

Telecom Operators in Virtual Heterogeneous Networks May Benefit from Dynamic Resource Replacement for Virtual Services Based on Flow Splitting: FSB-DReViSeR

Umakant Bhaskar Gohatre^{1,*}, Dr. Venkat P. Patil², and Dr. Sunil S. Chavan³

¹Department of Computer Science and Engineering, Smt. Indira Gandhi College of Engineering, Navi Mumbai, Mumbai University, Mumbai, Maharashtra, India

²Vice Principal, Smt. Indira Gandhi College of Engineering, Navi Mumbai, Mumbai University, Mumbai, Maharashtra, India

³Principal, Smt. Indira Gandhi College of Engineering, Navi Mumbai, Mumbai University, Mumbai, Maharashtra, India

Abstract. Virtual networks provide their services by aggregating groups of virtual devices that use an existing physical network to interact with one another. These services, unlike the underlying network, often need a large number of distinct resources (bandwidth, processing power, servers, etc.) to operate. The adaptability of the virtual network architecture is what makes it so appealing. A server is required for each virtual service; hence computers are categorized as virtual service resources. It is common practice to either supply resources for the virtual network or replace the virtual service resource by migrating the service to another node that offers the most suitable number of resources to do so (QoS) when a given resource is unable to meet quality of service requirements due to traffic variation caused by mobile users. Our flow-splitting technique allows for the dynamic redistribution of virtual service resources across several virtual connections. We take a different approach than the prevalent tree-based approaches in the current body of literature by basing our method on graph topology instead. The simulation results from this study demonstrate that our solution drastically decreases the time needed to replace virtual service resources when compared to existing methods.

1 Introduction

Network virtualization has gained popularity in recent years because it gives network operators (Cisco, Juniper, Amazon, etc.) a great deal of leeway in creating their networks. Virtualization technologies are becoming more popular among network service providers as a means to enhance customer service. As a result of virtualization's segregation effect, each virtual network may have its own topology, enabling finer-grained management of quality of

* Corresponding author: umakantbhaskar@gmail.com, venkat.patil@sigce.edu.in
sunil.chavan@sigce.edu.in

service and traffic. Isolated virtual devices may function autonomously to help with things like switching between locations or providing users with specialised virtual resources [1]. Modern network designs extensively take use of the freedom to move resources around. For example, cloud computing platforms are mentioned, since they provide customers with on-demand access to a variety of resources (including processing power, network throughput, and data storage). Even in the rapidly growing field of IoT (the Internet of Things), where many devices are used and enormous amounts of data are consistently generated ([2, 3]), this holds true. The Internet of Things is illustrative of a diverse network design with limited resources. Changes in traffic patterns are an inevitable consequence of user mobility across networks and of the addition and removal of nodes. Quality of service in the network is dependent on the server's (the virtual service resource) ability to respond to changes in load. However, there is a chance that, in the long term, this virtual service resource won't be able to provide the necessary QoS. When this occurs, more server power may be obtained via the use of a cloud computing architecture [2, 4]. The service provider has to figure out how to meet the demand for these resources while still making a profit. Such a situation is ideal for illustrating the VNE problem [5]. Many studies have been conducted to find a solution to this problem ([5-7]), yet all of the proposed solutions waste space or other resources. The virtual service resource may move to a new source node with more available resources to prevent this from happening. The second strategy, which has not been adequately explored previously [8, 9], is the focus of our investigation and requires no additional funding. In contrast to the graph topology of real-world networks, the offered approaches employ a tree topology.

2 Literature Review

While mobile users present particular difficulties for wireless networks, another study [10] offered some recommendations for managing network resources to optimize the number of satisfied users. However, these techniques focus only on bandwidth and disregard other critical QoS elements like jitter and packet loss. This is particularly true in the more constrained setting of virtual networks, where such an event simply does not happen. Many different players (providers of services and infrastructure) are often involved in such settings. Resource allocation must take into account the dynamic nature of network topologies due to the adaptability of virtual networks and the freedom members have in forming their own networks [11]. The communication spectrum [12] must also be taken into account during the building of optical networks. However, optical networks are beyond the scope of this research.

Sensor networks and the Internet of Things are examples of highly specialised wireless network environments that need even more power. Products (both storage and CPU) fail because not enough effort and money were put into their design. Quality of Service (QoS) may be improved by allocating cloud resources for IoT data processing based on user traffic patterns, as suggested by researchers in [13]. To account for variations in user requirements and IoT technology, [13] proposes a mathematical method. However, this approach puts calculating the optimal allocation configuration ahead of taking into account how long consumers have to wait for calculations to finish.

Researchers Koyanagi and Tachibana [9] developed a replacement technique that maintains quality of service (QoS) when a new node is introduced to a network. In a tree-like approach, the virtual service is replicated on the new host node. The approach continues on to the final destination if and only if the Quality of Service (QoS) criteria are satisfied at each of the intermediary nodes.

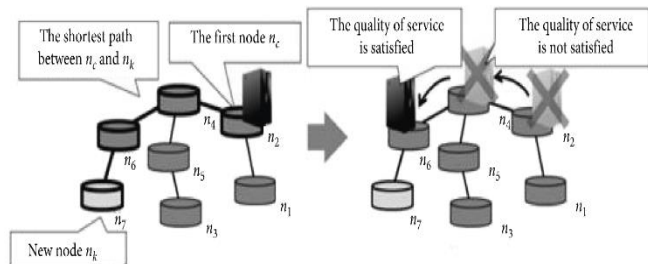


Fig. 1: We are able to swap out the associated virtual service resources by moving the virtual network service one hop.

