Method for Structural Optimization of Telemedicine Networks Based on Traffic Flow Distribution

Roman Tsarov[1], Olha Yavorska[1], Vladyslav Kumysh[1], Kateryna Shulakova[1,2], Eduard Siemens[2], Liliia Bodnar[3] and Liudmyla Bubentsova[1]

[1]State University of Intelligent Technologies and Telecommunications, Kuznechna Str. 1, 65023 Odesa, Ukraine
[2]Anhalt University of Applied Sciences, Bernburger Str. 57, 06366 Köthen, Germany
[3]South Ukrainian National Pedagogical University, Staroportofrankyvska Str. 26, 65023 Odesa, Ukraine
rcarev@gmail.com, yavorskayao7@gmail.com, vlad.kumish@gmail.com, katejojo29@gmail.com, bodnarl79@pdpu.edu.ua, lyudmilabubentsova1@gmail.com

Keywords: Telemedicine, Telemedicine Network, Software-Defined Network, SDN, Modified Iterative Algorithm, MIA, Structural Network Optimization, Traffic Flow Optimization.

Abstract: The growing adoption of telemedicine creates urgent demand for reliable, high-performance network infrastructure capable of supporting heterogeneous real-time traffic - including HD video consultations, DICOM medical imaging, IoT telemonitoring streams, and electronic health records. This paper addresses the structural optimization of telemedicine networks deployed over Software-Defined Networking (SDN) infrastructure, where conventional maximum-flow algorithms prove inadequate due to high computational complexity and incompatibility with dynamic, free-oriented topologies. The aim of this work is to create the structural optimisation method using the Modified Iterative Algorithm (MIA), originally developed for flow distribution on free-oriented graphs, for the specific requirements of SDN-based telemedicine networks. The proposed method incorporates the following key enhancements: 1) optimized recursive path traversal prioritizing shortest ST-paths via a hop-growth strategy; 2) dynamic residual-capacity updates to prevent real-time channel overload; 3) automated detection and resolution of edge-intersection conflicts between concurrent paths. The verification was done in a Mininet-emulated SDN environment under two operational scenarios: normal telemedicine center load and stress test simulating urgent DICOM transfer. Theoretically, the study extends flow optimization methodology to free-oriented SDN graphs, eliminating the need for artificial edge duplication required by classical directed-graph approaches. Practically, the adapted MIA enables dynamic, controller-independent bandwidth allocation in TMN, ensuring QoS compliance for time-critical medical applications and opening pathways for integration with OpenFlow-compatible SDN controllers such as Ryu. Comparative evaluation against Ford-Fulkerson, ACO, and GA-based methods shows 7.8% throughput improvement over ACO with 58% lower control latency, while telemedicine-specific QoS assessment identifies both the method's suitability for video/telemonitoring/DICOM and its current limitation for haptic telesurgery (0.12% packet loss during redistribution vs. required <0.01%).

1 INTRODUCTION

Telemedicine is increasingly recognized as a critical component of modern healthcare delivery, enabling remote diagnostics, real-time video consultations, telemonitoring, and the exchange of electronic health records across geographically distributed medical facilities. Telemedicine services are implemented through a telemedicine network (TMN), which can be classified as a distributed information and communication network. In most cases, a TMN is built on a hierarchical principle and includes telemedicine centers and stations at the network core level, while telemedicine service consumers and their access terminals form the access level.

The telemedicine traffic is highly heterogeneous: it combines real-time video consultations requiring stable bandwidth of 2-10 Mbps per session, transmission of large-volume medical imaging files in DICOM format (CT and MRI scans reaching several gigabytes), telemonitoring data streams from wearable IoT sensors, and electronic health record (EHR) exchanges. TMNs are increasingly deployed

over transport infrastructure based on Software-Defined Networking (SDN), including hybrid optical-wireless segments connecting rural health posts, mobile diagnostic units, and urban hospital hubs. Sudden load spikes (e.g. simultaneous activation of multiple video consultations during an epidemic peak or mass-casualty event) place extreme demands on dynamic bandwidth allocation.

