Enhanced Derived Fast Reroute Techniques in SDN

Michal Hraska
University of Zilina,
Zilina, Slovakia
michal.hraska@fri.uniza.sk

Jozef Papan
University of Zilina,
Zilina, Slovakia
jozef.papan@fri.uniza.sk

***Abstract*—In software-defined networks, proactive and reactive response procedures to line or node outages or failures are used, often not reflecting the escalating demands of deployed applications. With the expanding deployment of software-defined network in the Internet of Things, medical equipment, and military infrastructure, the requirements for stable data transmission are increasing. This also includes fast network recovery, thus minimizing the time-of-service unavailability. When designing the software-defined network topology, considering the scale, it is necessary to choose suitable mechanisms that guarantee stable availability and restoration of the connection after its interruption. Many of the mechanisms have their basis in IP networks, where they are already successfully deployed. This paper presents advanced fast network recovery techniques that show good statistical characteristics in their adaptation in an software-defined network environment. Some of them are designed to complement existing network recovery techniques in an OpenFlow environment.**

I. Introduction

With growing demands for data transfer due to an increasing number of devices connected to the internet network and Quality of Service (QoS), more demanding data transfer requirements are imposed [1]–[3]. Different types of devices are connected to the network and require different types of connection. In connection with the integration of technologies working in real-time, for example, Smart city solutions and their monitoring or medical devices, the requirement for high availability of these services is necessary.

The combination of innovative requirements and demands on low operating cost [4] often results in implementing new mechanisms, which conceptually improves existing procedures and sets new trends in the given areas. It is not different in telecommunication services and services that provide the network technologies for aforementioned segments - Smart city, medical devices, and others. For many segments, the transfer of competencies to the online or cloud space represents cost savings. The technology of particular abstraction of network components also helps in the mentioned trend, thanks to which it is possible to save costs again. It is the virtualization of network components, where the main computing operations are left to one or a smaller number of nodes which makes decisions instead of virtualized components thanks to the local topology knowledge.

Virtualized devices do not have to have such high computing power, and thus their operating costs can be lower. In general, networks designed in this way are called software-defined networks (SDN) [5]–[10]. Current and concrete developments also prove that IoT and Smart City solutions increasingly use the conceptual design of SDN [11].

However, when routing network traffic, there are situations when network devices have to reevaluate the routing path of transmitted data and databases [12]–[14]. These situations can occur, for example, because of an error at the physical layer of packer switching. In such a case, it is necessary to find a new path to the destination of packet routing in the shortest possible time, as seen in Fig. 1. convergence process is the effort to recover the network as quickly as possible to the stage when packets can be routed again. The convergence process is the effort to recover the network as quickly as possible to the stage when packets can be routed again. Networks of telecommunication providers require connection recovery times when searching alternative paths, often below 50ms [15] so it is possible to talk about a fast connection recovery, fast reroute, or fast failover mechanisms [9], [16], [17].

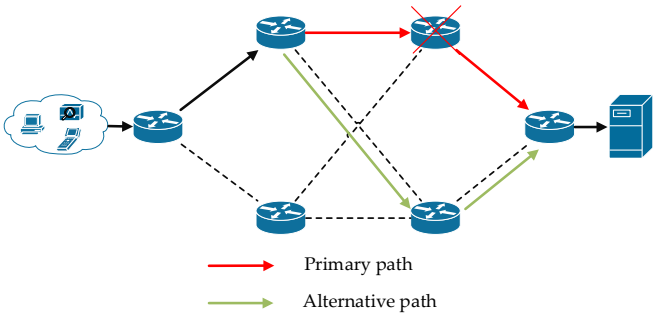

Fig. 1. Reroute strategy in standard IP networks

In order to detect the outage itself, support mechanisms are introduced in the networks so that the correspondents are often informed about the calculation of replacement work.

Outage detection and route reconstruction are one of the main criteria that Fast reroute mechanisms must include. As part of the reconstruction, among other things, the calculation of the backup route takes place and also its introduction into the routing records of network devices.

These techniques include monitoring physical layer parameters such as link or node status, power levels, and bit error rates. By detecting failures at the physical layer, Fast reroute can quickly reroute traffic around failed links or nodes, minimizing network downtime and ensuring high availability. Additionally, physical layer detection techniques can also be used to identify potential failures before they occur, allowing for proactive maintenance and reducing the likelihood of unexpected downtime.

Other techniques are based on protocols such as Internet Control Message Protocol (ICMP), Bidirectional Forwarding Detection (BFD), or LSP ping to detect link or node failures. When sending periodic packets or messages, Fast reroute can detect failures and quickly reroute traffic around them. Moreover, Fast reroute can use protocols like OSPF or IS-IS to exchange topology information and detect changes in the network. This allows Fast reroute to adapt to changes and maintain network availability quickly.

