International Journal of Research in Computer Science
ISSN 2249-8257 Volume 1 Issue 1 (2011) pp. 75-88

www.ijorcs.org

Fast IP Network Recovery using MRC

T.K Rajesh[1], N. Sreenivasulu[2], K.V.Ragavender[3]

[1]Department of CSE, TKR College of Engineering & Technology, Hyderabad
Email: tk.rajesh006@gmail.com
[2]Alfa College Of Engineering College And Technology, Allagadda, Kurnool
E Mail: Sreenivas006@Yahoo.Com
[3]Associate Prof, TKR College of Engineering & Technology, Hyderabad
E Mail: raghukotla@gmail.com

Abstract

Internet takes vital role in our communications infrastructure, due to slow convergence of routing protocols after network failure become a budding problem. To assure fast recovery scheme from link and node failure in networks, we present a new recovery scheme called **Multiple Routing Configuration** (MRC). Our anticipated scheme guarantees recovery in all single failure scenarios, using a mechanism to handle both link and node failures, and without knowing the root cause of the failure. MRC is strictly connectionless, and assumes only destination based hop-by-hop forwarding. MRC is based on keeping additional routing information in the routers, and allows packet forwarding to continue on an alternative output link immediately after the detection of a failure. In this paper we present MRC, and analyze its performance with respect to load distribution after a failure. We also show how an estimate of the traffic demands in the network can be used to improve the distribution of the recovered traffic, and thus reduce the chances of congestion when MRC is used.

Keywords: MRC, Availability, computer network reliability, communication system routing, protection.

I. Introduction

In recent years the Internet has been transformed from a special purpose network to an omnipresent platform for a wide range of everyday communication services, in the same manner the demands on Internet reliability and availability have increased accordingly.

A. Functions

i) Internet communication must continue despite loss of networks or gateways.
ii) The Internet must support multiple types of communications service.
iii) The Internet architecture must accommodate a variety of networks.
iv) The Internet architecture must permit distributed management of its resources.
v) The Internet architecture must be cost effective.
vi) The Internet architecture must permit host attachment with a low level of effort.
vii) The resources used in the internet architecture must be accountable

The ability to recover from failures has always been a central design goal in the Internet. IP networks are essentially robust, since IGP routing protocols like OSPF are designed to update the forwarding information based on the changed topology after a failure. This re-convergence assumes full distribution of the new link state to all routers in the network domain. When the new state information is distributed, each router individually calculates new valid routing tables.

B. Demerits in Existing Technologies

i. The IP re-convergence is a time consuming process, and a link or node failure is typically followed by a period of routing instability. During this period, packets may be dropped due to invalid routes.
ii. The IGP convergence process is slow because it is *reactive* and *global*. It reacts to a failure after it has happened, and it involves all the routers in the domain.

C. Proposed Scheme:

We present a new scheme for handling link and node failures in IP networks. Multiple Routing Configurations (MRC) is a *proactive* and *local* protection mechanism that allows recovery in the range of milliseconds MRC allows packet forwarding to continue over pre-configured alternative next-hops immediately after the detection of the failure. Using MRC as a first line of defense against network failures, the normal IP convergence process can be put on hold. This process is then initiated only as a result of non-transient failures. Since no global re-routing is performed, fast failure detection mechanisms like fast hellos or hardware alerts can be used to trigger MRC without compromising network stability. MRC guarantees recovery from any single link or node failure, which constitutes a large majority of the failures experienced in a network. MRC makes no assumptions with respect to the *root cause of failure*, e.g., whether the packet forwarding is disrupted due to a failed link or a failed router.

The main idea of MRC is to use the network graph and the associated link weights to produce a small set of back-up network configurations. MRC assumes that the network uses shortest path routing and destination based hop-by-hop forwarding. This gives great flexibility with respect to how the recovered traffic is routed. The backup configuration used after a failure is selected based on the failure instance, and thus we can choose link weights in the backup configurations that are well suited for only a subset of failure instances.