In virtual heterogeneous multimedia networks, jitter (packet delay variation) is a major issue, yet the time needed to replace the service does not satisfy the criterion for acceptable QoS. This happens because the QoS of each node along the alternative route is monitored. (ii)Concurrent examination of additional replacement choices are precluded by the utilised tree design. By proposing a replacement mechanism that checks the QoS satisfaction only after the service has been completely replaced by the new host node, Horiuchi and Tachibana [8] enhanced the technique of Koyanagi and Tachibana [9]. This strategy also greatly reduces the downtime caused by component replacement. However, this method does not solve the tree topology issue. No service migration strategy is presented, and there are a lot of new components to learn. The last step of the Horiuchi and Tachibana replacement procedure is to designate a node as the new virtual service resource, as seen in Fig 2

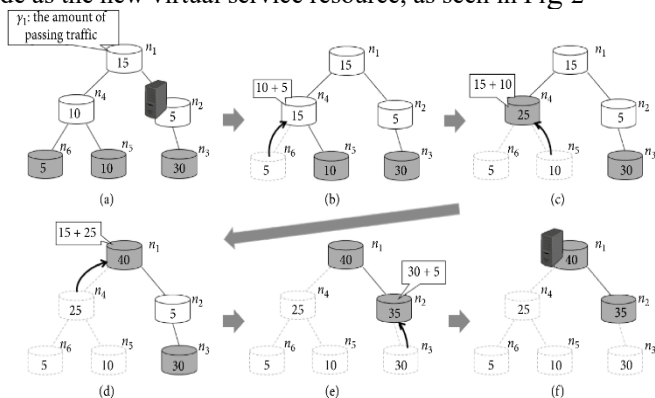


Fig. 2 Horiuchi and Tachibana's method for a virtual service that just requires a single path to function in lieu of its actual infrastructure.

To address the problems highlighted in [8], we had earlier presented [14] a tree-based topology replacement approach that included a data movement mechanism. To circumvent the issue of all leaf nodes having the same traffic weight, we suggested in [8] traversing the tree until reaching a level where the parent nodes have distinct traffic weights. If there is a tie, it will be broken by the node that had the most visits. The proper procedure is shown in Figure 3. In this respect, it is worth noting that Horiuchi and Tachibana [8] were unable to find a solution when the tree leaves had the same traffic weights

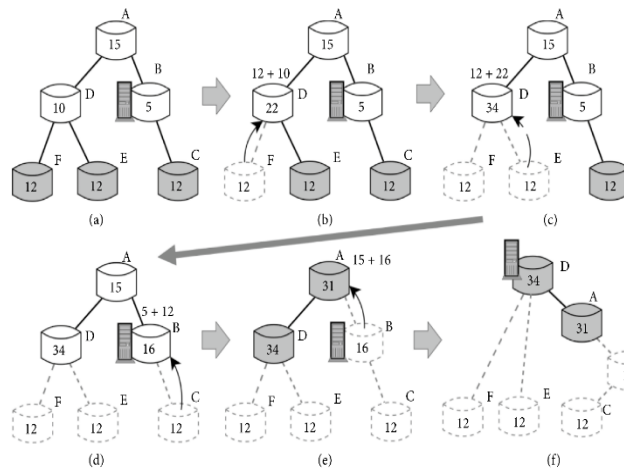


Fig 3_ How to solve the problem of all leaf nodes being equally busy

We suggested in [14] that decreasing the rate at which virtual service resources are swapped out might help decrease packet delay variance. In layman's terms, this helps postpone replacements that aren't strictly essential. At long last, the solution has a well-thought-out strategy for data migration. The strategy proposed in [8] for replacing virtual with physical services is enhanced by each of these changes. The problems solved in [14] can be utilised with no other data structure.

Recently, Liu et al. [15] reported a virtual machine migration strategy in an SDN-OpenFlow multicontroller system that is quite similar to the virtual service resource replacement investigated here. SDN's (Software Defined Networking) central control of massive heterogeneous networks has been shown. [16] To determine the optimal route for relocating a virtual machine, the source machine will query the controller. The controller will choose the path with the greatest throughput if more than one exists for the migration packets. Still, there are a couple problems with this strategy: Figure 4 depicts the results of unwisely selecting one of two routes in a network with equal maximum throughput. Here, we need to relocate a virtual machine from to. The first path offers 9 Gbps, the second 8 Gbps, and the third also offers 9 Gbps. The migration routes 1 and 3 with the maximum available bandwidth are prioritised using the Liu et al. approach. Option 3, however, outperforms Choice 1 when hop counts are taken into account. Poor migration route selection in a network with intermittent connections: Figure 5 displays the results of Liu et al.'s analysis, which concludes that Route 1 is the optimal option for virtual machine migration since its bandwidth is five times more than that of Paths 4, 3, and 2. However, when we take into consideration the lengths of the connections and the number of hops along the various routes, the following results become clear: The overall distance of Path1 is 7.5, the total number of hops is 5, and its capacity is 5 Gbps. When using a bandwidth of 2 Gbps, the total distance travelled through (b)Path2 is 2.5 miles; multiply this by three to obtain the total number of hops, which is 8. The 3 Gbps throughput of Path3 is spread over its 5-hop, 1-mile length. Total distance for Path4 (capable of 4 Gbps) = 2 + 1 + 1 + 2 = 6. Even though the highest possible bandwidth is on Route 1, the best options may really be Routes 2, 3, or 4. Therefore, bandwidth is not the only factor that should be considered while selecting a migration strategy

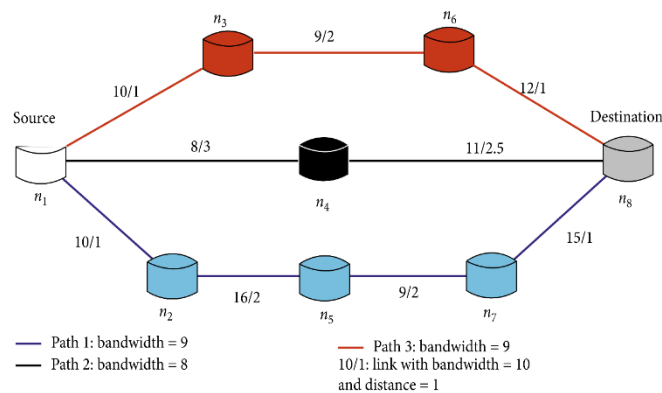


Fig 4 _ Mistakenly selecting one method out of many that all have the same throughput.

