An Enhanced stable and scalable routing methodology in Named data networks

K. Kranthi kumar ,P.Durgaprasad

***Abstract* -** The named data network is a new technology in routing were its forwarding plane helps us to recover the data on its own when the network failures occur.in this NDN the major issue are when the data is sent from one network to the other the data flow is not stable. By major survey we came to know that routing protocols plays a major role for the stable and scalable data flow. This routing protocols and be classified by forwarding process and also network topologies were the protocols have the data that can be retrieved back whenever the network faults occur. In this paper we proposed that hybrid flow architecture, network topologies and routing protocols are improved to get the stable and scalable routing when the data flow is interrupted.

***KEYWORDS* -**

NDN, routing, stability and scalability, forwarding process, probing, data packets, interest packets.

1. INTRODUCTION

Named data networks is a new network architecture that helps us to change the basic network services from delivering packets to the given destination to retrieve the data back with a given name. NDN communication is like a receiver driven data consumer sends interest packets carrying the names of desired data by any node in the network which can return data packets that have matching names to satisfy the interests [1]. This is a two -way interest data packet that exchange in opposite directions on same network path. Today's IP networks put all intelligence into routing, which disseminates topology and policy information, computes routes detects and recovers from failures while the data plane merely forwards packets according to the FIB. When the data plane has its own adaptability, are routing protocols still needed? If so, for what purpose and to what extent? If some of routing's tasks can be offloaded to forwarding, would that bring positive impact on routing protocols' design and operation, e.g., making routing more scalable and stable [2].

In this paper we explain the role of routing in NDN networks. Through analysis, design, and extensive simulation, we find that routing is important in ip networks and the forwarding plane for data retrieval, as well as for efficiently getting the new links. However, NDN routing does not need to converge fast following network changes, which can be handled by adaptive forwarding more promptly.

This enables one to significantly improve the scalability and stability of the routing system using larger keep-alive timer values that ignore short-term failures. Furthermore, routing algorithms that would not work well in current networks may work fine in NDN due to its reduced role of bootstrapping adaptive forwarding.

FORWARDING PLANE

In routing forwarding plane is also called as data plane. the router architecture that decides what to do with packets arriving on an interface. Most commonly, it refers to a table in which the router looks up the destination address of the incoming packet and retrieves the information necessary to determine the path from the receiving element, through the internal forwarding fabric of the router, and to the proper outgoing interface(s) [3]. The IP Multimedia Subsystem architecture uses

Author name: Dr.K. Kranthi Kumar, Associate professor
In Srinidhi institute of science and technology, INDIA,phno:9848624931
Krantikathula@gmail.com
Coauthor: P. Durgaprasad, currently pursuing master's degree program in Srinidhi institute of science and technology,INDIA,Phno:7893919694
Durgaprasad0237@gmail.com

the term transport plane to describe a function roughly equivalent to the routing control plane.

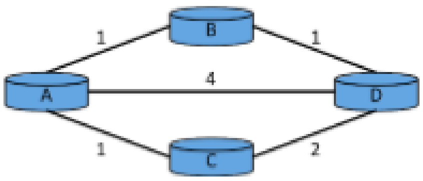


Fig.1. :A network node example

II. Literature survey

FailureRecovery: NDN's symmetric traffic flow enables fast fault detection. Routers can calculate RTT for each Interest-Data exchange, which can be used as a prediction for future Interests. After forwarding an Interest, a router starts a timer based on the average of previous RTTs; potential network problems can be detected if no Data is received before the timer expires. With Interest NACKs, fault detection and notification is even faster. When network problems are detected, routers can explore alternative paths freely without worrying about loops, since loops can be detected by checking the nonce field carried in Interests. Fast fault detection and loop-free forwarding are the two unique features that make NDN's forwarding plane smart and adaptive – routers are able to handle network faults such as prefix hijacking, failures and congestion locally at the forwarding plane [4]. We use the simple example in Figure 1 to illustrate how NDN routers handle link failures. The costs of links are marked in the figure2; routers rank the interfaces using the cost of their best paths towards the destination. When there is no failure in the network, A uses B as its primary next hop for content provided by D. Interface A-B will be marked Green as long as Data continues to flow from B to A. When link B-D fails, A will keep sending Interests to B at first. However, B cannot satisfy the Interests due to the failure, so it will send NACKs back to A [5]. Upon receiving a NACK, A will mark A-B Yellow and retry the next best interface, in this case A-C. Since there is no failure on this path, Data will flow back through path D-C-A. Will then marks interface A-C Green and start using C as the primary next hop.