II. MRC Overview

MRC is based on building a small set of backup routing configurations that are used to route recovered traffic on alternate paths after a failure. The backup configurations differ from the normal routing configuration in that link weights are set so as to avoid routing traffic in certain parts of the network.

MRC approach is threefold:

i. We create a set of backup configurations.
ii. A standard routing algorithm like OSPF is used to calculate configuration specific shortest paths and create forwarding tables in each router.
iii. We design a forwarding process that takes advantage of the backup configurations to provide fast recovery from a component failure.

We construct the backup configurations so that for all links and nodes in the network, there is a configuration where that link or node is not used to forward traffic. Thus, for any single link or node failure, there will exist a configuration that will route the traffic to its destination on a path that avoids the failed element. Also, the backup configurations must be constructed so that all nodes are reachable in all configurations, i.e., there is a valid path with a finite cost between each node pair.

Using a standard shortest path calculation, each router creates a set of configuration-specific forwarding tables. For simplicity, so that a packet is forwarded according to a configuration, meaning that it is forwarded using the forwarding table calculated based on that configuration. In this paper has a separate forwarding table for each configuration, but more efficient solutions can be found in a practical implementation. It is important to stress that MRC does not affect the failure-free original routing, i.e., when there is no failure, all packets are forwarded according to the original configuration, where all link weights are normal. Upon detection of a failure, only traffic reaching the failure will switch configuration. All other traffic is forwarded according to the original configuration as normal.

III. GENERATING BACKUP CONFIGURATIONS

Configurations Structure:

MRC configurations are defined by the network topology, which is the same in all configurations, and the associated link weights, which differ among configurations. We formally represent the network topology as a graph G = (N, A), with a set of nodes N and a set of unidirectional links (arcs) A. In order to guarantee single-fault tolerance, the topology graph

G must be bi-connected.

In generating backup configuration we will first detail the requirements that must be put on the backup configurations used in MRC. Then we propose an algorithm that can be used to automatically create such configurations. The algorithm will typically be run once at the initial start-up of the network, and each time a node or link is permanently added or removed. We use the notation shown in Table.1.

Table 1: Notation

G = (N,A)	Graph comprising nodes N and directed links (arcs) A
Ci	The graph with link weights as in configuration i
Si	The set of isolated nodes in configuration Ci
Bi	The backbone in configuration Ci
A(u)	The set of links from node u
(u, v)	The directed link from node u to node v
$p_i(u, v)$	A given shortest path between nodes u and v in Ci

N(p)	The nodes on path p
A(p)	The links on path p
Wi(u, v)	The weight of link (u, v) in configuration Ci
Wi(p)	The total weight of the links in path p in configuration Ci
Wr	The weight of a restricted link
n	The number of configurations to generate (algorithm input)

Definition: A *configuration* Ci is an ordered pair (G,wi) of the graph G and a function wi : A → {1, . . . ,wmax,wr,∞} that assigns an integer weight wi(a) to each link a € A.

Algorithm 1

Creating backup configurations.

1. **for** i € {1 . . . n} **do**
2. Ci ← (G, w_0)
3. Si ← Ø
4. Bi ← Ci
5. **end**
6. Q_n ← N
7. Q_a ← Ø
8. i ← 1
9. **while** Q_n ≠ Ø **do**
10. u ← first (Q_n)
11. j ← i
12. **repeat**
13. **if** connected (B_i \ ({u}, A (u))) **then**
14. C_{tmp} ← isolate (C_i, u)
15. **if** C_{tmp} ≠ null **then**
16. C_i ← C_{tmp}
17. S_i ← S_i Ụ{u}
18. B_i ← B_i \ ({u},A(u))
19. i ← (i mod n) + 1
20. **until** u € S_i **or** *i=j*
21. **if** u not € S_i **then**
22. Give up and abort
23. **end**

The number and internal structure of backup configurations in a complete set for a given topology may vary depending on the construction model. If more configurations are created, fewer links and nodes need to be isolated per configuration, giving a richer (more connected) backbone in each configuration. On the other hand, if fewer configurations are constructed, the state requirement for the backup routing information storage is reduced. However, calculating the minimum number of configurations for a given topology graph is computationally demanding. One solution would be to find all valid configurations for the input consisting of the topology graph G and its associated normal link weights w0, and then find the complete set of configurations with lowest cardinality.

