IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 January 2024, accepted 13 March 2024, date of publication 18 March 2024, date of current version 3 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3379150

RESEARCH ARTICLE

The New Label Bit Repair Fast Reroute Mechanism

JOZEF PAPAN, TOMAS CHOVANEC, IVANA BRIDOVA, AND MICHAL KVET, (Member, IEEE)
University of Žilina, 010 26 Žilina, Slovakia

Corresponding author: Jozef Papan (Jozef.Papan@uniza.sk)

This research was funded by the Slovak Grant Agency VEGA project Fast Reroute, No. 1/0316/24.

ABSTRACT The paper is devoted to a research area focused on solving problems associated with outages and connection interruptions in modern IP networks. The goal is to design and implement such a mechanism that ensures the change of the original route through an exact order of routers in the alternate route. The restoration of the connection through an alternative pre-calculated path will thus take place faster than it was in the case of the original B-REP (Bit Repair Fast Reroute Mechanism) mechanism, as the influence of unwanted loops is eliminated. Fast recovery is crucial in today's IP networks because fast recovery minimizes service interruptions that occur during normal network convergence. Increasing the stability and recovery speed of modern IP networks has the potential to positively impact the provision of critical services that require reliable and continuous connectivity. Solutions like Label B-REP Fast Reroute (FRR) are therefore important for progress in this area and for improving the properties of future networks. The functionality of the proposed Label B-REP mechanism is verified through tests in the OMNeT++ simulation environment.

INDEX TERMS Fast Reroute, FRR, label B-REP, B-REP.

I. INTRODUCTION

The research area focused on solving the problems associated with drops and interruptions of connection in modern IP networks and deals with finding new and better ways to improve the stability and reliability of connection in IP networks. This area addresses sub-research problems such as:

Network convergence occurs when a link or router in a network goes down when routing protocols must rebuild paths for data transmission. Convergence time depends on the size and topology of the network, used routing protocols, load, etc. Therefore, it is not possible to determine an unambiguous convergence time valid for all cases. The speed of convergence is a critical factor, as services may be unavailable or limited during this time. In order to minimize the convergence time and improve the stability of the network, techniques such as Fast Reroute (FRR) [1], route backup, and redundant routers are currently used, which try to respond quickly to outages and minimize their negative impact on users and provided services.

FRR [2], [3] are technologies and mechanisms that are designed to minimize network recovery time after a network outage through an alternative pre-calculated route, which helps to minimize connection interruptions and reduce negative impacts on the services provided [4], [5], [6].

Routing protocols and mechanisms: research focuses on improvements to existing routing protocols such as OSPF (Open Shortest Path First) [7], [8], IS-IS (Intermediate System to Intermediate System) [9], or BGP (Border Gateway Protocol) [10], [11], [12], but also on designing new mechanisms IPFRR (Internet Protocol Fast Reroute) [13], which would ensure effective routing [14] and thus minimize outages [4], [15].

Detection and diagnosis of outages: Research work seeks to improve methods and tools for the detection and diagnosis of network outages, which is important for rapid repair and restoration of connectivity [16], [17], [18].

Backup and redundancy are looking for solutions so that in the event of a device or line failure, traffic is automatically redirected to backup devices [19], [20].

In current IP networks, the biggest problems occur in the form of - stability, availability, and performance, and the aforementioned areas are devoted to how to achieve higher

The associate editor coordinating the review of this manuscript and approving it for publication was Xiao-Sheng Si.

stability, availability, and performance in modern IP networks, which is key to the effective functioning of current ICT.

The presented paper discusses a new approach that will allow increasing connection stability in IP networks through FRR technology and the design of a new Label B-REP(Bit Repair Fast Reroute Mechanism)**FRR mechanism.** Searching for new solutions for the current problem, i.e., quick restoration of the connection in the network after an outage, is of great practical importance in IP networks.

The main goal of the article is the design, implementation, and verification of the new Label B-REP (Bit Repair Fast Reroute Mechanism) mechanism in the OMNeT + simulation environment, which can ensure faster network recovery compared to the B-REP mechanism, as it ensures the change of the original route through an unambiguous order of routers in the alternate route, thereby eliminating the phenomenon of unwanted loops that arose during the B-REP mechanism.

The paper itself is divided into eight sections. The second section focuses on introducing the reader to the issue of FRR. Here, we will describe the principle of FRR mechanisms, their basic properties, and their parameters. The reader will also be familiar with the terminology used within the framework of FRR mechanisms.

The first section presents an introduction to the issue. The second section provides an overview of related research works dealing with the issue of FRR and supporting mechanisms for fast convergence in the network. In the third section, the Label B-REP design methodology is presented. The fourth presents the implementation of the modified Label B-rep in the OMNET++ simulation environment. The fifth section describes the testing and verification of the Label B-REP mechanism in the OMNeT++ simulation environment.

The sixth section provides a discussion of the results achieved, and the seventh section provides conclusions.

II. RELATED WORKS

Nowadays, when network operation is a key part of many business processes, streaming services, or other real-time services, it is important to ensure that the network is reliable and resistant to outages.

The section presents the latest FRR solutions that are currently available and enable fast network recovery in the event of outages.

FRR solutions are based on the network convergence process [21], [22], [23], [24]. In the event of a link or router failure in the network, the neighboring routers inform each other about the change in the network topology and recalculate new routes to the specified destinations. During this process, the flow of data from the source to the destination is interrupted - packets are usually lost (dropped). Convergence in an OSPF network can take from a few seconds to a few minutes, depending on the size and complexity of the network. To minimize downtime and data loss, it is important that routers are properly configured and that an appropriate error detection method is used. All the mentioned factors vary depending on the routing protocol used [23], [25].

The critical factor in the case of network convergence is the convergence time itself. There is an effort to make it as short as possible, and it is necessary to know and estimate this time already during the design of the network itself [26]. Convergence time depends on the protocol used, timer settings, and network configuration. However, in general, the convergence period can consist of the following time periods:

- Fault Propagation - The time it takes for information about the fault to be transmitted to other routers in the network and for neighboring nodes to learn about the link failure. It usually takes 10 ms to 100 ms per next-hop router [25], [27].
- Calculation of new routing information - The time required to calculate new routing information for the affected routers in the network. It varies depending on the routing protocol used and the size of the network. For link-state routing protocols using Dijkstra's algorithm, the calculation usually takes several ms [25].
- Updating the routing table - The time of this phase is directly dependent on the hardware equipment of the router, its performance, the implementation of the routing protocol, and the number of prefixes affected by the error, but it can be in the order of hundreds of milliseconds [25], [27].

IPFRR is a fast-forwarding mechanism that, when deployed, ensures that incoming packets are not lost during the convergence process [25], [27].

FRR can be implemented in different ways, which differ from each other in the calculation of the alternative path [28], [29].

When activating the Fast ReRoute mechanisms, early detection of the outage and its repair using the deployed FRR mechanism is important. Due to the fact that at the time of connection failure, the repair mechanism already has backup routes prepared in advance, so their actual deployment takes less time than detecting a specific line failure on the router. Therefore, it is necessary to focus on outage detection techniques that can minimize this detection time [30].

The Fast Reroute Mechanism (FRR) has two critical components that need to be protected. It is a line and a node [27].

Repair Coverage is the ability of the Fast ReRoute mechanism to restore a route after a failure and expresses the effectiveness of individual FRR mechanisms. Indicates the percentage of network routes that can be restored using the Fast ReRoute mechanism. Patch coverage can be affected by various factors such as network topology, link capacity, network throughput, patch algorithm, etc. [27], [31].

A proactive approach within FRR mechanisms means that elements in the network do not wait for a problem to arise, but are actively prepared for an outage [32].

It is characteristic of ReRoute that thanks to the pre-calculated alternative route, the speed of their recovery after a line or node failure depends only on the time of failure detection, which is also considered to be their significant advantage [27].

Each mechanism is unique, and they differ from each other in the way or improvements in how they approach finding an alternative path, as well as in the conditions they can tolerate [33], [34].

The existing IPFRR mechanisms differ in principle in how and by what criteria they calculate the alternative route. Among the most used FRR mechanisms are Loop-Free Alternate (LFA) and its extended version, Remote LFA (RLFA), or Traffic Engineered TI-LFA [35]. Other IPFRR mechanisms include:

- Equal-cost multi-path (ECMP) [36], [37] – uses the principle of a duplicate path with the same metric.
- Multiple Routing Configurations (MRC) [29], [31], [37], [38], [39], [40], [41] – use the principle of multiple routing configurations for specific outages in networks.
- Not-Via Addresses [42] special IP addresses to explicitly indicate network outages.

Further mechanisms based on tunneling, Maximally Redundant Trees (MRT) [43], [44], which will use alternative tree configurations and other IPFRR mechanisms with a similar method [44], [45], [46].

Fibbing is a new network technology that can provide flexible routing in IP networks and redirect flows to desired paths. In a fibbing network, the network controller is in place to generate fake messages and nodes by sending OSPF link state messages (LSAs). Fibbing was developed to achieve centralized control over distributed routing [38], [39].

The convergence time of the protocol is mainly dependent on the speed transfer time (FPT - Flood Pacing Time) of the routers in the network. Because the controller identifies the exact location of the failed link through monitoring cycles, it sends new Type 5 LSAs to reconfigure recovery paths. When a router receives LSA messages, it does not immediately send them to neighboring nodes. The router accumulates them and packs them into one packet, which it forwards only after the FPT timer expires. Since the failure recovery time depends on the FPT, the FPT time was set to 5 ms in the investigated system to reduce the network reconfiguration time [34].

MPLS is an OSI/ISO layer 2 and 3 technology that is widely used in backbone networks and Wide Area Networks (WANs). The reason for its dominance is its high performance and ability to quickly redirect.

MPLS consists of two key parts. **Multi Protocol** means it can carry any network protocol (network/layer 3) such as IP, IPv6, IPX, X.25, AppleTalk, **etc. Label Switching,** i.e., switching, not routing based on labels [47].

MPLS-TE FRR is only developed for MPLS-enabled networks, so it cannot be implemented without MPLS support. The prerequisite for IPFRR is an IGP, such as OSPF or IS-IS.

Conventional IP routing includes de-encapsulation and performs packet inspection, which affects additional processing [48] and slowdown. MPLS inserts its header between the layer 2 and 3 protocol headers, and at each hop in the network, the "label" value in the header changes. **This is different from IP packet forwarding.**

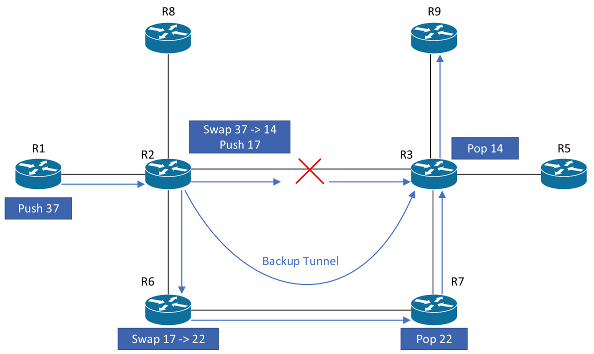


FIGURE 1. **MPLS - backup tunnel.**

MPLS [49] forwarding decisions are made by creating and exchanging MPLS labels and consulting the LIB (Label information base) forwarding tables. MPLS forwarding table entries map labels to the next hops. Each entry in the MPLS [6] forwarding table points to an entry in the MPLS next-hop table, which is typically an interface.

Fast ReRoute provides LSP (Label Switched Path) connection protection. This allows all traffic carried by the LSP that bypasses the fault to be rerouted. The router that connects the faulty connection makes the decision about redirecting the traffic completely locally. The tunnel end station is also informed of the link failure via IGP or RSVP; the end station then attempts to establish a new LSP that bypasses the failure.

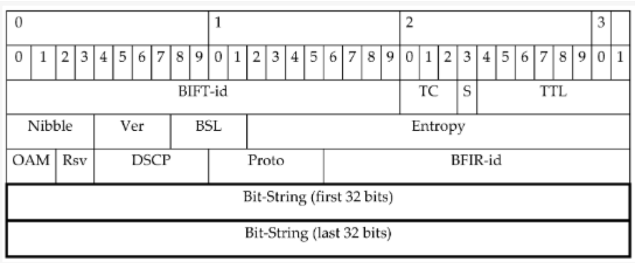
The example in Fig. 1 illustrates how Fast ReRoute is used to protect the flow carried in the TE tunnel between routers R1 and R9 as it traverses the link between R2 and R3. The TE tunnel from R1 to R9 is considered the primary tunnel and is defined by tags *37, 14,* and *Pop.* A backup tunnel is created to protect this connection. It leads from R2, R6, R7 and R3. This backup tunnel is defined by labels *17, 22,* and *Pop* [50], [51], [52].