A. Types of protection

Link and node protection are two key techniques used in Fast reroute to ensure network availability and minimize downtime in the event of a link or node failure.

Link protection involves precomputing backup paths for each primary path in the network. These backup paths are used to reroute traffic around failed links. When routing, the network device uses the same next-hop node as the primary route but bypasses the failed link.

On the other hand, Node protection means precomputing backup paths for each node in the network. Node protection uses techniques like Remote LFA (RLFA) [18], Topology Independent Loop-Free Alternates (TI-LFA) [19], and Node Segment Protection (NSP). The alternative route may include other routes on the node failure bypass path.

B. Repair coverage

Repair coverage measures the effectiveness of Fast Reroute techniques in protecting against link or node failures in a network. It refers to the percentage of traffic that can be successfully rerouted around a failed component using Fast reroute techniques.

As we mentioned earlier, precomputing backup paths for both links and nodes in the network, and using appropriate detection techniques to identify failures quickly are essential when creating new network mechanisms.

By achieving high repair coverage, Fast reroute can significantly improve network availability and minimize downtime in the event of a failure. This is particularly important in networks that support critical services, such as financial transactions, emergency services, or healthcare applications, where even brief interruptions can have serious consequences.

C. Example of used detection mechanisms

One of the mechanisms used is Bidirectional Forward Detection (BFD) [20]–[22]. Each of the routing protocols uses some method of failure detection. BFD is an independent and fast mechanism in which neighboring nodes exchange messages, as seen in Fig. 2. It can be deployed together with protocols like OSPF, EIGRP, BGP, and MPLS. A node is declared unavailable if the requesting device does not receive a response within the specified time. The advantage of BFD is the fact that messages can be evaluated by the line interface and not by the processor, as is the case with other detection mechanisms.

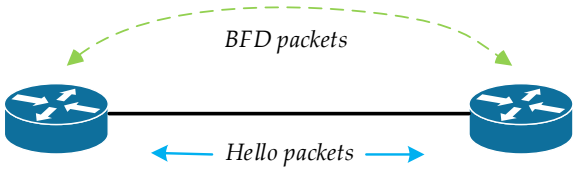

Fig. 2. BFD mechanism

BFD defines a number of states that describe the status of a BFD session between two devices. There are four transition states of the BFD mechanism.

- "Down" state is the initial state of a BFD session, indicating that no communication has been established between the two devices. In this state, BFD sends periodic packets to the remote device to try to establish communication.

- "Init" state is reached when the remote device responds to the BFD packets sent in the Down state. BFD establishes communication and begins to negotiate session parameters.

- "Up" state is the final state of a BFD session, indicating that communication has been successfully established between the two devices. Mechanism continuously sends and receives packets to monitor the status of the communication path.

Beside that, there is "AdminDown" state, which indicates that the session has been manually disabled by an administrator and BFD does not send or receive any packets.

The "AdminDown" and "Down" states are considered inactive states, while the "Init" and "Up" states are considered active states. BFD can also enter a "Diagnostic" state if a diagnostic test is being performed, and a "Down (remote)" state if the remote device has been detected as down.

In recent years, real-time data transmission has become a critical requirement. As we mentioned, due to the involvement of new technologies connected to the internet network, there is a need to ensure high availability in addition to alternative routing routes. Fast reroute is an important aspect of the concept of SDN networks also due to the expanding application in IoT [23], medical [24] and military network infrastructure.

II. CURRENT SDN IMPLEMENTATION AND OPENFLOW

The implementation of a specific SDN solution depends on the software with which it is implemented and, therefore, may differ, especially from the protocol and the way devices communicate at the given layers [25]. One of the best-known and most used solutions is OpenFlow [26]–[29].

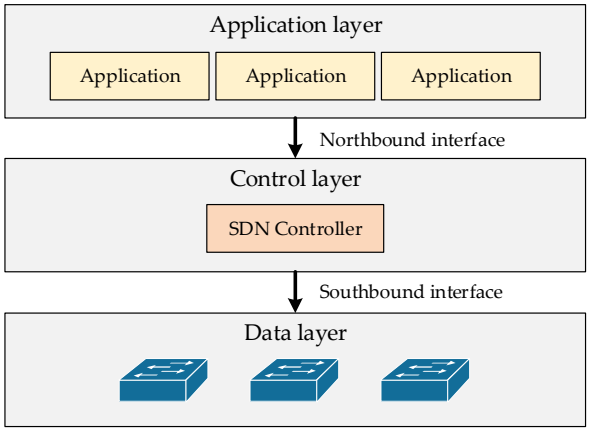

Fig. 3. General SDN structure

Software-defined networks can be designed in a multi-layer scheme, as seen in Fig. 3, where virtualized software network devices are placed in the data layer and handle packet routing based on rules. The number of devices on each layer varies depending on the application of the SDN solution.