Description: Algorithm 1 loops through all nodes in the topology, and tries to isolate them one at a time. A link is isolated in the same iteration as one of its attached nodes. The algorithm terminates when either all nodes and links in the network are isolated in exactly one configuration, or a node that cannot be isolated is encountered. We now specify the algorithm in detail, using the notation shown in Table. 1.

Main loop: Initially, n backup configurations are created as copies of the normal configuration. A queue of nodes (Qn) and a queue of links (Qa) are initiated. The node queue contains all nodes in an arbitrary sequence. The link queue is initially empty, but all links in the network will have to pass through it. Method first returns the first item in the queue, removing it from the queue.When a node u is attempted isolated in a backup configuration Ci, it is first tested that doing so will not disconnect Bi according to definition. The connected method at line 13 decides this by testing that each of us neighbors can reach each other without passing through u, an isolated node, or an isolated link in configuration Ci.

IV. Local Forwarding Process

When a packet reaches a point of failure, the node adjacent to the failure, called the *detecting node*, is responsible for finding a backup configuration where the failed component is isolated. The detecting node marks the packet as belonging to this configuration, and forwards the packet. From the packet marking, all transit routers identify the packet with the selected backup configuration, and forward it to the egress node avoiding the failed component.

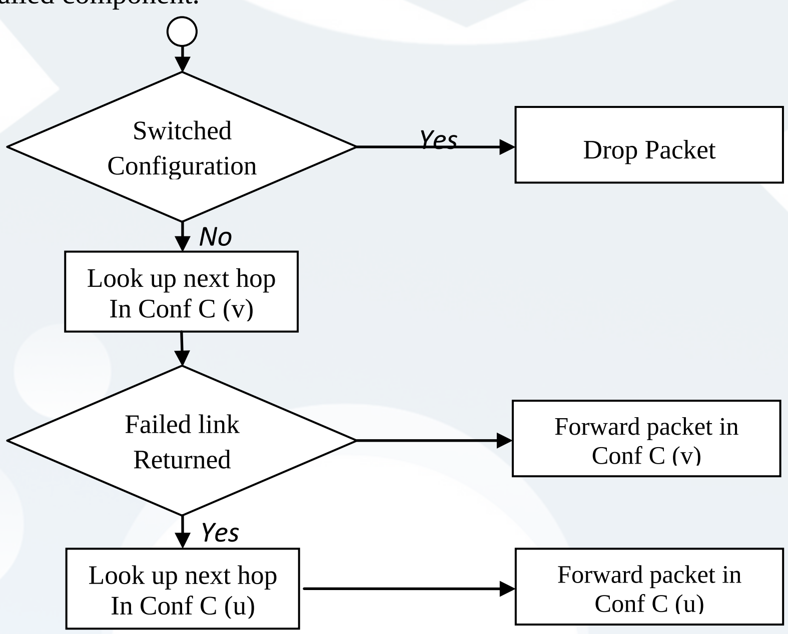

Fig. 1: Packet forwarding state diagram.

Implementation issues:

The forwarding process can be implemented in the routing equipment as presented above, requiring the detecting node u to know the backup configuration C(v) for each of its neighbors. Node u would then perform at most two additional next-hop look-ups in the case of a failure. However, all nodes in the network have full knowledge of the structure of all backup configurations. Hence, node u can determine in advance the correct backup configuration to use if the normal next hop for a destination d has failed. This way the forwarding decision at the point of failure can be simplified at the cost of storing the identifier of the correct backup configuration to use for each destination and failing neighbor. The Table 2 also shows how many nodes that are covered by LFAs, and the number of configurations needed when MRC is used in combination with LFAs. Since some nodes and links are completely covered by LFAs, MRC needs to isolate fewer components, and hence the number of configurations decreases for some topologies.