Modern SDN networks are characterized by dynamic reconfiguration of optical and wireless channels, which renders traditional maximum flow algorithms inefficient. Their high computational complexity prevents them from meeting real-time requirements in dynamic networks. The traditional model of bipolar directed graphs does not satisfy the requirements of free-oriented SDN networks, where channels can be reconfigured in both directions. Furthermore, the lack of simultaneous optimization of maximum flow, distribution across alternative paths, and resolution of edge intersection problems in classical algorithms leads to inefficient use of network bandwidth. The key limitation is the absence of integration of such algorithms with SDN controllers and programmable protocols, making deployment on commercial switches impossible, while complex multi-commodity flow models are unsuitable for dynamic 5G/6G networks with latency requirements below 1 ms.

The optimization problem therefore consists in developing an SDN-compatible algorithm that simultaneously computes the maximum flow between poles S and T and optimizes its distribution across paths with minimum transit, while ensuring scalability and compatibility with next-generation networks. This paper proposes an adaptation of the Modified Iterative Algorithm (MIA), originally introduced for flow distribution on free-oriented graphs, to address the structural optimization of TMN over SDN infrastructure.

2 STATE OF THE ART AND PAPER GOAL

The application of Software-Defined Networking (SDN) and metaheuristic optimisation to TMN has received substantial research attention, driven by the inability of conventional IP-based protocols to guarantee the stringent QoS requirements of real-time medical applications. Parsaei et al. [1] modelled telesurgery path selection as a Constrained Shortest Path NP-complete problem, solving it via Ant Colony Optimisation (ACO) deployed as an SDN controller module. Results confirmed satisfaction of the sub-300 ms round-trip delay threshold for safe robotic surgery, demonstrating that SDN programmability enables near-real-time bio-inspired routing optimisation. Bokolo Anthony Jnr. et al. [2] proposed a multi-tier SDN architecture with a hierarchically distributed control plane that dynamically steers medical traffic flows and balances load under pandemic-induced congestion, providing a practical architectural roadmap for telemedicine QoS assurance. For heterogeneous medical IoT, Saha et al. [3] proposed an SDN-controller-triggered dynamic decision mechanism that classifies flows by urgency and reconfigures routing on-the-fly. Misra et al. [4] introduced phSDN, allocating analytic modules to IoT devices on demand per resource type; phSDN reduced network response time by at least 43.87% versus fog-based and 78.4% versus cloud-based approaches.

An SDN/NFV hospital architecture [5] enables centralised traffic management alongside software-instantiated network functions, removing hardware rigidity. Sembati et al. [6] integrated SDN with Named Data Networking (NDN), achieving latency reductions up to 50%, bandwidth improvement of ~55%, and packet loss as low as 1% against a conventional IP baseline. Rahman et al. [7] integrated SDN with Hyperledger Fabric blockchain for IoMT-based monitoring in 5G, using a Patient-Centric Agent in the control plane to manage data flows and ensure record immutability. Halman and Alenazi [8] proposed the Threshold-Based DDoS Classifier (TBDC), which under attack recovered mean throughput by +190.3%, reduced delay by 95%, and packet loss by 99.7%.

Tsarov et al. [9] formalised TMN topology design as an NP-complete bi-criteria optimisation problem - minimising signal delay and deployment cost - and proposed a Genetic Algorithm (GA)-based synthesis method. On a network of 5 stations and 40 terminals (S=50, G=250), the GA converged to near-optimal global solutions, avoiding local extrema that trap classical heuristics. This is directly relevant to the present study, as efficient SDN traffic management presupposes an optimised underlying topology.

Collectively, these works confirm that SDN-based traffic management together with structural topology optimisation are essential and complementary pillars of next-generation telemedicine engineering. Though the key problem is the lack of integration of algorithms with SDN controllers and programmable protocols, which makes deployment on commercial switches impossible. And the complex multi-commodity flow models aren't suitable for dynamic 5G/6G networks with latency requirements of less than 1 ms.