A. SDN layers

- **Application layer** consists of the network applications that run on top of the SDN controllers. These applications can be used for network monitoring, security, traffic engineering, and other network management tasks.
- **Control layer** is the most interesting in a concept of a SDN design. This layer is responsible for managing and configuring the network devices in the infrastructure layer. It includes the SDN controllers, which receive network topology information from the infrastructure layer and use it to make forwarding decisions.
- **Data layer** includes the virtual network devices such as switches, routers, and access points that form the foundation of the network. These virtual devices are placed physically on a computational node, for example, servers.

Beside the application, control and data layer, there are interfaces to communicate between each layer.

- **Northbound API** is the interface between the application layer and the control layer. It allows network applications to communicate with the SDN controller and access network topology information.
- **Southbound API** is the interface between the control layer and the data layer. It allows the SDN controller to communicate with the network devices and configure them according to the network policies and rules.

Rules are stored in Flow tables, Group tables, and other structures [30], as seen in Fig. 4. These records are received from a superior device called a controller, which is logically located in the control layer. In addition, the SDN concept can also include the applications themselves, which are located in the application layer. Communication between devices at different levels is realized by interfaces - southbound and northbound interface.

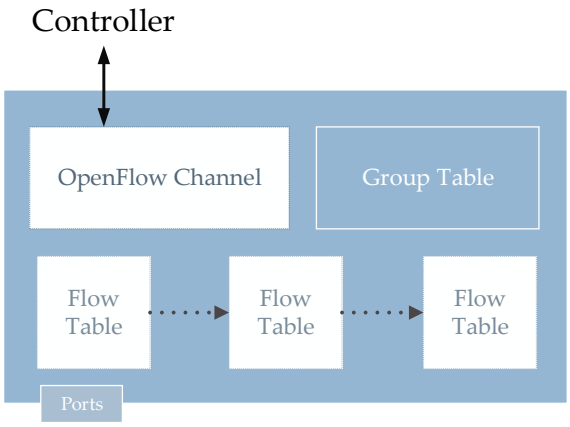

Fig. 4. OpenFlow switch structure

Expanding topologies also within SDN technology requires a more effective solution to connection failures or service unavailability. The fact that the software switch must always contact the controller when the routing record is missing can also be a disadvantage in the event of a line failure. Some advanced techniques from classic IP networks can help solve fast reroute in SDN. Delays and problems in recovering from outages can pose a serious risk to data traffic [31].

III. New SDN Fast Reroute concepts and mechanisms

In addition to some extensions offered by OpenFlow in terms of network recovery and routing techniques the event of an outage, new mechanisms are emerging not only for outage detection [32], but also for themselves, for example, local routing solutions in SDN.

One of the original mechanisms that accelerate the decision-making of SDN networks is Fast failover groups, which allow leaving part of the routing decision to the switches themselves. In addition, the OpenState plugin for OpenFlow was introduced, offering even better network recovery times than Fast failover groups [33].

A. Crankback in SDN

Crankback signaling, in the sense of classical IP networks, is used as an extension for Multiprotocol Label Switching (MPLS) and Generalized Multiprotocol Label Switching (GMPLS) Resource Reservation Protocol - Traffic Engineering (RSVP-TE) networks [34].

A crankback mechanism is a scheme that allows data involved in the construction of a network path to return from a certain point. The intention is not to lose said information and additionally to gain knowledge of the problem when creating a path, for example, in the event of a network link failure. In this way, it is possible to purposefully construct new paths from the source to the destination when routing packets. The advantage of the targeted creation of alternative routes is the possibility to skip the routes on which we know that a crankback has occurred

- and, therefore, the routing along the given route has failed. The principle of the mechanism is shown in Fig. 5.

In understanding SDN networks, the idea of using a crankback mechanism is considered in a very similar context. The basic assumption [35] is that crankback is not used in path building for packets, and thus that packets would be marked for path building. The difference from marking in classic IP networks is that the routed data itself receives the flags.

When detecting the unavailability of a link or a node using the crankback mechanism, two situations can occur. The first situation is the node can detect the failure and forward the packets to a new route by itself. The second method is when it is necessary to reroute the packets along their predefined path until a suitable network node capable of handling the forwarding request is found. The advantage, if the first option occurs, is that the fast failover groups that were described above can ensure the switching of packets without the need to contact the controller. In the second variant, when it is necessary to find a suitable node for routing, the situation is different. Considering the forwarding of data through several network nodes, it is necessary to include a controller in the overall process. The latter will ensure the distribution of packet marking for other network nodes as well.