When R2 detects that the link between it and R3 is no longer available, it redirects traffic destined for R3 through the backup tunnel. This is done by pushing label 17 (Push 17) on packets destined for R3 after the normal swap operation (which replaces label 37 with label 14) has been performed. By adding label 17 to the packets, it redirects them along the backup tunnel through the other interface to router R6, routing traffic around the faulty link. The forwarding of packets from the primary tunnel to the backup tunnel is decided exclusively by R2 after detecting a link failure.

MPLS TE provides protection against any of the following events:

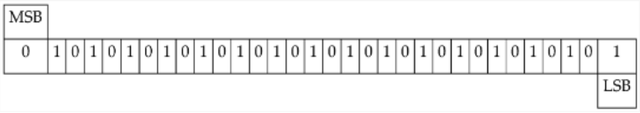
- One route failure.
- Single node failure or multiple path failure.

The Fast ReRoute and LSP mechanism for MPLS can provide a failure recovery time of less than 50ms when switching from a primary to a secondary LSP. The experiments carried



0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
BIFT-id																				TC			S	TTL							
Nibble				Ver				BSL				Entropy																			
OAM		Rsv		DSCP						Proto						BFIR-id															
Bit-String (first 32 bits)																															
Bit-String (last 32 bits)																															

FIGURE 2. BIER header.



MSB																															
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
																															LSB

FIGURE 3. The bit-string field in BIER.

out in the article [6] achieved a success rate of 87.5% in finding an alternative path within 50 ms.

A. OUR RESEARCH IN THE FIELD OF FRR

In the Framework of Related Works, we are also adding a section where we will present our progress so far in the field of FRR. We have been researching FRR since 2015, when our first Multicast Repair (M-REP) mechanism was developed [53]. In other works, efforts were devoted to its extension [54] and simulations of other advanced mechanisms for fast network recovery [13], [55].

B-REP FRR is a new Fast ReRoute mechanism designed in cooperation with our Faculty of Management and Informatics, University of Žilina (Slovakia) and Department of Infocommunication Engineering, Kharkiv National University of Radio Electronics (Ukraine) published in the article [55]. B-REP (Bit Repair) is designed to repair all network failures (so-called 100% Repair Coverage). B-REP belongs to the group of FRR mechanisms that calculate backup routes in advance but with low computational complexity. The algorithm is able to calculate backup paths based on common line metrics or completely ignore them and calculate paths without being limited by metrics. This means that if a path exists, the algorithm will find it and use it. The advantage is also the possibility to manually set your own backup path by the network administrator [55].

Unlike several other Fast ReRoute mechanisms, B-REP uses a standardized BIER header. This BIER header shown in Fig. 2 includes one special Bit-String field. This special field consists of bit field, and it is shown in Fig. 3. The Bit-String in the B-REP mechanism is used to define a backup path when routing packets during network outage bypass. B-REP is primarily intended for deployment in communication networks with a link-state routing protocol.

A bit string is a special data structure used to store information about routers in the BIER domain. This string consists of an array of bits, with each bit representing a specific router. Bits are ordered from least significant (LSB) to most significant (MSB), allowing efficient storage and processing of information. The value of the bit string specifies the routers to be used in the event of an error to deliver the packet in the correct direction.

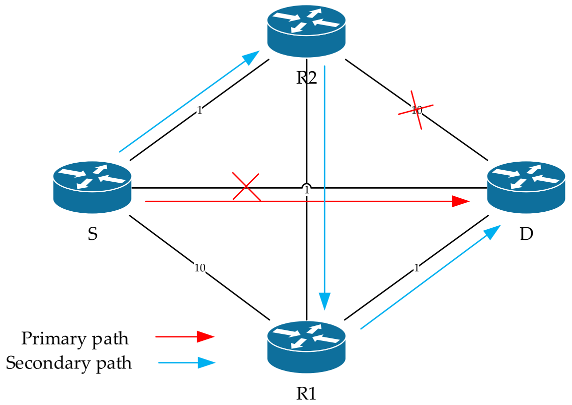


FIGURE 4. The routing problem of B-REP.

While testing the original B-REP mechanism, we discovered one serious problem. In the case of a specific topology (Fig. 4), where router S has as direct neighbors both routers (R1 and R2) that are on an alternative path, it may happen that it chooses the wrong one. When router S sends the packet to R1 (instead of R2), R1 sends the packet to R2, and then R2 will drop the packet because the link to router D is dead.s Bitstring tells the router which routers formed the alternate path but does not say in which order.

Therefore, we focused on the development of a new mechanism that would clearly identify which routers participate in the alternative path and in what order.

B. SUMMARY AND CONCLUSION OF THE ANALYSIS

Fibbing is a new technology for routing in IP networks that focuses on a combination of distributed and centralized approaches.

The goal of Fibbing is to achieve the desired routing paths at the output. Fibbing brings smaller and more efficient memory and CPU usage on routers, even when many fake nodes are introduced into the network. Fibbing takes approximately constant time to generate many records in the network. Ultimately, the crash recovery time seems to depend on the set FTP time.

MPLS-TE is commonly used to optimize data flow in highly interconnected networks where quality of service (QoS) and minimum delay requirements must be met. It allows redirecting flows to optimal paths under current conditions, higher network reliability, and, last but not least, high flexibility. It is a rather complex mechanism that requires experts to manage and maintain the network, and its deployment can also be quite expensive due to the need for devices with MPLS support.

B-REP is capable of interoperating with any link-state routing protocol. Its main task is to find and calculate the shortest

TABLE 1. Comparison of existing FRR solutions from the point of view of critical factors.

	100% repair coverage	Precomputing	Packet modification	Link-state dependency	Computational complexity
ECMP FRR	No	Yes	No	No	Low
Directed LFA	Yes	Yes	Yes	Yes	High
LFA	No	Yes	No	No	Low
MPLS-TE FRR	No	Yes	Yes	No	High
MRC	Yes	Yes	Yes	Yes	High
MRT	Yes	Yes	Yes	Yes	High
Not-Via Addresses	Yes	Yes	Yes	Yes	High
Remote LFA	No	Yes	Yes	Yes	Average
TI-LFA	Yes	Yes	Yes	Yes	High
OPFRR	Yes	Yes	No	Yes	High
PSFRR	Yes	Yes	No	Yes	High
LFRR	Yes	Yes	No	Yes	High
B-REP	Yes	Yes	Yes	Yes	High

alternative path to temporarily bypass a network failure. A big advantage is that it uses a standardized way of marking the backup path using Bit-String for implementation. It provides easy implementation into the existing architecture and defines your own alternative paths. The comparison of recent existing solutions in FRR from the point of view of critical factors is depicted in Table 1.

Analysis of critical factors:

The following requirements are imposed on FRR mechanisms:

- Low computational complexity of the algorithm (precomputing).
- Repair of all possible errors in the network (Repair coverage) approaching 100%.
- Simple integration into the existing topology.
- Protection against line and router failure.
- Fast recovery of communication within 50 ms.
- Success rate approaching 100% of saved packets.
- Support for multicast technology.
- Fast detection of network errors.

These properties can also be identified as the basic problem areas of the Fast ReRoute mechanisms, as each of the current FRR mechanisms meets only some of these properties.

We have identified the following problem areas for the mentioned FRR mechanisms.

1) MODIFICATION OF PACKETS

Quick detection of an outage and subsequent notification to the remaining routers affected by the outage is key to fast network recovery technologies. In some FRR mechanisms, information about a specific line failure is propagated:

- by modifying special bits in the IPv4 header,
- by encapsulating the packet with another header or
- based on the interface through which the packet was received.

2) PRELIMINARY CALCULATIONS

FRR mechanisms work on a principle that is based on fast detection of a link failure with a neighboring router and precomputing alternative paths.

The computational complexity of individual FRR mechanisms increases with the growing number of routers in the network. These calculations must be re-implemented if there is a change in topology in the network and are usually performed on routers as specific low-priority processes when the router's CPU is idle.

Calculations of FRR mechanisms thus take time and system resources of the router.

3) DEPENDENCE ON LINK-STATE ROUTING PROTOCOLS

Another important fact is that several analyzed FRR mechanisms require topological information about the network from the database of link-state routing protocols to calculate the alternative path. This feature limits the application of FRR mechanisms only to networks where a primarily link-state routing protocol is deployed. Currently, most existing FRR mechanisms are dependent on information from link-state routing protocols.

Currently, several companies (e.g. Cisco Systems, Juniper Networks) are working on the implementation of fast network recovery mechanisms into existing operating systems in routers. Mainly, LFA and Remote LFA mechanisms found commercial applications in both companies due to their simplicity.

In general, from the information obtained, it is not possible to say unequivocally which of the mechanisms is the best. Each has its advantages and disadvantages, and it is up to the network architect to decide which one to use.

Our goal is to continue to improve the mechanisms that will ensure faster recovery in the network. Since we found the emergence of unwanted servers that increase the convergence time with the B-REP mechanism, we want to eliminate this unwanted phenomenon with the new Label B-REP mechanism through a clear order of routers in the alternative path.

In general, from the information obtained, it is not possible to say unequivocally which of the mechanisms is the best. Each has its advantages and disadvantages, and it is up to the network architect to decide which one to use.

Our goal is to continue to improve the mechanisms that will ensure faster recovery in the network. Since we found the emergence of unwanted servers with the B-REP mechanism, which prolong the convergence time, we want to eliminate this unwanted phenomenon with the new Label B-REP mechanism through a clear order of routers in the alternative path.

III. METHODOLOGY OF DESIGN OF LABEL B-REP

The section provides an overview of the design methodology. Label B-REP brings new possibilities and advantages for the design of IP networks and their stability. This section describes in more detail the individual parts and components of the B-REP mechanism, including our modifications to the Label B-REP mechanism.

B-REP is based on the principle of unicast delivery of IP packets, described in more detail in the publication [55]. B-REP focuses on protecting the unicast data flow and uses a standardized solution to describe an alternative path using a Bit-String. After a deeper examination of the origin B-REP FRR mechanism, it was found that under certain circumstances, such as the specific network design and the corresponding order of bits in the Bit-String, either a routing loop could occur, or the order of routers would not be precisely defined in the given situation.

Based on the finding that the B-REP and EB-REP mechanisms can cause a measurement loop under certain circumstances, we came to the decision that it would be more appropriate to store the order of routers on the backup path in a precisely determined order in the Bit-String method of the stack data structure. I.e., the next-hop router on the backup path would always be on top of the imaginary stack. "Two new parameters are introduced for the correct operation of the B-REP mechanism" [55].

In the following parts, its analysis and implementation in the OMNeT ++ simulation environment and subsequent simulation on the model topology created by us with the running OSPF routing protocol will be presented.

A. B-REP ROUTER-ID

In order to ensure the correct functioning of the B-REP mechanism in a common network domain, it is essential that each router has a unique identifier (ID). Therefore, the use of a new router identifier (the first mentioned parameter) was proposed, designated as *B-REP router-ID*(B-REP R-ID), which is similar to the BFR-ID used in networks with BIER support. The process of assigning this identifier is not well defined and can be done either manually by the administrator or derived from another existing and already assigned unique router identifier. Once an ID is assigned to a router, no other router within the domain can use it, thus ensuring the uniqueness of the identifiers [55].

B. BIT STRING

Another parameter is a variable of the type "bit-string", which represents a special variable-length field named as Bit-String. It allows the user to define the B-REP Router-ID of routers with B-REP support. Subsequently, the order of the individual B-REP Router-IDs is precisely defined, thus enabling the entire hop-by-hop route backup to be determined. The transport network is assumed to have one of the link-state routing protocols enabled and active, which provides accurate topological information to each router in the area about the other routers in the shared area. Thanks to this criterion, B-REP can calculate and accurately determine the value of the Bit-String that defines the backup path for various failures.

TABLE 2. Comparison of B-REP R-ID.

Router	B-REP R-ID	Bit-String B-REP R-ID (original)	Bit-String B-REP R-ID (modified)
R1	1	00001 (LSB)	00001 (LSB)
R2	2	00010 (LSB)	00010 (LSB)
R3	3	00100 (LSB)	00011 (LSB)

TABLE 3. Table with assigned B-REP R-IDs.