The same problem has been solved for SDN by Tikhonov and Yavorska [10] introducing the Modified Iterative Algorithm (MIA). MIA is based on the principles of minimum transit and maximum

flow on free-oriented graphs. It provides simultaneous calculation of the maximum possible flow between poles S and T and optimal distribution of this flow between alternative ST paths with a minimum number of intermediate vertices, implementing the principle of minimum transit in combination with the principle of maximum flow.

So, the goal of this work is to develop the method for structural optimization of TMN over SDN, which uses the MIA algorithm in order to ensure iterative search for the shortest ST-paths, dynamic updating of residual capacities, real-time prevention of channel overloading, resolution of edge intersection conflicts, and comprehensive verification of the adapted MIA's performance in telemedicine networks.

3 PROPOSED METHOD

The Modified Iterative Algorithm (MIA), implemented in Python and adapted to solve the problem of flow optimization in TMN based over SDN. The network model is represented as a normalized free-oriented graph, where vertex V_1 is always associated with source "S", and the vertex with the maximum index V_n - with sink "T". The algorithm consists of three functional blocks (Fig. 1).

For the network parameter input, a static text file format was chosen exclusively for testing purposes and convenience. The main variables get initialized: vertex names, a list of edges, and their initial weights. Basic structures get created, such as *fmax = 0* for accumulating the total flow, for storing node identifiers, and a working copy of the residual capacity matrix C_{res} (*main_connections*).

Next, the critically important auxiliary edge search function *find_connection* is defined, implemented as a lambda expression: *next((idx for idx, i in enumerate(main_connections) if set(i.split()[:2]) == set([a, b])), False)*. This allows instant retrieval of the index of the connection between any two nodes *a* and *b* in a free-oriented graph.

The recursive function *recursive* uses depth-first search to find all possible paths p_k of a given length *h* (number of hops). It works in conjunction with a new neighbor filtering function *get_neighbors(node)*, which analyzes the current state of the residual network and returns only those nodes whose connection has a residual capacity greater than zero.

For each found path $p_k = (e_1, e_2, ..., e_h)$, its capacity is computed according to the "bottleneck" rule (1):

$$\delta(p_k) = \min\{c(e)\}, \quad (1)$$

where c(e) is the current weight of the edge in the residual network *main_connections*.