Routing in IP

IP's routing plane is intelligent and adaptive, but its forwarding plane is stateless and strictly follows routing. Therefore, the routing plane is also regarded as the control plane. Routing is responsible for building the routing table and maintaining it in face of network changes, including both long-term topology and policy changes as well as short-term churns. When there is a change in the network, routers need to exchange routing updates with each other in order to reach new global consistency [6]. The time period after a change happens and before all routers agree on the new routing state is called the routing convergence period. IP routing protocols need to converge fast in order to reduce packet loss and resume packet delivery after network changes.

However, fast routing convergence is challenging in large operational networks. The fundamental reason is that it conflicts with other design goals for routing protocols, i.e., routing stability and scalability. Routing stability ensures stable routing paths within the network. It is important for applications that suffer from RTT fluctuation; it also helps routers achieve traffic engineering goals. Routing scalability is essential for supporting a large number of nodes, links and prefixes1 in the network. For link-state routing, each router knows the entire topology. These protocols can converge fast, but at the cost of poor stability and limited scalability. For distance/path-vector routing, routers do not have a full knowledge of the topology [7]. They are able to achieve better scalability, but the convergence time may be as long as tens of minutes. Below we use link-state routing as an example to explain the issues with today's IP routing protocols. In summary, it is hard to achieve fast convergence, stability and scalability simultaneously in a routing protocol. If failures can be handled without global routing convergence, the requirement on fast convergence can be relaxed, making it possible to improve routing stability and scalability [8].

Routing in NDN

In NDN, the forwardin1g plane is the actual control plane since the forwarding strategy module makes forwarding decisions on its own. This fundamental change prompts us to rethink the role of routing in NDN [9]. The first question is whether NDN still needs routing protocols. Conventionally, routing protocols are responsible for disseminating topology and policy information, computing routes and handling short-term network changes. For NDN to work without routing, routers need to be able to do the following things efficiently: 1) retrieve Data when the network is stable; 2) handle link failures; and 3) handle link recovery.

III. Classification of existing system

Link-State Routing

Link-state routing protocols store the entire network topology in the link-state database (LSDB), making it possible to compute optimal interface ranking. Suppose a node N has n interfaces 1 In. For Data provided by node M, we rank these interfaces using CM N,k, which is the cost of the best path from N to M through interface Ik.

One simple method to compute CN, k for all destinations through Ik is to remove all interfaces except Ik from N's LSDB and run Dijkstra's algorithm to compute the shortest paths. This may not be the best method since it will end up calling Dijkstra's algorithm once for every interface. It is just used to illustrate how interface ranking can be done in link-state routing. Optimization of the algorithm is possible but out of the scope of this paper.