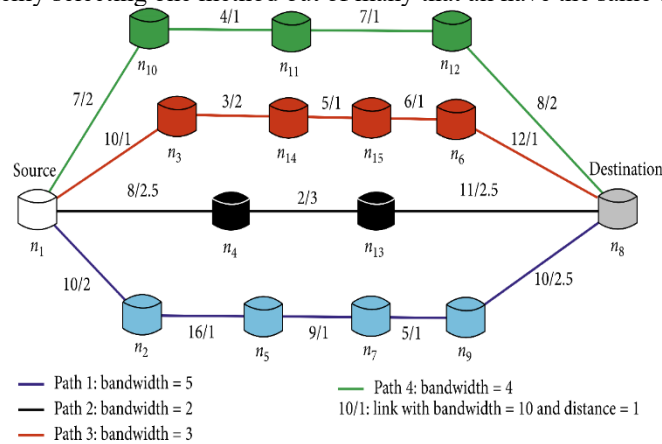


Fig 5 _ Distribution of available bandwidth plan.

Once a migration path is decided upon, the total time necessary to complete it may be estimated. According to the linear model (1) proposed by Kherbache [17], the amount of time it takes to migrate a virtual machine (VM) increases linearly with the amount of RAM and bandwidth available to the VM. While the copy speed of the virtual machine's files is described in detail by the interruption time, it has minimal effect on the overall migration time [17]. However, (1) does not take into account the duration of connections or the total number of nodes visited.

3 Principal Contribution to the Working Model

The major objective of this article is to provide a framework for enhancing Quality of Service (QoS) in heterogeneous virtual networks amidst variable traffic caused by user mobility. We give our own remedies to the problems with the work of Horiuchi and Tachibana, focusing on tree topology. Our contributions may be broken down into two categories: To allow for quick virtual service resource replacement, I services files are sent across multiple connections to a target node. The traffic-splitting technique ([18, 19]) is the foundation of this approach of dealing with node and connection failures. Bandwidth, network latency, and connection lengths are just a few of the metrics used to choose the best path. (ii) We enhance our method by considering factors other than bandwidth while selecting the best path to pursue. Despite the fact that there may not be enough capacity in the link to carry the data flow, we may still choose a route as the ideal replacement choice if the bandwidth does not account for the real available flow space in the connections.

The remaining parts of this work are organised as follows. The traffic-splitting method is broken down in Part 2. In Section 3, we explain our first major contribution to the study of flow splitting-based replacement, which entails determining the best method for a given set of conditions, such as throughput, network topology, and path lengths. Section 4 builds upon Section 3 by refining a replacement approach based on flow splitting with an emphasis on link throughput. The results of our simulations are discussed in Section 5. The report concludes with a summary of the results and any required inferences. The Thinking Behind Traffic Segmentation This section explains the method for segmenting traffic and its key requirements

3.1 The Descriptive statements

Traffic splitting allows for the distribution of a flow's weight over many lines, improving overall quality of service [18, 19]. Common scenarios in which traffic splitting is used include: Distributing the original flow's burden across a large number of connections might help prevent bottlenecks; one such way is resource gap among links and nodes [18]. Congestion brought on by mobile users and shifting network topologies [8, 9] and link or node failures [18, 20] leading to packet redistribution across connections are typical causes of the chasm. (ii) Underutilized network resources, such as mirror servers, are crucial for processing a high number of requests fast. Depending on the connection capacity [22], the flowlets (little flow formed by splitting) may have the same mass or different masses.

In certain circumstances, rerouting flows requires more work than the typical single-path method. Detailed Description of Duties for the Traffic Division 2.2 Reconciling the packets at the destination node is a significant challenge for the flow splitting method if the goal of the split is to redirect traffic [18]. Packets don't always get to their destination in the order in which they were sent due to the irregularity of the networks, making it difficult to reproduce the original flow. Reducing the number of flowlets [22] and assigning distinct labels to packets in the same flow ([18, 22]) are two solutions presented in the literature to this problem. Another challenging issue is finding the best routes to advance the flow lets. The total length of these paths is proportional to the number of flows let (sub flows) created. Here, we investigate what considerations must be made while selecting the most effective paths for traffic forwarding. A variety of criteria might be taken into account: To maximise flowlet throughput, we give preference to routes that bring the source and destination nodes as near together as feasible (i) in terms of link length. However, we may have the shortest-possible connection, which has little bandwidth and takes a great deal of forwarding time. This can't happen if we care about maintaining QoS. (ii) Bandwidth. (iii) Throughput.

In this research, these three considerations are included into the optimum route selection technique for relocating virtual service resources. Method for Splitting Traffic to Substitute Virtualized Services We present our expanded version of Horiuchi and Tachibana's [8] traffic-splitting-based solution. As compared to [8], this approach takes into account the graph topologies and data movement mechanisms of a virtual network's services.

3.2 Example of hypothesis and Notation

Before we discuss our method, let's make some assumptions: (i) The network is represented by the graph $G(N, L)$, where N is the number of nodes and L is the number of links. Each pair of nodes may be connected to another because (i) there are always two routes to go from

one node to another, and (ii)links can travel in either direction. (iii) Heterogeneous networks need a software-defined network controller. (iv)Each connection's bandwidth, length, or throughput is considered independently, and the relative importance of the nodes is disregarded. (v)At any given moment, the lag in copying services is the same for all nodes

We Make Substitutions The original migrated traffic is divided into numerous smaller flowers and transmitted over several parallel channels in order to restore the virtual service resource at the destination node. They all go the same path. Since migration traffic can only be segmented by a single virtual service resource on the network, the flowlets will not be segmented again until they reach the final node. The solution to the topic of how to divide up migration traffic might depend on two factors.