Router	Router (OSPF) ID	B-REP R-ID	Bit-String value
R1	1.1.1.1	1	00001 (LSB)
R2	2.2.2.2	2	00010 (LSB)
R3	3.3.3.3	3	00011 (LSB)
R4	4.4.4.4	4	00100 (LSB)
R5	5.5.5.5	5	00101 (LSB)

Unlike the original B-REP router-ID allocation, where one bit in the Bit-String represented one router, in our new Label B-REP mechanism, we will encode individual routers using multiple bits. This behavior can be understood as a representation of a decimal number (B-REP R-ID) in binary form. The comparison is shown in Table 2.

Dijkstra's algorithm is used to calculate the alternative route, while B-REP requires access to the LSDB database of the deployed link-state routing protocol. The B-REP mechanism can run as a specific, low-priority process during router CPU idle time, thus not burdening it with calculating backup routes at busy times. It records the resulting alternative path in the B-REP BackupTable (B-REP BP table) [55].

C. B-REP COOPERATION WITH OSPF

In this part, the cooperation of the new Labal B-REP mechanism and the above-described components with the OSPF routing protocol deployed in the network is rebuilt.

In the OSPF system, it is necessary that each active router has a unique identification code (router ID), called OSPF Router-ID. This code can be set manually or automatically when assigned based on one of the router's local IP addresses. If it is necessary to ensure uniqueness, it is possible to use the alternative option of assigning B-REP R-ID according to the already determined OSPF R-ID. The process of mapping these identification codes is not strictly prescribed, but different algorithmic methods can be used. Such an example of mapping is shown in Table 3 referring to the values belonging to the topology shown in Fig. 5.

Thanks to the OSPF protocol and its LSDB routing information database, all routers in the common area have the same information. This allows each router with the B-REP mechanism enabled to assign a unique B-REP R-ID to itself and to other routers. Subsequently, after assigning a B-REP R-ID, each router remaps this identifier (or identifiers) to a Bit-String value and stores them in the B-REP table, as shown

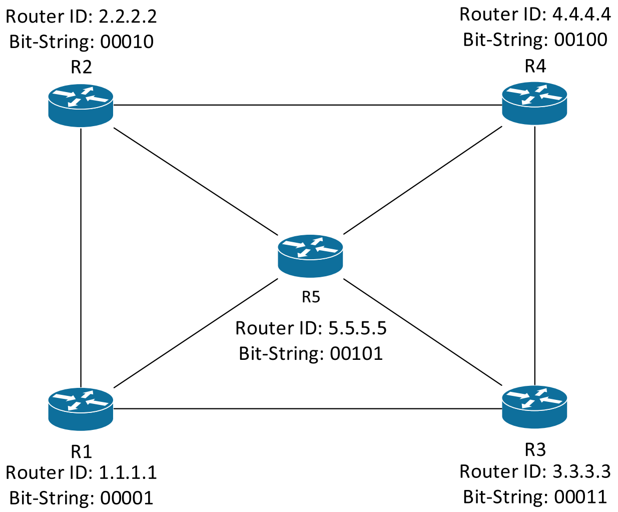


FIGURE 5. Allocation of bit strings by router ID.

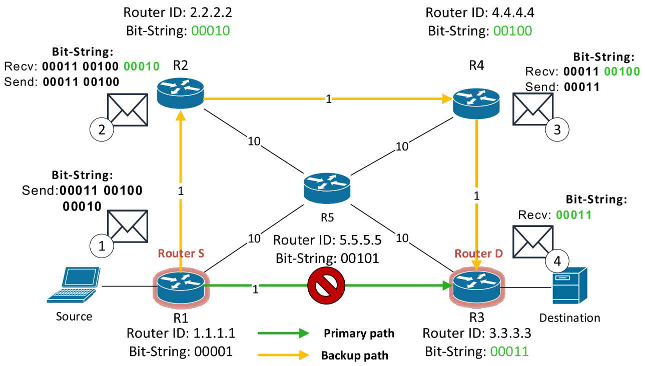


FIGURE 6. Fault detection with B-REP mechanism response.

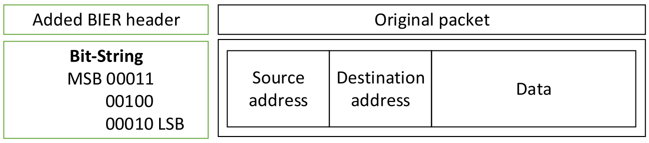


FIGURE 7. B-REP additional header.

in Table 3. As a result, every router with an activated B-REP mechanism should have the same table since it is based on the same information provided by the LSDB database.

The table is compiled in such a way that all OSPF Router-ID records are sorted in ascending order, and a unique identifier is assigned to such sorted records, which is then converted into a Bit-String representation (Table 3). At the moment of detection of an outage, the contents of the B-REP table are frozen so that inconsistent information does not occur during the creation of a backup route. After the convergence of the IGP routing protocol (in our case, OSPF) is completed, the process of rebuilding the B-REP table is started on each router separately.

The concept of most FRR mechanisms is to precalculate backup routes to bypass a failed router or link. Adhering to this concept, the B-REP mechanism, thanks to the pre-calculated backup routes, is able to ensure that the routing mechanism of the router can quickly deploy the backup route with minimal delay and loss. All routers with an active B-REP mechanism calculate based on the protected connection. In FRR terminology, this protected link (or protected line) is referred to as a link to be protected by a backup path in the event of its failure or the failure of a directly connected neighbor. Additionally, the B-REP mechanism uses Dijkstra's algorithm, which computes backup alternative routes to all destinations by setting the protected link metric to infinity.

If the router detects a connection failure with the primary router through which the packet is supposed to reach the given destination, it becomes router S. The target of the alternative path from router S in terms of bypassing the failure is router D, whose role in this example is played by router R3 (Fig. 6) [55].

At the time of failure of the primary path, which is set as a protected link, router S has already prepared a backup hop-by-hop route to the destination router D. The B-REP mechanism is activated on router S. The latter takes control over packet routing and starts encapsulating packets of the original unicast flow with the additional BIER header. It sets a Bit-String field in the BIER header (Fig. 7) with a calculated backup path to which packets will be forwarded up to the destination router D.

If another outage occurs along the backup route, it is assumed that the new router S modifies the Bit-String field of the packet with a new backup route and routes the packets already along it. The biggest problem in this case is indicated by the fact that the new router S may not yet know about the first drop (since OSPF convergence is still in progress) and can route packets back to the location of the first drop. It is important to note that this feature is currently not well-tested and is under investigation [56].

The following example shows the passage of a packet along an alternative path (Fig. 8). As can be seen, the backup path is built through routers **S** →R2→R4→ **D.** Router R1 detected the failure of its primary path and became Router S, which began encapsulating incoming packets with the destination address of Router D with a temporary BIER header. In the header, he set the Bit-String field to the value corresponding to the calculated backup path (00011 00100 00010 LSB), which in words would look like this (D R4 R2 LSB) for a better idea.

When the router participating in the backup route (also referred to as **N_i**) receives a packet with a BIER header, it checks the value of the Bit-String field in it. According to the length of its B-REP R-ID converted to binary form, it takes a specified number of bits from the top of the imaginary stack (Bit-String in the BIER header), compares them with its Bit-String and then removes this item from the Bit-String field in the header BIER. This will ensure that the router that received the packet with the BIER header will not receive it again, thus avoiding the formation of loops.

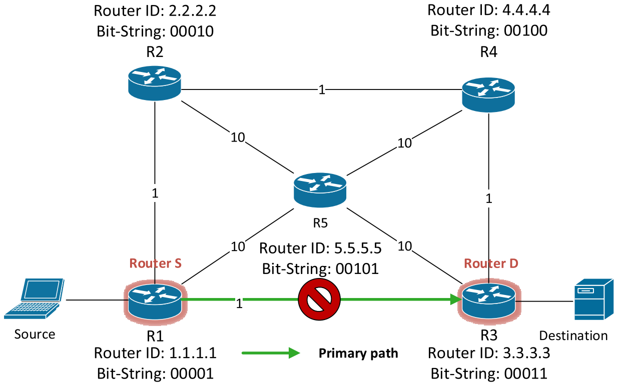


FIGURE 8. Hop-by-hop packet traversal along the backup path.

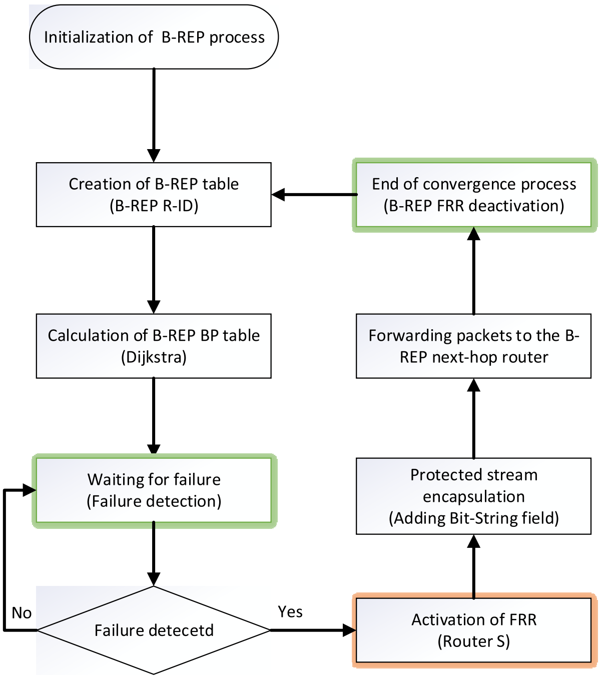

Initialization of B-REP process
Creation of B-REP table
(B-REP R-ID)
End of convergence process
(B-REP FRR deactivation)
Calculation of B-REP BP table
(Dijkstra)
Forwarding packets to the B-REP next-hop router
Waiting for failure
(Failure detection)
Protected stream encapsulation
(Adding Bit-String field)
No
Failure detecetd
Yes
Activation of FRR
(Router S)

FIGURE 9. Flow diagram of activation of the B-REP mechanism during an outage.

In other words, router R2 receives a packet with a Bit-String value (00011 00100 00010 LSB), removes its B-REP R-ID (00010), and sends it to the next router in the sequence (R4) with the following Bit-String value in the BIER header (00011 00100 LSB). However, before sending it to the next-hop router, it checks whether such a next-hop router is directly connected at all. Other routers on the backup path repeat this procedure until the Bit-String is empty. After removing its B-REP R-ID, the last router D, also removes the B-REP header and further leaves the routing of the packet to the routing protocol.

For a better understanding, the functioning of the B-REP mechanism can also be illustrated with the help of flowcharts. The first diagram (Fig. 9) shows the activation process of the B-REP mechanism and its subsequent behavior in the event of an error. The second diagram (Fig. 10) shows how packets

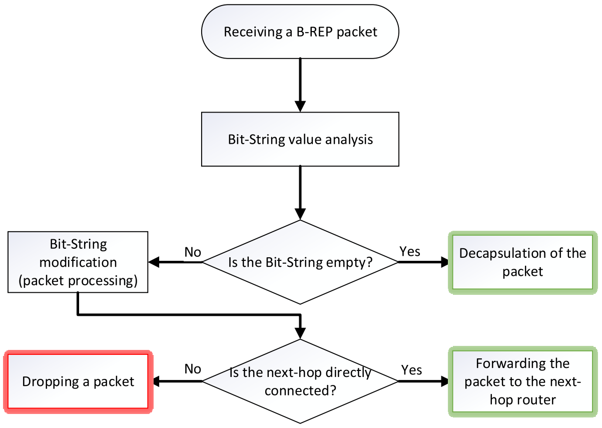

Receiving a B-REP packet
Bit-String value analysis
Bit-String modification (packet processing)
No
Is the Bit-String empty?
Yes
Decapsulation of the packet
Dropping a packet
No
Is the next-hop directly connected?
Yes
Forwarding the packet to the next-hop router

FIGURE 10. Flow diagram - processing of a packet with a BIER header.

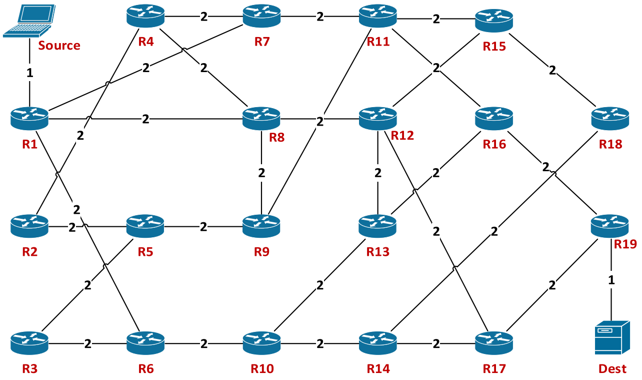


FIGURE 11. Topology with a line rating.

with a BIER header are processed on each router with the B-REP mechanism deployed.