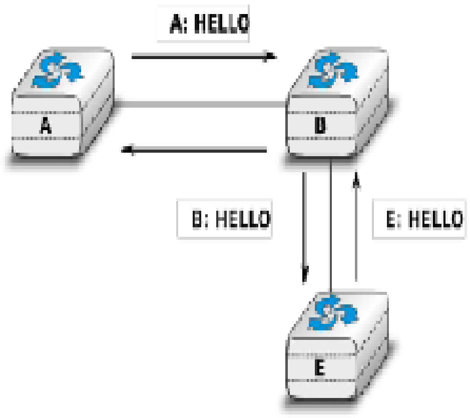


Fig .2. : The exchange of hello message

Distance/Path-Vector Routing

In distance-vector or path-vector routing, routers announce cost of the complete routing path towards each destination to their neighbors. When router N receives a routing announcement for Data provided by M from interface Ik, it simply adds the link cost of Ik to the received path cost to obtain its path cost CM N, k. The interfaces are then ranked by the path costs to M through them. Note that a router may not receive routing announcement from all interfaces, since these routing protocols often incorporate split-horizon route announcement to prevent routing loops. If router N learns a route towards M through interface Ik, it will not advertise its route. Interfaces that do not receive routing announcement are assigned infinite cost to ensure they stay at the end of the ranked interface list [10].

They will only be used as the last resort if all higher-ranked interfaces fail to retrieve Data. These interfaces are useful in many situations. For example, in BGP if a provider P uses a customer C as the next hop, it will not make routing announcement to C. If C's best path fails, it will not have an alternative path until routing converges, in which case P will announce its alternative path to C. RBGP [11] is proposed to address this issue by allowing P to announce its alternative path to C even without failures. NDN, on the other hand, is able to achieve the same effect without changing the routing protocol.

FUNCTIONS

PREVENTION OF LOOP

The creation of loop can be avoided in path vector routing. A router receives a message it checks to see if its autonomous system is in the path list to the destination if it is looping is involved and the message is ignored.

POLICY ROUTING

When a router receives a message, it can check the path, if one of the autonomous systems listed in the path against its policy, it can ignore its path and destination it does not update its routing table with this path or it does not send the messages to its neighbors [12].

OPTIMUM PATH

A path to a destination that is the best for the organization that runs the autonomous system [13].

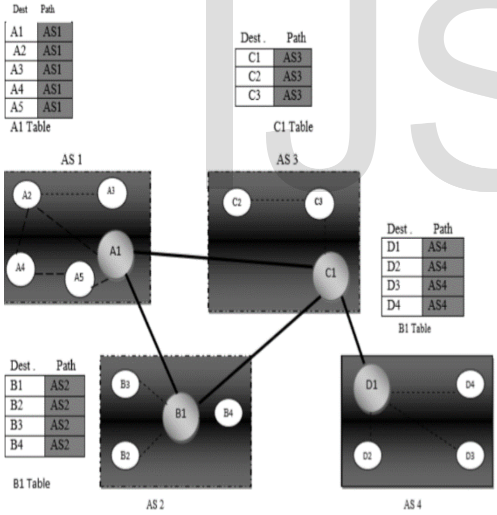


Fig.3.: Initial routing in path vector routing

Probing

It has been shown that NDN routers can handle link failures locally at the forwarding plane [14]. In this subsection we answer the question of whether the same applies to link recovery. Routers can detect link failures quickly by observing Interest-Data exchanges or Interest NACK. However, there is no explicit signal for link recovery from the forwarding plane. again, let's take Figure 1 as an example. After interface B-D recovers from a failure, interface A-B becomes the best interface for A to retrieve data from D. However, A will continue using interface A-C because the forwarding strategy prefers Green interfaces over Yellow ones. In this case, A needs to probe interface A-B by sending a copy of an Interest to it. If the probing Interest successfully brings Data back, interface A-B will be marked Green and be used to forward subsequent Interests to D. There is a research question of when to perform probing. An Interest copy is used for probing so that regular Data retrieval will not be affected if probing is unsuccessful. However, this causes extra Interest and Data in the network. There is a trade-off between how fast a link recovery is detected and the amount of overhead caused by probing. In CCNx [15], a

prototype implementation of NDN, routers probe alternative interfaces periodically in order to detect better paths. This enables routers to detect link recovery at the forwarding plane. Fast recovery detection is achievable through aggressive probing. However, it will incur significant overhead.