A change from the classic approach of the crankback mechanism is offered by the modification by the authors in [35]. This modification involves redirecting network traffic after detecting a failure based on packet marking. Also, this change will cause a new detour to be defined for other network traffic as well. Therefore, only initially routed packets are returned from the node that detected the failure using the crankback principle. Every other one is routed by a new route, which the given switches receive as a table state change distributed by OpenFlow.

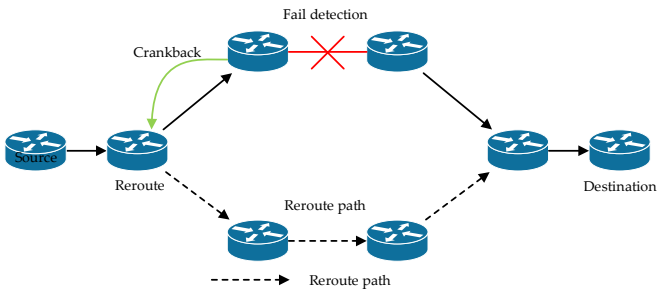

Fig. 5. Crankback mechanism working principle

B. SD-FAST architecture

SD-FAST is a scheme designed to redirect packets in case of link failure. The implementation of the mechanism is solved on each switch in the data layer of the network infrastructure. Uses BFD to monitor line status. SD-FAST is designed to affect only the traffic affected by the network link failure.

In case of failure detection, SD-FAST uses data modification in the flow table, which it obtains from Open vSwitch Database Management Protocol (OVSDB). If there is a change in the overall state, for example, in the case of a link failure and subsequent failure detection, the switch also modifies the affected packets. This fact helps to prevent backtracking of another, alternative route.

The SD-FAST architecture specifically extends the application [36], control, and data plane with its components. The route planner participates in the construction of roads, which creates connections between given network nodes. Based on the established routes, the route configurator then selects the primary and backup output ports of the devices for routing data along the selected route to the destination. Only primary routes are included in the flow tables, and alternative routes are selected from the backup table in case of the detection of a fault. The crankback mechanism described above can also be used in the SD-FAST principle.

C. FLR and BLR mechanisms

When designing new mechanisms for the reconstruction of routing paths in SDN, it is necessary to take into account the software limitations of virtual switches. For example, in an OpenFlow switch, routing records are stored in Ternary Content Addressable Memory (TCAM), which is limited in size. In practice, these memory locations are limited to 1500 records. The problem arises in the commercial deployment of the OpenFlow solution because the amount of routing records can reach tens of thousands per second [37].

The mentioned amount of routing records is also related to the fact that additional routing records are needed for one data stream when solving network recovery using a proactive approach.

With proactive protection of the node, the calculation of alternative routing paths is introduced in the event of failure of the primary route for data transmission. This means that the TCAM memory must contain extra records at the cost of faster network recovery. Even though the OpenFlow switch remembers the routing data it receives from the controller, it discards old routing records when it runs out of memory, which means re-loading the controller and additional requests for older data flows.

The basic prerequisite for deploying the Forward Local Rerouting (FLR) and Backward Local Rerouting (BLR) mechanisms is the implementation in OpenFlow. The creators [38] use the `OPTIONAL ACTION` field, which will carry an attribute about the backup output port on the device. This port will be used in case of failure of the primary line, and thanks to the use of the value in the `OPTIONAL ACTION` attribute, the number of routing records in the TCAM memory will be reduced.

The principle of the FLR mechanism is the creation of a series of backup routes when in the event of a failure, this network traffic is routed directly from the point of failure to the next switch on the primary path. The next node with the smallest number of other nodes to the destination switch is selected. An important aspect when choosing backup routes is to ensure loop-freeness.

The algorithm for finding backup routes creates a backup route for each line on the source-to-destination route that passes

through as few other nodes as possible. The flow direction of the original route remains respected. The shortest possible link is used to determine the path. In this way, a set of backup routes is created for each node in the direction of the data flow on the primary link. Subsequently, the alternative connection that contains the fewest switches is selected from the set. In exceptional cases, routing with a loop can occur in a set of alternative connections where the same node has a different output rule for two different cases.