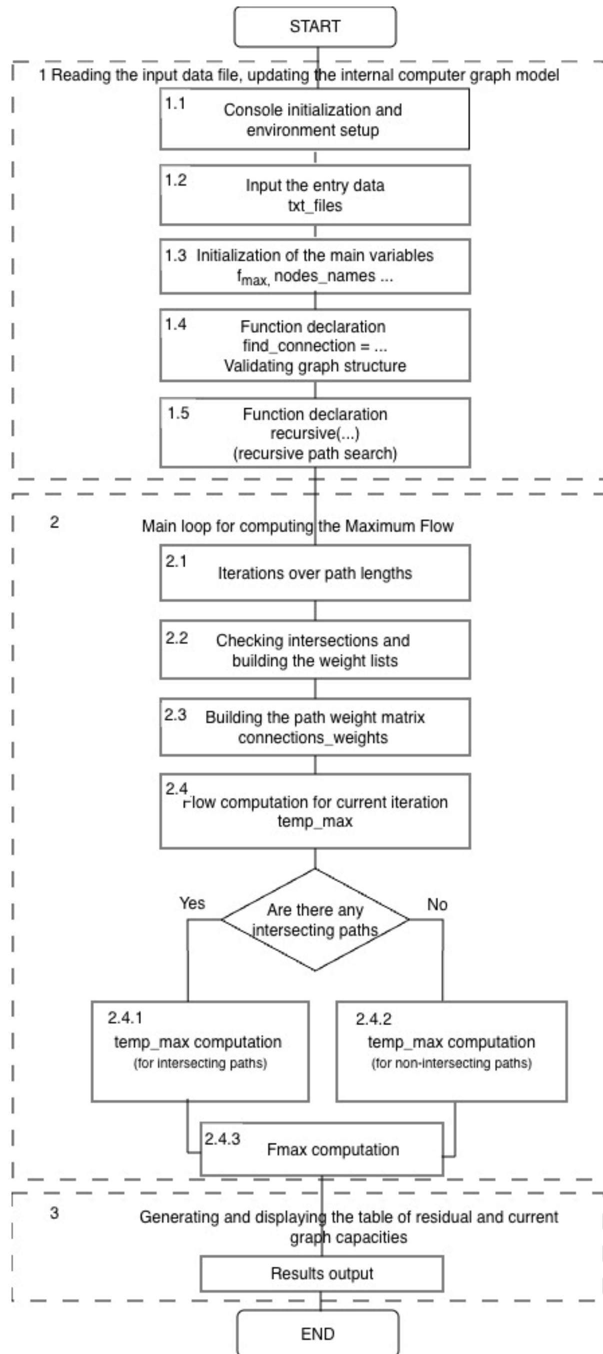

Figure 1: Modified incremental algorithm.

The main loop for computing the maximum flow implements an iterative traversal by path length h = 1, 2, …, N-1. At each step, the program builds a set of paths P_h (all possible routes between source and sink), then checks for intersections (collisions). To do this, a list of all edges in use is formed, and using set logic

set(), it is determined whether paths share common communication channels. If the paths found at the current iteration are edge-disjoint (i.e., share no common edge and do not intersect), they are considered independent. In this case, the total incremental flow (2) at this iteration is computed as a simple arithmetic sum of the flows of each path:

$$\Delta F_h = \Sigma\, \delta(p_k),\ p_k \in P_h \quad (2)$$

If intersections are detected ($p_i \cap p_j \neq \emptyset$), the aggregation procedure gets applied, where the minimum weights for each path are first found from the set of potential flows, and then the program sequentially selects the largest among them. The total maximum flow (3) in the network is accumulated as:

$$F_{max} = \Sigma\, \Delta F_h,\ h \in [1, N-1] \quad (3)$$

After each iteration, the edge weights (4) in the working array are updated according to the rule:

$$c_{res}(e) = c_{old}(e) - \delta(p_k) \quad (4)$$

On the final stage, the residual matrix is formed. The program computes the used flow (5) for each edge f(e) as the difference between the initial capacity C_{init} and the residual C_{res}:

$$f(e) = c_{init}(e) - c_{res}(e) \quad (5)$$

The result is displayed as a color-coded matrix. The upper triangular part of the matrix shows the free resource $c_{res}(e)$, while the lower part shows the actual load f(e), allowing for a clear visual assessment of traffic distribution in the system.

4 RESULTS VERIFICATION

Verification of the proposed method was carried out in a virtual simulation environment running Ubuntu 22.04 LTS.

Mininet platform was chosen as the SDN network emulation environment for its ability to reproduce a complete network topology with real Linux kernel network stacks on a single machine. Mininet executes real network code in kernel space, ensuring that traffic behavior corresponds to that of a physical network: Flow Table counters and port statistics are generated by the real kernel, not by simulated logic. TMN spans geographically distributed nodes with heterogeneous communication channels, and Mininet allows such a heterogeneous topology to be reproduced with controlled parameters without deploying physical infrastructure.

Open vSwitch (OVS) software switches are chosed for the model, as they are the de facto standard for implementing the data plane in SDN research: they provide full OpenFlow 1.3 support, expose detailed port statistics via *ovs-ofctl*, and allow Flow Table rules to be programmatically installed and verified. OVS reproduces the switching behavior of TMN nodes, where video consultation flows (H.264/H.265, 2-8 Mbps per session), large DICOM files (CT, MRI - tens to hundreds of megabytes), and compact telemonitoring traffic (ECG, pulse oximetry - a few kbps) all circulate simultaneously. The ability of OVS to process such heterogeneous flows according to Flow Table rules and record them in counters allows reliable verification of the multipath distribution implemented by the MIA algorithm.