Table 2: Number of backup configurations for selected real world

Network	Nodes	Links	Confs	LFA	Confs
Sprint US (POP)	32	64	4	17	4
Sprint US (R)	284	1882	5	186	5
Geant	19	30	5	10	4
COST239	11	26	3	10	2
German Telecom	10	17	3	10	-
DFN	13	37	2	13	-

The results show that the number of backup configurations needed is usually modest; 3 or 4 is typically enough to isolate every element in a topology. No topology required more than six configurations. In other words, Alg. 1 performs very well even in large topologies. The algorithm fails only if it meets a node that if isolated disconnects the backbone in each of the n backup configurations. The algorithm often goes through all network nodes without meeting this situation even if n is low, and is more successful in topologies with a higher average node degree. The running time of our algorithm is modest; about 5 seconds for the router level Sprint US network.

V. Recovery Load Distribution

MRC recovery is local, and the recovered traffic is routed in a backup configuration from the point of failure to the egress node. This shifting of traffic from the original path to a backup path affects the load distribution in the network, and might lead to congestion. In our experience, the effect a failure has on the load distribution when MRC is used is highly variable. In this section, we describe an approach for minimizing the impact of the MRC recovery process on the post failure load distribution. If MRC is used for fast recovery, the load distribution in the network during the failure depends on three factors:

(a) The link weight assignment used in the normal configuration C0,

(b) The structure of the backup configurations, i.e., which links and nodes are isolated in each Ci € {C1, . . . , Cn},

(c) The link weight assignments used in the backbones B1, . . . , Bn of the backup configurations.

Algorithm 2

Load-aware backup configurations.

```
for i € {1 . . . n} do
C_i ← (G,w_0)
S_i ← Ø
end
Q_n ← N
assign_C_T (Q_n, γ, ascending)
Q_a ← Ø
while Q_n ≠ Ø do
u ← first (Q_n)
i = C_T (u)
j ← i
repeat
if connected (B_i \ ({u},A (u))) then
C_tmp ← isolate (C_i, u)
if C_tmp ≠ null then
C_i ← C_tmp
S_i ← S_i Ụ {u}
B_i ← B_i \ ({u},A(u))
else
i ← (i mod n) + 1
until u € S_i or i=j
if u not € Si then
Give up and abort
end
```

A. Evolution

To evaluate our load aware construction algorithm, we compute the worst case load on each link after a link failure, and compare it to the results achieved by the original algorithm. We focus on the most important contributions aimed at restoring connectivity without a global re-convergence. Tab. 3 summarizes important features of the different approaches

TABLE 3: Conceptual comparison of different approaches for fast IP Recovery

Scheme	Guaranteed in bi-connected	Node faults	Pre-configured	Link faults	Connection less	Failure agoinstic	Last hop
MRC	YES	YES	YES	YES	YES	YES	YES
Not via tunneling	YES	YES	YES	YES	YES	YES	YES
Local rerouting	no	no	yes	no	yes	N/A	N/A
FIR	YES	no	YES	YES	YES	N/A	N/A
FIFR	YES	YES	YES	YES	YES	YES	no
LFA	no	YES	YES	YES	YES	YES	YES
MPLS FRR	YES	YES	YES	YES	no	no	N/A
Rerouting OSPF	YES	YES	YES	no	YES	YES	YES

VI. Results & Discussions

Screenshots:

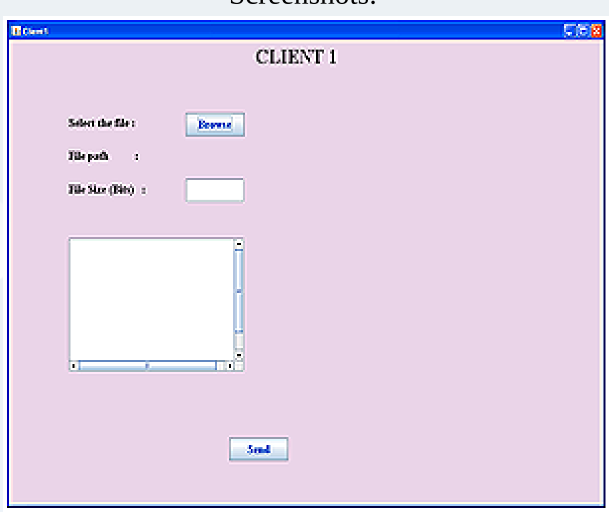

Fig 2:Client 1tranceives the data

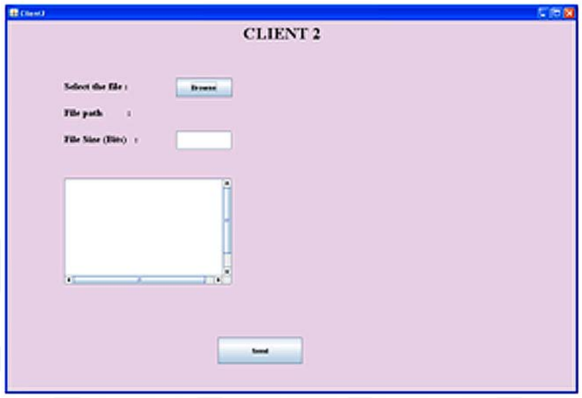

Fig 3: Client 2 tranceives the data

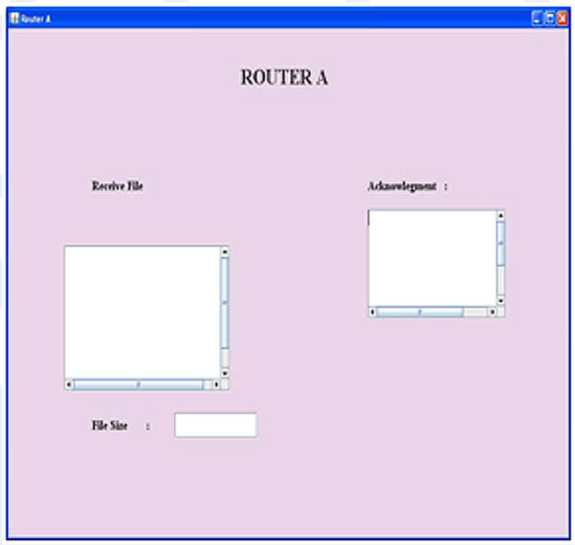

Fig 4: Router A is a one of the node in a topology

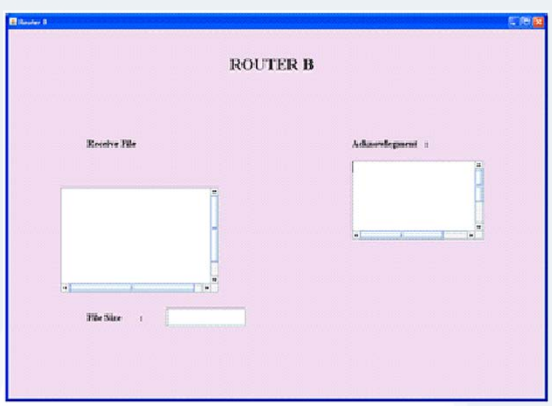

Fig 5: Router B is a one of the node in a topology