Routing Stability and Scalability

Link-state routing protocols exhibit poor stability and scalability in IP due to the fast routing convergence requirement. However, there is a simple method to address these issues in NDN. Since NDN routers can handle network failures at the forwarding plane, short-lived failures can be masked from the routing protocols. Research shows that the duration of network failures follows a long-tailed distribution, and over 50% of failures last less than one minute [16, 17]. Therefore, the number of routing events can be significantly reduced if routing protocols do not need to react to the short failures.

As a result, the bandwidth and CPU cycles consumed by routing updates can be reduced, and there will be less routing fluctuation. In addition, since there is no fast routing convergence requirement, larger networks and more name prefixes become affordable. In summary, both routing stability and scalability can be significantly improved. For link-state routing, we can implement the idea by increasing the HELLO and DEAD interval. For example, if we set the DEAD interval to be one minute, over 50% of the link failures will be ignored by the routing protocol. Alternatively, we can increase the suppression timer for routing update generation and SPF computation to achieve the same effect. Although this idea looks simple, it can be applied to any existing IP routing protocol to improve its stability and scalability. We will evaluate the effectiveness of this method in the next section [18].

Topologies	Before processing		After processing	
C3600 Router	52	168	10	11
C7200 Router	284	1232	N/A	N/A

Table.1. : -Topologies used in simulation

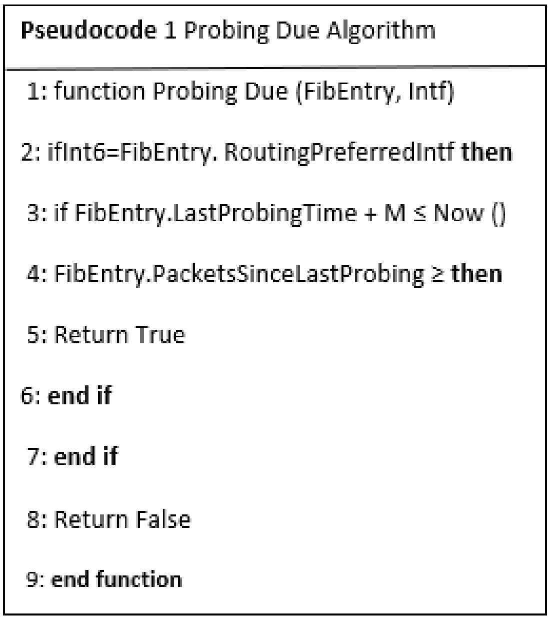
IV. Proposed system

In this section we use extensive simulations to show that NDN's packet forwarding performance under network failures is hardly affected by routing convergence time; by masking short-lived failures from routing protocols, one can effectively reduce routing overhead while maintaining high packet delivery performance in NDN networks.

simulation Setup

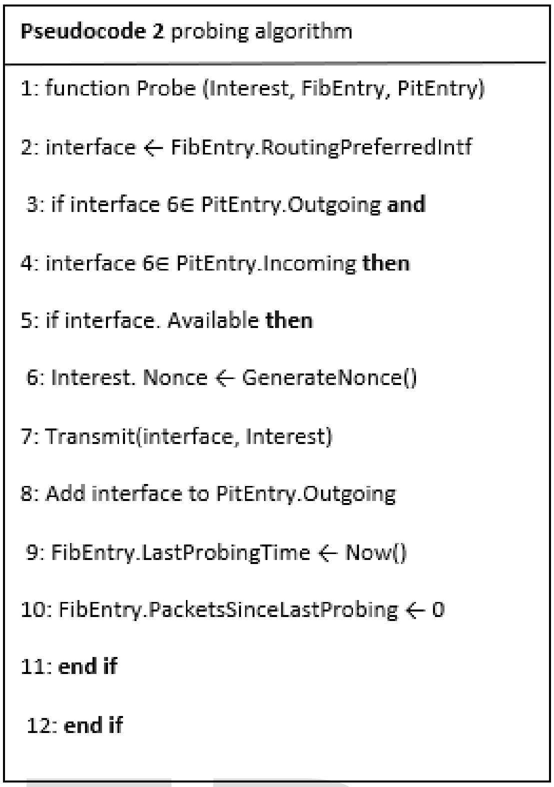
Unless otherwise specified, we run experiments in the GNS3 simulator [19] which provides complete implementations of OSPF and RIP routing protocols. We implement basic NDN operations and the forwarding strategy presented in [20] in the simulator. We also make necessary changes to the routing protocols as described in Section 4.1We use the Abilene topology [21] and selected Rocket fuel topologies in the experiments. A summary of the topologies is presented in Table 1. We process the first three topologies to remove all single-homed nodes, because if the link of a single-homed node fails, the node will be disconnected from the network and thus cannot provide any insightful result. For OSPF, we use propagation delay as the cost of the links. Unless otherwise specified, we report results from the AS1239-PoP topology.

