International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 3, Issue 4(April 2014), PP.33-38

FCP with Source Path Routing to eliminate Convergence

Dnyaneshwar Dhangar, Saina Ismail Patel,
Apurva Ashok Gawad, Shruti Santosh Dudwadkar,
(Computer, Rajiv Gandhi Institute of Technology/ Mumbai University, India)
(Computer, Rajiv Gandhi Institute of Technology/ Mumbai University, India)
(Computer, Rajiv Gandhi Institute of Technology/ Mumbai University, India)
(Computer, Rajiv Gandhi Institute of Technology/ Mumbai University, India)

Abstract :- Current distributed routing paradigms (such as link-state, distance-vector, and path-vector) involve a convergence process. Due to the convergence process the router load is increased, outages and transient loops are introduced, and it results in slow reaction to failures. We propose a new routing paradigm where the goal is not to reduce the convergence time but rather to eliminate the convergence process completely. We propose a technique called Failure-Carrying Packets (FCP) that allows data packets to find a working path without requiring completely up-to-date state in routers. But this involves computational overhead at each router. The techniques to reduce the computational overhead involve a lot of state being maintained at each router. To this end, we propose a slight extension to the FCP algorithm called FCP with Source-Path Routing to reduce the computational overhead of FCP without keeping any heavy amount of state at the routers.

Keywords: - Convergence, **F**ailure carrying packet, network map, source path routing.

I. INTRODUCTION

The current Internet is an enormous size network consisting of thousands of Autonomous Systems (AS) operated by different institutions, such as the Internet Service Providers (ISP), companies, universities etc. It is now used as a general-purpose network for commercial purposes. Such evolution of the Internet has seen a large number of applications being deployed on it for commercial purposes. Many of these applications, such as VoIP, gaming etc, have stringent delay and loss requirements. Such stringent requirements call for a stable routing environment in the Internet.

Stable routing demands routing stability in case of failure or up gradation of any network component. When any failure occurs, the router adjacent to the failure has the responsibility of informing every other router in the routes avoiding the failure. Other routers, in response to the failure, update their routing tables computing new routes avoiding the failure. This process, in which every router involves itself in computing the new view of the network is called routing convergence.

Sometimes loops can be formed during routing. Such loops can lead to delay in routing packets or even loss of packets, resulting in serious performance degradation of the applications.

We will see failure carrying packets technique to remove convergence. We will also see techniques that solve the overhead problem of FCP. We propose our technique that is most efficient in removing computational overhead in each router and reducing load at router. This technique is called FCP Using Source Path Routing.

II. FAILURE CARRYING PACKETS

In this [1], we propose a completely different approach, than the traditional routing protocols for dealing with failures in the network. Instead of converging after the failure, this protocol eliminates convergence period altogether. Under FCP, the router detecting the failure simply reroutes the data packet around the failure, inserting the information of the failure within the packet header.

Thus other routers receiving the packet use this information locally to compute the path to the packet's destination avoiding the failed component. This eliminates the need for immediate propagation of the failure information by the detecting router. We describe the FCP protocol in detail in the following section.

2.1 FCP Concept

In case of no failure FCP reduces to link state protocol, where every router maintains a consistent view of the potential set of links which is called as the Network Map. This set of potential links consists of only those links that are operational over a long period of time. FCP uses this map to compute a stable path to all the destinations in the absence of failure. FCP behaves quite differently when failure occurs.

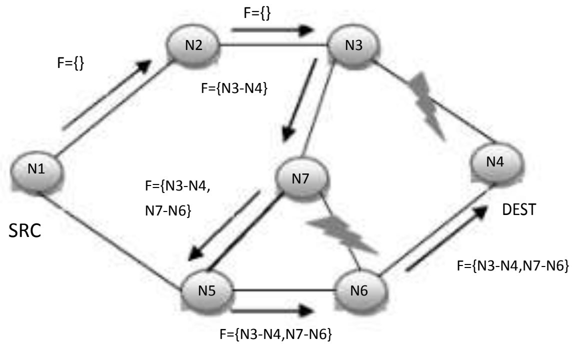

Fig. 1: An example illustrating FCP routing

The main intuition behind FCP is that, since all routers maintain the same Network Map at a particular time, all which is needed by the routers for dealing with a failure(s) is to know the set of link(s) that have currently failed in the network. Any router having this information will be able to compute a path to any destination, avoiding the failure, if a path exists. The best part about FCP is that, instead of sending separate protocol messages for propagating the failure information to the other routers, it adds this failure information in the packet header and computes new shortest path and sends the packet according to new path. In this way the convergence period is completely removed because the failure information is passed through the packet itself.