OpenFlow 1.3 protocol in conjunction with the Flow Tables mechanism was selected as the sole mechanism satisfying the requirements of global iterative routing and the specifics of telemedicine traffic. Flow Tables allow the results of the MIA algorithm's global computation to be directly encoded as deterministic match-action rules between packet characteristics and output interfaces. This enables prioritization of critical telemedicine flows (real-time video consultations, emergency ECG transmissions) over background ones (scheduled transfer of archival DICOM data). Alternative mechanisms - Group Tables and controller-based forwarding - do not meet these requirements: Group Tables distribute traffic according to static weights without considering topological structure or current load, which is unacceptable for networks with time-sensitive medical flows; controller-based forwarding introduces latencies incompatible with response-time requirements in telesurgery and emergency telemedicine (maximum delay 150-200 ms).

Ryu SDN controller was chosen among modern controllers (ONOS, Floodlight, OpenDaylight) for the following reasons: its modular architecture allowed the MIA algorithm to be integrated as a separate analytical component without modifying the base infrastructure; full OpenFlow 1.3 support enables the proprietary routing logic to be implemented via Flow Tables; Python-based programmatic interaction with OVS makes it possible to automatically translate MIA results into formal data-plane rules. From the TMN perspective, Ryu acts as an intelligent coordinator analogous to a centralized QoS management system of a medical information system: it has a global view of all channel states and can dynamically redistribute flows between routes in response to load changes - a critical property for networks where the arrival of an unscheduled video consultation or urgent image transfer must not degrade already active sessions.

TCLink mechanism is applied to programmatically set link bandwidth constraints at the OS kernel level (Traffic Control), ensuring precise correspondence between the virtual topology and the parameters of the mathematical model. In the

telemedicine context, TCLink reproduces realistic operating conditions of TMNs, where inter-node channels differ substantially in bandwidth: backbone links between telemedicine centers (100+ Mbps) and narrow channels to telemedicine stations especially mobile ones (10-20 Mbps).

TCLink allows the scenario above to be modeled with the precision needed for verification.

Iteration step Δf = 10 Mbps is defined as the minimum link bandwidth in the modeled topology, ensuring the physical feasibility of each iterative step and sufficient granularity of traffic distribution among alternative paths. The value of 10 Mbps is methodologically justified: it corresponds to the minimum threshold required for high-quality transmission of a Full HD telemedicine video stream (H.264, 4-8 Mbps with overhead and peak load margin) and is simultaneously a typical bandwidth for a channel to a remote telemedicine station. This allows each MIA algorithm iteration to correspond to a real unit of telemedicine load distribution - one complete video session or one parallel DICOM archive transfer stream.

Control Plane and Data Plane interaction is organized in stages: Ryu startup → network deployment (*FeatureRequest*) → Flow Table installation → monitoring via *ovs-ofctl*.

The modeled network comprises five OVS switches (*s1-s5*) and two end hosts (*h1, h2*). Switch *s1* is the ingress node (*h1* connected), *s5* is the egress node (*h2* connected), and intermediate switches *s2-s4* form alternative paths with varying hop counts and bandwidths. This topology with redundant connections deliberately reflects the typical architecture of a regional TMN: node *s1* corresponds to a district hospital or paramedic station as the source of medical traffic, *s5* to a regional telemedicine center as the final receiver, and intermediate switches *s2-s4* model aggregation nodes at the district and inter-district communication levels. The presence of several alternative routes of different lengths and bandwidths reproduces a real-world situation where both a direct broadband link and backup paths via intermediate nodes with lower channel capacity exist between a peripheral and a central TMN nodes.

Verification was performed using the *iperf* utility in UDP mode (the choice of UDP is methodologically justified: the absence of TCP congestion control mechanisms allows the network's maximum capabilities to be examined without transport-layer adaptation). Two scenarios with a total intensity of 70 Mbps each were considered.