3.3 Criteria for Choosing the Best Migration Path

In order to speed up the process of replacing virtual service resources, we will be directing traffic down the most direct routes. Our route selection technique, which accounts for bandwidth, connection length, and number of hops, alleviates the limitations of the Liu et al. [15] approach, which is bound to the selection of the shortest migration path. Time required for each flowlet to complete its migration is determined before a route set is selected. The migration time per unit of distance is assumed to be given by (1), which implies that for link, the migration time is provided by

$$d_{ik} = w_{ik} \cdot d_{mig} \tag{1}$$

$$TM_i = \sum_{k=1}^{n-1} d_{k(k+1)} \tag{2}$$

If there are n potential migration paths, the best one is the one that takes the least amount of time overall, or (1 n). Although there are other migration paths that might be used with our replacement method, these are the best ones that are consistent with (3). Figure 6 shows the consequences of using [15] and choosing the incorrect virtual machine migration path while having identical bandwidths

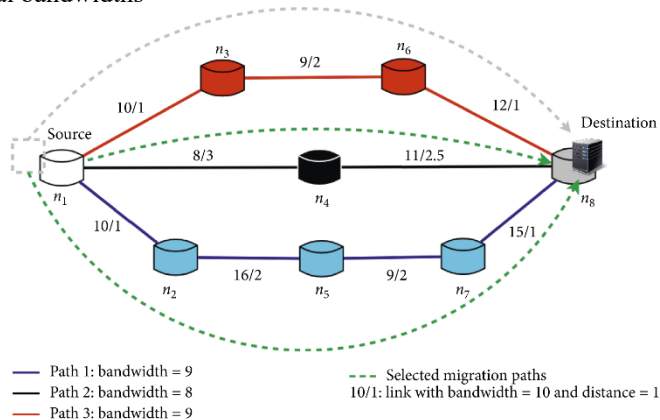


Fig 6 The process of determining the best available alternate paths for connections with a given bandwidth.

Take note of it. Replacement Paths 2 and 3 are best for bidirectional traffic since they switch traffic over quickly. When pathways 3 and 1 have the same bandwidth, the single-path

migration technique encourages selecting path 3 as the optimal migration path, as opposed to the single-path migration strategy proposed by Liu et al[15]. Figure 7 shows that when compared to the method given in [15], which has the same issue of making a poor decision when the bandwidths of the available pathways are different, our method for migration route selection performs much better.

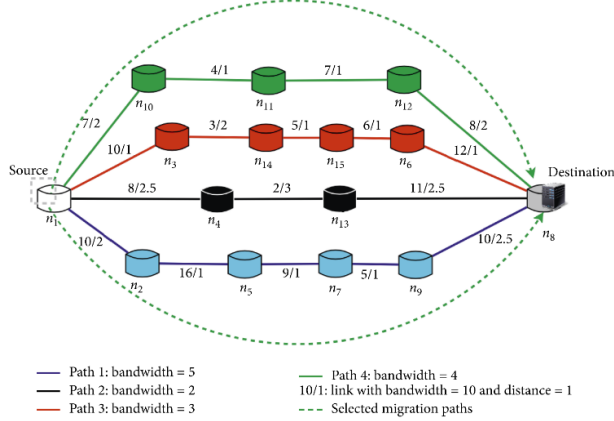


Fig 7 Bandwidth-adaptive alternative route selection.

Estimates are established for each route based on research into the likely replacement timeframes for virtual service Based on these data, it seems that Replacement Paths 1 and 4 are preferable to the Single Replacement Path 1 selected by Liu et al. [15].

4 System for Transferring Information

When moving virtual machines from one physical host to another, it is essential that their associated virtual service resource data remain in the same condition [23]. We need a safe method of exchanging information in order to do this. The most well-known methods for moving virtual machines are: (i)In cold migration, also known as "stop and copy" [24], the virtual machine is powered down and its data is copied to the new host. The virtual machine is started after service on the designated node has been restored. The time it might take to complete bootstrapping on the target node is one of the method's limitations. (ii)Precopy, postcopy, and hybrid postcopy are the three main approaches to live migration [24, 25]. Before starting up, a full copy of the virtual machine's RAM is sent to the remote host. When transferring a large number of pages, the service on the source computer is terminated and only the pages that have been modified are transmitted to the destination host. The biggest issue with this method is that the target node could not obtain the most recent memory pages if the maximum interruption interval is too short [17]. Therefore, it may take a considerable amount of time to move. The postcopy migration technique involves pausing the virtual machine on the source node before copying it to the target node. When the data transfer to the target node is complete, the virtual machine is restarted. In the event of a failed migration on either the source or the destination node, the virtual machine's memory will become corrupted, necessitating a new deployment. Hybrid postcopy [26] was developed to fix the inefficiencies of traditional postcopy. Transferring memory pages from one node to another without compromising availability is the goal of incremental memory page copying (precopy; see Figure 8). When a certain event occurs, the service is suspended on the original machine and restarted with the most up-to-date data on the target machine (postcopy). To drastically cut down on migration time and disruption, we advise using a hybrid strategy.

We're transitioning from physical to virtual infrastructure utilising a variety of ways. All service data, traffic, and requests for the current execution state, as well as any changed pages of memory, are routed during the pre copy phase.

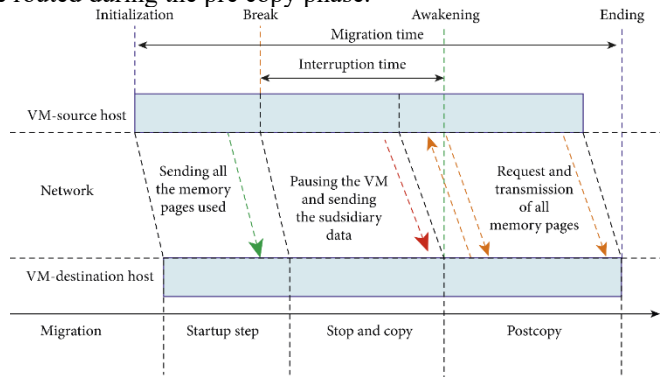


Fig 8: Algorithm II for hybrid post-copying. Combining link throughput with traffic splitting for optimal route selection is an efficient method for substituting virtual resources for physical ones.

