Research Article/Araştırma Makalesi

GAZİ
JOURNAL OF ENGINEERING SCIENCES

Impact Analysis of Path Selection Strategies over Blockchain-based Routing in Multi-Domain SDN

Evrim Guler [*a]

Submitted: 08.09.2022 Revised: 26.11.2022 Accepted: 03.11.2022 doi:10.30855/gmbd.0705040

Keywords: QoS, SDN, Blockchain, Inter-Domain, Path Selection

[a,*] Bartin University,
Faculty of Engineering,
Architecture and Design,
Dept. of Computer Engineering
74100 - Bartin, Türkiye
Orcid: 0000-0002-7226-4748
e mail: evrimguler@bartin.edu.tr

[*]Corresponding author:
evrimguler@bartin.edu.tr

ABSTRACT

Over the past decade, one of the most commonly utilized inventions in the realm of computer communications is Software-Defined Networking (SDN). Among many other lauded advantages, the architecture of SDN supported by OpenFlow communication protocol potentially provides an End-to-End (E2E) path throughout multiple networks under the consideration of Quality of Service (QoS) metrics while supporting network administrators to manage the granular flows. We previously introduced QoSChain, which combines the benefits of SDN and blockchain technology to provide QoS for inter-networks. For the aim of enabling policy automation in cross-networks, E2E visibility, assurance, validation, and control, this framework orchestrates a software-driven traffic management system. In this study, our main focus is on the influence of various path computation options on overall QoSChain performance. In particular, we assess the effectiveness of five simple yet effective solutions for path selection: First Feasible Path Selection (FFPS), Random Feasible Path Selection (RFPS), Minimum Hop Path Selection (MHPS), FFPS with the Border Gateway Protocol (BGP) shortest path at inter-network level (FFPS_BGP), and MHPS with the BGP shortest path at inter-network level (MHPS_BGP). Our experimental results indicate that path selection is crucial to overall performance while minimizing hop counts to deliver superior performance at the expense of initially longer setup times.

Çok Alanlı YTA'da Blok Zinciri Tabanlı Yönlendirme Üzerinden Yol Seçim Stratejilerinin Etki Analizi

ÖZ

Son on yılda, Yazılım Tanımlı Ağ Oluşturma (YTA), bilgisayar iletişimi alanında en yaygın kullanılan buluşlardan biri haline geldi. OpenFlow tarafından etkinleştirilen YTA mimarisi, diğer pek çok övgüye değer avantajın yanı sıra, ağ yöneticilerine ağlar arasındaki akışlar için Uçtan Uca (E2E) Hizmet Kalitesi (QoS) garantili yollar sağlama konusunda yardımcı olma potansiyeline sahiptir. Daha önceki çalışmamızda, ağlar arası QoS provizyonu için blok zincir teknolojisi ve YTA'nın faydalarını birleştiren QoSChain'i tanıtmıştık. Bu çerçeve, ağlar arası politika otomasyonu, güvence, E2E görünürlüğü, kontrol ve doğrulamayı etkinleştirmek için yazılım odaklı bir trafik yönetim sistemini düzenler. Bu çalışmada, yol seçim stratejilerinin genel QoSChain performansı üzerindeki etkisine odaklanıyoruz. Spesifik olarak, beş basit ama etkili yol seçim stratejisinin performansını değerlendiriyoruz: İlk Uygun Yol Seçimi (FFPS), Rastgele Uygun Yol Seçimi (RFPS), Minimum Atlama Yolu Seçimi (MHPS), Sınır Ağ Geçidi Protokolü (BGP) ile ağlar arası düzeyde FFPS (FFPS_BGP) ve Sınır Ağ Geçidi Protokolü (BGP) ile ağlar arası düzeyde MHPS (MHPS_BGP). Deneysel sonuçlarımız, başlangıçta daha uzun kurulum süreleri olmasına rağmen üstün performans sağlayan atlama sayısı minimizasyonu ile yol seçiminin genel performans için çok önemli olduğunu göstermektedir.

Anahtar Kelimeler: Servis Kalitesi, Blokzincir, YTA, Çoklu Alan, Yol Seçimi

To cite this article: E. Guler, “Impact Analysis of Path Selection Strategies over Blockchain-based Routing in Multi-Domain SDN,” *Gazi Journal of Engineering Sciences,* vol. 8, no. 3, pp. 524-539, 2022. doi:10.30855/gmbd.0705040

1. Introduction

With the proliferation of various Internet applications (e.g., VoIP, video conferencing, online gaming, etc.), more sophisticated and efficient routing mechanisms are required to meet the QoS demands and requirements of the applications. However, due to different unsolved concerns including the limited global view of network infrastructures, per-hop decisions, and limited Quality of Service (QoS) abilities for network flows, End-to-End (E2E) routing in today's traditional networking is an ossified problem. In current network architecture, Software-Defined Networking (SDN) and OpenFlow protocol offer to promise and at the same time forward-looking solutions for routing problems. The OpenFlow protocol and Software-Defined Networking (SDN) provide a potential and promising solution to figure out the QoS-based E2E routing issues of the existing networking architecture. The SDN decouples control and data planes through the use of a logically centralized controller component, providing several options for routing capabilities and enabling QoS. QoS-based E2E routing per service flow both inter-networks and intra-networks with the help of SDN and OpenFlow becomes more simple, scalable, and time-efficient than traditional network systems [1].

In recent years, a new technology called Blockchain (BC) has emerged, drawing considerable interest from researchers and practitioners, and being recommended for implementation in a variety of application scenarios [2]. Some of these research studies achieve network-infrastructure research by integrating BC infrastructure [3-7]. The authors in [3] provide an Ethereum-based approach to implementing a smart contract for service creation with QoS parameters. The authors of [4] suggest a unique BC-integrated orchestration structure for content dissemination networks to prevent significantly increasing loads. For example, [5] provides a plausible routing strategy for acquiring E2E path information by using blockchain nodes, but in a wireless sensor network, a model powered by reinforcement learning technique is utilized to assist in efficiently selecting routing links connecting dynamically picked routing nodes. The authors of [6] provide a safe BC-enable Border Gateway Protocol (BGP)-based routing technique that maintains a common knowledge of the Internet's routing mechanisms, avoids BGP hijacking, and prevents unauthorized use of BGP. Decentralized route discovery to a gateway or destination device in a delay-tolerant Internet of Things (IoT) network, the study in [7] represents a BC-based contractual routing protocol.

These BC and SDN-focused publications present networking-focused investigations, however, they do not examine the implications of path selection algorithms in their recommendations. In this study, we investigate the performance of our previously suggested BC-enhanced QoS-based inter-domain routing system, which is distinct from the works described above, using various path selection algorithms.

In our previous work, by incorporating BC technology into SDN networks, we introduced a revolutionary QoS-enabled inter-network routing system., namely *QoSChain* [8]. We explore extensively in this work a topic that was left unaddressed in the original work as the impact of several path selection techniques on the QoSChain's overall performance. Our recent work [9] represented several fundamental path selection strategies that are called *First Feasible Path Selection (FFPS)*, *Random Feasible Path Selection (RFPS)*, and *Minimum Hop Path Selection (MHPS)*. In this paper, we define new path selection strategies, namely, *FFPS with Border Gateway Protocol (BGP) at the inter-network level (FFPS_BGP)*, and *MHPS with BGP at the inter-network level (MHPS_BGP)* to compare the performance of required time to set up a flow, the number of exchanging and processing messages, the bandwidth-hop count product, and a composite metric of bandwidth and delay of the chosen paths are used in turn to approximate the network resource consumption. In the experimental results, we indicate that MHPS achieves better performance than the other approaches while having slower initial connection request acceptance in various network topologies.

In the rest of the paper, we provide SDN and BC background in Section 2. The literature review of blockchain-enabled routing is introduced in Section 3. Path selection tactics, the workflow of the path selection framework, and a BC-enhanced, QoS-aware cross-domain routing framework are all investigated in Section 4. Sections 5 and 6 explain the experimental results of our study and finalize the paper, respectively.

2. Architecture of Blockchain Enhanced Software Defined Networking

The Software Defined Network (SDN) consists of data, control, and application planes as shown in Figure 1.

The data plane, which makes up most of the bottom plane, comprises various network components, including virtual and physical switches and routers, access points, and so on. Through Controller-Data Plane Interfaces, SDN controllers can interact with and control these devices (C-DPIs). The OpenFlow communication protocol [10] is extensively used C-DPI standard to support the interaction between data plane devices and controllers, and packet forwarding is a crucial and basic data plane function.