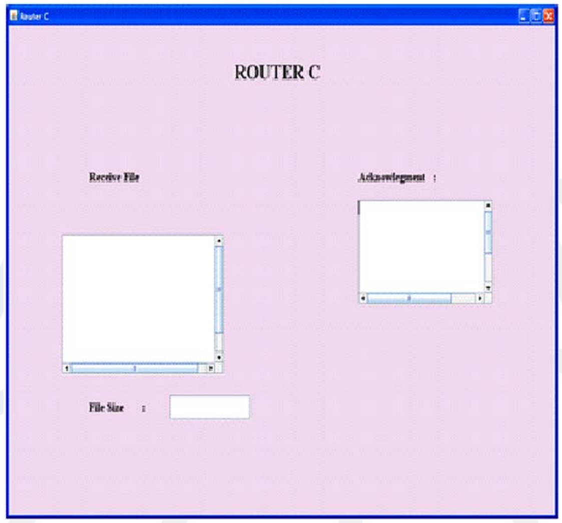

Fig 6: Router C is a one of the node in a topology

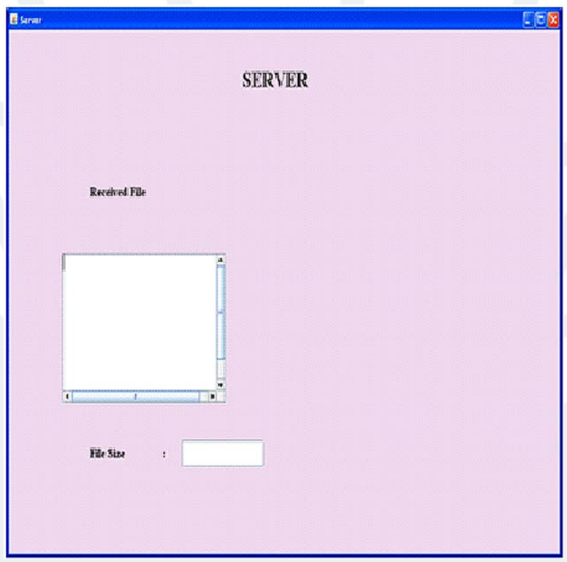

Fig 7: Server is a one of the node in a topology

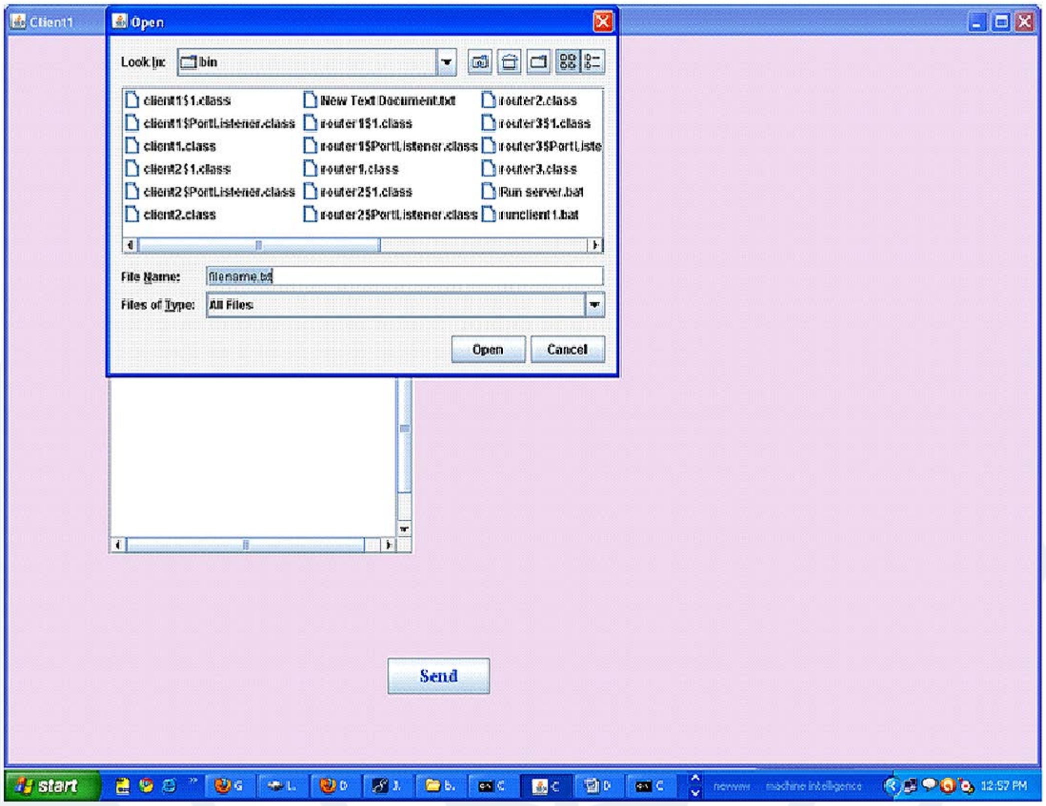

Fig 8: Selecting a file from database
It selects content of File from database to transfer from source to destination