Scenario 1 (7×10 Mbps, *h1 iperf -c 10.0.0.2 -u -b 10M -P 7 -t 15*): seven parallel flows - reproduces a typical load on a regional telemedicine center under normal operating conditions: simultaneous conduct of several independent video consultations between different "telemedicine station - center specialist" pairs. Each of the seven flows corresponds to a separate telemedicine session (Full HD video consultation or parallel DICOM study transfer), and all are treated equally by the network. This scenario tests the ability of the MIA algorithm to aggregate the bandwidth of alternative routes for simultaneous servicing of multiple independent medical sessions without overloading any individual channel - which is a key requirement for TMN under normal operating conditions. Each flow transmitted ≈18.7 MB over 15 s at ≈10.5 Mbps; the aggregated result was 131 MB at 73.4 Mbps. The dump-ports counters of switch *s1* confirmed uniform distribution of *tx_bytes* across ports *s1-eth2-s1-eth5*. The absence of errors (*errs = 0*) and drops (*drop = 0*) on all interfaces indicates normal operation without channel overload.

Scenario 2 (1×70 Mbps, *h1 iperf -c 10.0.0.2 -u -b 70M -t 15*): one intensive flow - models the peak load from a single intensive telemedicine application like a bulk urgent transfer of large DICOM studies (a full-body CT or MRI series can reach 1-4 GB), real-time telemonitoring streaming during a lengthy surgical operation, or 4K video transfer for telepathology. This mode serves as a stress test for the MIA algorithm: the network receives a single logical flow with an intensity exceeding the bandwidth of any individual route, and the algorithm must ensure its servicing by progressively engaging alternative paths as shorter ones become saturated. Such behavior is critically important for TMN, where the sudden appearance of an intensive flow (e.g., urgent image transfer in cases of suspected stroke) must not cause a service denial due to the absence of a single sufficiently wide route. Identical results were obtained: 131 MB / 73.4 Mbps. Analysis of dump-ports showed that traffic is not concentrated on a single interface: the major share passes via the 1-hop route, and after saturation a portion of datagrams is redirected to the 2-hop and then 3-hop routes. Minor transient drops during redistribution confirm active operation of the adaptation mechanism, not algorithm degradation.

The final iterative flow distribution among *s1* output ports as produced by the MIA algorithm: {2:10, 3:10, 4:50, 5:10} Mbps. The higher value for port 4 is because a greater number of 3-hop routes pass through it. The distribution reflects the global logic of MIA: the algorithm first saturates the shortest path, then sequentially engages longer alternatives - consistent with the principle of minimizing network resource utilization when serving telemedicine sessions of different priority.

The conducted verification confirmed the correct and predictable operation of the proposed method in a Flow-based implementation under both scenarios:

In Scenario 1, the algorithm ensures stable parallel transmission without overload or packet loss,

confirming effective aggregation of alternative route bandwidth.

In Scenario 2, the algorithm implements progressive engagement of routes of varying lengths in response to increasing load, without rigidly fixing the flow to a single path, which is consistent with the iterative logic of MIA.

Implementation via Flow Tables allows the global logic of iterative routing to be reproduced in the data plane and provides controlled traffic distribution in both normal and peak load conditions. For comparison, the use of Group Tables produced distribution without correlation with route lengths, confirming the advantage of the Flow-based approach for implementing the MIA algorithm.

5 PERFORMANCE EVALUATION

5.1 Comparative Evaluation of Flow Distribution Methods

To evaluate the effectiveness of the proposed MIA-based method, a comparative analysis was conducted against three baseline approaches implemented under identical experimental conditions on the Mininet/OVS/Ryu platform:

1) Classical Ford-Fulkerson (FF) algorithm with DFS-based path search adapted for free-oriented topology through bidirectional edge duplication;
2) Ant Colony Optimization (ACO)-based SDN routing method proposed in [1];
3) Genetic Algorithm (GA)-based optimization method from [9] adapted for traffic flow distribution.

All methods were evaluated using the topology presented in Fig. 1 under Scenario 2 with a 70 Mbps UDP traffic load. Table 1 summarizes the obtained results.