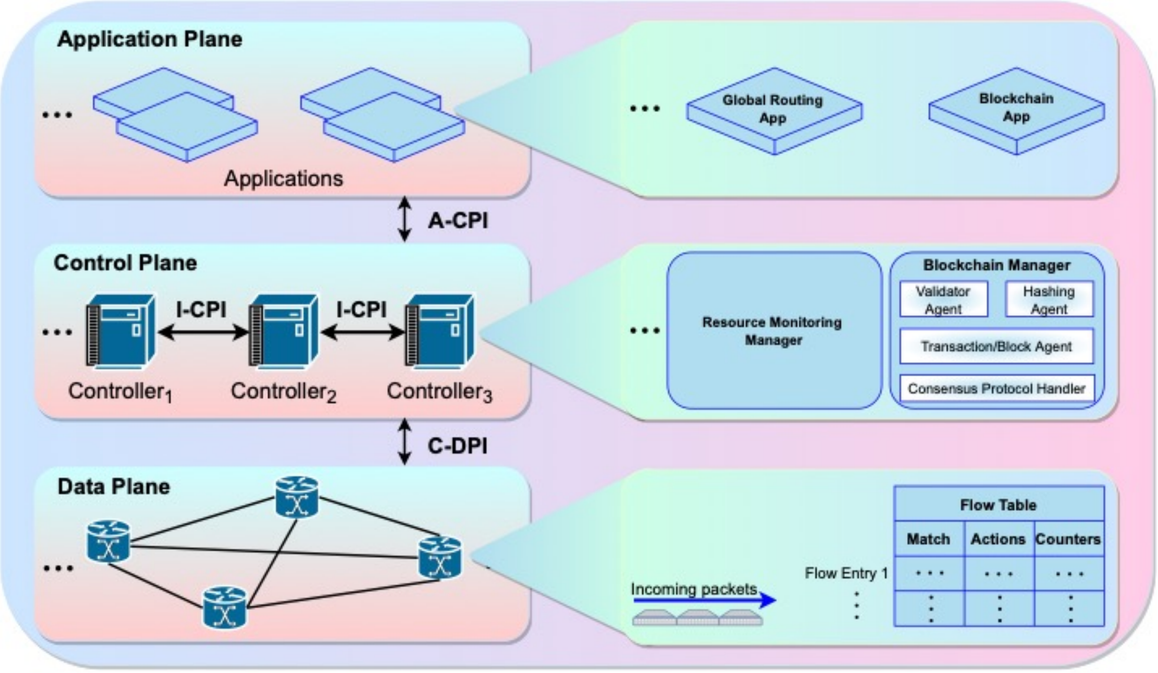


Figure 1. SDN architecture

An SDN controller's middle plane is mainly composed of one or more software-based SDN controllers, depending on the architecture, that offers control capabilities by using a C-DPI to observe network forwarding behavior. The control plane includes the interfaces between controllers in the plane (i.e., the Application Controller Plane Interface, or A-CPI) as well as the interfaces between controllers and network devices (i.e., the Intermediate-Controller Plane Interface, or I-CPI). The I-CPI is intended to transfer information between controllers but is not standardized. Interaction between the controller(s) and network applications is made possible by an A-CPI for network management, security, and other reasons (or services). To control controller behavior, controllers have a variety of functional parts (e.g., a topology manager, a virtualizer, etc.).

The application plane, which is made up of network applications, is the top plane of an SDN. These applications interact with controller(s) via an open A-CPI and use an abstract view of the network to make decisions to perform specific network functions (e.g., REST API).

A data plane router or switch that supports OpenFlow [11] provides the process of transmitting network packets by considering user-defined flow entries in a number of flow tables. Each flow entry in the table is consisting of *Counters*, *Actions*, and *Match* that are used in TCP/IP to establish the flow entry's primary focus on a specific packet header, to apply entities required in the Match field to a packet, and to keep specified information (such as packets, flows, networks, etc.).

Figure 2 indicates the architecture of a block and blockchain. The structure of a block is made up of a list of transactions (T_x) in a block body and a block header to specify various data for the block. The block header may include various transaction details such as a timestamp, an identification number, a difficulty variable, the encrypted value of a transaction in a Merkle tree, the hashed value of the previous/parent block, and a changing variable in each computation, depending on the blockchain use cases and consensus protocols.

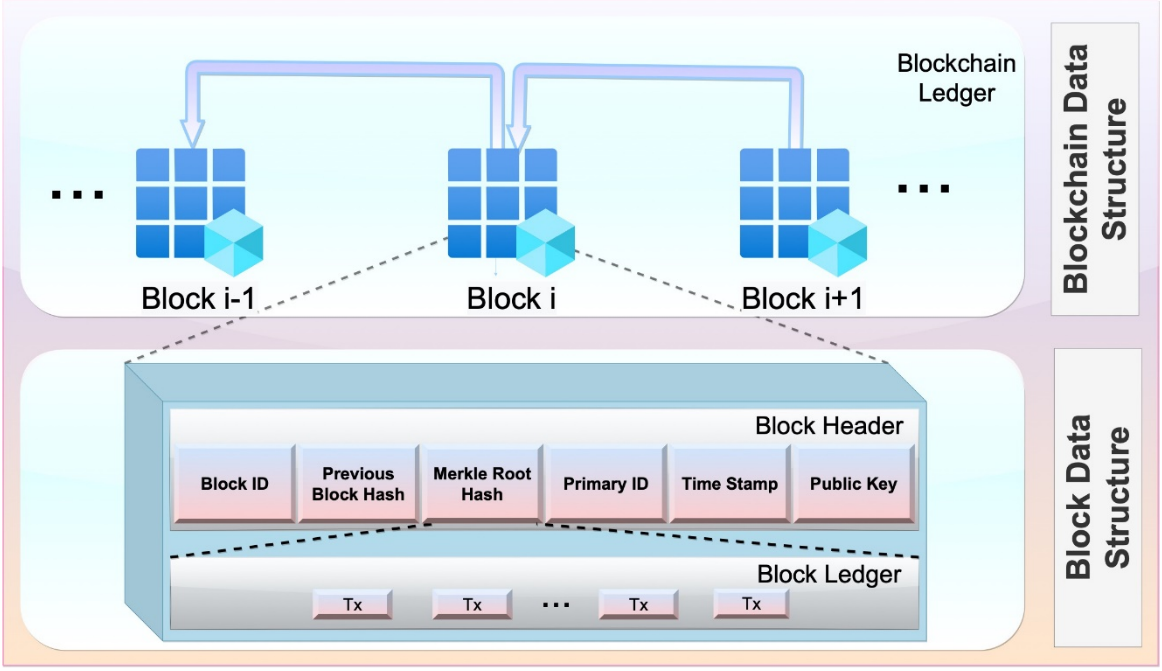


Figure 2. Blockchain and block data structures

Each user has a set of private and public keys that they can use to conduct transactions on the network. To gain access to the network, users must authenticate using private keys. Each user examines the published transactions on the blockchain and discards any blocks that include incorrect transactions. The hashed value of the previous block needs to be checked until reaching the *Genesis Block* as the first block. Any member attempting to alter data in their network is not permitted to edit earlier blocks. As a result, a blockchain's structure is impenetrable, and All nodes on the network must agree on a consensus protocol (e.g., Proof-of-Work) with various features ensuring the integrity of data in order to add a new block to the chain. [12].

3. Related Work

In the literature, there are numerous studies that introduce BC applications that are used in a variety of fields (e.g., IoT, cloud computing, supply chains, and healthcare systems). For this purpose, various research studies focus on BC-based routing frameworks in the literature [3-7, 13, 14] whereas none of these studies considers QoS-based E2E path selection framework over multi-domain SDN ISPs. There is a close study in [15] that proposes a multi-domain latency-aware routing scheme in SDN networks. Every pair of connected ISPs' round-trip times is periodically measured by the suggested architecture, which then stores the results in a distributed, decentralized BC network. The BC network's latency measurement data is processed by SDN controllers, who also verify data integrity and oversee latency-aware routing for real-time data flows. On the other hand, except in our previous works [8, 16] that are used in this research as a main part of the routing framework, QoS-aware E2E path determination over multi-domain SDN ISPs is not well-studied, especially path selection efficacy on the utilization of network resources.

