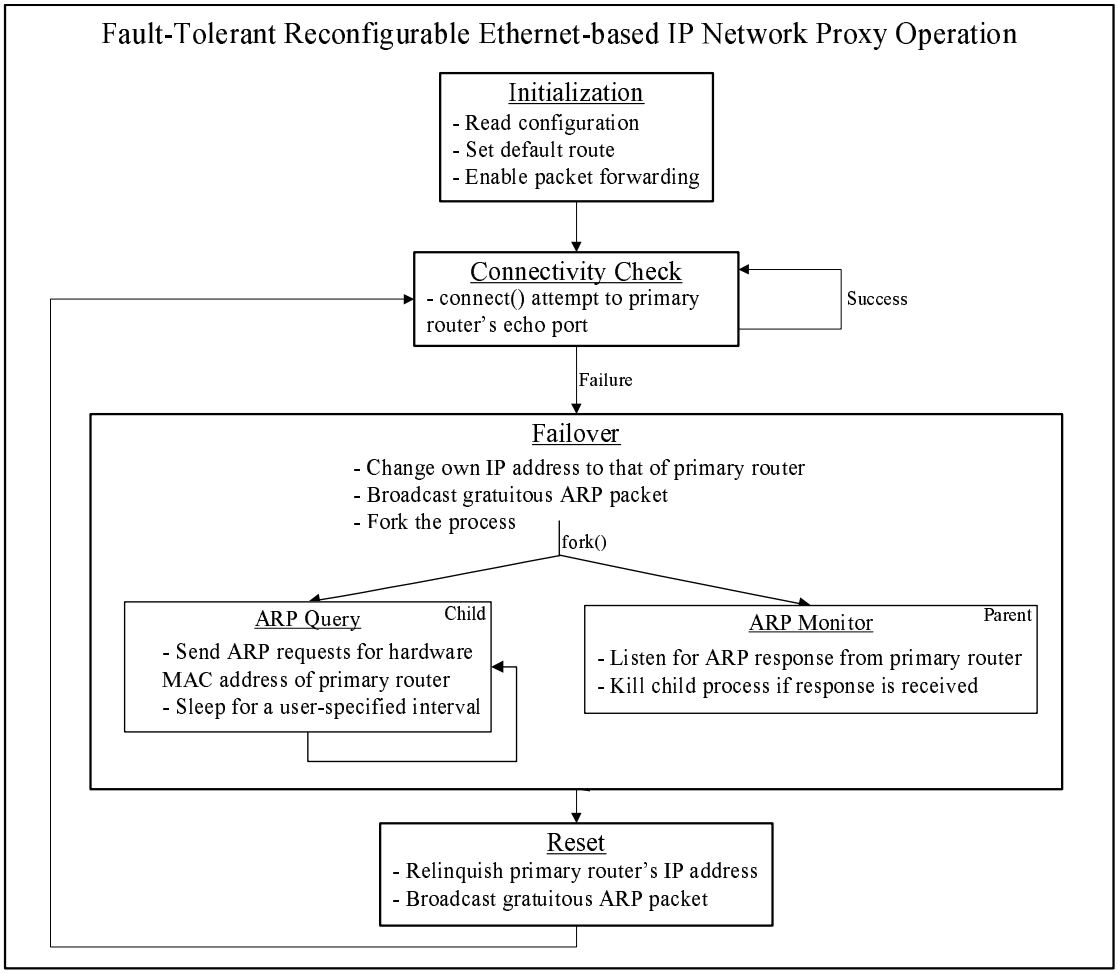

Figure 4. Sequence of operations performed by the FREP.

Thus, connectivity to the primary router is verified if the FREP is either allowed to establish a connection or if it receives the "Connection refused" error. If the connectivity check succeeds the software continues to perform checks at the user-specified interval. Otherwise, the FREP enters failover mode and its network interfaces are reconfigured to mimic the router.

3.2 "Failover" Mode

Once the switch to failover mode occurs, the *eth0* interface is assigned the IP address of the primary router. Since the primary router's IP address is now associated with a different hardware MAC address, this change must be propagated throughout the network. This operation is performed by sending a gratuitous ARP packet[5] to the Ethernet broadcast address. This ensures that the ARP caches of all listening network nodes are updated with the new MAC address. At this point, all outbound packets sent to the primary router are now being delivered directly to the FREP. Once the packets are received by the FREP, a routing decision is made by the kernel. Since no static routes exist in the routing table, outbound packets are sent via the default route, which was inserted earlier. Thus, without changing the IP address of the router, all packets sent to the primary router on a particular subnet are now forwarded to a backup router via the transparent proxy.

3.3 Recovery

While operating in failover mode, the FREP constantly monitors the status of the primary router. The monitoring operation is performed by two separate processes spawned via a `fork()` system call. The child process is responsible for sending out ARP queries for the MAC address of the router at specified intervals. The ARP queries are sent via the low level packet interface of the Linux kernel and are constructed using the `arphdr` structure and addressed using a `sockaddr_ll` structure. The parent process lis-

[5] A gratuitous ARP packet is an ARP request or reply that forces all listening nodes to update their ARP cache. The packet sender and target addresses are both set to the IP address of the cache entry that is to be updated; the sender hardware address is set to the hardware address to which the entry should be updated. [7]

tens for a response from the router using various methods from the pcap packet capture library [8]. Once a response is received, the child process is killed and control is returned to the caller. The FREP then relinquishes the router's IP address and resumes performing TCP connectivity checks.