Fig 9: Client 1 data selection

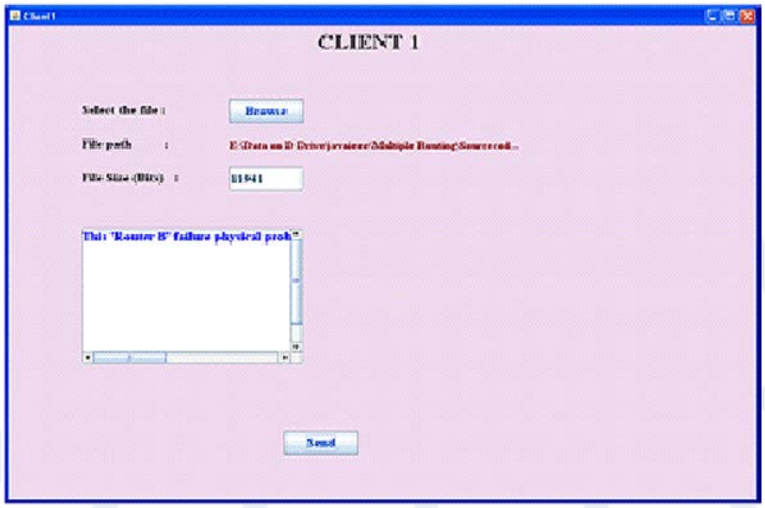

Fig 10: Client1 has acknowledge from MRC

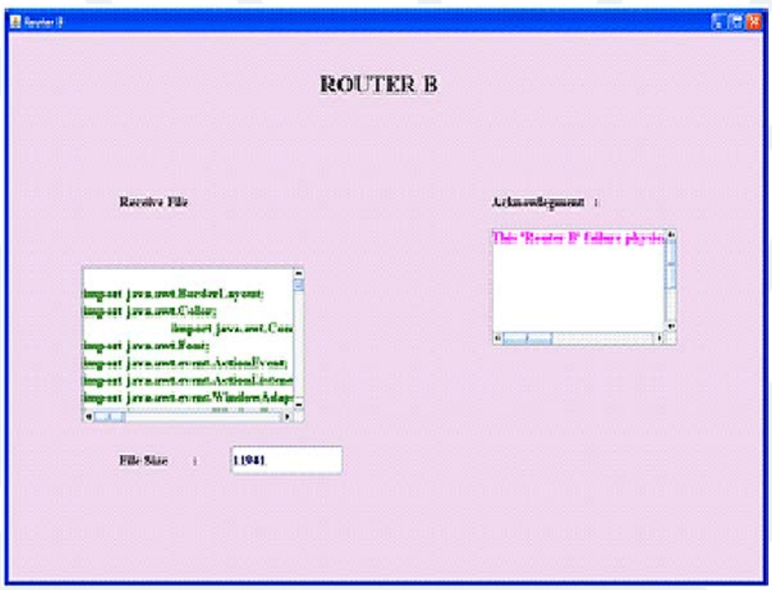

Fig 11:Router B Failed by Physical Problem

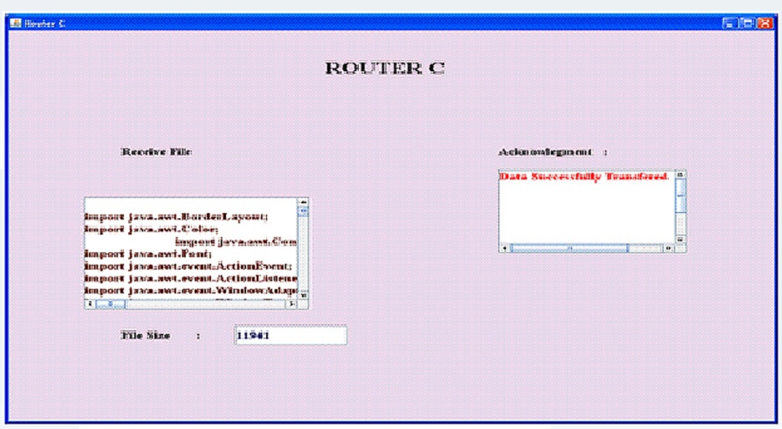

Fig 12: MRC send s data via Router C

Fig 13: Server receive the data

VII. Conclusion

We have presented Multiple Routing configurations as an approach to achieve fast recovery in IP networks. MRC is based on providing the routers with additional routing configurations, allowing them to forward packets along routes that avoid a failed component. MRC guarantees recovery from any single node or link failure in an arbitrary bi-connected network. By calculating backup configurations in advance, and operating based on locally available information only, MRC can act promptly after failure discovery.

MRC operates without knowing the root cause of failure, i.e., whether the forwarding disruption is caused by a node or link failure. This is achieved by using careful link weight assignment according to the rules we have described. The link weight assignment rules also provide basis for the specification of a forwarding procedure that successfully solves the last hop problem.

The performance of the algorithm and the forwarding mechanism has been evaluated using simulations. We have shown that MRC scales well: 3 or 4 backup configurations is typically enough to isolate all links and nodes in our test topologies. We have evaluated the effect MRC has on the load distribution in the network while traffic is routed in the backup configurations, and we have proposed a method that minimizes the risk of congestion after a link failure.

VIII. References