The authors in [17] introduce SDN-enabled networking architecture with blockchain technology while integrating the security and autonomy management layers to advance multi-layer communication in SDN networks. The authors in [18] develop a framework that consists of trust and verifying QoS compliance for E2E routing over multi-domain SDNs. To store and exchange the various types of trust data needed to provision and validate E2E QoS compliance of the domains, TRAQR effectively takes advantage of blockchain specifications such as the tamper-proof and decentralized infrastructure. In [19], the authors employ a cross-domain routing framework to implement the trusted relationship for various SDN controllers in a multi-domain network. The authors of [15] propose SDN instances processed latency measurement data that are periodically posted and stored in blocks for validating the integrity of data and managing real-time data flows by considering latency-aware routing in a blockchain network. In [20], controller(s) create blocks including flow rules-based transactions for incoming flows, and send them to all switches under its control after validating based on the network topological view. The authors of [21] propose a blockchain-enable infrastructure to transmit correlated flows between social network users while focusing on minimal link overlapping in a set of paths by using a topology manager module, flow association, and path selection modules over SDN controllers.

4. BC-Enhanced QoS-Aware Inter-Network Routing Framework with Various Path Selection Strategies

This study analyzes the implications of alternative path selection strategies on the underlying network by leveraging and enhancing our blockchain-enhanced Quality of Service (QoS)-aware SDN-based inter-network routing framework proposed in our earlier study [8]. In this section, we introduce a recap of the routing architecture, an explanation of the various path selection algorithms employed, and the overall framework's workflow to accomplish this objective and make the investigation self-contained and reader-friendly.

The Internet is made up of interconnected entities that send data from the origin to a target. They connect geographically separated communication devices and networks whose IP prefixes are given to an Internet Service Provider (ISP) with predetermined routing policies. The organization known as the Internet Assigned Numbers Authority (IANA) assigns a specific ISP Number to each ISP on the Internet, which is used in inter-ISP routing, also known as inter-domain routing, as an identity [22]. An ISP is in charge of data transmission between its networks and networks hosted by neighboring ISPs. To that end, an ISP uses the Interior Gateway Protocol (IGP) to connect its inner devices to its domains and the Exterior Gateway Protocol (EGP) to connect to nodes in neighboring ISPs. Inter-ISP relationships are formed between ISPs and can be peering or customer-provider interaction.

4.1. BC-Enhanced QoS-Aware inter-network routing framework

An illustrated network architecture is shown in Figure 3 that consists of five SDN-based inter-networks or ISPs and a blockchain network between the network controllers is used to keep track of the network's status for E2E routing regarding transactions and created blocks. In Figure 3, cylindrical objects with interconnecting links (i.e., thick black solid lines) represent border nodes of the networks, and through various networks, hexagonal objects without interconnection links show the core network devices.

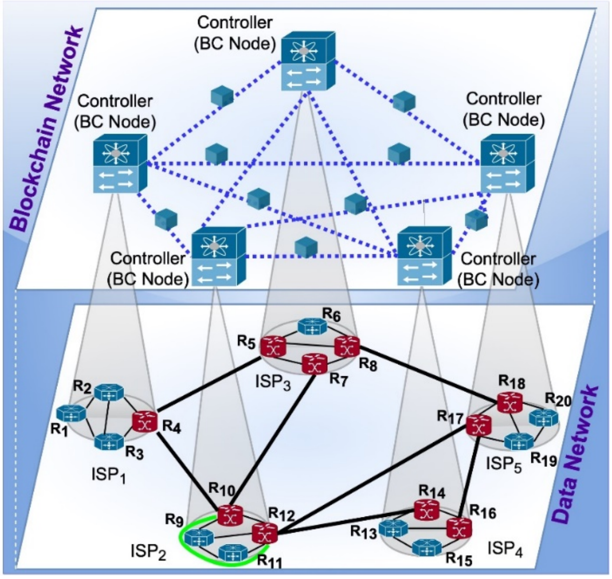


Figure 3. A blockchain-enabled SDN model

Blockchain-Enhanced SDN Controller: For blockchain capabilities in the SDN controller utilized in the routing architecture, Figure 1 depicts the new controller modules, as well as the existing ones and network applications that have been implemented. The Blockchain Manager (BM) module in a network controller is in charge of all blockchain-related operations. Based on the blockchain's block validation rules, the *Validator Agent* is to validate the blocks that are incoming from other controllers. Before being sent to the blockchain network, transactions and blocks are hashed by the *Hashing Agent* module. The implementation of the transactions and/or blocks that make up the blockchain network, as well as the blockchain network's consensus algorithm, is handled by the *Transaction/Block Agent* and *Consensus Protocol Handler*, respectively. It is the responsibility of the Resource Monitoring Manager (RMM) to keep an eye on network resources like bandwidth and delay and to alert the BM module to set up the appropriate transaction(s) when anything changes. Global Routing App (GRA) is in charge of putting inter-network routing functionalities into place when the controller receives an inter-network service request. The Blockchain Application (BA) module assists with handling service request messages as well as transferring blocks between the blockchain network and its clients.

Pathlet: A pathlet is a section of a clear path connecting two border node pairs at the *Ingress* and *Egress* points. The pathlet end-points are *Ingress* and *Egress* nodes that are in the same network. For instance, the pathlet between the two border nodes is shown as a partial green pathlet in Figure 3.

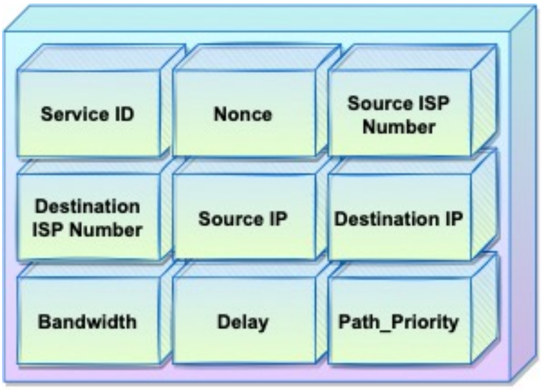


Figure 4. Data structure of Service Request (SR)

Service Request: According to the routing framework, a *Service Request (SR)* is a request for the provisioning of connectivity between users (i.e., computers) on the same or different networks using specific QoS parameters, such as bandwidth and delay. Theoretically, users may ask for any rate of service (bandwidth and/or delay), and a continuous-rate network must be able to support arbitrary (service) requests. The *SR* data structures in the framework are shown in Figure 4. An *SR* message contains the following information:

- Service ID: The persistent service identifier for a service.
- Nonce: Randomly generated distinct *Service Request ID*.
- Source and Destination ISP Number: ISP numbers of the source/destination ISPs, respectively.
- Source and Destination IP: The source and destination computers' IP addresses, respectively.
- Bandwidth: The bandwidth demand of a service request over the E2E path.
- Delay: The acceptable delay for a service request over the E2E path.
- Path_Priority: The prioritized parameters of the E2E path regarding QoS parameters to indicate path selection preference order (i.e., *FFPS*, *RFPS*, *MHPS*, *FFPS_BGP*, and *MHPS_BGP*), whose details are provided in Section 4.2.

The SR message is generated by a relevant program on a user's computer and is then forwarded to the relevant (source) network controller.

On the blockchain network, using cryptographic processes with a set of public and private keys, each node on the network runs its own blockchain instance and correlates to a network controller in the routing architecture. In this article, the terms *blockchain node* and *ISP controller* will be used interchangeably. An IP address and a pair of public and private keys are both present on a blockchain node for cryptographic operations. Peer-to-Peer (P2P) full-mesh networking is used by blockchain nodes to communicate over the Internet. To establish a peering relationship, ISP controllers trade their public keys and bind them to distinctive identifiers of blockchain nodes (i.e., ISP Number). The controllers broadcast their public keys together with their digitally signed networking data and ask other nodes for the same info (e.g., IP addresses, ISP numbers, lists of border nodes, etc*.). Keep-alive* messages are used by blockchain nodes to manage whether or not their peers are still alive. The other peer ends the peering connection with the blockchain node if a peer does not respond.

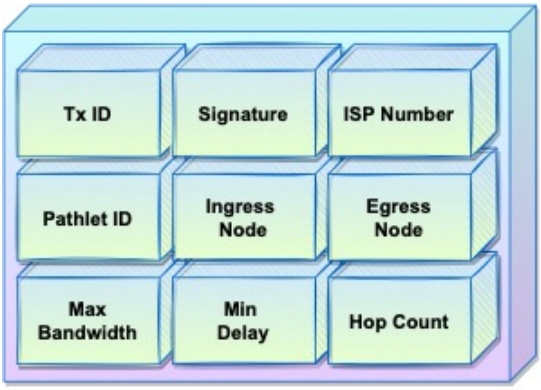


Figure 5. The data structure of a transaction in BC-enabled QoS-aware inter-network routing

Transaction: In the routing framework, blockchain nodes generate transactions from pathlets and their QoS values (i.e., controllers). Network controllers generate unique pathlets for each pair of border nodes in their respective networks in order to add them to the blockchain ledger. Figure 5 demonstrates a transaction data structure within the routing framework. A transaction contains the following information:

- Tx ID: The transaction's distinct ID
- Signature: The blockchain node (i.e., ISP controller) that created the transaction's digital signature by utilizing its private key.
- ISP Number: The unique ISP number, which is used as an identifier in inter-ISP routing.
- Pathlet ID: The unique ID of a distinct pathlet.
- Ingress and Egress Node: The ending points (i.e., start and end node) of a pathlet in an ISP, respectively. Each ISP will share its border node IDs with the other ISPs participating in the blockchain in advance.
- Max Bandwidth and Min Delay: The pathlet's satisfying maximum bandwidth and providing minimum delay.
- Hop Count: The number of hops in a corresponding pathlet in an ISP.