Results for other topologies are similar and lead to the same conclusions. The AS1239-Router topology is only used to show the improvement of routing scalability 3. We inject random link failures into the topologies. A shifted Pareto distribution is used to generate time-to-failure and time-to-recover values for each link independently. We use 120 seconds as the mean-time-to-recover, and 1000 seconds as the mean-time-to-fail; the scale parameter of the Pareto distribution is set to be 208 so that 50% of the failures last less than one minute [22]. When a link fails, both directions of the link stop working. With this model, multiple network events (failures and recovery) can happen concurrently.



Pseudocode 1 Probing Due Algorithm

```
function Probing Due (FibEntry, Intf)
  ifInt6=FibEntry. RoutingPreferredIntf then
    if FibEntry.LastProbingTime + M ≤ Now ()
    FibEntry.PacketsSinceLastProbing ≥ then
      Return True
    end if
  end if
  Return False
end function
```



Pseudocode 2 probing algorithm

```
function Probe (Interest, FibEntry, PitEntry)
  interface ← FibEntry.RoutingPreferredIntf
  if interface 6∈ PitEntry.Outgoing and
  interface 6∈ PitEntry.Incoming then
    if interface. Available then
      Interest. Nonce ← GenerateNonce()
      Transmit(interface, Interest)
      Add interface to PitEntry.Outgoing
      FibEntry.LastProbingTime ← Now()
      FibEntry.PacketsSinceLastProbing ← 0
    end if
  end if
```

Hybrid flow architecture

In hybrid flow networks, flows occupy an entire wavelength and users flow data at the optical line rate without windowing. Sessions are of fixed duration. Thus, if a session expires and the receiver has not acknowledged all frames without error, a reflow is required. The hybrid flow data center architecture differs from a traditional data center network in two main response.

First, the hybrid flow architecture includes a control plane that allows for dynamic path setup, network monitoring, and resource allocation. Second, in the hybrid flow architecture, elephant flows are forwarded on all-optical switches and do not leave the optical domain on intermediary switches.

The application layer informs the transport layer of the size of a flow. The transport layer is responsible for session initiation with a control plane scheduler. It is also responsible for keeping track of multiple sessions in the event that all frames are not received during a single session and more session requests are required. Based on the flow size and the network loading, the transport layer decides whether the flow should be sent over a traditional EPS architecture or over the optical flow architecture [23].