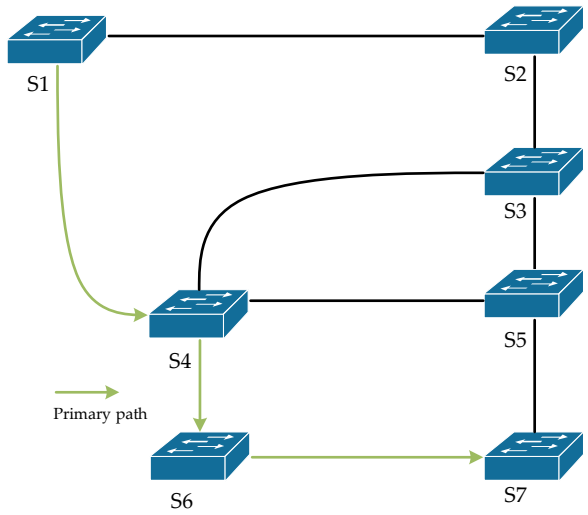

Fig. 6. Model topology

An example of the functioning of the FLR mechanism can be visualized using the topology in Fig. 6. The data flow has its source in the device S1 and goes to S7, where the shortest path is marked as {S1, S4, S6, S7}. However, it is necessary to proactively anticipate possible router or line failure. The algorithm using FLR will calculate suitable alternative routes for each route to the destination. The set that the algorithm will include is {S1-S4}, {S4-S6}, {S6-S7}. The replacement set, in case of failure, will contain pre-calculated alternative routes for the corresponding lines as follows:

- {S1-S4}: {S1, S2, S3, S5, S7}
- {S4-S6}: {S1, S4, S5, S7}
- {S6-S7}: {S1, S4, S5, S7}

However, the given an example does not yet take into account the solution of loops that can and do occur in the network.

In order to avoid deploying two identical routing records with different outputs, the algorithm checks the occurrence of the same node in the set of backup routes when searching for an alternative route.

The BLR mechanism calculates, like the FLR, a backup route, but only one. In the event of a failure on the primary route, the traffic is forwarded back to the source and then routed through the backup line. Compared to FLR, BLR is less complex for calculations. Its use is more appropriate in smaller topologies. The disadvantage is the additional load when the failed data flow must first be routed back to the source. In addition, it is necessary to mark the failed data with a flag, the processing of which may cause further delay.

IV. Discussion

The third part of the work was devoted to advanced techniques, the intention of which is to provide experience from classic IP networks in the field of recovery after line failure for use in software-defined networks. The intention was to provide an overview of new mechanisms that are not only in the stage of theoretical formulation but also at least in simulation implementation.

The use of the Crankback principle in the SDN network is concretely implemented, for example, using the SD-FAST design. However, it requires a more complicated design structure for the necessary modifications to the classic SDN topology built on the Open flow protocol.

However, the SD-FAST approach itself, according to measurements, indicates a several times improved packet routing time in case of link failure since it is faster than Crankback [34] in finding return paths. In general, SD-FAST reduced the total delay by 64% [36] compared to the Crankback mechanism implementation solution. Testing was done in the Mininet 2.3 environment using the Ryu controller and OVS switches in a data plane. USNET topology with 60 nodes was chosen to evaluate SD-FAST, where 12 devices were always selected for testing. The process of testing was repeated five times to choose a unique pair of nodes every time. Packet traffic was simulated with ICMP ping and iPerf, also the exchange of multimedia files between node pairs was done. As we mentioned, the delay was one of the evaluated metrics among network throughput and backtracking time.

Among the mechanisms with advanced fast reroute options, we can also include the principle based on Forward and Backward Local Rerouting algorithms. The main advantage of implementing FLR and BLR is reducing the number of necessary entries in the TCAM memory. Thanks to this, any necessary requests to the controller are reduced due to the demand for knowledge of the routing records. However, the alternative path search algorithm guarantees loop-freeness, eliminating another major problem in routing packets in the network.

During testing, it was shown that by using the BLR mechanism in combination with the attribute in the `OPTIONAL ACTION` field, it is possible to reduce the number of additional routing records by up to 75% compared to the use of the BLR mechanism and records in the Group table structure [38]. Compared with the existing OpenFlow Segment Protection (OSP) principle [39], a 40% reduction in records was achieved.

Deploying the mentioned mechanisms on a model SDN topology reduced the number of additional routing records by 65% when comparing the FLR algorithm with the `OPTIONAL ACTION` attribute and OSP. The evaluation included the result of the experiment, where the topology included 5 devices, which are the most commonly used models in practice [40]. The Mininet environment with random and Internet2 topology was used to study performance. The testing environment includes

sufficient computational capacity [38]. Three performance metrics were involved in the overall evaluation: average number of additional rules, average number of additional hops, and backup bandwidth sharing efficiency.

In the future of this research, it is possible to analyze the deployed mechanisms at the level of telecommunication service providers and compare them with theoretically presented models. SD-WAN networks, which are adapted to ensure manageability, also form a large domain, but where the abstraction and virtualization of network devices is not so strong, and these solutions are closer to traditional IP networks.

V. Conclusion

SDN is still a relatively new technology that finds its application, for example, in the networks of telecommunication providers or as a solution for the background of larger data centers. With the advent of IoT, which incorporates potentially large-scale networks, it is necessary that these networks are resilient to outages. This characteristic is also important for other mission-critical applications where we cannot afford delays or link outages.