A blockchain sends a transaction to a node connected to the blockchain network, which verifies the transaction's validity. Transactions that are invalid are discarded. Other connected nodes receive valid transactions that were previously unknown to the node. The transactions will eventually reach every node in the network after these further validate them and send them to their peers. The transaction data validation rules determine which data is required to represent a transaction. Each blockchain node in the framework needs to validate transactions using a set of rules that ensure: (i) the transactions are required to be digitally signed, (ii) QoS-related bandwidth and delay fields of a transaction are positive values, (iii) ISP number needs to be a valid number and available, (iv) ingress and egress node IDs in the transaction belong to the network, where the transaction is generated.

Table I shows transactions created by ISP2 controller for pathlets between R10 and R12 border devices in ISP2 network shown in Figure 3. The table shows transactions only for pathlets among R10 and R12. The transaction with ISP2_6 ID is created for the interconnecting link from R10 to R4 in ISP2 and ISP1. When an ISP joins the blockchain network, it begins by propagating the first transactions to the appropriate blockchain nodes for the pathlets among the border network devices shown in the table. The controller produces a new transaction (referred to as an *update transaction*) reflecting the state change on the pathlet whenever there is a QoS-related network state change, such as a bandwidth update in a link. To illustrate how to create a transaction in Table I, once ISP2 participates in the blockchain network, the ISP begins to create its initial transactions that hold corresponding data for the distinct pathlets among the border devices R10 and R12. The initial transactions with IDs ISP2_1, ISP2_2, and ISP2_3 are for different pathlets having their IDs {R10_R12_1 (R10-R12), R10_R12_2 (R10-R9-R12), and R10_R12_3 (R10-R9-R11-R12) between R10 and R12, respectively. Only the appropriate owner controller is able to reach the pathlet's details (i.e., complete network device list across a pathlet as ISP2 controller in this case). The controller initiates update (new) transactions for the pathlets when a bandwidth update of a link occurs in the network, assuming that the bandwidth increased from 5 Mbps to 15 Mbps in the connection between R10 and R9. As seen in the table, the transactions with IDs ISP2_4 and ISP2_5 are the newly updated transactions for previous transactions with IDs ISP2_2 and ISP2_3 while reflecting the updated available bandwidth for the pathlets with IDs R10_R12_2 and R10_R12_3 as 20 Mbps, respectively. Therefore, they have the same pathlet IDs.

Table 1. Transactions generated by ISP2 controller for pathlets between R10 and R12 border devices in ISP2. Transaction with ISP2_6 ID is created for the inter-connecting link from R10 to R4 in ISP2 and ISP1, respectively. Transaction with ISP2_4 Tx ID is an update for the pathlet with ISP2_2 Tx ID and R10_R12_2 Pathlet ID. Similarly, the transaction ISP2_5 Tx ID is an update for the pathlet with ISP2_3 Tx ID.

Tx ID	Signature	ISP Number	Pathlet ID	Ingress Node	Egress Node	Max Bandwidth	Min Delay	Hop Count
ISP2_1	0000kbxfg...	ISP2	R10_R12_1	R10	R12	15	5	1
ISP2_2	0000asx34...	ISP2	R10_R12_2	R10	R12	5	12	2
ISP2_3	0000fdxr4...	ISP2	R10_R12_3	R10	R12	10	15	3
ISP2_4	0000ytx6j...	ISP2	R10_R12_2	R10	R12	20	12	2
ISP2_5	0000erfg4...	ISP2	R10_R12_3	R10	R12	20	15	3
ISP2_6	0000uprth...	ISP2	R10_R4_1	R10	R4	40	30	5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Block: Figure 2 depicts the data structures of a block employed in this study's routing scheme. A framework block is comprised of two components as the block header and transactions. Each block header consists of Block ID, Previous Block Hash, Merkle Root Hash, Primary ID of the controller that creates a block in a time interval of block generation, Timestamp, and Public Key of the block-generator controller as follows:

- Block ID: The distinct number of a block.
- Previous Block Hash: The hashed value of the previous block.
- Merkle Root Hash: A data structure where each transaction's hash is merged with all the others to create a single root hash.
- Primary ID: In a block generation interval (epoch), the unique ID of the major blockchain node (ISP controller).
- Timestamp: The time point that the block was released.
- Public Key: The public key of the controller (i.e., the BC node) creating the block.

These data columns can be changed based on the blockchain use case and consensus process. At each epoch, the primary's responsibilities are passed from the list of nodes to the next node. The principal node creates a new block by following the steps outlined below:

- Its transaction pool is where all new transactions are collected.
- The invalid transactions are rejected according to the validation rules of the transaction.
- If time limitations are available, it checks that block creation restrictions are being followed.
- A block is created that includes all valid transactions and is signed with the primary node's private key.
- The other blockchain nodes receive the newly generated block through propagation.

Once the other nodes receive the new block broadcast by the primary, they validate it as follows:
1. After receiving the block, they check that:

- The block was created through a current epoch's primary node that does not produce any other blocks.
- Block is properly created and signed.

2. The transaction validation rules are checked, and the block is created within the block generation limits.
3. If block verification is successful, the new block will be added to the node's blockchain.
4. If block validation was unsuccessful, the block is rejected and sent a bad block transaction.

The blockchain node that produced the faulty block may be blocked or removed from the list of peering nodes if it continues to produce similar blocks.

4.2. Routing framework path selection strategies

In this subsection, we explain the path selection techniques implemented in this research. To determine the QoS-based E2E path of an *SR*, the proposed blockchain-enabled inter-ISP routing framework by satisfying QoS requirements scans all available transactions in the blockchain. A network's path selection approach is as important as its path calculation algorithm/protocol for achieving optimal network resource usage and user quality of experience. For that purpose, we examine and contrast several E2E path selection algorithms, including *First Feasible Path Selection (FFPS)*, *Random Feasible Path Selection (RFPS)*, *Minimum Hop Path Selection (MHPS)* [9], *FFPS with Border Gateway Protocol (BGP) at inter-network level (FFPS_BGP)*, and *MHPS with BGP at inter-network level (MHPS_BGP)*, to investigate the effects on underlying network resources and scalability.

First Feasible Path Selection (FFPS): This strategy by using all available transactions in the blockchain gives preference to choose the first possible E2E path computed at time t to fulfill the *SR*. If we define $P_{E2E}^{s-d,t} = \{P_i^{s-d,t}, 1 \le i \le n\}$ as the set of E2E pathways from s to d that fulfill the *SR* at time t as $(P_i^{s-d,t})$, where n is the number of feasible paths and $P_{E2E}^{s-d,t}$ is never equal to zero, then the strategy will choose the path that corresponds to the first computed feasible path $(P_1^{s-d,t} \in P_{E2E}^{s-d,t})$.

Random Feasible Path Selection (RFPS): This technique grants preference, when using all of the transactions that are now accessible in the blockchain, to the path that is randomly chosen from among all of the possible E2E paths that have been computed for the *SR* at time t. When we define $P_{E2E}^{s-d,t} = \{P_i^{s-d,t}, 1 \le i \le n\}$ as the set of E2E pathways from s to d that fulfill the *SR* at time t as $(P_i^{s-d,t})$, where n is the number of feasible paths and $P_{E2E}^{s-d,t}$ is never equal to zero, then the strategy will randomly pick a path $(P_i^{s-d,t} \in P_{E2E}^{s-d,t}, 1 \le i \le n)$ among all available E2E paths.

Minimum Hop Path Selection (MHPS): Using all of the transactions that are currently available in the blockchain, this selection method determines the path that will result in the fewest hops taken out of all of the possible E2E routes that lead from the source to the destination. If we define $P_{E2E}^{s-d,t} = \{P_i^{s-d,t}, 1 \le i \le n\}$ as the set of E2E pathways from s to d that fulfill the *SR* at time t as $(P_i^{s-d,t})$, where n is the number of feasible paths and $P_{E2E}^{s-d,t}$ is never equal to zero, then the strategy picks the E2E path (i.e., $min_{\forall i}\, L(P_i^{s-d,t})$) that has the minimum number of hops among all feasible E2E paths. $L(P_i^{s-d,t})$ is the length (i.e., number of hops) of an E2E path and can be defined as:

$$L\left(P_i^{s-d,t}\right) = 1 + \sum\nolimits_{\forall e_j^{s-d,t} \in P_i^{s-d,t}} 1\,, 1 \le i,j \le n \qquad (1)$$

where $e_j^{s-d,t}$ is a link over the E2E path $P_i^{s-d,t}$.