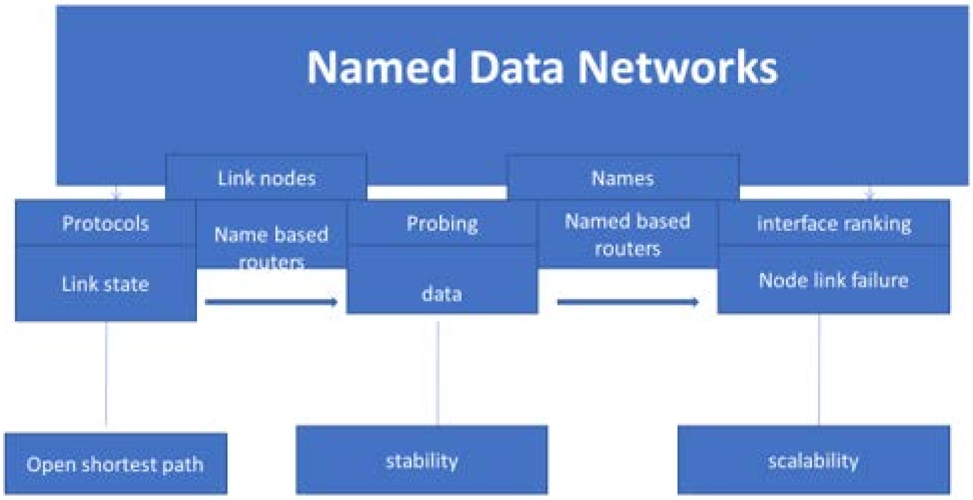
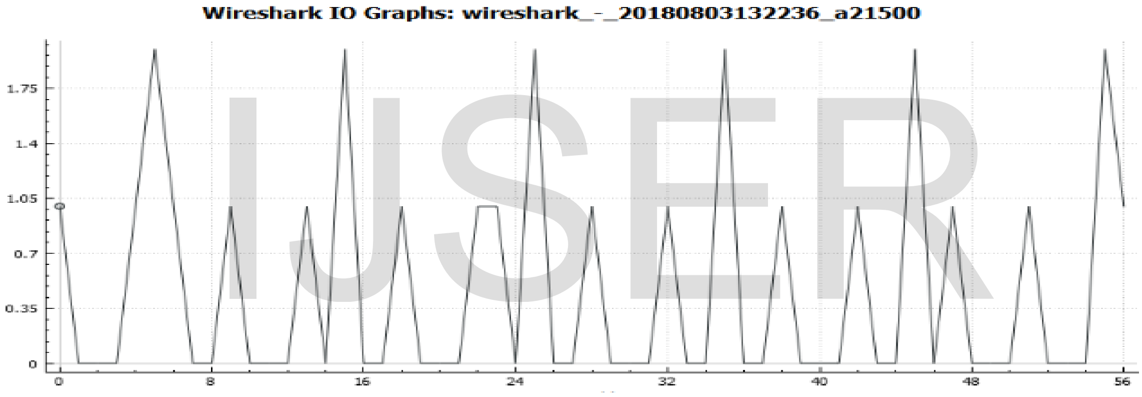
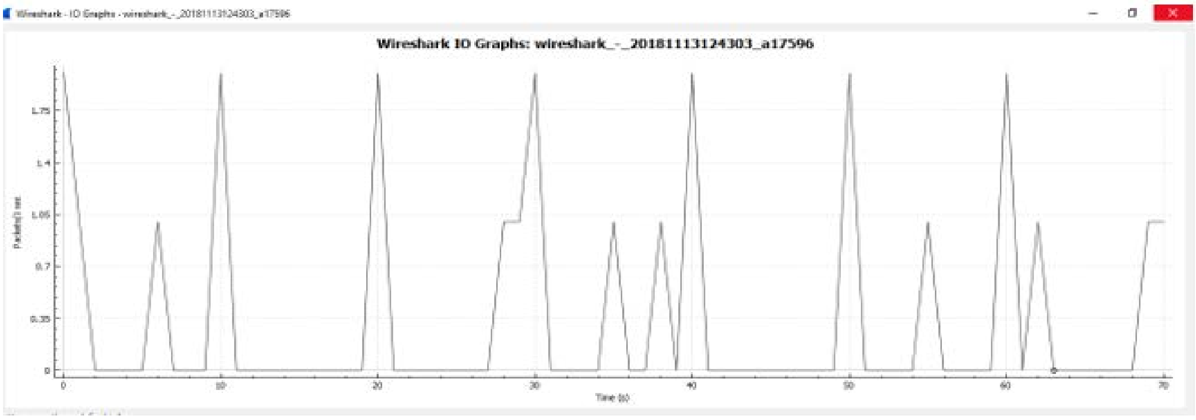