We provide a flow-splitting-based solution that considers both available bandwidth and the length of the connection to select the shortest viable alternative way as a replacement strategy for virtual service resources. It is possible that the shortest pathways will be chosen based on bandwidth even if the linked connections do not permit service data movement, since both connection data flow and bandwidth fluctuate over time. We detail how we determined the shortest route selection criterion for a virtual service resource replacement, with emphasis on throughput and connection length.

4.1 Challenges of Using Data Flow to Select

The primary difficulty with utilising data flow as a criterion for choosing a data forwarding technique is that it is very dynamic, being unique for each connection and constantly adjusting to meet the needs of the network at any given moment. Some issues that must be addressed are as follows: In order to establish the current link flow rate at any given moment, which is required to answer question I, a full overview of the network and its traffic is required. Software-defined networking (SDN) is used as a workaround since virtual service resources do not have this functionality. Given that a better route at one time may not be a better road at another time due to variances in flow data rate, (ii)figuring out how to produce a substitute route decision that can be depended upon to maintain flows within an acceptable time range.

Our Path Selection Algorithm is 4.2) Information Flow-Driven In order to solve the traffic mapping issue, the SDN controller might use the information it collects about the whole network, including its topology, failures, and traffic. The controller's network accountability procedure provides access to this information. The term "network accountability" refers to the method through which a network's infrastructure provider may track user activity for the purpose of discovering abnormalities or extracting useful information to increase the network's profitability [27]. Therefore, using this accounting, we can strengthen the system's reliability and security [28]. If our solution determines that a virtual service resource has to

be replaced, the transmitting node will communicate this information to the controlling node, which will subsequently reroute the relevant traffic flows. This map may then be used to figure out which of the available detours is the most time- and energy-efficient.

The controller estimates the minimum and maximum changes in data flow using historical data and the network traffic map. The controller then takes these variables into account and picks a path. Since then, we have specified the minimum and maximum replacement durations as

$$\begin{aligned} d_{ik(\min)} &= \frac{m_{\text{mig}}}{v_{\min}} + i_{(\text{mig})ik}, \\ d_{ik(\max)} &= \frac{m_{\text{mig}}}{v_{\max}} + i_{(\text{mig})ik}, \end{aligned} \quad (3)$$

How long does it take for information to go across several nodes? From equations (3)-(5), we can determine that and are the minimum and maximum total replacement times for a given route, respectively.

$$\begin{aligned} \text{TM}_{i(\min)} &= \sum_{k=1}^{n-1} d_{k(k+1)\min}, \\ \text{TM}_{i(\max)} &= \sum_{k=1}^{n-1} d_{k(k+1)\max}, \end{aligned} \quad (4)$$

A route selection technique based on would be unable to adjust to an increase in the flow data rate. As the flow data rate increases, a selection based on the maximum replacement time becomes insufficient. In particular, we suggest using the flow average determined by

$$d_{ik} = \frac{d_{ik(\min)} + d_{ik(\max)}}{2}. \quad (5)$$

This median is predicted to reduce the instances of connection flow data rates that cannot be accounted for.

Below Fig 9 illustrates how throughput may be used as an alternative to bandwidth when determining the most efficient migration strategy. In this picture, we see how different replacement durations are determined by factoring in the average throughput of each route:

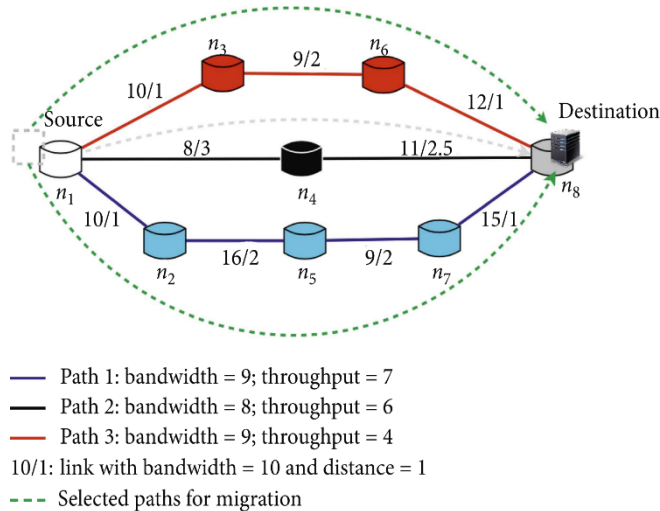


Fig 9: Using bandwidth rather than throughput, we find that changing routes 3 and 1 is superior than replacing pathways 3 and 2

5 Discussion and Analysis of Simulated Results

The simulation results used to evaluate the effectiveness of our methods are shown. We analyse the potential role that bandwidth plays in the decision to switch to a different route and the effect that flow splitting has on the need to replace virtual server resources. In Section we compare and evaluate several migration strategies that take into account bandwidth and flow data rate. We utilised the latest version (5.0) of OMNET++, a discrete event network simulator, to simulate the system. As an example of the scalability of our approaches, we simulated two networks, one with 20 nodes and 31 connections and the other with over a thousand nodes and hundreds of thousands of linkages (a total of 60 nodes and 90 links). Quality-of-Service Effects of Traffic-Splitting with Dynamic Route Bandwidths The time it takes to switch out a virtual service resource is compared between our graph-based FSB-DReViSeR method, Horiuchi and Tachibana's [8] method without a migration methodology, and Horiuchi and Tachibana's updated method with a migration strategy. Figure 10 displays the results for Network 1, whereas Figure 11 displays the results for Network 2. When moving from one host to another, [8] employs the hybrid virtual machine migration approach [26]. Through simulation, we were able to determine the typical replacement intervals for each tactic.