IV. IMPLEMENTATION OF MODIFIED LABEL B-REP IN OMNET++

The modified LABEL B-REP was implemented in the OMNET++ simulation environment.

For simulation purposes, a test topology was created on which the mechanism we proposed was tested.

The simulations themselves are based on the dynamic routing link-state protocol OSPF, which discovers new neighbors through Hello messages and maintains a connection with them. All OSPF routers in the network exchange complete topological information and store it in a Link State Database (LSDB). Thanks to this, the routers know the entire topology of the network, but for proper functioning, it is necessary that all routers in the network have the same database, in which the information is updated dynamically. OSPF uses Dijkstra's algorithm to calculate the shortest paths.

An example of the proposed test topology can be seen in Fig. 11. The most beautiful path calculated by the OSPF protocol from the source to the destination is Source -> R1 -> R7 -> R11 -> R16 -> R19 -> Dest.

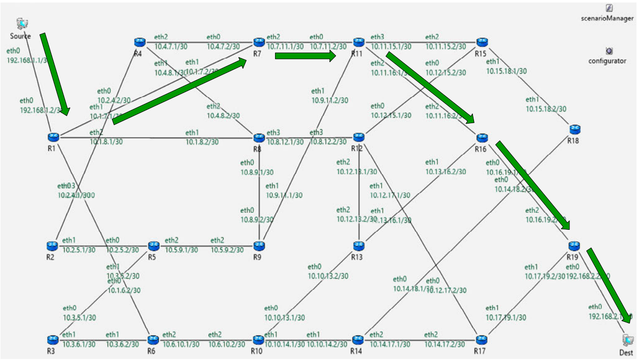


FIGURE 12. Test topology with primary path and addressing.

TABLE 4. User packet path through the network - OSPF.

	Sim. time [s]	Source/Destination	Transmission type
1	40.00001172	Source → R1	UDPBasicAppData-0
2	40.0000381	R1 → R7	UDPBasicAppData-0
3	40.00006448	R7 → R11	UDPBasicAppData-0
4	40.00009086	R11 → R16	UDPBasicAppData-0
5	40.00011724	R16 → R19	UDPBasicAppData-0
6	40.00014362	R19 → Dest	UDPBasicAppData-0

V. SIMULATIONS IN OMNET++

The section will present the process of testing and verifying our implemented Label B-REP mechanism in the OMNeT ++ environment. The simulations were performed in a test topology named *dense_topology*. When designing the topology, we took into account the density of links between routers, so that there are still alternative routes even in the event of various line failures. The generated user traffic was from the computer named *Source*(192.168.1.1) to the destination computer *Dest*(192.168.2.1).

For clarity within the topology, we implemented the IP addressing between the routers in the following form 10. *first_router.second_router*.x. It can be seen in Fig. 12 in the topology during the running simulation. Also shown here is the primary path in a stable situation in the network used by the OSPF protocol from the *source* of user traffic *to the destination*. The primary path passes through the routers R1→R7→R11→R16→R19, which is also confirmed by Table 4, showing the jumps of user data flow packets in the simulation.

The Source Table 4 shows exactly what time the first data packet passes through the network.

A. SIMULATION SCENARIO NO. 1 – LINE FAILURE

With the first scenario, we will test the resistance and reaction of the mechanism to a line failure in the network. We expect an outage in the simulation time of 60 seconds when we consider the network fully functional and all routers have prepared alternative paths.

In this scenario, we simulate a link failure on the primary path between routers R11 and R16 (Fig. 13). We expect an immediate response of the R11 router to the outage, which leads to the encapsulation of the packet with the IPFRR header together with the backup route in Bit-String format from the B-REP BackupTable. Thus, at this time, the B-REP mechanism starts to work. Based on the metrics, the second shortest path is selected as the alternative path (shown in orange color in Fig. 13). Router R11 forwards the encapsulated packet to the nearest next-hop neighbor on the backup path. The routers on the backup path (R15, R12, R17, R19) successively process the IPFRR packet and, from the Bit-String field, determine the next-hop neighbor to which they forward the packet. When the IPFRR packet is processed by the last router on the backup route (R19), it removes the last entry in the Bit-String from it. Subsequently, the packet becomes de-encapsulated and is further routed to the destination by the standard routing protocol (*Dest*).

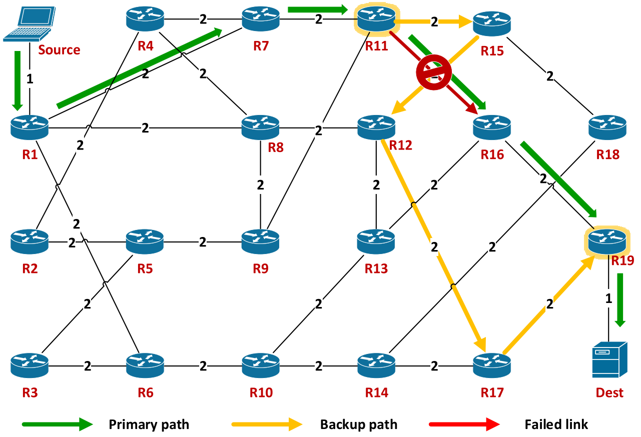
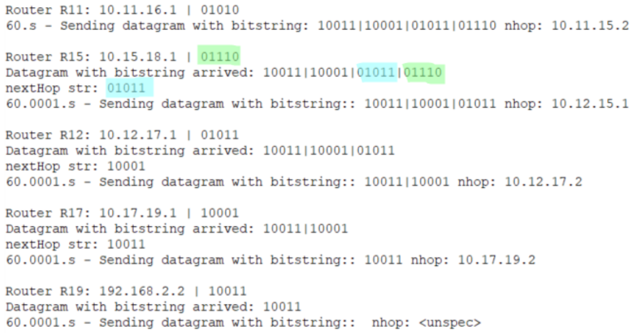


FIGURE 13. Scenario 1 - line failure.



```
Router R11: 10.11.16.1 | 01010
60.s - Sending datagram with bitstring: 10011|10001|01011|01110 nhop: 10.11.15.2

Router R15: 10.15.18.1 | 01110
Datagram with bitstring arrived: 10011|10001|01011|01110
nextHop str: 01011
60.0001.s - Sending datagram with bitstring:: 10011|10001|01011 nhop: 10.12.15.1

Router R12: 10.12.17.1 | 01011
Datagram with bitstring arrived: 10011|10001|01011
nextHop str: 10001
60.0001.s - Sending datagram with bitstring:: 10011|10001 nhop: 10.12.17.2

Router R17: 10.17.19.1 | 10001
Datagram with bitstring arrived: 10011|10001
nextHop str: 10011
60.0001.s - Sending datagram with bitstring:: 10011 nhop: 10.17.19.2

Router R19: 192.168.2.2 | 10011
Datagram with bitstring arrived: 10011
60.0001.s - Sending datagram with bitstring::  nhop: <unspec>
```

FIGURE 14. Console statements generated by the B-REP mechanism - scenario 1.

In the simulation, the B-REP mechanism is disabled when the source router (R11) detects that the protected interface is in the UP state again.

The expected behavior described above can be verified in more detail with the implemented console statements. We also actively used them during the actual mechanism testing during development, shown in Fig. 14.

It is clear from the console listing that router R11 with OSPF Router-ID *10.11.16.1* and B-REP Router-ID in Bit-String format *01010* detected the failure of its directly

TABLE 5. **Path of a user packet through the network - B-REP - scenario 1.**

	Sim. time [s]	Source/Destination	Transmission type
1	60	Source → R1	UDPBasicAppData-20
2	60.00001466	R1 → R7	UDPBasicAppData-20
3	60.00002932	R7 → R11	UDPBasicAppData-20
4	60.00004398	R11 → R15	arpREQ
5	60.00004984	R15 → R11	arpREPLY
6	60.00005570	R11 → R15	UDPBasicAppData-20
7	60.00007036	R15 → R12	arpREQ
8	60.00007622	R12 → R15	arpREPLY
9	60.00008208	R15 → R12	UDPBasicAppData-20
10	60.00009674	R12 → R17	arpREQ
11	60.00010260	R17 → R12	arpREPLY
12	60.00010846	R12 → R17	UDPBasicAppData-20
13	60.00012312	R17 → R19	arpREQ
14	60.00012898	R19 → R17	arpREPLY
15	60.00013484	R17 → R19	UDPBasicAppData-20
16	60.00016122	R19 → Dest	UDPBasicAppData-20

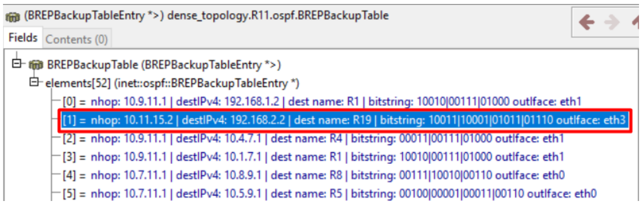
connected link on the primary path and activated the B-REP mechanism. From the BREP BackupTable, he extracted a prepared entry with an alternative path also in Bit-String format, encapsulated a packet with a backup path, and sent it to the next-hop router with IP *10.11.15.2*.

This next-hop router (R15) received the IPFRR packet along with the Bit-String of the backup route as we can see in Fig. 14. First, he removed his BREP Router-ID from the desired Bit-String to avoid loops (highlighted in green in Fig. 14). From the packet, he read the B-REP Router-ID of the next next-hop router on the backup path (currently the last item in the Bit-String - highlighted in turquoise in Fig. 14), looked up this entry in his B-REP table and found out the IP address of the next- router hop on this backup path. It also discovered its output interface to which the designated next-hop router is directly connected. Finally, it forwarded the packet through the detected output interface with the modified Bit-String.

The described process is performed by all other routers on the backup route until the packet reaches the last router (R19). After removing his B-REP Router-ID, he discovers that the Bit-String has remained empty and unencapsulates the packet from the IPFRR header. The packet is further routed through the network using the standard routing protocol (OSPF). Since in our case, the destination station is directly connected to the R19 router, we see *the <unspect> tag as the next-hop.*

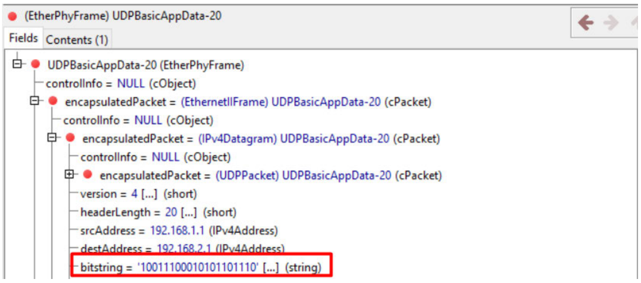
The behavior described above can also be observed within the simulation from the graphical packet forwarding table in the OMNeT++ console. The part of this table where the packet goes through the network via the backup path is shown in Table 5.

In Fig. 15 is an entry from the BREP BackupTable that router R11 used to bypass the intruded local fault. Fig. 16 shows the encapsulated packet sent by this router along with the backup path in Bit-String format.