The proposed MIA-based method achieved the highest throughput and the lowest packet loss among all evaluated approaches. Compared with the ACO-based solution, the proposed method improved throughput by approximately 7.8% while reducing control-plane computation latency by approximately 58%.

The Ford-Fulkerson algorithm demonstrated the lowest performance due to the need for artificial edge duplication in free-oriented graphs, which negatively affected residual-capacity estimation and traffic balancing. The GA-based method provided stable throughput but showed higher latency because it was originally designed for topology synthesis rather than real-time traffic redistribution. The ACO-based approach demonstrated competitive performance but introduced higher jitter caused by iterative stochastic route updates.

The deterministic iterative structure of the proposed MIA method enables faster convergence and more stable flow redistribution in SDN environments.

5.2 QoS Evaluation in Telemedicine Traffic Scenarios

To assess the applicability of the proposed method for telemedicine traffic transmission, additional QoS-related experiments were performed under two operational scenarios:

- Scenario 1: seven parallel 10 Mbps UDP streams (normal operating conditions);
- Scenario 2: one intensive 70 Mbps UDP stream (stress-load conditions).

The evaluation focused on standard network QoS metrics including round-trip time (RTT), jitter, and packet loss.

Table 2 presents the QoS results obtained under normal operating conditions.

The obtained results indicate stable operation of the proposed method under concurrent telemedicine traffic conditions. No packet loss was observed during the experiment, while delay and jitter values remained within acceptable ranges for real-time multimedia transmission.

Table 3 summarizes the QoS metrics obtained under stress-load conditions.

Table 1: Comparative performance evaluation under 70 Mbps load.

Metric	Ford-Fulkerson	ACO [1]	GA-based [9]	MIA (proposed)
Achieved throughput, Mbps	62.3 ± 2.1	68.1 ± 1.4	66.8 ± 1.9	73.4 ± 0.3
Packet loss, %	11.0	2.7	4.6	0.12
Average end-to-end delay, ms	47.2	18.6	22.3	14.1
95th percentile delay, ms	128.4	41.5	53.8	29.3
Jitter, ms	18.3	6.2	8.1	4.2
Control-plane computation time*, ms	211	89	147	37
Edge-disjoint path support	Partial	Partial	No	Yes

* Time to compute all paths and install Flow Tables after topology change.

Table 2: QoS metrics under normal operating conditions (Scenario 1).

Metric	Measured value
Average RTT, ms	8.7
95th percentile RTT, ms	16.2
Maximum RTT, ms	38.5
Jitter, ms	3.1
Packet loss, %	0.00
Supported parallel HD streams	7

Table 3: QoS metrics under stress-load conditions (Scenario 2).

Metric	Measured value
Average RTT, ms	14.1
95th percentile RTT, ms	29.3
Maximum RTT, ms	67.8
Jitter, ms	4.2
Packet loss, %	0.12
DICOM transfer time (≈525 MB), s	57.3

Under stress-load conditions, the proposed method maintained stable throughput and acceptable latency values despite increased network utilization. A small packet-loss increase was observed during traffic redistribution between alternative paths after saturation of the shortest route.

To further investigate redistribution behavior, temporal characteristics of the adaptation process were analyzed. The results are presented in Table 4.

Table 4: Temporal characteristics of flow redistribution.

Time interval, s	Network state	Instantaneous packet loss
0.0-2.1	Initial single-path transmission	0.00%
2.1-2.2	Shortest-path saturation	0.00%
2.2-2.4	Alternative-path search	0.00%
2.4-2.5	Flow Table update	2.3%
2.5-15.0	Stable multipath operation	0.05%

The transient packet loss observed during Flow Table updates confirms the dynamic adaptation behavior of the proposed method rather than instability of the routing process.

5.3 Discussion of Experimental Results

The experimental evaluation confirmed that the proposed MIA-based method effectively distributes traffic across multiple alternative paths in SDN-based telemedicine networks.