To understand FCP better, consider the example in Fig. 1, a network with unit link weights. Assume N_1 sends a message to N_4, and that links N_3–N_4 and N_7–N_6 are down. Since only nodes adjacent to the failed links know about the failure, the packet is forwarded along the shortest path in the original graph, (N_1, N_2, N_3, N_4), until it reaches the failed link N_3–N_4. At this point, N_3 computes a new shortest path to N_4 based on the map minus link N_3–N_4, and includes the failed link N_3–N_4 in the header. Let us assume that this path is (N_3, N_7, N_6, N_4). When the packet reaches N_7, N_7 adds the failed link N_7–N_6 to the header, and computes a new shortest path that does not include the two failed links.

2.2 FCP Algorithm

```
F: failed link field
1.      Initialize F=null;
2.      When the packet arrives to a router
a.      If (F!=null)
Compute a new route to the destination removing the
failed link
If (new route does not exist)
Abort
Else If (next hop on the path has failed link)
        Add that link to F and go to step 2(a).
Else
Forward the packet to the next hop router.
```

Fig. 2 FCP Algorithm

As the Figure2 above shows, when a packet arrives at a router, its next-hop is computed using the network map minus the failed links in the packet header. If this next-hop would send the packet out an interface that has a failed link, then the router inserts the failed link in the header

2.3 FCP Properties

2.3.1 Guaranteed reach ability

This property says that a packet p entering a network at a certain time t_1 will be delivered to the destination by the time $t_{2,}$ provided (1) there are at most f failures during [t_1,t_2], where f is an upper bound on the number of failures in the network during the interval, and (2) the network remains connected during [t_1,t_2].

2.3.2 Path Isolation

The path isolation property says that a malicious node cannot impact the path followed by a packet unless it is already on that path i.e. off-path malicious nodes cannot affect the routing process. This directly follows from the fact that an off-path node cannot contaminate the routing state of the nodes along the packet's path as these nodes compute the route solely based on the disseminated map and the list of failed links in the packet's header.

2.4 FCP Challenges

Challenges in FCP include:

Computational overhead: FCP presents an overhead that every router on the failure-carrying packet's path has to compute new routes to the packet's destination.

Map dissemination and updates: As FCP relies on all routers having a consistent view of the network map, there is a map dissemination and update protocol.

2.5 Overcoming FCP Challenges

2.5.1 Nodes precompute backup path

Pre compute "backup next-hop" for each destination. This saves router from re computation of new route to a packet's destination on failure. Thus re computation is required only when failures happen on primary and backup paths.

2.5.2 Caching

This includes computation and maintenance of caching information of paths in case of failures seen on both primary and backup paths.

2.5.3 . Disadvantages of these methods:

Keeping backup paths for every destination at a router doubles the router state Cached paths to route around failures on primary and backup paths add even more to router state.

III. FAILURE CARRYING PACKETS USING SOURCE PATH ROUTING

In order to reduce the computational overhead- if the backup path for each destination is pre- computed and stored, and incase if there is failure on backup path ,then maintaining cache paths for each combination of failures seen in the packet header – incur a lot of state being maintained at each router. Moreover the state maintained are mostly hard state which is unnecessary considering that most failures occurring in the network are short-lived and transient.

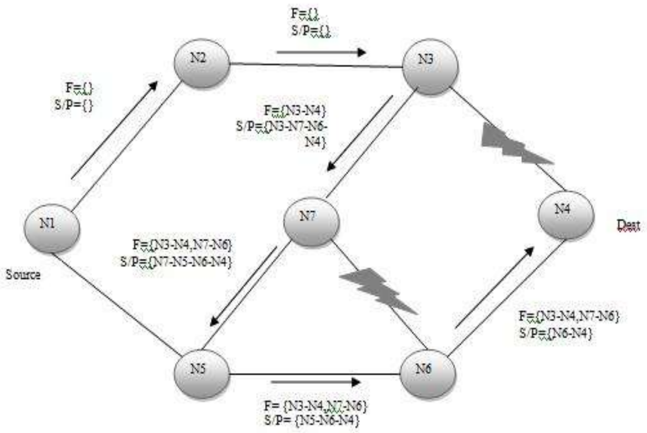

Fig. 3. An example illustrating FCP routing with source path

To this end, we propose an extension to the FCP algorithm called Failure Carrying Packets with Source-Path Routing extension which aims at reducing computational overhead of FCP without maintaining heavy amount of state at the routers. Below we discuss the extension along with the algorithm.
Consider again the same example given below, reproduced here for convenience.