4 Performance Analysis

The performance of the FREP software and hardware is highly dependent upon the user defined parameters that are set in the program's configuration file. The following parameters can be changed by the user and have a direct effect on the operating characteristics of the FREP:

- Interval between TCP connectivity checks
- Maximum allowed timeout of TCP connectivity check
- Maximum number of failures allowed for TCP connectivity check before a switch to failover mode occurs
- Interval between recovery checks (ARP requests for the primary router's MAC address)

By modifying these parameters the user has total control over the behavior of the FREP. If high availability, for example, is not an issue the interval between connectivity checks can be increased to a higher value. This setting prevents the FREP from frequently attempting to connect to the router, while still providing redundancy in the event of a failure. Thus, the trade-off that the user is faced with is response time versus network overhead. Although the overall network footprint of the FREP device is minimal, performing frequent connectivity checks on a busy network may not be desired.

Performance of the FREP was measured using the Tcpdump packet capture tool [9], which is available with most standard Linux distributions. The effects of a network failure and a recovery performed by the FREP device were observed using TCP, UDP, and ICMP traffic. All of these tests were performed using a 1 sec interval between TCP connectivity checks. It should be noted that increasing this parameter by a particular amount has the effect of increasing the minimum values reported here by that same amount.

The time required for the FREP to complete the reconfiguration operation was measured as the time between the initial loss of connectivity to the primary router and the time of arrival of the first packet of traffic at the FREP. This time was measured to be 1.55 sec and is representative of the time required by the FREP to forward traffic to the backup router from the initial time of failure. It was observed, however, that the apparent interruption of network service to the local subnet was significantly greater and a result of the dynamic routing protocol employed to perform inter-network routing. The tests described here were performed in a testbed network that consists of three subnet/router pairs as shown in Fig. 1. By scaling down the network to a two router/subnet pair architecture, the 1.55 sec. reconfiguration time can be observed between any two nodes and is the "raw" reconfiguration rate of the FREP. To more closely mimic a production network, however, we have included results from our larger testbed network, which produces additional delays.

4.1 ICMP Traffic

The characteristics of the duration of the loss of network service were first analyzed using ICMP pings. The pings were performed from a local node to a distant node on another subnet and from a distant node back into the local subnet. 64-byte packets were sent at a rate of 1 packet/sec. The maximum average interval between a ping request and a ping reply during reconfiguration to failover mode was 10.22 sec[6]. These data reflect the duration of the apparent loss of service in our testbed network. Once packets are forwarded to the backup router of a subnet by the FREP, a further delay is introduced by the routers on the backbone ring of the network. The additional delay introduced by the routers is directly dependent on the amount of time it will take for a link state update to propagate properly through the network.

The maximum average delay during the recovery operation was measured to be 0.017 sec. When an ARP reply is received from the primary router during failover operation, the FREP software reconfigures a network interface to allow the router to regain control of the subnet. The initial destination of network packets, however, is ultimately controlled by the sending devices. Thus, after connectivity to the router has been re-established, the FREP continues to receive and forward packets until the sending devices update their ARP caches with the hardware MAC address of the router. As a result of this behavior, an almost instantaneous switchover occurs when a sending device updates its ARP cache and resumes sending data directly to the router. Recovery operation rate was determined to be independent of network topology or routing protocol. Thus, long convergence times of dynamic routing protocols will not affect the duration of unavailability of network connectivity during the FREP recovery operation.

4.2 TCP and UDP Traffic

Initial experiments performed with TCP and UDP traffic showed that network performance varies greatly with the specific application used. For example, all TCP-based protocols tested suffered only from an interruption of service and were able to maintain a connection after connectivity was re-established, despite the loss of some packets during the failover operation. UDP protocols suffered from packet loss and some datagrams required re-transmission. Most UDP-based applications, however, possess error-detection

[6]This data was acquired from our testbed network while running the OSPF routing protocol with one routing area and the following parameters: hello interval = 1 sec., dead-interval = 4 sec., spf-delay = 5 sec., spf-holdtime = 10 sec.

and are able to re-transmit data and recover from an interruption of network services. Thus, it was concluded that TCP and UDP-based traffic will suffer from an interruption of network connectivity for an average of 10.22 sec, but no assumptions can be made about any actual data loss due to application dependence.

5 Conclusion

5.1 Alternative Methods

Some alternative methods of achieving fault-tolerance via a proxy were also researched. These methods included ICMP redirects, hardware MAC address takeover, and an active proxy scheme. The current method was chosen based on its ease of implementation and efficiency.