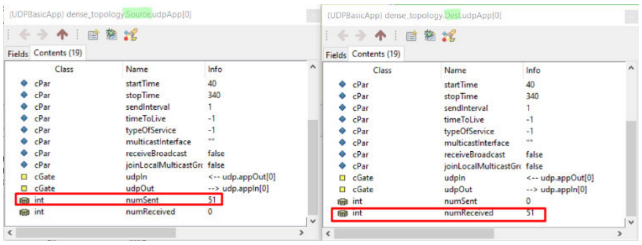
```
(BREPBackupTableEntry *>) dense_topology.R11.ospf.BREPBackupTable
Fields  Contents (0)
⊟ BREPBackupTable (BREPBackupTableEntry *>)
  ⊟ elements[52] (inet::ospf::BREPBackupTableEntry *)
     [0] = nhop: 10.9.11.1 | destIPv4: 192.168.1.2 | dest name: R1 | bitstring: 10010|00111|01000 outIface: eth1
     [1] = nhop: 10.11.15.2 | destIPv4: 192.168.2.2 | dest name: R19 | bitstring: 10011|10001|01011|01110 outIface: eth3
     [2] = nhop: 10.9.11.1 | destIPv4: 10.4.7.1 | dest name: R4 | bitstring: 00011|00111|01000 outIface: eth1
     [3] = nhop: 10.9.11.1 | destIPv4: 10.1.7.1 | dest name: R1 | bitstring: 10010|00111|01000 outIface: eth1
     [4] = nhop: 10.7.11.1 | destIPv4: 10.8.9.1 | dest name: R8 | bitstring: 00111|10010|00110 outIface: eth0
     [5] = nhop: 10.7.11.1 | destIPv4: 10.5.9.1 | dest name: R5 | bitstring: 00100|00001|00011|00110 outIface: eth0
```

FIGURE 15. **BREP BP table entry used to bypass the fault.**



```
● (EtherPhyFrame) UDPBasicAppData-20
Fields  Contents (1)
⊟ ● UDPBasicAppData-20 (EtherPhyFrame)
   controlInfo = NULL (cObject)
   ⊟ ● encapsulatedPacket = (EthernetIIFrame) UDPBasicAppData-20 (cPacket)
      controlInfo = NULL (cObject)
      ⊟ ● encapsulatedPacket = (IPv4Datagram) UDPBasicAppData-20 (cPacket)
         controlInfo = NULL (cObject)
         ⊞ ● encapsulatedPacket = (UDPPacket) UDPBasicAppData-20 (cPacket)
         version = 4 [...] (short)
         headerLength = 20 [...] (short)
         srcAddress = 192.168.1.1 (IPv4Address)
         destAddress = 192.168.2.1 (IPv4Address)
         bitstring = '10011100010101101110' [...] (string)
```

FIGURE 16. **Sent IPFRR packet from router R11 to R15.**



(UDPBasicApp) dense_topology.Source.udpApp[0]

Fields Contents (19)

Class	Name	Info
cPar	startTime	40
cPar	stopTime	340
cPar	sendInterval	1
cPar	timeToLive	-1
cPar	typeOfService	-1
cPar	multicastInterface	""
cPar	receiveBroadcast	false
cPar	joinLocalMulticastGr	false
cGate	udpIn	<-- udp.appOut[0]
cGate	udpOut	--> udp.appIn[0]
int	numSent	51
int	numReceived	0

(UDPBasicApp) dense_topology.Dest.udpApp[0]

Fields Contents (19)

Class	Name	Info
cPar	startTime	40
cPar	stopTime	340
cPar	sendInterval	1
cPar	timeToLive	-1
cPar	typeOfService	-1
cPar	multicastInterface	""
cPar	receiveBroadcast	false
cPar	joinLocalMulticastGr	false
cGate	udpIn	<-- udp.appOut[0]
cGate	udpOut	--> udp.appIn[0]
int	numSent	0
int	numReceived	51

FIGURE 17. **Total number of data packets sent and received.**

In a simulation time of about 75 seconds, the convergence of the OSPF routing protocol was completed, and both OSPF and B-REP mechanism recalculations took place.

At the end of the testing, we add the number **of sent** packets from the source of data traffic (*Source*) and the number of **received** packets to the destination station *Dest*. We made this recording in a simulation time of 90 seconds, i.e., 30 seconds after the line failure. In Fig. 17, it can be seen that we have sent a total of 51 packets, and on the other hand, we have requested the same number (51) of packets. By implementing the B-REP mechanism, we ensured that there was no loss during the line outage.

B. SIMULATION SCENARIO NO. 2 – ROUTER FAILURE

The second scenario tested the mechanism's resistance and reaction to a router outage in the network. We expect an outage in the simulation time of 60 seconds when we consider the network to be fully functional and all routers have prepared alternative paths.

In this scenario, we simulate the outage of router R11 on the primary path (Fig. 18). We expect the R7 router to react immediately to the outage, which leads to the encapsulation of the packet with the IPFRR header together with the backup route in Bit-String format from the B-REP BackupTable.

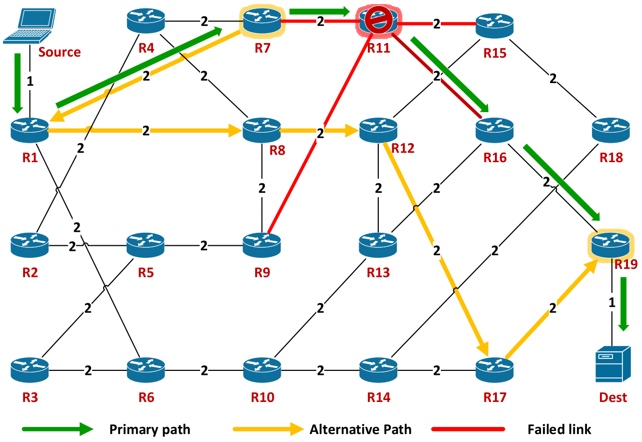
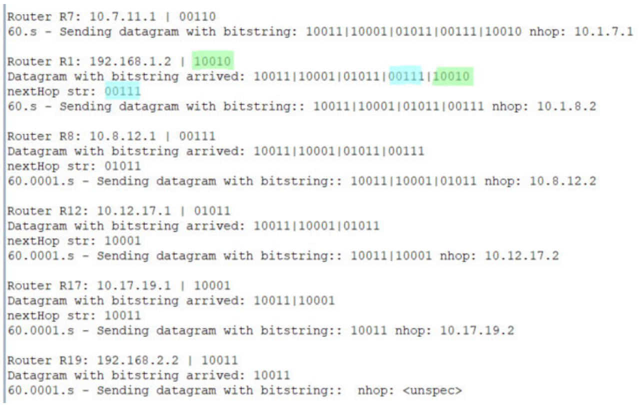


FIGURE 18. Scenario 2 - router outage.



```
Router R7: 10.7.11.1 | 00110
60.s - Sending datagram with bitstring: 10011|10001|01011|00111|10010 nhop: 10.1.7.1

Router R1: 192.168.1.2 | 10010
Datagram with bitstring arrived: 10011|10001|01011|00111|10010
nextHop str: 00111
60.s - Sending datagram with bitstring:: 10011|10001|01011|00111 nhop: 10.1.8.2

Router R8: 10.8.12.1 | 00111
Datagram with bitstring arrived: 10011|10001|01011|00111
nextHop str: 01011
60.0001.s - Sending datagram with bitstring:: 10011|10001|01011 nhop: 10.8.12.2

Router R12: 10.12.17.1 | 01011
Datagram with bitstring arrived: 10011|10001|01011
nextHop str: 10001
60.0001.s - Sending datagram with bitstring:: 10011|10001 nhop: 10.12.17.2

Router R17: 10.17.19.1 | 10001
Datagram with bitstring arrived: 10011|10001
nextHop str: 10011
60.0001.s - Sending datagram with bitstring:: 10011 nhop: 10.17.19.2

Router R19: 192.168.2.2 | 10011
Datagram with bitstring arrived: 10011
60.0001.s - Sending datagram with bitstring::  nhop: <unspec>
```

FIGURE 19. Console statements generated by the B-REP mechanism - scenario 2.

Based on the metrics, the second shortest path is selected as the alternative path (shown in orange color in Fig. 18). Router R7 forwards the encapsulated packet to the nearest next-hop neighbor on the backup path. The routers on the backup path (R1, R8, R12, R17, R19) successively process the IPFRR packet and, from the Bit-String field, determine the next-hop neighbor to which they forward the packet. When the IPFRR packet is processed by the last router on the backup route (R19), it removes the last entry in the Bit-String from it. Subsequently, the packet becomes de-encapsulated and is further routed to the destination by the standard routing protocol (*Dest*).

The expected behavior described above can be verified in more detail by the implemented console statements (Fig. 19). Here, it can be seen that router R7 with OSPF Router-ID 10.7.11.1 and B-REP Router-ID in Bit-String format 00110 has detected the failure of its directly connected line on the primary path and activated the B-REP mechanism. From the BREP BackupTable, he extracted a prepared entry with an alternative path also in Bit-String format, encapsulated a packet with a backup path, and sent it to the next-hop router with IP 10.1.7.1.

TABLE 6. Path of a user packet through the network - B-REP - scenario 2.

	Sim. time [s]	Source/Destination	Transmission type
1	60	Source → R1	UDPBasicAppData-20
2	60.00001466	R1 → R7	UDPBasicAppData-20
3	60.00002932	R7 → R1	UDPBasicAppData-20
4	60.00004398	R1 --> R8	arpREQ
5	60.00004984	R8 --> R1	arpREPLY
6	60.00005570	R1 --> R8	UDPBasicAppData-20
7	60.00007036	R8 --> R12	arpREQ
8	60.00007622	R12 --> R8	arpREPLY
9	60.00008208	R8 --> R12	UDPBasicAppData-20
10	60.00009674	R12 --> R17	arpREQ
11	60.00010260	R17 --> R12	arpREPLY
12	60.00010846	R12 --> R17	UDPBasicAppData-20
13	60.00012312	R17 --> R19	arpREQ
14	60.00012898	R19 --> R17	arpREPLY
15	60.00013484	R17 --> R19	UDPBasicAppData-20
16	60.00016122	R19 --> Dest	UDPBasicAppData-20

This next-hop router (R1) received the IPFRR packet along with the Bit-String of the backup route as seen in Fig. 19. First, he removed his B-REP Router-ID from the desired Bit-String to avoid loops (highlighted in green in Fig. 19). From the packet, he read the B-REP Router-ID of the next next-hop router on the backup path (currently the last item in the Bit-String - highlighted in turquoise in Fig. 19), looked up this entry in his B-REP table and found out the IP address of the next- router hop on this backup path. It also discovered its output interface to which the designated next-hop router is directly connected. Finally, it forwarded the packet through the detected output interface with the modified Bit-String.

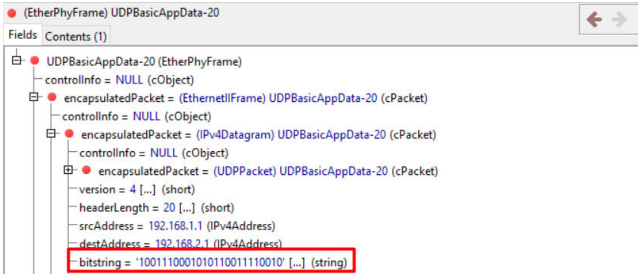
The described process is performed by all other routers on the backup route until the packet reaches the last router (R19). After removing his B-REP Router-ID, he discovers that the Bit-String has remained empty and unencapsulates the packet from the IPFRR header. The packet is further routed through the network using the standard routing protocol (OSPF).

The behavior described above can also be observed within the simulation from the graphical packet forwarding table in the OMNeT++ console. The part of this table where the packet goes through the network via the backup path is shown in Table 6.

Fig. 20 is the entry from the BREP BackupTable that router R7 used to bypass the intruded local fault. Fig. 21 shows the encapsulated packet sent by this router along with the backup path in Bit-String format.

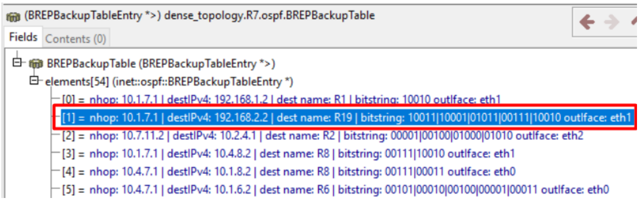
In a simulation time of about 75 seconds, the convergence of the OSPF routing protocol was completed, and both OSPF and B-REP mechanism recalculations took place. From this time, router R1 found a new route in the converged routing table, and the packet from Source *was* already routed outside router R7 (Table 7).