All the links are assumed to have unit weights. Assume N_1 sends a message to N_4 which is the destination, and that links N_3−N_4 and N_7−N_6 are down. Since only the adjacent nodes know about the failed links, the packet is forwarded along the shortest path in the original graph, (N_1,N_2,N_3,N_4), until it reaches the failed link N_3-N_4. At this point, N_3 computes a new shortest path to N_4 based on the map minus link N_3-N_4. In the original design, FCP will include the failed link N_3-N_4 and insert it into the packet header, and forwards the packet along the newly computed route. But in this extension, the router not only adds the failed link N_3-N_4 in the packet, but also adds the newly computed shortest route to the destination which is $N_3 \rightarrow N_7 \rightarrow N_6 \rightarrow N_4$ in the header and forwards it to N_7 . Now node N_7 knows that there is link failure at N_7-N_6 and hence it adds this failed link in the packet header. Now N_7 will compute the shortest path to the destination minus link N_7-N_6 from the network map. The shortest route $N_7 \rightarrow N_5 \rightarrow N_6 \rightarrow N_4$ which is computed is now inserted in the packet header. Subsequent routers N_5 and N_6 will forward the packet along the path inserted by N_7. Finally the packet reaches the destination N_4. Thus only nodes encountering failures on the primary path or source path need to recompute the shortest path to destination and other nodes simply forward packets along the computed shortest path. Since most of the failures occurring are only single link failures, very few nodes may encounter a failure combination where a link from both primary as well as source-path have failed and thus may need to perform recomputation to route a packet encountering failures along its journey to destination. Moreover this recomputation needs to be performed only for the first packet that were to pass through the failed link(s), as all the routers cache the newly computed paths to route around the failure.

```
F: failed link field
S/P: source path field
1.          Initialize F=null, S/P=null;
2.          When the packet arrives to a router
a.          If (F!=null)
Compute a new route to the destination removing
the failed link and add the source path in packet
header.
If (new route does not
exist) Abort
Else If (next hop on the path has failed link)
            Add that link to F and go to step 2(a).
Else
Forward the packet to the next hop router.
```

Fig. 4. FCP with source path routing Algorithm

IV. IMPLEMENTATION APPROACH

FCP adopts a link-state approach to routing and hence it is implemented by modifying Open Shortest Path First (OSPF) protocol which is also a link-state routing protocol currently used in the Internet for Intra-domain routing. Modifications are also required to be made to Internet Protocol (IP) protocol as FCP's forwarding functionality differs from IP. In the following section we detail the approach to be taken to simulate the behavior of FCP using OSPF and IP.

4.1 Map dissemination of FCP

FCP's map dissemination approach differs quite from that of OSPF. OSPF keeps the routers updated with the current network state by making each router periodically propagate link-state, whereas FCP consists of a centralized coordinator that periodically floods all the routers only with long-term changes made to the network. To simulate such behavior with OSPF, the link-state of the network is propagated only when the network boots and subsequent update messages need to be suppressed. Only if a change to the network is deemed permanent then the protocol should propagate an update message. This can be achieved by making changes to the timers associated with the OSPF link-state update messages.

4.2 Packet forwarding approach of FCP

Conventional packet forwarding requires a destination IP address lookup operation to be made on the forwarding table by IP and determine the outgoing interface for the packet and forward the packet. Incase if there is link failure the forwarding operation remains the same with the only change that the detecting router locally reroutes the packet avoiding the failure. Through routing protocol messages the other routers are informed about the failure. Thus IP has nothing to do with the failure. But in case of FCP, since the failure information and source-path information is carried in the IP header, IP has a role to play here. The forwarding engine here examines the packet for failure or source-path information. In case source-path is present IP forwards the packet along the source-path if the adjacent link on the source path is alive. If there is failure in the adjacent link then IP needs to invoke the services of FCP that computes the new route to destination, if any exists. IP inserts the new route and the failure information in the packet header and forwards the packet along the newly computed path.

4.3 Changes to be made to IP Header

Extra state needs to be incorporated in the IP header such as failure and source-path information. This can be accomplished by use of Options field of the IP header.

Changes to be made to IP header:

IP header needs to carry failure info and source-path. Use of IP Options field can be made. The length of the Option field is variable, and the end of a packet header has to be aligned to a 32-bit boundary, so an additional padding field of the appropriate length is added (and set to 0 by default).