FFPS with Border Gateway Protocol (BGP) at inter-network level (FFPS_BGP): This technique provides preference to selecting the BGP shortest path at inter-network level [23], while taking the first available path from intra-network level for E2E path computed to satisfy the *SR* at time t by leveraging all of the accessible transactions in the blockchain. Let $I_{E2E}^{s-d,t} = \left\{I_j^{s-d,t}, 1 \le j \le k,\ j,k \in Z^+\right\}$ and $P_{E2E}^{s-d,t} = \left\{P_i^{s-d,t}, 1 \le i \le n\right\}$ be the sets of E2E shortest inter-network level ISPs (i.e., selected ISPs by BGP-based shortest path) and paths from s to d satisfying the *SR* at time t, respectively, as $(P_i^{s-d,t})$ with n as the feasible paths for each intra-network I_j by using BGP protocol at inter-network level, where $P_{E2E}^{s-d,t} \ne \emptyset$, the strategy will then choose the first feasible path that was calculated $(P_1^{s-d,t} \in P_{E2E}^{s-d,t}$ for each $I_j \in I_{E2E}^{s-d,t})$.

MHPS with BGP at inter-network level (MHPS_BGP): This selection strategy chooses the path that has the minimum hops among all feasible paths in intra-network level by using all available transactions in the blockchain while picking the BGP-based shortest path at inter-network level from source ISP to destination ISP of a *SR*. Let $I_{E2E}^{s-d,t} = \left\{I_j^{s-d,t}, 1 \le j \le k\right\}$ and $P_{E2E}^{s-d,t} = \left\{P_i^{s-d,t}, 1 \le i \le n\right\}$ be the sets of E2E shortest inter-network level ISPs (i.e., selected ISPs by BGP-based shortest path) and paths from s to d satisfying the *SR* at time t, respectively, as $(P_i^{s-d,t})$, where $P_{E2E}^{s-d,t} \ne \emptyset$, then the strategy picks the E2E path (i.e., $min_{\forall i}\, L(P_i^{s-d,t})$) that has the minimum number of hops among all feasible E2E paths for each selected $I_j \in I_{E2E}^{s-d,t}$. $L(P_i^{s-d,t})$ is the length (i.e., number of hops) of an E2E path and can be defined as:

$$L\left(P_i^{s-d,t}\right) = 1 + \sum_{I_p \in I_{E2E}^{s-d,t}} \sum_{\forall e_j^{s-d,t} \in P_i^{s-d,t}} 1\,, \quad 1 \le i,j \le n,\ \ 1 \le p \le k \qquad (2)$$

where $e_j^{s-d,t}$ is a link over the E2E path $P_i^{s-d,t}$.

4.3. Workflow of path selection framework

Figure 6 indicates the process followed by the E2E path selection method that is based on blockchain technology and prioritizes the QoS of an *SR*. In Step 2, after receiving an SR from a user, the network controller calculates an E2E path using its blockchain ledger. Taking into account the QoS parameters and path selection priorities (i.e., Path_Priority) mentioned in the *SR* message, the E2E path is made up of pathlets connecting a source network's edge node to an endpoint on the target network's edge. After calculating available E2E paths, the QoS-based blockchain routing framework rejects the service request by sending a `Reject` message to the user when there is no available E2E path in Step 3. If any E2E path satisfies the required QoS parameters of the *SR*, the controller of the source ISP in Step 4 initiates the process of sending pathlet request messages to every ISP controller that is contributing a pathlet to the calculated E2E path. In Step 5, if all pathlets are successful in satisfying the requirements of QoS, after receiving messages of pathlet requests in Step 4, the framework initiates the process of disseminating all responses to each network controller over an E2E path. In Step 6, the framework gives a service response message that says "Accept" on the basis of the corresponding QoS requirements.

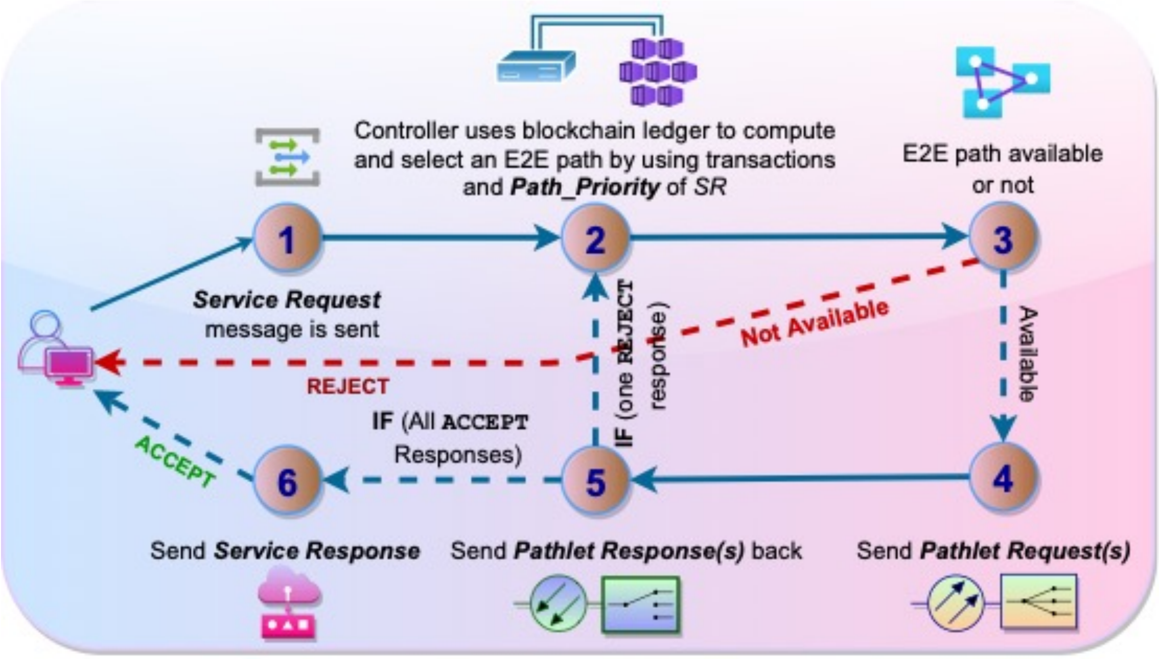

Controller uses blockchain ledger to compute and select an E2E path by using transactions and Path_Priority of SR
E2E path available or not
1
2
3
Service Request message is sent
Not Available
Available
IF (one REJECT response)
REJECT
IF (All ACCEPT Responses)
ACCEPT
6
5
4
Send Service Response
Send Pathlet Response(s) back
Send Pathlet Request(s)

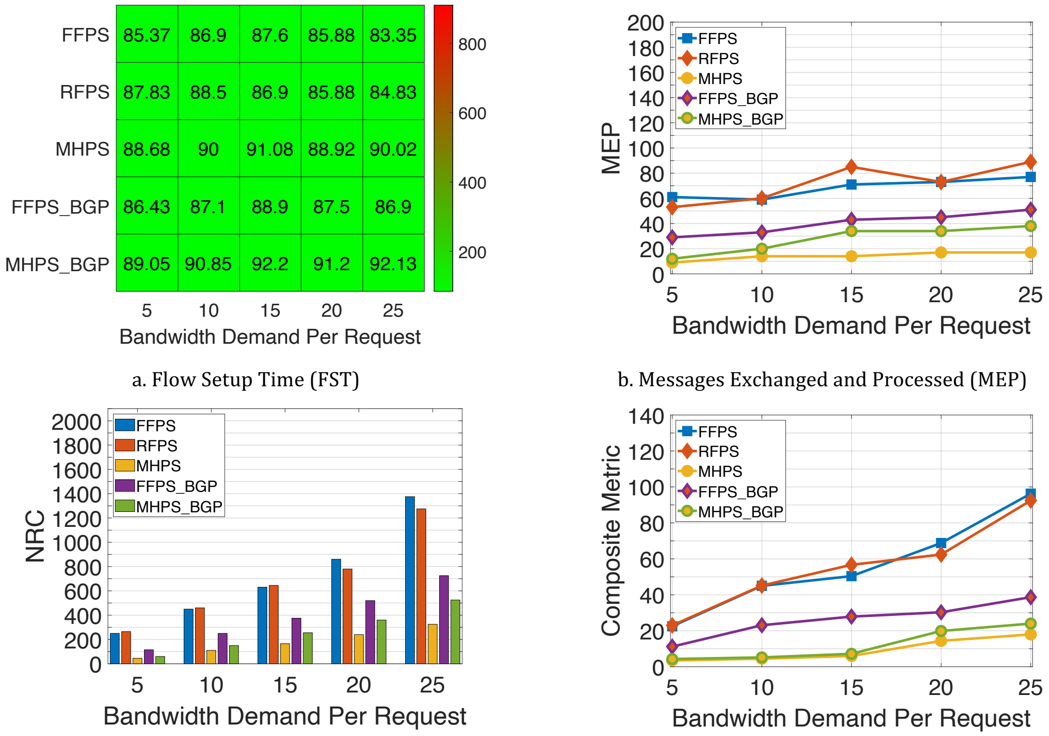
Figure 6. Workflow of inter-network routing with blockchain-enabled procedures