At the end of the testing, we also checked the number of sent and received packets on the *Source* and *Dest* devices in



```
● (EtherPhyFrame) UDPBasicAppData-20
Fields  Contents (1)
⊟ ● UDPBasicAppData-20 (EtherPhyFrame)
  ├ controlInfo = NULL (cObject)
  ⊟ ● encapsulatedPacket = (EthernetIIFrame) UDPBasicAppData-20 (cPacket)
    ├ controlInfo = NULL (cObject)
    ⊟ ● encapsulatedPacket = (IPv4Datagram) UDPBasicAppData-20 (cPacket)
      ├ controlInfo = NULL (cObject)
      ⊞ ● encapsulatedPacket = (UDPPacket) UDPBasicAppData-20 (cPacket)
      ├ version = 4 [...] (short)
      ├ headerLength = 20 [...] (short)
      ├ srcAddress = 192.168.1.1 (IPv4Address)
      ├ destAddress = 192.168.2.1 (IPv4Address)
      ├ bitstring = '10011100010101100111110010' [...] (string)
```

FIGURE 20. BREP BP table entry used for fault bypass - scenario 2.



```
(BREPBackupTableEntry *>) dense_topology.R7.ospf.BREPBackupTable
Fields  Contents (0)
⊟ BREPBackupTable (BREPBackupTableEntry *>)
  ⊟ elements[54] (inet::ospf::BREPBackupTableEntry *)
    ├ [0] = nhop: 10.1.7.1 | destIPv4: 192.168.1.2 | dest name: R1 | bitstring: 10010 outIface: eth1
    ├ [1] = nhop: 10.1.7.1 | destIPv4: 192.168.2.2 | dest name: R19 | bitstring: 10011|10001|01011|00111|10010 outIface: eth1
    ├ [2] = nhop: 10.7.11.2 | destIPv4: 10.2.4.1 | dest name: R2 | bitstring: 00001|00100|01000|01010 outIface: eth2
    ├ [3] = nhop: 10.1.7.1 | destIPv4: 10.4.8.2 | dest name: R8 | bitstring: 00111|10010 outIface: eth1
    ├ [4] = nhop: 10.4.7.1 | destIPv4: 10.1.8.2 | dest name: R8 | bitstring: 00111|00011 outIface: eth0
    ├ [5] = nhop: 10.4.7.1 | destIPv4: 10.1.6.2 | dest name: R6 | bitstring: 00101|00010|00100|00001|00011 outIface: eth0
```

FIGURE 21. Sent IPFRR packet from router R7 to R1.

TABLE 7. New path after OSPF convergence - scenario 2.

	Sim. time [s]	Source/Destination	Transmission type
1	76.00001172	Source → R1	UDPBasicAppData-0
2	76.00002638	R1 → R8	UDPBasicAppData-0
3	76.00004104	R8 → R12	UDPBasicAppData-0
4	76.00005570	R12 → R17	UDPBasicAppData-0
5	76.00007036	R17 → R19	UDPBasicAppData-0
6	76.00008502	R19 → Dest	UDPBasicAppData-0

the simulation time of 90 seconds, and the result is the same as in Fig. 17, so there was no loss of packets.

C. SIMULATION SCENARIO NO. 3 – MULTIPLE OUTAGES

In the third scenario, the resistance and response of the mechanism to the outage of the router in the network and the subsequent outage of the line on the backup path were tested. We expect the beginning of testing in a simulation time of 60 seconds when we consider the network to be fully functional and all routers have prepared alternative paths. The second outage is expected at a simulation time of 65 seconds on the alternate path at the time of network convergence after the first outage.

In this scenario, we simulate an outage of router R11 on the primary path (Fig. 22 Failure A) for 60 seconds. The second outage starts in the simulation time of 65 seconds when the line on the backup route between routers R12 and R17 goes down (Fig. 22 Failure B).

We expect the R7 router to react immediately to the outage, which leads to the encapsulation of the packet with the IPFRR header together with the backup route in Bit-String format from the B-REP BackupTable. Based on the metrics, the second shortest path is selected as the alternative path (shown in orange color in Fig. 22). Router R7 forwards the encapsulated packet to the nearest next-hop neighbor on the backup path. The routers on the backup path (R1, R8, R12, R17, R19) successively process the IPFRR packet and, from the Bit-String field, determine the next-hop neighbor to which they forward the packet. When the IPFRR packet is processed by the last router on the backup route (R19), it removes the last entry in the Bit-String from it. Subsequently, the packet becomes de-encapsulated and is further routed to the destination by the standard routing protocol (*Dest*). The routing procedure is, therefore, the same as we described above for scenario 2 (*Simulation scenario no. 2 – Router failure)*.

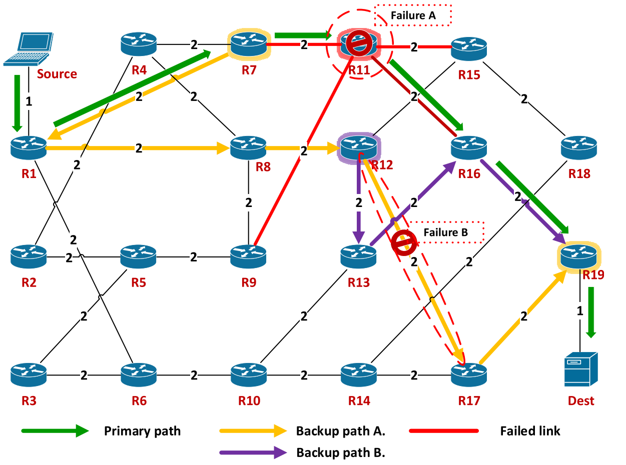


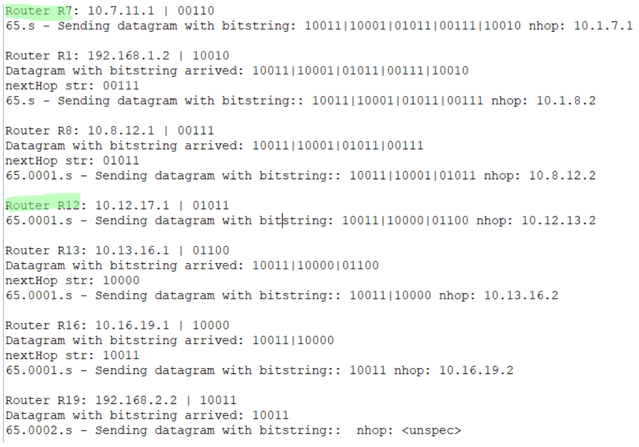
FIGURE 22. Scenario 3 - multiple outages.

However, in the simulation time of 65 seconds, a line failure occurs directly on the backup path (Fig. 22 Failure B). Here, an immediate response is expected from the router R12, which is part of the backup path and becomes the new router S. Its task is to search its B-REP BackupTable for a new backup path with the corresponding Bit-String. Based on the metrics, the second shortest path from the point of view of router R12 to destination R19 (shown in purple color in Fig. 22) is selected as an alternative path. Router R12 forwards the encapsulated packet to the nearest next-hop neighbor on the backup path. The routers on the backup path (R13, R16, R19) successively process the IPFRR packet and, from the Bit-String field, determine the next-hop neighbor to which they forward the packet. The entire process of processing the IPFRR packet by these routers is the same as we have already described, so we will not describe it again.

We can check the above process of expected behavior in more detail with the implemented console statements (Fig. 23).

The behavior from the console statements can also be observed within the simulation from the graphical packet forwarding table in the OMNeT++ console. The part of this table where the packet goes through the network via the backup path is shown in Table 8.

In the simulation time of 76 seconds, the network converged from the first network outage, and packets started to be routed by router R1 directly to R8, which became the primary



```
Router R7: 10.7.11.1 | 00110
65.s - Sending datagram with bitstring: 10011|10001|01011|00111|10010 nhop: 10.1.7.1

Router R1: 192.168.1.2 | 10010
Datagram with bitstring arrived: 10011|10001|01011|00111|10010
nextHop str: 00111
65.s - Sending datagram with bitstring:: 10011|10001|01011|00111 nhop: 10.1.8.2

Router R8: 10.8.12.1 | 00111
Datagram with bitstring arrived: 10011|10001|01011|00111
nextHop str: 01011
65.0001.s - Sending datagram with bitstring:: 10011|10001|01011 nhop: 10.8.12.2

Router R12: 10.12.17.1 | 01011
65.0001.s - Sending datagram with bitstring: 10011|10000|01100 nhop: 10.12.13.2

Router R13: 10.13.16.1 | 01100
Datagram with bitstring arrived: 10011|10000|01100
nextHop str: 10000
65.0001.s - Sending datagram with bitstring:: 10011|10000 nhop: 10.13.16.2

Router R16: 10.16.19.1 | 10000
Datagram with bitstring arrived: 10011|10000
nextHop str: 10011
65.0001.s - Sending datagram with bitstring:: 10011 nhop: 10.16.19.2

Router R19: 192.168.2.2 | 10011
Datagram with bitstring arrived: 10011
65.0002.s - Sending datagram with bitstring::  nhop: <unspec>
```

FIGURE 23. Console statements generated by the B-REP mechanism - scenario 3.

TABLE 8. Path of a user packet through the network - B-REP - scenario 3.

	Sim. time [s]	Source/Destination	Transmission type
1	65	Source → R1	UDPBasicAppData-20
2	65.00001466	R1 → R7	UDPBasicAppData-20
3	65.00002932	R7 → R1	UDPBasicAppData-20
4	65.00004398	R1 --> R8	UDPBasicAppData-20
5	65.00005864	R8 --> R12	UDPBasicAppData-20
6	65.00007330	R12 --> R13	arpREQ
7	65.00007916	R13 --> R12	arpREPLY
8	65.00008502	R12 --> R13	UDPBasicAppData-20
9	65.00009968	R13 --> R16	arpREQ
10	65.00010554	R16 --> R13	arpREPLY
11	65.00011140	R13 --> R16	UDPBasicAppData-20
12	65.00012606	R16 --> R19	arpREQ
13	65.00013192	R19 --> R16	arpREPLY
14	65.00013778	R16 --> R19	UDPBasicAppData-20
15	65.00015244	R19 --> Dest	UDPBasicAppData-20

TABLE 9. New path after OSPF convergence - scenario 3.

	Sim. time [s]	Source/Destination	Transmission type
1	81	Source → R1	UDPBasicAppData-0
2	81.00001466	R1 → R8	UDPBasicAppData-0
3	81.00002932	R8 → R12	UDPBasicAppData-0
4	81.00004398	R12 → R13	UDPBasicAppData-0
5	81.00005864	R13 → R16	UDPBasicAppData-0
6	81.00007330	R16 → R19	UDPBasicAppData-0
7	81.00008796	R19 → Dest	UDPBasicAppData-0

router. However, the B-REP mechanism was still running on R12, and OSPF convergence was running in the background. It was only in the simulation time of 81 seconds that the total recovery of the primary path was completed, and OSPF recovered from the second network outage. At 81 simulation seconds, user packets were fully routed by the OSPF routing protocol, as seen in Table 9.

At the end of the testing, we also checked the number of sent and received packets on the *Source* and *Dest* devices in the simulation time of 90 seconds, and the result is the same as in Fig. 17, so there was no loss of packets.

TABLE 10. Comparison of times.

The scenario	Event	Average packet transfer time [µs]	Packet loss [%]
No. 0	Without downtime	176.38	0
No. 1	Line outage	199.67	0
No. 2	Router outage	195.89	0
No. 3	Multiple outages	215.09	0

D. EVALUATION OF SIMULATIONS

In the simulations described above in the environment of the OMNeT++ simulation tool, we successfully managed to test the new Label B-REP mechanism. The simulations took place in three scenarios successively, from scenario 1 with a single link failure to scenario 2 with an entire router outage and scenario 3 with multiple outages. We tried to design all types of scenarios in such a way that they model a situation that could also occur in a real network. For each scenario, we first recorded the flow of packets routed along the primary path of the converged network and then simulated some kind of outage.

Testing has shown that the Label B-REP mechanism we implemented can behave effectively in various situations as expected. It should be noted that the existing FRR mechanisms are generally tested only in case of line or node failure. *Scenario 1* and *scenario 2* were focused on these two situations, from the results of which we can state the problem-free functionality of our implementation of the Label B-REP mechanism.

From the results of *scenario 3*, we conclude that the deployment of the Label B-REP mechanism in its current form is suitable for multiple outages in the network. The behavior of the mechanism in such a situation is very dependent on the problem in the network.

The Label B-REP mechanism calculates alternative paths ahead, similar to most FRR mechanisms today. For this reason, after the detection of an outage, the deployment of a backup path takes a minimum of time. Table 10 also confirms the average time delay of packet transmission over the network from the Source *to* the *Dest.*

It should be taken into account that the packets were sent in the simulation on idle devices and lines with a set delay of 10 μs. The time data are only informative and should show that the deployed B-REP mechanism caused almost no time delay in the transmission of packets. The times differ only minimally, and that is mainly due to the number of hops through which the packets had to go.