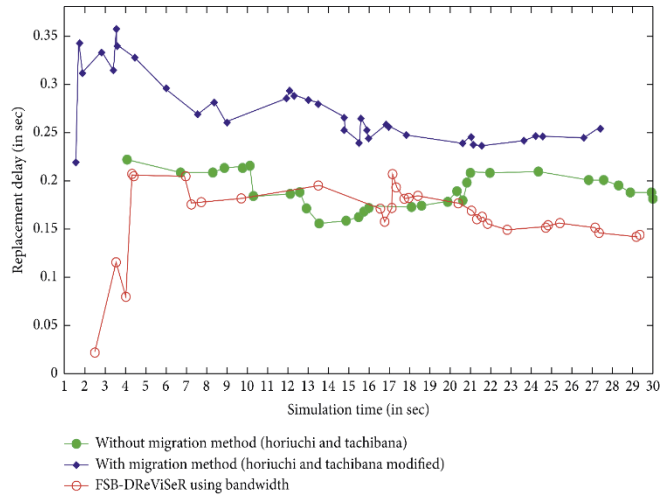


Fig 10: Replacement delays for a 20-node network.

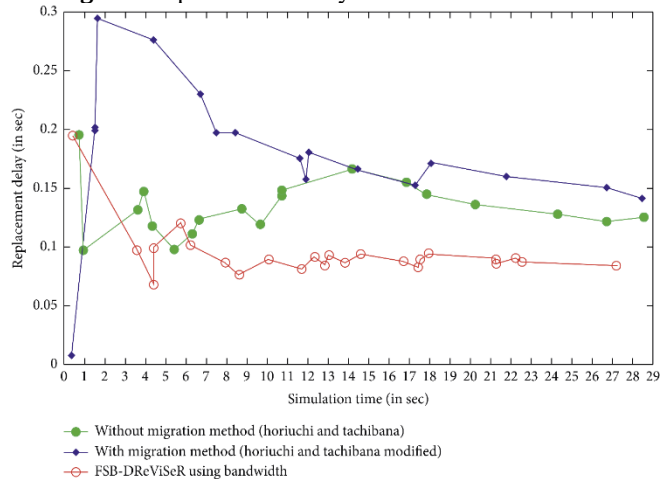


Fig 11: Latency in replacing 60 nodes in a network

When compared to Horiuchi and Tachibana's replacement technique, which relies on unused bandwidth, ours (bandwidth-using FSB-DReViSeR) achieves better throughput in the small network and has shorter replacement delays. We account for this success by routing the replacement traffic through the most efficient paths. Although the approach in [8] often results in lower replacement delays, there are a few occasions when it does not, such as between the $t = 11$ s and $t = 16$ s simulation periods in network2. Since the single-path replacement traffic management approach described by Horiuchi and Tachibana does not need segregation, this finding follows logically.

Compared to two tree topology-based implementations of the method in [8], the replacement times produced by our bandwidth FSB-DReViSeR strategy are more fascinating. The time needed to generate a minimal spanning tree from the topology of a graph increases the replacement delay greatly when utilising the Horiuchi and Tachibana method. The single-route solution proposed in [8] lengthens the time needed to totally replace services

5.1 Flow Splitting's Impact on Service Quality Data Flow Rate-Dependent

Our FSB-DReViSeR method, which takes into account link throughput (FSB-DReViSeR using flow data rate), is compared to a method that only takes into account bandwidth (FSB-DReViSeR using bandwidth) to determine the extent of the effect. We incorporated persistent random oscillations into the data flow (0, 1) to more closely approximate natural phenomena. The number of replacements was used to calculate average migration durations in order to establish which strategy provides the best quality of service even while under high demand. FSB-DReViSeR provides fewer replacement delays in a congested network (Figure 12) because it prioritises bandwidth. When working with a large network, the difference between FSB-DReViSeR and a protocol that prioritises throughput is negligible (see Figure 13). This may be because, as traffic density rises, the connection throughput on big networks approaches the link bandwidth. Therefore, after determining the gaps between and, we often get solutions that are quite near to the bandwidth. At high network sizes (more than 60 nodes), this finding implies that the performances of bandwidth-focused FSB-DReViSeR and throughput-focused FSB-DReViSeR are comparable.

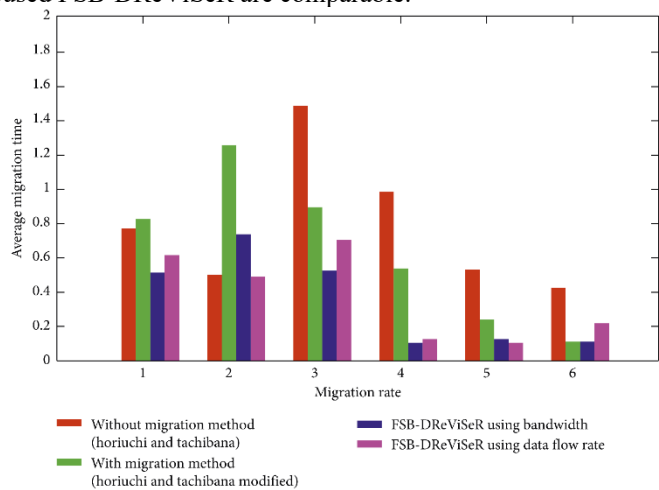


Fig 12 Time spent migrating as a percentage of the overall migration rate for a network of 20 nodes.