In Step 5, the controller of the source network can start looking for another E2E path that meets the same criteria if any network controllers along the E2E path send the source network controller a `Reject` response. With the help of their intra-network routing, the controllers of the source network and the destination network are responsible for overseeing the segments of the E2E path between the user and the border node of the source network, and between the border node of the destination network and the host at the other end of the path.

5. Experimental Results

In this section, we demonstrate the experimental results of our simulations that indicate the effectiveness and viability of the various QoS-based E2E path [8] finding strategies that were previously introduced, including *FFPS, RFPS, MHPS* [9], *FFPS_BGP*, and *MHPS_BGP*, in terms of utilizing and contrasting required time to compute E2E path as Flow Setup Time (FST), overhead of communication messages for setting up the QoS-based E2E path as the number of Messages Exchanged and Processed (MEP), required network capacity for entire data dissemination from source to destination hosts as Network Resource Consumption (NRC), and the impact analysis of simultaneously network capacity and latency as Composite Metric of a Path (CMP). To assess the effect of network varieties on the inter-network communication, we implement the NSFNET and US Backbone network topologies at the inter-ISP level and vary the number of switches in intra-networks in the range of [5, 10]. All intra-network topologies are generated with the degree connectivity of 0.8 by using Erdos-Renyi. For each service request, we define the request with a bandwidth demand in the range of [5, 25]. The physical links in the intra-networks provide the bandwidth capacity in the range of [5, 55] while supporting the bandwidth capacity for each inter-connecting physical links between the inter-ISPs in the experiments is quite enough (i.e., 10 Gbps) to eliminate ignoring service request rejection due to the limitations of the network resource.

In Figures 7-9 and 8-10, according to our calculations, the average number of switches in a network is 7, and the average amount of bandwidth required for each service request is 10.



a. Flow Setup Time (FST)

b. Messages Exchanged and Processed (MEP)

c. Network Resource Consumption (Bandwidth)

d. Composite Metric (Bandwidth and Delay)

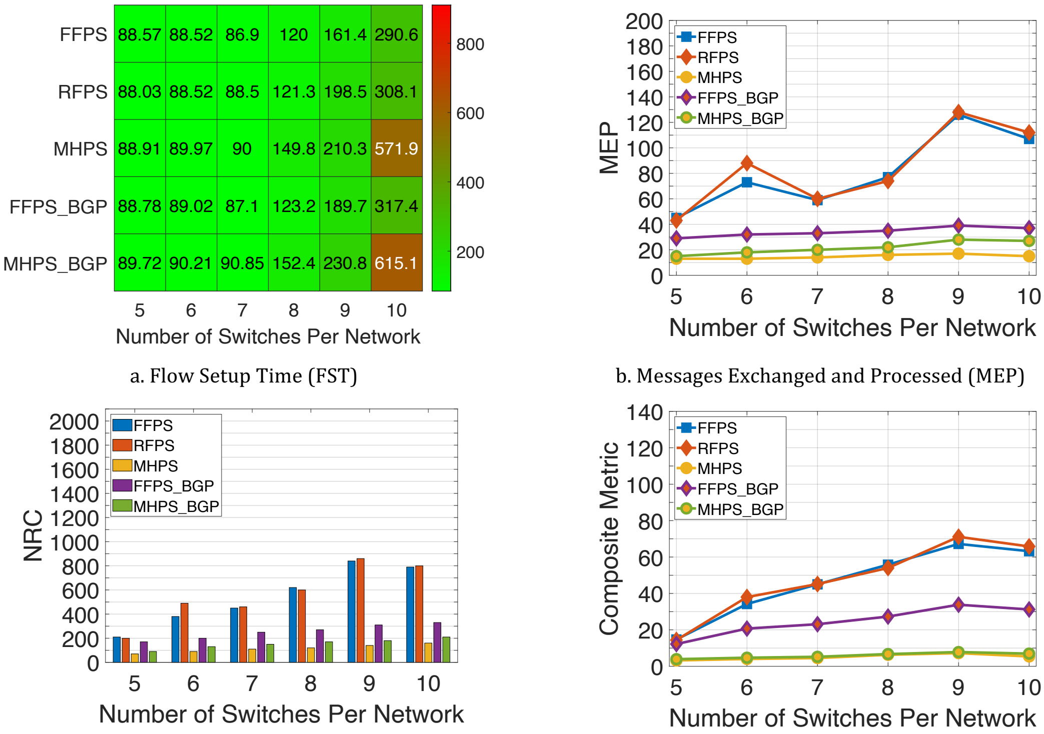
Figure 7. Increasing bandwidth demand in a Service Request (NSFNET ISP Network)

5.1. End-to-end flow setup time

Flow Setup Time (FST) is the amount of time that must pass before the reservation state information may be established along the E2E path that uses QoS. Since it takes into consideration propagation delay, processing time, and path computation, the FST metric is an advantageous statistic that can be used to evaluate the routing and scalability of SDN networks [8].

The impacts of rising bandwidth demand in a service request and the number of switches per network on FST are represented in Figures 7a-9a and 8a-10a, respectively. This is because increasing bandwidth demand restricts the available network resources and reduces the number of possible transactions to choose an E2E path for a request. The proposed path selection strategies have similar flow establishment times in Figure 7a. In contrast, as shown in Figure 8a, a lower total number of switches produces correspondingly lower FST values for networks with fewer than 9 switches. The FST grows larger as the number of switches does, with *MHPS* being the contributor that makes the most significant contribution to this growth. This is because the latter will have to do a lot more work to pick the path with the fewest hops due to having a higher number of available transactions in the blockchain with a more available network resource.

Figure 9a indicates that the proposed *MHPS* and *MHPS_BGP* approaches have higher FST to satisfy a service request. This is because the strategies look for available transactions to pick a smaller number of hops that are used for QoS-based E2E path from source host to destination host whereas the other approaches pick first or randomly available pathlets from available transactions in the blockchain. Similarly, when the number of switches is increasing, the blockchain will have a higher number of available transactions with distinct pathlets. As seen in Figure 10a, the *FFPS, RFPS*, and *FFPS_BGP* strategies outperform the minimum hop path selection strategies because the *MHPS* and *MHPS_BGP* seek a higher number of transactions in the blockchain to find an E2E path with less number of hops.



a. Flow Setup Time (FST)

b. Messages Exchanged and Processed (MEP)

c. Network Resource Consumption (Bandwidth)