The solutions considered for link reconstruction after an outage are often associated with the type of SDN implementation. A look at advanced network recovery techniques offers a broader view of other options that are already implemented in IP networks. However, these mechanisms are many times theoretical in the notion of software-defined networks. The analysis we present shows some of the possibilities that can be transferred from IP networks to the SDN world.

The use of Crankback, a novel mechanism for fast recovery in SDN, has shown significant promise in improving network resilience and reducing recovery time in case of failures.

SD Fast leverages the inherent flexibility of SDN architecture to proactively detect and isolate network faults, allowing for a swift recovery of network services. By optimizing packet forwarding paths and dynamically adjusting flow rules, SD Fast ensures a seamless and efficient transition to alternate paths, thereby minimizing the impact of the failure on network performance.

Forward Local Rerouting is a novel fast recovery technique that has shown tremendous potential in improving network performance by quickly rerouting traffic around link or node failures. By utilizing FLR in SDN architectures, network administrators can proactively detect and isolate faults, providing rapid recovery and preventing service disruption. Additionally, the use of FLR helps to reduce the need for manual intervention during network recovery, enabling network operators to respond to faults quickly and efficiently.

With the increasing reliance on network infrastructure for critical applications and services, fast recovery mechanisms such as Crankback, SD Fast, FLR, and BLR are becoming essential for ensuring business continuity and maintaining customer satisfaction.

VI. Acknowledgement

This publication was realized with support of the Operational Programme Integrated Infrastructure in frame of the project: Intelligent systems for UAV real-time operation and data processing, code ITMS2014+: 313011V422 and co-financed by the European Regional Development Fund.

VII. References

[1] G. Koman, M. Kubina, M. Holubčík, and J. Soviar, “Possibilities of application a big data in the company innovation process,” *Communications in Computer and Information Science*, vol. 877, pp. 646–657, 2018, doi: 10.1007/978-3-319-95204-8_54, ISBN: 9783319952031.

[2] M. A. El-Serafy, A. M. Elsayed, M. H. Aly, E.-S. A. El-Badawy, and I. A. Ghaleb, “Multiple Routing Configurations for Datacenter Disaster Recovery Applicability and Challenges,” in *2014 International Conference on Computer and Communication Engineering*, 2014, pp. 146–149, doi: 10.1109/ICCCE.2014.51, ISBN: 978-1-4799-7635-5.

[3] S. Ren, W. Dou, and Y. Wang, “A deterministic network calculus enabled QoS routing on software defined network,” 2017, vol. 2017-Janua, pp. 182–186, doi: 10.1109/ICCSN.2017.8230102.

[4] G. Caiza, S. Chiliquinga, S. Manzano, and M. V. Garcia, “Software-Defined Network (SDN) Based Internet of Things within the context of low-cost automation,” *IEEE International Conference on Industrial Informatics (INDIN)*, vol. 2020-July, pp. 587–591, Jul. 2020, doi: 10.1109/INDIN45582.2020.9442180, ISBN: 9781728149646.

[5] J. Olorunfemi Abe and H. A. Mantar, “Multipath routing and brokering in inter-domain or inter-as with SDN: A model,” *Proceedings - 2017 Advances in Wireless and Optical Communications, RTUWO 2017*, vol. 2017-January, pp. 192–197, Dec. 2017, doi: 10.1109/RTUWO.2017.8228532, ISBN: 9781538605851.

[6] “Network Softwarization and Virtualization in Future Networks: The promise of SDN, NFV, MEC, and Fog/Cloud Computing,” *Multimedia Streaming in SDN/NFV and 5G Networks*, pp. 99–118, Dec. 2022, doi: 10.1002/9781119800828.CH6.

[7] H. Hasan, J. Cosmas, Z. Zaharis, P. Lazaridis, and S. Khwandah, “Development of FRR mechanism by adopting SDN notion,” in *2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2016, pp. 1–7, doi: 10.1109/SOFTCOM.2016.7772133.

[8] M. A. Moyeen, F. Tang, D. Saha, and I. Haque, “SD-FAST: A Packet Rerouting Architecture in SDN,” *15th International Conference on Network and Service Management, CNSM 2019*, Oct. 2019, doi: 10.23919/CNSM46954.2019.9012703, ISBN: 9783903176249.

[9] C. Cascone, D. Sanvito, L. Pollini, A. Capone, and B. Sansò, “Fast failure detection and recovery in SDN with stateful data plane,” *International Journal of Network Management*, vol. 27, no. 2, pp. 1–14, Mar. 2017, doi: 10.1002/nem.1957, ISSN: 10991190.