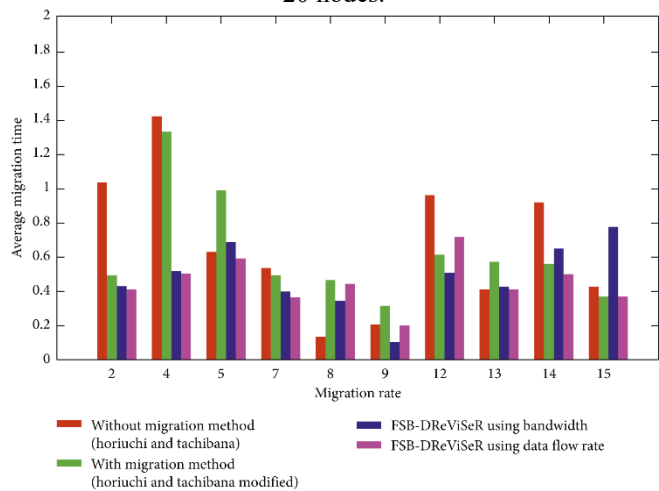


Fig 13: How much of a 60-node network's time is devoted to migration.

5.2 How Flow Separation Affects Post-Upgrade Service Stability

In the second part, we discussed how flow splitting strategies might make it challenging to reassemble the flow after it reaches its final destination. For this reason, it is preferable to maintain a modest splitting rate. Here, we assess how well our method works to maintain service availability after virtual service resources have been swapped out. In order to have a better grasp of the problem, we compare the rate of data transmission loss to the number of backups. In an effort to minimise the number of branching paths, we settled on a two-way flow split. The findings are summarised in Figures 14 and 15. Figures 14 and 15 show that the two variants of the Horiuchi and Tachibana method reduce the rate of data loss regardless of the size of the network. This is a direct consequence of the flow splitting technique included into our migration procedure. However, when the number of resource replacements is vital in a wide network, our method minimises the packet loss rate. In cases when equivalence points do exist, our throughput-focused FSB-DReViSeR technique nonetheless often results in better replacement delays.

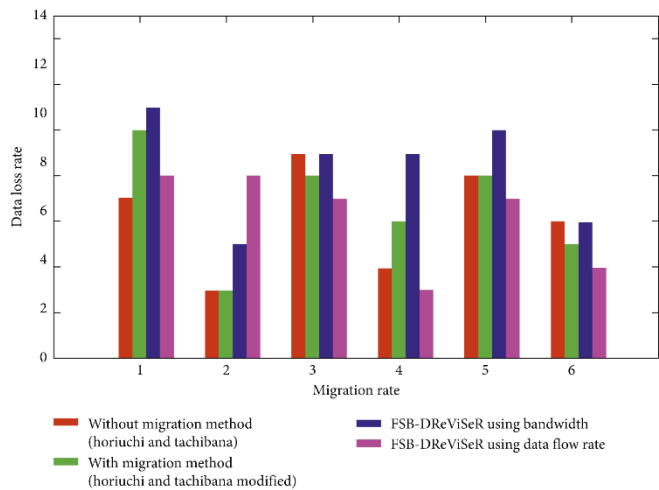


Fig 14: Data loss rate versus migration rate for a 20-node network.

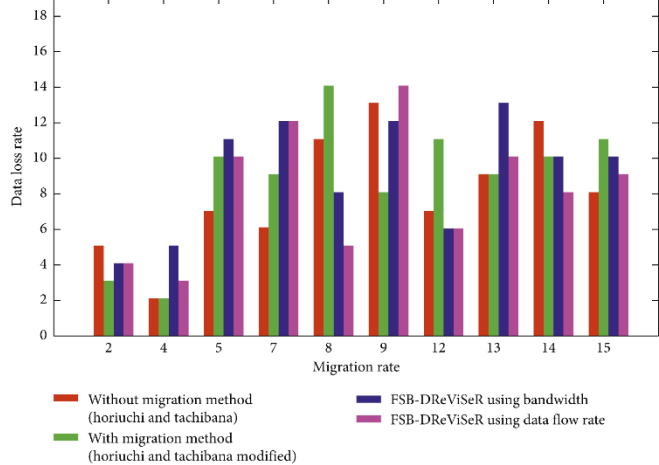


Fig 15: Data loss and network migration in a 60-node system, graphically shown. According to the reported data, when the network size and the number of replacements are both big, our flow splitting-based dynamic virtual service resource replacement technique

outperforms the approach in [8]. In addition, FSB-DReViSeR performs better than the other suggested approaches since it considers throughput while choosing which migration route to adopt.

6 Conclusion

The former uses bandwidth and network link length as key criteria to select the fastest replacement paths. The throughput-focused strategy was shown to be more trustworthy and dependable in simulations of both small and big networks. When it comes to satisfying QoS criteria in a graph network, our replacement technique is superior to that of Horiuchi and Tachibana, among others. What happens when a backup server isn't available to replenish the resources of a virtualized service in a specific logical cloud is a topic for further study. In conclusion, the research on "Telecom Operators in Virtual Heterogeneous Networks May Benefit from Dynamic Resource Replacement for Virtual Services Based on Flow Splitting: FSB-DReViSeR" has highlighted the potential advantages of employing dynamic resource replacement techniques in virtual networks to enhance the performance and efficiency of telecom operators. The study explored the concept of Flow Splitting (FS) and Dynamic Resource Replacement (DReViSeR) as a means to optimize virtual service provisioning in heterogeneous networks. By intelligently distributing traffic flows across multiple network paths and dynamically reallocating resources based on demand, the proposed FSB-DReViSeR framework demonstrated promising results in terms of improved network utilization, reduced latency, and increased overall quality of service. The research findings shed light on the importance of considering virtualization technologies and dynamic resource management strategies in the context of telecom operators operating in heterogeneous network environments. With the ever-growing demand for bandwidth-intensive applications and services, the ability to efficiently allocate and manage network resources becomes crucial for operators to meet customer expectations and maintain a competitive edge. Moreover, the research contributes to the ongoing discussions surrounding the future of telecommunications and the evolution of network architectures. By showcasing the benefits of dynamic resource replacement for virtual services, telecom operators can harness the advantages of virtualization, enabling greater flexibility, scalability, and cost-effectiveness. It is worth noting that while the FSB-DReViSeR framework shows promise, further research and practical implementation are necessary to validate its effectiveness in real-world scenarios. Additionally, considerations such as network security, traffic management, and interoperability should be taken into account when deploying such dynamic resource allocation mechanisms. In conclusion, the study presents a compelling argument for the integration of Flow Splitting and Dynamic Resource Replacement techniques in virtual heterogeneous networks. By embracing these innovative approaches, telecom operators can optimize their resource utilization, enhance service delivery, and adapt to the evolving demands of the telecommunications landscape.