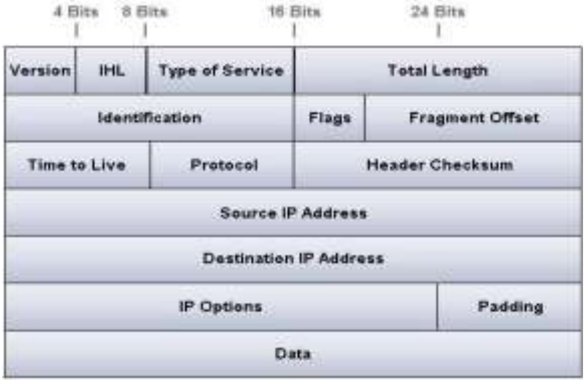

Fig 5. IP Header

4.3.1. IP Options and Source Routing

Normally, IP datagram's are routed without any specific instructions from devices regarding the path a datagram should take from the source to the destination. It's the job of routers, using routing protocols, to figure out those details. In some cases, however, it may be advantageous to have the source of a datagram specify the route a datagram takes through the network. This is called source routing.

There are two IP options that support source routing. In each, the option includes a list of IP addresses specifying the routers that must be used, to reach the destination. When strict source routing is used, this means that the path specified in the option must be used exactly, in sequence, with no other routers permitted to handle the datagram at all. In contrast, loose source routing specifies a list of IP addresses that must be followed in sequence, but having intervening hops in between the devices on the list is allowed.

4.3.2. IP Datagram Options and Option Format:

All IP datagram's must include the standard 20-byte header, which contains key information such as the source and destination address of the datagram, fragmentation control parameters, length information and more. In addition to these invariable fields, the creators of IPv4 included the ability to add options that provide additional flexibility in how IP handles datagram's. Use of these options is, of course, optional. However, all devices that handle IP datagram must be capable of properly reading and handling them.

The IP datagram may contain zero, one or more options, which makes the total length of the Options field in the IP header variable. Each of the options can be either a single byte long, or multiple bytes in length, depending on how much information the option needs to convey. When more than one option is included they are just concatenated together and put into the Options field as a whole. Since the IP header must be a multiple of 32 bits, a Padding field is included if the number of bits in all options together is not a multiple of 32 bits.

IP Option Format

Each IP option has its own subfield format. For most options, all three subfields are used: Option Type, Option Length and Option Data. For a few simple options, however, this complex substructure is not needed. In those cases, the option type itself communicates all the information required, so the Option Type field appears alone, while the Option Length and Option Data subfields are omitted.

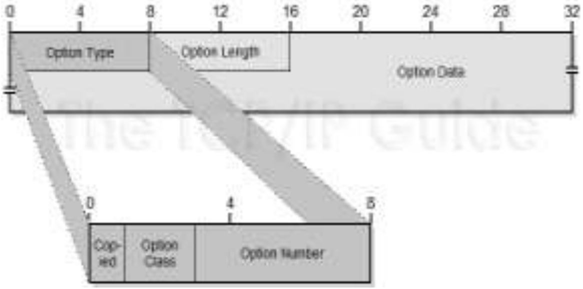

Fig 6. Options Field

V. CONCLUSION AND FUTURE WORK

Although the proposed extension aims at reducing computational overhead and router state of FCP, it is really to needed to quantify the savings achieved. Moreover the techniques are needed to be evaluated on some real topologies and failure instances.

Thus the future work of the project would include implementation of original FCP protocol as it is proposed by the authors first and its evaluation and then implementation of the extended version and its comparative evaluation with the original one.

The parameters of evaluation would be amount of router state, computational requirements and packet overhead for both the versions of the protocol.

ACKNOWLEDGMENT

We wish to express our sincere gratitude to Dr. U. V. Bhosle**,** Principal and Prof. S. B. Wankhade**,** H.O.D of Computer Department of RGIT for providing us an opportunity to do our project work on “Removing Convergence using FCP with source path routing ".

This project bears on imprint of many people. We sincerely thank our project guide Prof. D. J. Dhangar for his guidance and encouragement.

REFERENCES

[1] P. Francois and O. Bonaventure, “Avoiding transient loops during IGP convergence in IP networks,” In Proc. INFOCOM, 2005.

[2] C. Alaettinoglu, V. Jacobson, and H. Yu, “Towards Millisecond IGP Convergence”, IETF Internet draft 2000.

[3] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne, “Fast IP Network Recovery using Mumtiple Routing Configurations”,

[4] K. K. Lakshminarayanan, M. C. Caesar, M. Rangan, T. Anderson, S. Shenker, and I. Stoica, “Achieving Convergence-Free Routing using Failure-Carrying Packets,” In SIGCOMM, 2007.

[5] S. Rai, B. Mukherji, and O. Deshpande, “IP Resilience within an Autonomous System: Current Approaches, Challenges, and Future Directions,” IEEE Communications Magazine, vol. 43, no. 10, pp. 142-149, Oct. 2005