[1] Basu.A and J. G. Riecke, “Stability issues in OSPF routing,” in *Proceedings of SIGCOMM*, San Diego, California, USA, Aug. 2001, pp. 225–236.

[2] Boutremans.C, G. Iannaccone, and C. Diot, “Impact of link failures on VoIP performance,” in *roceedings of International Workshop on Network and perating System Support for Digital Audio and Video*, 2002, pp. 63–71

[3] Clark.D.D “The design philosophy of the DARPA internet protocols,” *SIGCOMM, computer Communications Review*, vol. 18, no. 4, pp. 106–114, Aug. 1988.

[4] Francois.P, C. Filsfils, J. Evans, and O.Bonaventure, “Achieving sub-second IGPconvergence in large IP networks,” *ACM SIGCOMM Computer Communication Review*, vol.35, no. 2, pp. 35 – 44, July 2005.

[5] Labovitz.C, A. Ahuja, A. Bose, and F.Jahanian, “Delayed Internet Routing Convergence,” *IEEE/ACM Transactions on Networking*, vol. 9, no. 3, pp. 293–306, June2001.
[6] Markopoulou.A G. Iannaccone, Bhattacharyya, C.-N. Chuah, and C. Diot, “Characterization of failures in an IP backbone network,” in *Proceedings INFOCOM*, Mar. 2004.
[7] Nelakuditi.S, S. Lee, Y. Yu, Z.-L. Zhang, and C.-N. Chuah, “Fast local rerouting for handling transient link failures,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 359–372, apr, 2007.
[8] Przygienda.T N. Shen, and N. Sheth, “M-ISIS: Multi topology (MT) routing in IS-IS,” Internet Draft (work in progress), Oct. 2005, draft iet f-isis-wg-lti-topology-11.txt.
[9] Rai.S, B. Mukherjee, and O. Deshpande, “IP resilience within an autonomous system: Current oaches, challenges, and future directions,” *IEEECommunications Magazine*, vol. 43, no. 10, pp. 142–149, Oct. 2005.

How to cite

T.K Rajesh, "Fast IP Network Recovery using MRC". *International Journal of Research in Computer Science*, 1 (1): pp. 75-88, September 2011. doi:10.7815/ijorcs.11.2011.007