Fig .4.: Hybrid flow architecture

RESULTS



Graph 1: Packet delivery performance in IP under different HELLO interval


Wireshark - IO Graphs - wireshark_-_20181113124303_a17596
Wireshark IO Graphs: wireshark_-_20181113124303_a17596
Packets/s sec
1.75
1.4
1.05
0.7
0.35
0
0
10
20
30
40
50
60
70
Time (s)

Graph 2: link failures and hop count of packet

V. OBSERVATION

Routing is a necessary subsystem for any large-scale network. Like IP, NDN itself does not dictate what kinds of routing algorithms or protocols to use. However, one can take advantage of NDN's adaptive forwarding plane to improve the stability and scalability of existing routing protocols, as well as enable routing protocols that are deemed difficult to adopt in IP networks. Traditional Routing Protocols: With adaptive forwarding, routing in NDN only assumes a supporting role. It provides a reasonable starting point for forwarding which can then effectively explore different choices [24]. The job of routing becomes more of disseminating topology and policy information than distributed computation of best paths. This new division of labor between routing and forwarding makes routing protocols simpler and more scalable. Traditional routing protocols such as OSPF, RIP, and BGP can benefit greatly from NDN's adaptive forwarding plane. They can be tuned for synchronizing among routers long-term topology and policy information without handling short-term churns.

VI. CONCLUSION

In this paper we study the role of routing in NDN. NDN's adaptive forwarding plane leads to a new division of labor between routing and forwarding planes. While the latter can detect and recover from link failures quickly independent from the former, the former helps bootstrap adaptive forwarding and handle link recovery. We specify how NDN routing coordinates with forwarding through interface ranking and probing mechanisms. Our analysis and simulations show that NDN routing protocols can benefit from the forwarding plane due to the relaxed requirement on timely detection of failures and convergence delay. Consequently, NDN routing stability and scalability can be greatly improved. Moreover, the adaptive forwarding plane also enables new routing schemes that may not work well in IP to be used in an NDN network.

VII. FUTURE WORK

A massive amount of research has been conducted on how to gracefully accommodate routing changes with minimum impact on packet delivery in IP networks. One category of solutions relies on routing protocols to adapt to the changes. Francois et al. show that sub-second link-state routing convergence in large intra-domain networks is achievable by tuning various timers. But this method incurs extra routing overhead and may also cause routing instability. Fast reroute (FRR) mechanisms handle link failures by pre-computing alternative paths. MPLS FRR mechanisms provide backup paths in MPLS-enabled networks to protect specific links from failures. Similarly, IPFRR mechanisms (e.g., [8]) provide temporary alternative paths before routing convergence in pure IP networks. However, it is hard for the FRR mechanisms to cover all possible failure scenarios; nor can they handle multiple link failures well.

VIII. REFERENCES

[1]Inchan hwang, Dabin Kim, Young-Bae Ko, "Analysis of NDN repository architecture and its improvement intensive applications", *Advanced Communication Technology (ICACT) 2017 19th International Conference on*, IEEE pp. 203-209, 2017.

[2] mahmud rashih celenliouglu, Haci ali mantar, **"**A scalable routing and admission control model in sdn-based networks**"** ANCS '14 Proceedings of the tenth ACM/IEEE symposium on Architectures for networking and communications systems.