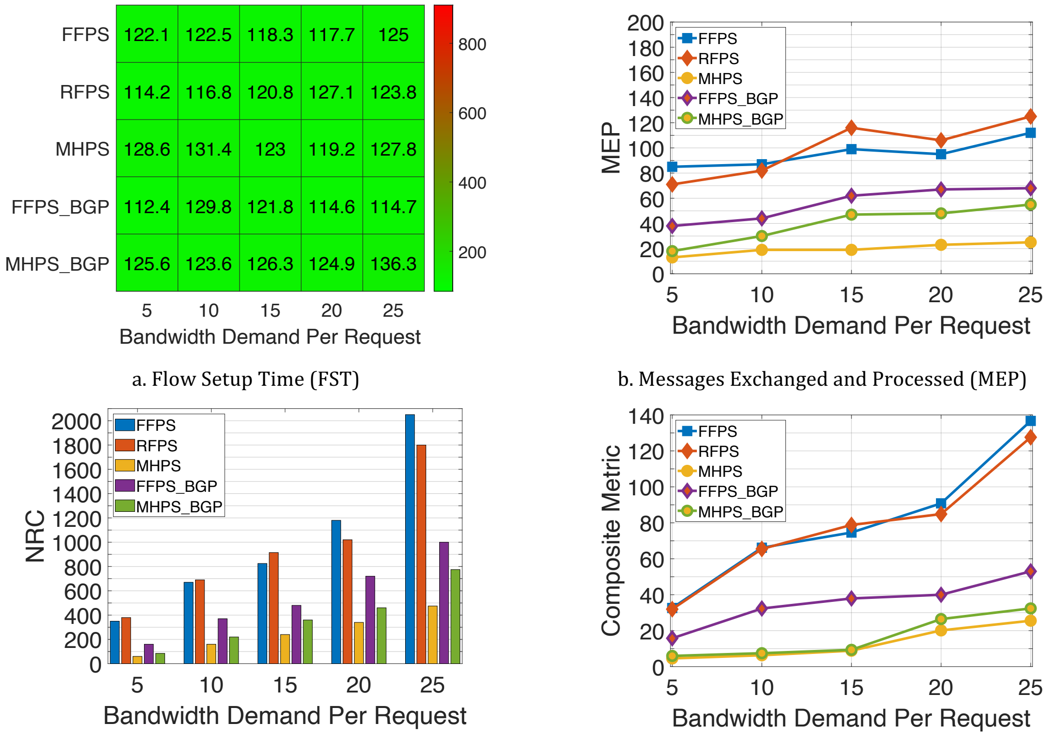
d. Composite Metric (Bandwidth and Delay)

Figure 8. Increasing the number of switch(es) per network (NSFNET ISP Network)

5.2. Messages exchanged and processed

While the network size increases in terms of the number of switches that are used in each network, the network controllers may have to cope with an increase in the number of flow requests and related messages that are processed and sent between network devices (e.g., network devices, hosts, etc.). Additionally, in order to establish a QoS-based E2E link that spans many networks, the controllers communicate with one another and with other controllers that are already present in the network. The controllers of ISP networks may become a bottleneck point due to the limited computational network resources such as CPU and memory as a result of this message exchange and handling operations. In order to provide a QoS-based E2E path for a flow of service requests, controllers make an effort to reduce the total amount of messages that need to be processed and exchanged. As a result, for the purpose of analyzing the effectiveness of routing frameworks in multi-domain SDN networks, the Messages Exchanged and Processed (MEP) is another important indicator to contribute to the overall scalability of a network.

Figures 7b-9b and 8b-10b represent MEP plots for *FFPS, RFPS, MHPS, FFPS_BGP*, and *MHPS_BGP* in a configuration analogous to that used in the FST trials, with variable amounts of bandwidth required per service request and the number of switches used in each network for the NSFNET and US Backbone network topologies, respectively, at the inter-ISP level. As can be seen in the figures, the *MHPS* approach has better performance than all other routing techniques with regard to the MEP measure and particularly as the demand for bandwidth and the number of switches in each network both increase. Furthermore, the *FFPS_BGP* and *MHPS_BGP* have a better performance of MEP than the *FFPS* and *RFPS* strategies because the *FFPS_BGP* and *MHPS_BGP* approaches use the BGP-aware shortest path at network-level. However, the *MHPS_BGP* strategy cannot guarantee that each BGP-based path at network-level has a smaller number of hops at intra-ISP while the *MHPS* approach forces to pick a QoS-based E2E path from the source host to the destination host by using all available transactions. In other words, in contrast to the other routing systems, which do not need sending additional messages to other networks, the *MHPS* strategy takes an active, explicit approach to the consideration of resource availability when determining the optimal number of hops to take in order to build an E2E path. Both *FFPS* and *RFPS* result in more communications since they select the first and the random feasible path, respectively. In a sense, *MHPS* makes up for the initial longer amount of time spent creating the routes by eventually achieving a higher level of effectiveness in terms of the MEP measure.



a. Flow Setup Time (FST)

b. Messages Exchanged and Processed (MEP)

c. Network Resource Consumption (Bandwidth)

d. Composite Metric (Bandwidth and Delay)

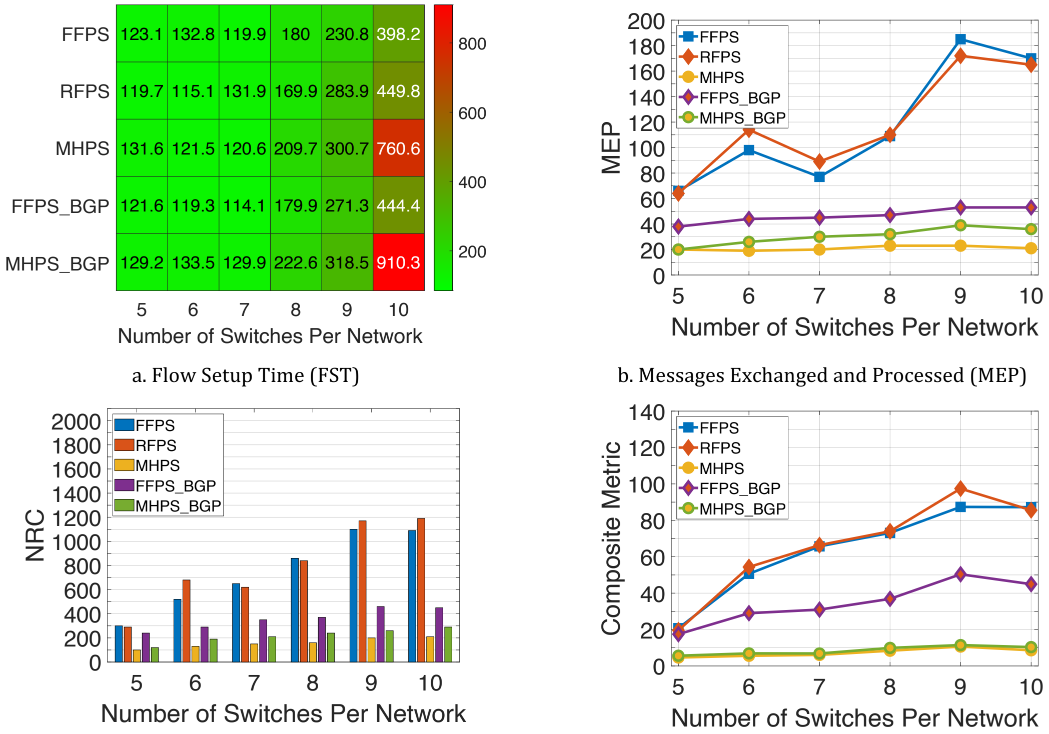
Figure 9. Increasing bandwidth demand in a Service Request (USNET ISP Network)

5.3. Network resource consumption

Using the resources available on the network to set up, administer, and monitor the quality of service for end-to-end path requests is another indication of how the network is being implemented. One of the best ways to address network performance issues and QoS for various services and applications while having limited visibility to establish an E2E path in inter-ISP networks is to monitor network flow (i.e., bandwidth). Therefore, to analyze path selection strategies, we use Network Resource Consumption (NRC), which is another crucial performance metric, through a condensed bandwidth-hop count product as a rough estimate.

Figures 7c-9c and 8c-10c represent the total network resource consumption while varying the requested bandwidth for a service demand and the number of switches for each network with NSFNET and US Backbone inter-ISP network topologies, respectively. As can be seen from the numbers, the *MHPS* has a noticeably reduced NRC while simultaneously building a QoS-based E2E link that spans many networks. This is due to the fact that, as was mentioned earlier, the *MHPS* searches through all of the available transactions to find the one with the lowest hop count when choosing a path for a service request, whereas the other approaches select the first available and randomly appropriate transaction in order to set up the path more quickly. In Figures 7c and 9c, the *MHPS* approach outperforms the other approaches by at least 70% when the bandwidth demand of a service request is higher than 10.

Moreover, the *MHPS* and *MHPS_BGP* are seen to use similar path selection in intra-ISPs whereas the *MHPS* considers picking an E2E path with a smaller number of hops for the entire network by using all available transactions in the blockchain. Similarly, in Figures 8c and 10c, the *MHPS* has better performance than other approaches as much as 50% when there are more than 6 switches in an intra-network.



a. Flow Setup Time (FST)

b. Messages Exchanged and Processed (MEP)

c. Network Resource Consumption (Bandwidth)

d. Composite Metric (Bandwidth and Delay)

Figure 10. Increasing number of switch(es) per network (USNET ISP Network)

5.4. Composite metric of a path

Another possible indicator of NRC is taking into account the amount of delay that the requests go through while the path is being established. In addition, this will serve to contribute to the general improvement of the user experience's quality. The Composite Metric of a Path (CMP) is our suggestion for the combination of the values for bandwidth and delay that are exhibited by the various paths that have been chosen. The CMP is computed as shown in Eq. (3).

$$C_{ij} = \alpha * \frac{1}{b_{ij}} + \beta * d_{ij},\ \forall ij \in L, \qquad (3)$$
$$\alpha + \beta = 1, \qquad \alpha, \beta \in \mathbb{R}^{+}$$

For each link ij in the entire network, C_{ij} is calculated in Eq. (3), where indicates the bandwidth capacity of a physical link, and the delay of a physical link is represented as d_{ij}. To determine the importance factor of QoS parameters, we use α and β values in the range [0, 1] in Eq. (3).