[10] P. Krongbaramee and Y. Somchit, “Implementation of SDN Stateful Firewall on Data Plane using Open vSwitch,” in *2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2018, pp. 1–5, doi: 10.1109/JCSSE.2018.8457354, ISBN: 978-1-5386-5538-2.

[11] L. Ogrodowczyk, B. Belter, and M. Leclerc, “IoT Ecosystem over programmable SDN infrastructure for smart city applications,” in *Proceedings - European Workshop on Software-Defined Networks, EWSDN* , 2017, vol. 2016-October, pp. 49–51, doi: 10.1109/EWSDN.2016.17, ISBN: 9781509061464.

[12] M. Kvet and M. Kvet, “Relational pre-indexing layer supervised by the DB-index-consolidator background process,” *Conference of Open Innovation Association, FRUCT*, vol. 2021-January, Jan. 2021, doi: 10.23919/FRUCT50888.2021.9347573, ISBN: 9789526924441.

[13] M. Kvet, “Relational data index consolidation,” *Conference of Open Innovation Association, FRUCT*, vol. 2021-January, Jan. 2021, doi: 10.23919/FRUCT50888.2021.9347614, ISBN: 9789526924441.

[14] M. Kvet, “Autonomous Temporal Transaction Database,” *Conference of Open Innovation Association, FRUCT*, vol. 2021-October, pp. 121–128, 2021, doi: 10.23919/FRUCT53335.2021.9599977, ISBN: 9789526924465.

[15] W. Gray, A. Tsokanos, and R. Kirner, "Multi-Link Failure Effects on MPLS Resilient Fast-Reroute Network Architectures," in *Proceedings - 2021 IEEE 24th International Symposium on Real-Time Distributed Computing, ISORC 2021*, 2021, pp. 29–33, doi: 10.1109/ISORC52013.2021.00015, ISBN: 9781665404143.

[16] O. Lemeshko, O. Yeremenko, B. Sleiman, and M. Yevdokymenko, "Fast reroute model with realization of path and bandwidth protection scheme in sdn," *Advances in Electrical and Electronic Engineering*, vol. 18, no. 1, pp. 23–30, 2020, doi: 10.15598/aeee.v18i1.3548, ISSN: 18043119.

[17] A. Ghannami, C. Shao, Aiman Ghannami, ChenXi Shao, A. Ghannami, and C. Shao, "Efficient fast recovery mechanism in Software-Defined Networks: Multipath routing approach," in *2016 11th International Conference for Internet Technology and Secured Transactions, ICITST 2016*, 2017, pp. 432–435, doi: 10.1109/ICITST.2016.7856747, ISBN: 9781908320735.

[18] P. Sarkar, S. Hegde, C. Bowers, H. Gredler, and S. Litkowski, "Remote-LFA Node Protection and Manageability," RFC8102, 2017, ISSN: 2070-1721.

[19] S. Litkowski *et al.*, "Topology Independent Fast Reroute using Segment Routing," Aug. 2020, http://www.ietf.org/internet-drafts/draft-ietf-rtgwg-segment-routing-ti-lfa-04.txt.

[20] S. M. Kim, G. Yang, C. Yoo, and S. G. Min, "BFD-based link latency measurement in software defined networking," in *2017 13th International Conference on Network and Service Management, CNSM 2017*, 2018, vol. 2018-Janua, pp. 1–6, doi: 10.23919/CNSM.2017.8256023, ISBN: 9783901882982.

[21] C. Pignataro, D. Ward, N. Akiya, M. Bhatia, and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)," RFC Editor, 2016, ISSN: 2070-1721.

[22] Di. Siqueira, T. Pinheiro, J. Dantas, and P. MacIel, "Dependability evaluation in a convergent network service using BGP and BFD protocols," *Conf Proc IEEE Int Conf Syst Man Cybern*, vol. 2019-October, pp. 2378–2383, Oct. 2019, doi: 10.1109/SMC.2019.8914368, ISBN: 9781728145693.

[23] J. Papan, P. Segec, O. Yeremenko, I. Bridova, and M. Hodon, "A New Bit Repair Fast Reroute Mechanism for Smart Sensors IoT Network Infrastructure," *Sensors*, vol. 20, no. 18, p. 5230, Sep. 2020, doi: 10.3390/s20185230, ISSN: 1424-8220.

[24] S. Badotra, Di. Nagpal, S. N. Panda, S. Tanwar, and S. Bajaj, "IoT-Enabled Healthcare Network with SDN," *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, pp. 38–42, Jun. 2020, doi: 10.1109/ICRITO48877.2020.9197807, ISBN: 9781728170169.

[25] P. Krongbaramee and Y. Somchit, "Implementation of SDN Stateful Firewall on Data Plane using Open vSwitch," *Proceeding of 2018 15th International Joint Conference on Computer Science and Software Engineering, JCSSE 2018*, Sep. 2018, doi: 10.1109/JCSSE.2018.8457354, ISBN: 9781538655382.