VI. DISCUSSION

Many efforts have been devoted to the development of various mechanisms in recent years, as they seem to be difficult in modern networks with sensitive traffic. Only a small

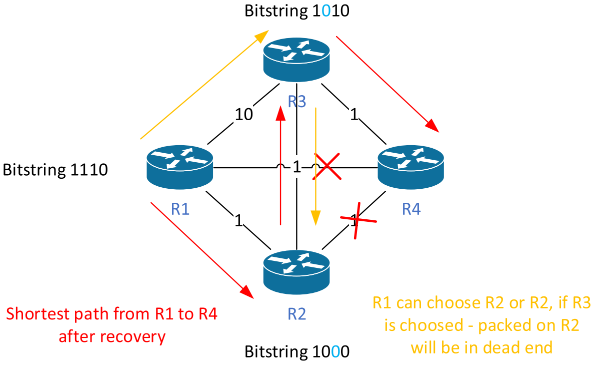


FIGURE 24. Example of the micro loop of B-REP.

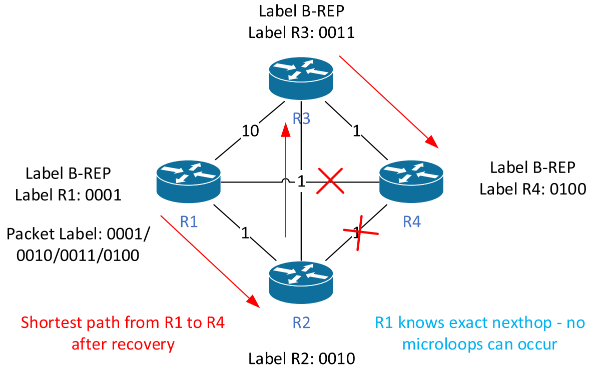


FIGURE 25. Example of solving micro loop with Label B-REP.

percentage of them are deployed by commercial companies in their operating systems. A large part of them, even good designs, are only tested in simulation tools and await either their success or possible improvement, but they represent a basic building block for the creation and research of new mechanisms.

Some provide a very high percentage of "repair coverage" (MRC, MRT mechanisms) but at the price of high demands on both hardware and software, which means that their implementation in existing networks is difficult.

Many of the developed mechanisms are computationally intensive because the backup routes are pre-calculated from the routing information of the IGP protocols they work with, and thus, it can be concluded that they are simply dependent on them.

The important part that this work dealt with was the verification and testing of the implemented FRR mechanism. It was necessary to design a suitable test topology of the network and then perform several tests on it in the form of simulated scenarios. During these tests, we monitored the behavior of the Label B-REP mechanism as well as the overall behavior of the network during its convergence after an outage. It was verified that with the implemented Label B-REP mechanism, no packets were lost during the outage.

Fig. 24 shows a specific topology where a microloop might occur. Router R1 can choose R2 or R3 as a next-hop. If router R1 is chosen as a primary next-hop, R3 instead of R2 microloop will occur. Fig. 25 shows examples of how the new Label B-Rep mechanism removed unwanted loops that were created by the original B-Rep mechanism and thus accelerated the convergence time of the network.

In conclusion, based on the performed tests, we can state that the mechanism succeeded both in the case of a line failure and also of the entire router in the network. According to the result, Label B-REP is faster by approximately 11% in comparison to B-REP.

The simulations were carried out in the OMNET ++ simulation environment. First, a test topology was created, and three different test scenarios were tested on it, where a link failure, a router failure, and a multiple outage were simulated.

The simulations took place in three scenarios successively, from scenario 1 with a single link failure to scenario 2 with an entire router outage and scenario 3 with multiple outages. Testing has shown that the implemented Label B-REP mechanism can behave effectively in various situations as expected.

TABLE 11. Comparison of existing FRR solutions to Label B-REP.

	100% repair coverage	Pre-computing	Packet modification	Link-state dependency	Computational complexity
ECMP FRR	No	Yes	No	No	Low
Directed LFA	Yes	Yes	Yes	Yes	High
LFA	No	Yes	No	No	Low
MPLS-TE FRR	No	Yes	Yes	No	High
MRC	Yes	Yes	Yes	Yes	High
MRT	Yes	Yes	Yes	Yes	High
Not-Via Addresses	Yes	Yes	Yes	Yes	High
Remote LFA	No	Yes	Yes	Yes	Average
TI-LFA	Yes	Yes	Yes	Yes	High
OPFRR	Yes	Yes	No	Yes	High
PSFRR	Yes	Yes	No	Yes	High
LFRR	Yes	Yes	No	Yes	High
B-REP	Yes	Yes	Yes	Yes	High
Label B-REP	**Yes**	**Yes**	**Yes**	**Yes**	**High**

TABLE 12. Speed comparison of FRR of existing FRR solutions to Label B-REP.

	B-REP	LFA	R-LFA	EM-REP	LABEL B-REP
Time of FRR process	0.035144	0.035044	0.035148	0.035358	0.031327

A. COMPARISON WITH OTHER SOLUTIONS

Label B-REP is more flexible in solving outages, setting up backup paths, and solving problem areas of the original

B-Rep mechanism, such as non-exact next-hop in specific topologies.

Existing solutions have a high computational complexity in calculating alternative paths. Classic FRR mechanisms such as LFA and R-LFA are implemented in Cisco and Juniper's operating systems. These solutions are implemented because of their low computational difficulty and load on the router's CPU, but they provide very low repair coverage.

According to our research in recent years, we conclude the basic information is in the following Table 11.

Most FRR mechanisms install alternative FRR paths after fast link failure detection. This detection usually takes 30 ms by BFD protocol, and because alternative paths are calculated in advance, installation of alternative routes is immediate. Therefore, the speed of all FRR solutions is approximately the same, including B-REP and Label B-REP.

These FRR solutions differ in repair coverage and CPU complexity.

Label B-REP mechanism takes advantage of the flexible mapping of alternative next-hops using labels and constructing alternative paths.

Table 12 compares the FRR process of existing solutions to Label B-REP.

According to the result, Label B-REP is faster by approximately 11% compared to B-REP.

VII. CONCLUSION

The goal of the contribution was the design and implementation of the new Label B-REP mechanism, which eliminated the deficiency of the previous B-REP mechanism.

When examining the B-REP, it was found that the B-REP mechanism could cause a routing loop under certain circumstances, so in the current research, we are dedicated to determining the exact order of routers on the backup path. Thus, we created the Label B-REP mechanism, which removes the deficiency of the B-REP mechanism and ensures that the change of the original route through the unambiguous order of the routers in the alternate route will take place faster since the connection restoration through the alternative route will take place faster than it was in the case of the B-REP mechanism, as the influence of unwanted loops is removed (the next-hop router on the backup path is always at the top of the imaginary stack).

As was proven in the work by simulations, the new Label B-REP mechanism removed the influence of negative loops and, compared to the B-REP mechanism, can restore the connection 11% faster in the event of a line failure, as well as in the case of an entire node failure.

In the future, our research will be directed towards finding a mechanism that would find application in SDN and WSN networks.

ACKNOWLEDGMENT

This research was funded by the Slovak Grant Agency VEGA project Fast Reroute, No. 1/0316/24.

REFERENCES

[1] M. Chiesa, R. Sedar, G. Antichi, M. Borokhovich, A. Kamisinski, G. Nikolaidis, and S. Schmid, "Fast ReRoute on programmable switches," *IEEE/ACM Trans. Netw.*, vol. 29, no. 2, pp. 637–650, Apr. 2021, doi: 10.1109/TNET.2020.3045293.

[2] D. Merling, S. Lindner, and M. Menth, "Comparison of fast-reroute mechanisms for BIER-based IP multicast," in *Proc. 7th Int. Conf. Softw. Defined Syst. (SDS)*, Apr. 2020, pp. 51–58, doi: 10.1109/SDS49854.2020.9143935.

[3] M. Chiesa, A. Kamisinski, J. Rak, G. Rétvári, and S. Schmid, "A survey of fast-recovery mechanisms in packet-switched networks," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1253–1301, 2nd Quart., 2021, doi: 10.1109/COMST.2021.3063980.

[4] K.-T. Foerster, A. Kamisiński, Y.-A. Pignolet, S. Schmid, and G. Tredan, "Improved fast rerouting using postprocessing," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 537–550, Jan. 2022, doi: 10.1109/TDSC.2020.2998019.

[5] H.-K. Tan and T.-W. Kuo, "Optimistic fast rerouting," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 1692–1697, doi: 10.1109/ICC45855.2022.9838260.

[6] W. Gray, A. Tsokanos, and R. Kirner, "Multi-link failure effects on MPLS resilient fast-reroute network architectures," in *Proc. IEEE 24th Int. Symp. Real-Time Distrib. Comput. (ISORC)*, Jun. 2021, pp. 29–33, doi: 10.1109/ISORC52013.2021.00015.

[7] M. Ashar and A. P. Wibawa, "IPv6 vs IPv4 performance simulation and analysis using dynamic routing OSPF," in *Proc. 4th Int. Conf. Comput. Informat. Eng. (IC2IE)*, Sep. 2021, pp. 452–456, doi: 10.1109/IC2IE53219.2021.9649228.

[8] R. Pal, R. Kushwaha, R. S. Tomar, and R. Tripathi, "Comparison of three routing protocols in terms of packet transfer using IPv6 addressing," in *Proc. 8th Int. Conf. Smart Comput. Commun. (ICSCC)*, Jul. 2021, pp. 164–169, doi: 10.1109/ICSCC51209.2021.9528111.

[9] T. Ashraf, S. S. W. Lee, M. Iqbal, and J.-Y. Pan, "Routing path assignment for joint load balancing and fast failure recovery in IP network," *Appl. Sci.*, vol. 11, no. 21, p. 10504, Nov. 2021, doi: 10.3390/app112110504.

[10] M. Aldaoud, D. Al-Abri, M. Awadalla, and F. Kausar, "N-BGP: An efficient BGP routing protocol adaptation for named data networking," *Int. J. Commun. Syst.*, vol. 35, no. 14, p. e5266, Sep. 2022, doi: 10.1002/dac.5266.

[11] H. S. Alotaibi, M. A. Gregory, and S. Li, "Multidomain SDN-based gateways and border gateway protocol," *J. Comput. Netw. Commun.*, vol. 2022, pp. 1–23, May 2022, doi: 10.1155/2022/3955800.

[12] V. Baggan, A. K. Sahoo, P. K. Sarangi, and S. P. Chaturvedi, "A comprehensive analysis and experimental evaluation of routing information protocol: An elucidation," *Mater. Today, Proc.*, vol. 49, pp. 3040–3045, Jan. 2022, doi: 10.1016/j.matpr.2020.10.676.

[13] J. Papan, P. Segec, and M. Kvet, "Enhanced bit repair IP fast reroute mechanism for rapid network recovery," *Appl. Sci.*, vol. 11, no. 7, p. 3133, Apr. 2021, doi: 10.3390/app11073133.

[14] C. Filsfils, K. Talaulikar, D. Voyer, A. Bogdanov, and P. Mattes, *Segment Routing Policy Architecture*, document RFC 9256, RFC Editor, 2022.

[15] K.-T. Foerster, J. Hirvonen, Y.-A. Pignolet, S. Schmid, and G. Tredan, "On the price of locality in static fast rerouting," in *Proc. 52nd Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2022, pp. 215–226, doi: 10.1109/DSN53405.2022.00032.

[16] Y. Chen, Z.-W. Liu, Y. Yu, B. Wang, W. Yao, H. Liu, R. Chen, and D. Li, "Distributed multiple line-outages detection in power grid with finite time observer," in *Proc. IEEE Int. Conf. Cyborg Bionic Syst. (CBS)*, Mar. 2023, pp. 196–201, doi: 10.1109/CBS55922.2023.10115367.

[17] A. Jain and S. Bhullar, "Network performance evaluation of smart distribution systems using smart meters with TCP/IP communication protocol," *Energy Rep.*, vol. 8, pp. 19–34, Nov. 2022, doi: 10.1016/j.egyr.2022.05.108.

[18] N. Bayat, K. Mahajan, S. Denton, V. Misra, and D. Rubenstein, "Down for failure: Active power status monitoring," *Future Gener. Comput. Syst.*, vol. 125, pp. 629–640, Dec. 2021, doi: 10.1016/j.future.2021.06.055.

[19] R. Kang, M. Zhu, F. He, and E. Oki, "Implementation of virtual network function allocation with diversity and redundancy in kubernetes," in *Proc. IFIP Netw. Conf.*, Jun. 2021, pp. 1–2, doi: 10.23919/IFIPNetworking52078.2021.9472200.

[20] J. Tang, S. Wei, D. Li, and X. Li, "Optimizing systemic redundancy of traffic sensor networks while maintaining resilience: New evidence from using graph learning," *IEEE Syst. J.*, 2023, doi: 10.1109/JSYST.2023.3257886.