References

1. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
2. R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: a survey," *IEEE Communications Magazine*, vol. 51, no. 11, pp. 24–31, 2013.

3. M. M. A. Khan, N. Shahriar, R. Ahmed, and R. Boutaba, "Multi-path link embedding for survivability in virtual networks," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 253–266, 2016.
4. H. Jiang, Y. Wang, L. Gong, and Z. Zhu, "Availability-aware survivable virtual network embedding in optical datacenter networks," *Journal of Optical Communications and Networking*, vol. 7, no. 12, pp. 1160–1171, 2015.
5. N. Shahriar, R. Ahmed, A. Khan, S. R. Chowdhury, R. Boutaba, and J. Mitra, "ReNoVatE: recovery from node failure in virtual network embedding," in *Proceedings of the 2016 12th International Conference on Network and Service Management (CNSM)*, pp. 19–27, IEEE, Montreal, Canada, October 2016.
6. N. C. Fernandes, M. D. Moreira, I. M. Moraes et al., "Virtual networks: isolation, performance, and trends," *Annales des Télécommunications*, vol. 66, no. 5-6, pp. 339–355, 2011.
7. I. Lee and K. Lee, "The internet of things (IoT): applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.
8. S. Horiuchi and T. Tachibana, "Dynamic replacement of virtual service resources based on tree topology for mobile users in virtual networks," *Journal of Computers*, vol. 13, no. 12, pp. 1335–1348, 2018.
9. Y. Koyanagi and T. Tachibana, "Dynamic resource allocation based on amount of traffic in virtual networks," Tech. Rep., IEICE, Miyazaki, Japan, 2014, Technical report.
10. L. Tan, Z. Zhu, F. Ge, and N. Xiong, "Utility maximization resource allocation in wireless networks: methods and algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 7, pp. 1018–1034, 2015.
11. D. Drutskey, E. Keller, and J. Rexford, "Scalable network virtualization in software-defined networks," *IEEE Internet Computing*, vol. 17, no. 2, pp. 20–27, 2012.
12. P. Lechowicz, S. Spadaro, R. Rumipamba-Zambrano, K. Walkowiak, and J. Perelló, "Migration planning from elastic optical networks to spectrally-spatially flexible optical networks," in *Proceedings of the 2018 Photonics in Switching and Computing (PSC)*, pp. 1–3, IEEE, Limassol, Cyprus, September 2018.
13. X. Li, L. Tan, and F. Li, "Optimal cloud resource allocation with cost performance tradeoff based on internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6876–6886, 2019.
14. J. F. Myoupo, Y. F. Yankam, and V. K. Tchendji, "On the dynamic replacement of virtual service resources for mobile users in virtual networks," *Journal of Computers*, vol. 15, no. 1, pp. 10–21, 2020.
15. J. Liu, Y. Li, and D. Jin, "SDN-based live VM migration across datacenters," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 583–584, 2015.
16. F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: from concept to implementation," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014.
17. V. Kherbache, "Ordonnancement des migrations à chaud de machines virtuelles," Université Côte d'Azur, Nice, France, 2016, Doctoral dissertation.
18. V. K. Tchendji, Y. F. Yankam, and J. F. Myoupo, "Conflict-free rerouting scheme through flow splitting for virtual networks using switches," *Journal of Internet Services and Applications*, vol. 9, no. 1, p. 13, 2018.

19. J. Leino and J. Virtamo, "Insensitive traffic splitting in data networks," in *Proceedings of the 19th International Teletraffic Congress (ITC19)*, pp. 1355–1364, Beijing, China, August 2005.
20. Y. Luo, B. Zhang, X. Wang, Z. Wang, Y. Sun, and H. Chen, "Live and incremental whole-system migration of virtual machines using block-bitmap," in *Proceedings of the 2008 IEEE International Conference on Cluster Computing*, Tsukuba, Japan, September 2008.
21. E. Keller, R. B. Lee, and J. Rexford, "Accountability in hosted virtual networks," in *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, Barcelona, Spain, August 2009.
22. L. Mekouar, Y. Iraqi, and R. Boutaba, "Incorporating trust in network virtualization," in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology*, Bradford, UK, June 2010.
23. T. S. Pham, J. Lattmann, J. L. Lutton, L. Valeyre, J. Carlier, and D. Nace, "A restoration scheme for virtual networks using switches," in *Proceedings of the 2012 4th International Congress on Ultra-Modern Telecommunications and Control Systems*, pp. 800–805, IEEE, St. Petersburg, Russia, October 2012.
24. M. S. Seddiki, "Allocation dynamique des ressources et gestion de la qualité de service dans la virtualisation des réseaux, Université de Lorraine, Nancy, France, 2015, Doctoral dissertation.
25. J. Veerasamy, S. Venkatesan, and J. C. Shah, "Effect of traffic splitting on link and path restoration planning," in *Proceedings of the 1994 IEEE GLOBECOM. Communications: The Global Bridge*, pp. 1867–1871, San Francisco, CA, USA, 1994.
26. F. Zhang, G. Liu, X. Fu, and R. Yahyapour, "A survey on virtual machine migration: challenges, techniques, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1206–1243, 2018.
27. U.. Keshavamurthy and H. S. Guruprasad, "VM migration: a survey," *Global Journal of Engineering Science and Researches*, vol. 2, pp. 219–224, 2015.
28. C. Clark, K. Fraser, S. Hand et al., "Live migration of virtual machines," in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation*, vol. 2, pp. 273–286, Berkeley, CA, USA, 2005.