5.1.1 ICMP Redirects

Using the method of ICMP redirects is a simple way of redirecting traffic destined for a router. One must ensure, however, that all of the equipment connected to the network is able to accept ICMP redirects and obey them, which may not always be possible. Thus, this technique of forwarding traffic violated the goal of designing a completely transparent redundancy scheme.

5.1.2 MAC Address Takeover

A form of MAC address takeover is implemented in our current scheme. ARP spoofing is used to re-associate IP addresses with different hardware addresses. A similar approach involves changing the MAC addresses of network devices themselves. This approach, however, still requires updates to be made to the ARP tables in devices such as Ethernet switches before any changes will take effect and is difficult to implement.

5.1.3 Active Proxy

Another alternative technique employs an "active" proxy to forward network traffic appropriately. The proxy is placed between the nodes on the local subnet and the two routers for that subnet. Based on connection status, packets are forwarded to the appropriate router by the proxy. This method, however, forces all traffic to be constantly piped through another device, regardless of connection status. The result is an additional amount of latency that is added to all traffic flowing into and out of the subnet. An advantage of the method, however, is that reconfiguration can take place with very little delay.

5.2 Applications

Our prototype implementation of the FREP was based on a standard PC. Production quality applications of this device, however, could be scaled down to a much smaller size. The FREP could, for example, be based on a small form factor PC or other small computer appliance and be specifically tailored for deployment in network closets and server rooms. Furthermore, as layer 3 network switches become more popular, additional functionality is readily being built into the devices. One such addition could be the implementation of a FREP directly inside of an Ethernet switch. This type of switch would be able to provide redundancy with even faster reconfiguration rates than the current FREP implementation by directly routing packets around a failed link.

5.3 Summary

The described proxy scheme provides an inexpensive and simple way to integrate redundancy into an existing IP over Ethernet network. It is not, however, an ideal solution for true high-availability networks. In these types of networks, more specific solutions such as [4] are often employed. Our proposed FREP scheme provides a transparent method of allowing nodes on a subnet to communicate with a backup router in the event that connectivity with the primary router is lost. The failover operation was shown to have an average duration of 1.55 sec. In a larger three subnet/router hybrid star-ring network, however, the downtime was increased to 10.22 sec due to routing delays. Complete functionality was restored with a minimal (< 0.02 sec) recovery operation taking place once connectivity to the primary router was re-established. The duration of the recovery operation was found to be independent of network topology or routing protocol. Using our scheme, additional redundancy can be provided to an Ethernet network without modification of existing networking equipment or software.

Appendix I: Configuration Options

The following configuration parameters are used by the FREP software:

- `Interface0IP` - IP address of FREP Ethernet interface *eth0*
- `Interface1IP` - IP address of FREP Ethernet interface *eth1*
- `Netmask` - Netmask of the particular subnet being monitored
- `RouterIP` - IP address of primary router
- `RouterMAC` - Hardware MAC address of primary router
- `BackupRouterIP` - IP address of backup router
- `CheckInterval` - Interval between TCP connectivity checks
- `PingTimeout` - Maximum allowable timeout for TCP connectivity check
- `AllowedFailures` - Number of connectivity checks that are allowed to fail before switching to failover mode

- `RecoveryInterval` - Interval at which ARP requests are sent by the FREP for the primary router's MAC address in failover mode

References

[1] L. L. Petersen and B. S. Davie, *Computer Networks: A Systems Approach, Second Edition*. (San Francisco, CA: Morgan Kaufman Publishers, 2000).

[2] S. Song, J. Huang, P. Kappler, R. Freimark, and T. Kozlik, "Fault-Tolerant Ethernet Middleware for IP-Based Process Control Networks," *Proc. 25th Annual IEEE Conference on Local Computer Networks*, Tampa, Florida, USA, 2000, 116-125.

[3] J. Huang, S. Song, L. Li, P. Kappler, R. Freimark, J. Gustin, and T. Kozlik, "An Open Solution to Fault-Tolerant Ethernet: Design, Prototyping, and Evaluation," *Proc. IEEE International Performance, Computing, and Communications Conference*, Phoenix/Scottsdale, Arizona, USA, 1999, 461-468.

[4] Linux-HA Development Team. High Availability Linux Project. [Online] Available: http://www.linux-ha.org

[5] R. Jonkman. NetSpec. [Online] Available: http://www.ittc.ku.edu/netspec/

[6] Internet Assigned Numbers Authority. Port Numbers. [Online] Available: http://www.iana.org/assignments/port-numbers

[7] Internet Engineering Task Force (IETF) and the Internet Engineering Steering Group (IESG). RFC 3220. [Online] Available: ftp://ftp.rfc-editor.org/in-notes/rfc3220.txt

[8] V. Jacobson, C. Leres, and S. McCanne. TCPDUMP Public Repository: libpcap-0.6.2. [Online]. Available: http://www.tcpdump.org/release/libpcap-0.6.2.tar.gz

[9] V. Jacobson, C. Leres, and S. McCanne. TCPDUMP Public Repository: tcpdump-3.6.2. [Online]. Available: http://www.tcpdump.org/release/tcpdump-3.6.2.tar.gz