[26] Y. Kuzmin and D. Volkanov, "Method for Packet Classification for OpenFlow Classification Table Using Graphics Processing Unit," *4th International Science and Technology Conference "Modern Network Technologies 2022", MoNeTec 2022 - Proceedings*, 2022, doi: 10.1109/MONETEC55448.2022.9960755, ISBN: 9781665472463.

[27] M. Borokhovich, L. Schiff, and S. Schmid, "Provable Data Plane Connectivity with Local Fast Failover Introducing OpenFlow Graph Algorithms," doi: 10.1145/2620728.2620746, ISBN: 9781450329897.

[28] V. Muthumanikandan, C. Valliyammai, and S. Harish, "Link Failure Detection and Alternate Path Tracing in OpenFlow Based Ethernet Networks," in *2017 9th International Conference on Advanced Computing, ICoAC 2017*, 2018, pp. 352–356, doi: 10.1109/ICoAC.2017.8441439, ISBN: 9781538643495.

[29] P. Thorat, S. Jeon, and H. Choo, "Enhanced local detouring mechanisms for rapid and lightweight failure recovery in OpenFlow networks," *Comput Commun*, vol. 108, pp. 78–93, Aug. 2017, doi: 10.1016/j.comcom.2017.04.005, ISSN: 01403664.

[30] C. Wang and S. Yan, "Scaling SDN network with self-adjusting architecture," *Proceedings of 2016 IEEE International Conference on Electronic Information and Communication Technology, ICEICT 2016*, pp. 116–120, Mar. 2017, doi: 10.1109/ICEICT.2016.7879664, ISBN: 9781509007288.

[31] N. Senthilkumaran, R. Thangarjan, and S. K. Nivetha, "Memory and Load-aware Traffic Rerouting (MLTR) in OpenFlow-based SDN; Memory and Load-aware Traffic Rerouting (MLTR) in OpenFlow-based SDN," 2019, https://www.opennetworking.org/, ISBN: 9781728110349.

[32] N. L. M. Van Adrichem, B. J. Van Asten, and F. A. Kuipers, "Fast recovery in software-defined networks," *Proceedings - 2014 3rd European Workshop on Software-Defined Networks, EWSDN 2014*, pp. 61–66, Dec. 2014, doi: 10.1109/EWSDN.2014.13, ISBN: 9781479969197.

[33] M. S. M. Zahid, B. Isyaku, and F. A. Fadzil, "Recovery of software defined network from multiple failures: Openstate vs openflow," in *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, 2018, vol. 2017-October, pp. 1178–1183, doi: 10.1109/AICCSA.2017.32, ISBN: 9781538635810.

[34] A. Farrel, A. Satyanarayana, A. Iwata, N. Fujita, and G. Ash, "Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE," RFC Editor, Jul. 2007, https://www.rfc-editor.org/info/rfc4920, ISSN: 2070-1721.

[35] A. Capone, C. Cascone, A. Q. T. Nguyen, and B. Sansò, "Detour planning for fast and reliable failure recovery in SDN with OpenState," in *2015 11th International Conference on the Design of Reliable Communication Networks, DRCN 2015*, 2015, pp. 25–32, doi: 10.1109/DRCN.2015.7148981, ISBN: 9781479977956.

[36] M. A. Moyeen, F. Tang, D. Saha, and I. Haque, "SD-FAST: A Packet Rerouting Architecture in SDN," *15th International Conference on Network and Service Management, CNSM 2019*, Oct. 2019, doi: 10.23919/CNSM46954.2019.9012703, ISBN: 9783903176249.

[37] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, pp. 267–280, 2010, doi: 10.1145/1879141.1879175, ISBN: 9781450300575.

[38] P. M. Mohan, T. Truong-Huu, and M. Gurusamy, "TCAM-Aware Local Rerouting for Fast and Efficient Failure Recovery in Software Defined Networks," pp. 1–6, Mar. 2016, doi: 10.1109/GLOCOM.2015.7417309.

[39] A. Sgambelluri, A. Giorgetti, F. Cugini, F. Paolucci, and P. Castoldi, "OpenFlow-based segment protection in Ethernet networks," *Journal of Optical Communications and Networking*, vol. 5, no. 9, pp. 1066–1075, 2013, doi: 10.1364/JOCN.5.001066, ISSN: 19430620.

[40] A. Giorgetti, F. Cugini, F. Paolucci, L. Valcarenghi, A. Pistone, and P. Castoldi, "Performance Analysis of Media Redundancy Protocol (MRP)," *IEEE Trans Industr Inform*, vol. 9, no. 1, 2013, doi: 10.1109/TII.2012.2186584.