[3] Aris cahyadi risdianto, Eueng Mulyana, "Implementation and analysis of control and forwarding plane for SDN" *Access IEEE*, vol. 6, pp. 12197-12205, 2015**.**

[4] Xu Zhang, weigang Hou, Failure recovery solutions using cognitive mechanisms for Software Defined Optical Networks, Access IEEE vol. 234, pp. 104, 2018**.**

[5] Yi Wu, Danio Daven Improved Detection of ACK/NACK Messages in the LTE Uplink

Control Channel" Access IEEE, vol.5, pp.5261-1502,2012.

[6] P.S. Ford ; Y. Rekhter ; H.-W. Braun Improving the routing and addressing of IP ,access IEEE vol.7 issue 3,2008.

[7] C. Alaettinoglu, Jacobson, and H. Yu. Towards Milli-Second IGP Convergence. Internet Draft draft-alaettinoglu-isis-convergence-00.txt, Nov. 2000.

[8]A.Atlas and A.Zinin. RFC 5286: Basic Specification for IP Fast Reroute: Loop-Free Alternates, 2008.

[9] A. Carzaniga, K. Khazaei, M.Papalini, and A. L. Wolf. Is information-centric multi-tree routing feasible? In Proceedings of ACM SIGCOMM ICN Workshop, 2013.

[10] R.Chiocchetti, D. Perino, G. Carofiglio, D. Rossi, and G. Rossini. Inform: A dynamic interest forwarding mechanism for information centric networking. In Proceedings of ACM SIGCOMM ICN Workshop, 2013.

[11] P. Francois, C.Filsfils, J. Evans, and O. Bonaventure. Achieving Sub-Second IGP Convergence in Large IP Networks. ACM SIGCOMM CCR, 35(3), July 2005.

[12] A. K. M. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang. Nlsr: Named-data link state routing protocol. In Proceedings of ACM SIGCOMM ICN Workshop, 2013.

[13] V. Jacobson, D. K.Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking Named Content. In Proceedings of ACM Conex, 2009.

[14] N. Kushman, S., D. Katabi, and B. Maggs. R-BGP: Staying Connected in a Connected World. In Proceedings of USENIX NSDI, 2007.

[15] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. C. Schmidt, and M. ahlisch. ICN Research Challenges. Internet draft, 2014.

[16] A. Kvalbein, A. Hansen, T. Cicic, S. Gjessing, and O. Lysne. Fast IP Network Recovery Using Multiple Routing Configurations. In Proceedings of INFOCOM, 2006.

[17] K. Lakshminarayanan, M. Caesar, M. Rangan, T. Anderson, S. Shankar, and I. Stoics. Achieving Convergence-Free Routing using Failure-Carrying Packets. In Proceedings of ACM SIGCOMM, 2007.

[18] S. Lee, Y. Yu, S. Nelakuditi, Z. li Zhang, and C. nee Chuah. Proactive vs Reactive Approaches to Failure Resilient Routing. In Proceedings of IEEE INFOCOM, 2004.

[19] J. Liu, A. Panda, A. Singla, B. Godfrey, M. Schapira, and S. Shankar. Ensuring Connectivity via Data Plane Mechanisms. In Proceedings of USENIX NSDI, 2013.

[20] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, Y. Ganjali, and C.Diot. Characterization of Failures in an Operational IP Backbone Network. IEEE/ACM Transactions on Networking, 16(4):749–762, August 2008.

[21] M. Motiwala, M. Elmore, N. Feamster, and S. Vempala. Path Splicing. In Proceedings of ACM SIGCOMM, 2008.

[22] J. Moy. RFC 2328: OSPF Version 2, 1998. http://www.ietf.org/rfc/rfc2328.txt.

[23] F. Papadopoulos, D.Krioukov, M.Bogua, and A. Vahdat. Greedy Forwarding in Dynamic Scale-Free Networks Embedded in Hyperbolic Metric Spaces. In Proceedings of IEEE INFOCOM, 2010.

[24]L.Saino, I. Psaras, and G. Pavlou. Hash-routing schemes for information centric networking. In Proceedings of ACM SIGCOMM ICN Workshop,2013.