In Figures 7d-9d and 8d-10d, we adjust the α and β values as 0.7 and 0.3, respectively, while also modifying the bandwidth requirement of a request and the number of switches in each network in a manner that is analogous to what was done in the earlier sections. To achieve the goal of minimizing the required composite metric while simultaneously constructing a QoS-based E2E channel over the inter-network, we calculate the composite metric of each link in the network that is capable of meeting the bandwidth requirement of the service request. Similar to minimizing the number of hops in an E2E path, the *MHPS* and *MHPS_BGP* strategies have better performance than other approaches while increasing the bandwidth demands of a service request and the number of switches per network. This is because *MHPS* and *MHPS_BGP* focus on picking an E2E path that satisfies QoS parameters with the minimum number of hops by using all available transactions. Thus, in Figures 7d and 9d, these *MHPS* and *MHPS_BGP* techniques have similar results while the bandwidth demand of a service request is up to 20. When the bandwidth demand is higher than 20, the *MHPS* approach has slightly better results than *MHPS_BGP* thanks to finding a minimum hop path by considering the entire network. Similarly, in Figures 8d and 10d, the *MHPS* and *MHPS_BGP* approaches outperform other strategies in the CMP metric by at least 100% while increasing the number of switches per network. This is because the *MHPS* and *MHPS_BGP* exploit select a path that has the minimum number of hops with the CMP metric.

6. Conclusion

We previously introduced *QoSChain*, an innovative blockchain-based QoS-enabled inter-network routing system, which was accomplished by combining the potent advantages of Blockchain (BC) technology with Software Defined Networking (SDN). As a further development of *QoSChain*, this work investigates the impact of three path selection strategies defined in [9], and two new Border Gateway Protocol-based inter-ISP routing strategies: *First Feasible Path Selection (FFPS), Random Feasible Path Selection (RFPS), Minimum Hop Path Selection (MHPS), FFPS with BGP at inter-network level (FFPS_BGP),* and *MHPS with BGP at inter-network level (MHPS_BGP)*. To evaluate the performance of proposed path selection strategies over NSFNET and US Backbone inter-network topologies, we use four different metrics: (i) the required time to set up a flow as Flow Setup Time (FST), (ii) the number of overhead messages of the network communication as Messages Exchanged and Processed (MEP), (iii) required network resource to establish a service request as Network Resource Consumption (NRC), and (iv) a correlation of network resource and latency for enabling an E2E path as Composite Metric of a Path (CMP). The experimental results indicate that the *MHPS* and *MHPS_BGP* approaches have better performance than the other approaches and exploit the effectiveness of reducing hop counts as much as possible to pick a QoS-based E2E path for MEP, NRC, and CMP metrics despite the fact that these results in a predominantly substantially longer initial FST.

Acknowledgment

This work is supported by the Scientific & Technological Research Council of Turkey (TUBITAK) under Grant No. 120E448.

Conflict of Interest Statement

The authors declare that there is no conflict of interest

References

[1] M. Karakus and A. Durresi, "Quality of service (qos) in software defined networking (sdn): a survey," *Journal of Network and Computer Applications*, vol. 80, pp. 200–218, 2017. doi:10.1016/j.jnca.2016.12.019

[2] A. A. Monrat, O. Schel´en, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117 134–117 151, 2019.

[3] P. Wang, X. Liu, J. Chen, Y. Zhan, and Z. Jin, "QoS-Aware service composition using blockchain-based smart contracts," *in ICSE, ser. ICSE '18. New York, NY, USA: Association for Computing Machinery, 2018,* pp. 296–297. doi:10.1145/3183440.3194978

[4] E. Ak and B. Canberk, "BCDN: A proof of concept model for blockchain-aided CDN orchestration and routing," *Computer Networks*, vol. 161, pp. 162–171, 2019.

[5] J. Yang, S. He, Y. Xu, L. Chen, and J. Ren, "A Trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks," *Sensors*, vol. 19, no. 4, pp. 970, 2019.

[6] M. Saad, A. Anwar, A. Ahmad, H. Alasmary, M. Yuksel, and A. Mohaisen, "RouteChain: towards blockchain-based secure and efficient BGP routing," *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC),* pp. 210-218, 2019. doi:10.1109/BLOC.2019.8751229

[7] G. Ramezan and C. Leung, "A Blockchain-Based contractual routing protocol for the internet of things using smart contracts," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 4029591, 2018. doi:10.1155/2018/4029591

[8] M. Karakus, E. Guler, and S. Uludag, "QoSChain: provisioning inter-AS QoS in software-defined networks with blockchain," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1706-1717, 2021. doi:10.1109/TNSM.2021.3060476.

[9] E. Guler, M. Karakus, and S. Uludag, "Evaluating path selection strategies with blockchain-based routing in Multi-Domain SDNs," *2022 International Balkan Conference on Communications and Networking (BalkanCom),* pp. 6-10, 2022. doi:10.1109/BalkanCom55633.2022.9900749.

[10] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008. doi:10.1145/1355734.1355746

[11] ONF, "OpenFlow Switch Specification (1.5.1)." Open Networking Foundation, March 2015. [Online]. Available: https://www.opennetworking.org/software-defined-standards/specifications. [Accessed: March 15, 2022].

[12] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access,* vol. 7, pp. 22328–22370, 2019. doi:10.1109/ACCESS.2019.2896108

[13] P. Kamboj and S. Pal, "QoS in software defined iot network using blockchain based smart contract: poster abstract," *in SenSys, ser. SenSys'19. New York, NY, USA: Association for Computing Machinery,* pp. 430–431. 2019. doi:10.1145/3356250.3361954

[14] Y. E. Oktian, E. N. Witanto, S. Kumi, and S. Lee, "ISP network bandwidth management: using blockchain and SDN," *2019 International Conference on Information and Communication Technology Convergence (ICTC),* pp. 1330-1335, 2019. doi:10.1109/ICTC46691.2019.8939811

[15] A. Arins, "Blockchain based inter-domain latency aware routing proposal in software defined network," *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE),* pp. 1-2, 2018. doi:10.1109/AIEEE.2018.8592203

[16] M. Karakus and E. Guler, "RoutingChain: a proof-of-concept model for a blockchain-enabled qos-based inter-as routing in SDN," *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 1-6, 2020. doi:10.1109/BlackSeaCom48709.2020.9235021

[17] P. Fernando and J. Wei, "Blockchain-Powered software defined network-enabled networking infrastructure for cloud management," *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC),* pp. 1-6, 2020. doi:10.1109/CCNC46108.2020.9045378

[18] P. Podili and K. Kataoka, "TRAQR: Trust aware End-to-End QoS routing in multi-domain SDN using Blockchain," *Journal of Network and Computer Applications,* vol. 182, pp. 103055, 2021.

[19] Q. Qiao, X. Li, Y. Wang, B. Luo, Y. Ren, and J. Ma, "Credible routing scheme of sdn-based cloud using blockchain," *In: Data Science. ICPCSEE 2019. Communications in Computer and Information Science, , Cheng, X., Jing, W., Song, X., Lu, Z. (eds), Springer, Singapore* vol. 1058, pp 189–206, 2019. doi: 10.1007/978-981-15-0118-0_15

[20] G. S. Aujla, M. Singh, A. Bose, N. Kumar, G. Han, and R. Buyya, "BlockSDN: blockchain-as-a-service for software defined networking in smart city applications," *in IEEE Network,* vol. 34, no. 2, pp. 83-91, 2020. doi:10.1109/MNET.001.1900151

[21] W. Hou, Z. Ning, L. Guo, and P. Guo, "SDN-based Optimizing Solutions for Multipath Data Transmission Supporting Consortium Blockchains," *2018 International Conference on Computer, Information and Telecommunication Systems (CITS),* pp. 1-5, 2018. doi:10.1109/CITS.2018.8440191.

[22] IANA "Internet Assigned Numbers Authority." [Online]. Available: https://www.iana.org. [Accessed: 2020-05-31].

[23] M. Caesar and J. Rexford, "Bgp routing policies in isp networks," *in IEEE Network,* vol. 19, no. 6, pp. 5-11, 2005. doi:10.1109/MNET.2005.1541715.