[21] W. Quan, *Emerging Networking Architecture and Technologies*, vol. 1696, 2023, doi: 10.1007/978-981-19-9697-9.

[22] Y. N. Krishnan and G. Shobha, "Performance analysis of OSPF and EIGRP routing protocols for greener internetworking," in *Proc. Int. Conf. Green High Perform. Comput.*, Mar. 2013, pp. 1–4, doi: 10.1109/ICGHPC.2013.6533929.

[23] J. Lv, X. Wang, M. Huang, F. Li, K. Li, and H. Cheng, "Accomplishing information consistency under OSPF in general networks," in *Proc. IEEE 22nd Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2016, pp. 278–285, doi: 10.1109/ICPADS.2016.0045.

[24] J.-R. Luttringer, Q. Bramas, C. Pelsser, and P. Mérindol, "A fast-convergence routing of the hot-potato," in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 1–10, doi: 10.1109/INFOCOM42981.2021.9488880.

[25] A. Kamisinski, "Evolution of IP fast-reroute strategies," in *Proc. 10th Int. Workshop Resilient Netw. Design Modeling (RNDM)*, Aug. 2018, pp. 1–6, doi: 10.1109/RNDM.2018.8489832.

[26] R. Petija, M. Michalko, F. Jakab, and P. Fecil'ak, "Convergence of routing protocols in real and simulated environments," in *Proc. 16th Int. Conf. Emerg. eLearning Technol. Appl. (ICETA)*, Nov. 2018, pp. 425–430, doi: 10.1109/ICETA.2018.8572184.

[27] M. Shand and S. Bryant. (2010). *IP Fast Reroute Framework*. RFC5714. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5714.txt

[28] M. Gjoka, V. Ram, and X. Yang, "Evaluation of IP fast reroute proposals," in *Proc. 2nd Int. Conf. Commun. Syst. Softw. Middleware*, Jan. 2007, pp. 1–8, doi: 10.1109/comswa.2007.382443.

[29] S. Cevher, M. Ulutas, and I. Hökelek, "Performance evaluation of multiple routing configurations," in *Proc. 21st Signal Process. Commun. Appl. Conf. (SIU)*, Apr. 2013, pp. 1–4, doi: 10.1109/SIU.2013.6531353.

[30] I. Nedyalkov, A. Stefanov, and P. Apostolov, "Modeling of the convergence time of an IP–based network with different traffic loads," in *Proc. 18th Int. Conf. Smart Technol.*, Jul. 2019, pp. 1–6, doi: 10.1109/EUROCON.2019.8861735.

[31] V. K. Pal and S. M. Ramteke, "A framework for fast IP rerouting," in *Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES)*, Feb. 2014, pp. 1–6, doi: 10.1109/ICICES.2014.7033905.

[32] S. Riaz, M. Rehan, T. Umer, M. K. Afzal, W. Rehan, E. U. Munir, and T. Iqbal, "FRP: A novel fast rerouting protocol with multi-link-failure recovery for mission-critical WSN," *Future Gener. Comput. Syst.*, vol. 89, pp. 148–165, Dec. 2018, doi: 10.1016/j.future.2018.06.029.

[33] (2022). *Fibbing: Central Control Over Distributed Routing*. Accessed: Jan. 3, 2022. [Online]. Available: http://fibbing.net/

[34] S. S. W. Lee, K.-Y. Chan, T.-S. Wong, and B.-X. Xiao, "A fast failure recovery scheme for fibbing networks," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1196–1212, 2020, doi: 10.1109/OJCOMS.2020.3018197.

[35] L. Roelens, Ó. G. D. Dios, I. D. Miguel, E. Echeverry, and R. J. D. Barroso, "Performance evaluation of TI-LFA in traffic-engineered segment routing-based networks," in *Proc. 19th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, Apr. 2023, pp. 1–8, doi: 10.1109/DRCN57075.2023.10108280.

[36] W. Tavernier, D. Papadimitriou, D. Colle, M. Pickavet, and P. Demeester, "Self-configuring loop-free alternates with high link failure coverage," *Telecommun. Syst.*, vol. 56, no. 1, pp. 85–101, May 2014, doi: 10.1007/s11235-013-9821-z.

[37] L. Csikor and G. Rétvári, "On providing fast protection with remote loop-free alternates," *Telecommun. Syst.*, vol. 60, no. 4, pp. 485–502, Dec. 2015, doi: 10.1007/s11235-015-0006-9.

[38] S. Cevher, M. Ulutas, and I. Hokelek, "Topology-aware multiple routing configurations for fault tolerant networking," *J. Netw. Syst. Manage.*, vol. 24, no. 4, pp. 944–973, Oct. 2016, doi: 10.1007/s10922-015-9358-4.

[39] S. Cevher, M. Ulutas, S. Altun, and I. Hökelek, "Multiple routing configurations for fast re-route in software defined networks," in *Proc. 24th Signal Process. Commun. Appl. Conf. (SIU)*, May 2016, pp. 993–996, doi: 10.1109/SIU.2016.7495909.

[40] M. A. El-Serafy, A. M. Elsayed, M. H. Aly, E. A. El-Badawy, and I. A. Ghaleb, "Multiple routing configurations for datacenter disaster recovery applicability and challenges," in *Proc. Int. Conf. Comput. Commun. Eng.*, Sep. 2014, pp. 146–149, doi: 10.1109/ICCCE.2014.51.

[41] Z. Limin, L. Zheqing, W. Hui, L. Peiyu, and C. Xi, "A new backup topology design method for IP fast recovery," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 1992–1997, doi: 10.1109/COMPCOMM.2016.7925050.

[42] M. Nagy, J. Tapolcai, and G. Rétvári, "Optimization methods for improving IP-level fast protection for local shared risk groups with loop-free alternates," *Telecommun. Syst.*, vol. 56, no. 1, pp. 103–119, May 2014, doi: 10.1007/s11235-013-9822-y.

[43] K. Kuang, S. Wang, and X. Wang, "Discussion on the combination of loop-free alternates and maximally redundant trees for IP networks fast reroute," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 1131–1136, doi: 10.1109/ICC.2014.6883473.

[44] A. Atlas, C. Bowers, and G. Enyedi, *An Architecture for IP/LDP Fast Reroute Using Maximally Redundant Trees (MRT-FRR)*, document RFC7812, 2016.

[45] A. H. M. Aman, A.-H.-A. Hashim, and H. A. M. Ramli, "Mathematical evaluation of context transfer and multicast fast reroute in multicast enabled network mobility management," *Int. J. Control Autom.*, vol. 10, no. 3, pp. 207–216, Mar. 2017, doi: 10.14257/ijca.2017.10.3.17.

[46] T. Elhourani, A. Gopalan, and S. Ramasubramanian, "IP fast rerouting for multi-link failures," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 3014–3025, Oct. 2016, doi: 10.1109/TNET.2016.2516442.

[47] R. Papneja, S. Vapiwala, J. Karthik, S. Poretsky, S. Rao, and J. L. Le Roux, *Methodology for Benchmarking MPLS Traffic Engineered (MPLS-TE) Fast Reroute Protection*, document RFC 6894, RFC Editor, 6894.

[48] T. Benhcine, H. Elbiaze, and K. Idoudi, "Fast reroute-based network resiliency experimental investigations," in *Proc. 15th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jun. 2013, pp. 1–4, doi: 10.1109/ICTON.2013.6603065.

[49] M. I. Goulamghoss and V. Bassoo, "Analysis of traffic engineering and fast reroute on multiprotocol label switching," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 2, pp. 2409–2420, Feb. 2021, doi: 10.1007/s12652-020-02365-5.

[50] O. Lemeshko and K. Arous, "Fast ReRoute model for different backup schemes in MPLS-network," in *Proc. 1st Int. Sci.-Practical Conf. Problems Infocommunications Sci. Technol.*, Oct. 2014, pp. 39–41, doi: 10.1109/INFOCOMMST.2014.6992292.

[51] D. Awduche, J. Malcolm, J. Agogbu, M. O'Dell, and J. McManus, *Requirements for Traffic Engineering Over MPLS*, document RFC 2702, 2019.

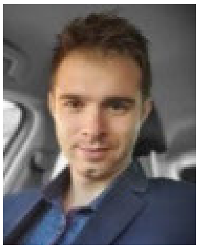
[52] O. Lemeshko and O. Yeremenko, "Linear optimization model of MPLS traffic engineering fast ReRoute for link, node, and bandwidth protection," in *Proc. 14th Int. Conf. Adv. Trends Radioelecrtronics, Telecommun. Comput. Eng. (TCSET)*, Feb. 2018, pp. 1009–1013, doi: 10.1109/TCSET.2018.8336365.

[53] J. Papan, P. Segec, P. Paluch, J. Uramova, and M. Moravcik, "The new multicast repair (M-REP) IP fast reroute mechanism," *Concurrency Comput., Pract. Exper.*, vol. 32, no. 13, p. e5105, Dec. 2018, doi: 10.1002/cpe.5105.

[54] J. Papan, P. Segec, O. Yeremenko, I. Bridova, and M. Hodon, "Enhanced multicast repair fast reroute mechanism for smart sensors IoT and network infrastructure," *Sensors*, vol. 20, no. 12, p. 3428, Jun. 2020, doi: 10.3390/s20123428.

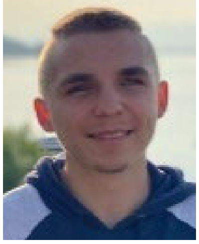
[55] J. Papan, P. Segec, O. Yeremenko, I. Bridova, and M. Hodon, "A new bit repair fast reroute mechanism for smart sensors IoT network infrastructure," *Sensors*, vol. 20, no. 18, p. 5230, Sep. 2020, doi: 10.3390/s20185230.

[56] (2023). *OMNeT++ Discrete Event Simulator*. Accessed: May 16, 2023. [Online]. Available: https://omnetpp.org/



JOZEF PAPAN received the Ph.D. and Doctorate degrees in applied informatics from the Faculty of Management Science and Informatics, University of Žilina, Slovakia, in 2015 and 2020, respectively.

Currently, he is the Head of the IP Fast Reroute Research Team, the Director of the Fortinet Network Security Academy, and a member of the Faculty of Management Science and Informatics and the Cisco Academy. He is the Teacher of the following subjects: securing networks with Fortinet (Fortinet Academy), principles of ICS (Cisco), and network architectures (Linux + Networks). He is the author or coauthor of more than 30 scientific papers published in scientific journals and presented at international conferences. His research interests include IP fast reroute, fault-tolerance, protocols and services in IP networks, WSN, the IoT, modeling, simulation of computer networks, smart sensors, wireless technology, portable devices, technical cybernetics, and cloud computing.



TOMAS CHOVANEC is a Research Student with the Faculty of Management Science and Informatics, University of Žilina, Slovakia. He is a Scientist with the Special Fast Reroute Research Team. His research interests include fault-tolerance, fast reroute, protocols and services in IP networks, WSN, the IoT, modeling, simulation of computer networks, and smart sensors.

IVANA BRIDOVA received the M.Sc. and Ph.D. degrees in telecommunications from the Faculty of Electrical Engineering, University of Žilina, Slovakia, in 2002 and 2010, respectively. She spent three months as a Researcher with the Department of Electrical and Computer Engineering, University of Patras, Greece, where she worked in optical access networks. Since 2020, she has been a Teacher and a Researcher with the Faculty of Management Science and Informatics. Her research includes IP fast reroute, fault-tolerance, protocols, services in IP networks, the IoT, modeling, simulation of computer networks, architectures of informatics systems, wireless technology, portable devices, technical cybernetics, and cloud computing. She has been a member of the Technical Program Committees for International Conference on Computational Collective Intelligence (ICCCI).

MICHAL KVET (Member, IEEE) became an Associate Professor of applied informatics from the Faculty of Management Science and Informatics, University of Žilina, Slovakia, in 2020. He is a recognized researcher, a conference speaker, and an Oracle ACE Alum. He is the author of several text-books and monography in temporal database processing. He is also the author of more than 70 scientific articles indexed in IEEE-Xplore, Scopus, and WOS. He is certified for SQL, PL/SQL, analytics, and cloud databases. He strongly participates with Oracle Academy and is part of multiple Erasmus+ projects. Besides, he is a consortium leader of the Erasmus+ Project dealing with the environmental analytics. He also organizes multiple database workshops annually. His research is devoted to the temporal databases, indexing, performance, analytics, and cloud computing.