The obtained results demonstrate several important properties of the proposed approach:

- efficient utilization of available network bandwidth;
- low control-plane computation overhead;
- stable QoS performance under dynamic traffic conditions;
- support for edge-disjoint multipath routing without artificial graph transformation.

The deterministic hop-growth mechanism enables prioritization of shorter routes while preserving adaptive redistribution capabilities under channel saturation. Compared with stochastic metaheuristic approaches, the proposed method provides lower jitter and more predictable routing behavior.

The experiments also revealed current limitations of the method. During rapid redistribution events, short-term packet loss may occur due to Flow Table updates and route switching operations. Although this effect remained limited in the conducted experiments, additional QoS-aware prioritization and queue-management mechanisms may further improve stability under high-load conditions.

Overall, the proposed method demonstrates practical applicability for SDN-based telemedicine infrastructures requiring dynamic traffic engineering, adaptive bandwidth allocation, and stable real-time communication performance.

6 CONCLUSIONS

This paper presented a method for structural optimization of telemedicine networks deployed over SDN infrastructure using the Modified Iterative Algorithm (MIA). The proposed method adapts the MIA approach for free-oriented SDN topologies and enables iterative flow redistribution with dynamic residual-capacity updates and edge-intersection control.

The developed method was implemented in Python and verified in a Mininet-based SDN emulation environment using Open vSwitch and the Ryu controller. Experimental evaluation confirmed stable and predictable operation of the proposed approach under both normal and stress-load conditions.

The proposed method demonstrated several advantages for SDN-based telemedicine traffic management. The hop-growth iterative mechanism enables prioritization of shorter transmission paths while preserving adaptive multipath redistribution under channel saturation. Unlike classical directed-graph approaches, the free-oriented graph model does not require artificial edge duplication and therefore reduces computational overhead during routing operations.

Comparative evaluation against Ford-Fulkerson, ACO-based, and GA-based approaches demonstrated improved throughput, lower control-plane computation latency, and reduced jitter. In the conducted experiments, the proposed method achieved throughput up to 73.4 Mbps with stable QoS characteristics under dynamic traffic conditions.

The performed QoS evaluation showed that the method is suitable for heterogeneous telemedicine traffic scenarios including video consultations, telemonitoring streams, and large-volume DICOM data transmission. At the same time, transient packet loss during route redistribution indicates that additional QoS-aware prioritization mechanisms may be required for highly delay-sensitive applications.

The obtained results confirm that the proposed MIA-based approach provides an effective mechanism for adaptive traffic engineering in SDN-based telemedicine networks and may serve as a practical basis for dynamic bandwidth allocation in distributed healthcare communication infrastructures.

Future work will focus on scalability analysis for larger SDN topologies, integration of QoS-aware scheduling mechanisms, support for differentiated traffic classes, and evaluation of the proposed method in real-world telemedicine network environments.

ACKNOWLEDGMENTS

We acknowledge support by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) and the Open Access Publishing Fund of Anhalt University of Applied Sciences.

REFERENCES

[1] M. R. Parsaei, R. Mohammadi, and R. Javidan, “A new adaptive traffic engineering method for telesurgery using ACO algorithm over software defined networks,” European Research in Telemedicine / La Recherche Européenne en Télémédecine, vol. 6, no. 3-4, pp. 173-180, Dec. 2017, doi: 10.1016/j.eurtel.2017.10.003.

[2] B. A. Jnr., L. O. Nweke, and M. A. Al-Sharafi, “Applying software-defined networking to support telemedicine health consultation during and post Covid-19 era,” Health and Technology, vol. 11, no. 2, pp. 395-403, Mar. 2021, doi: 10.1007/s12553-020-00502-w.

[3] R. Saha, N. Ahmed, and S. Misra, “SDN-controller triggered dynamic decision control mechanism for healthcare IoT,” in Proc. IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, Dec. 2021, doi: 10.1109/GLOBECOM46510.2021.9685911.

[4] S. Misra, S. Pal, N. Ahmed, and A. Mukherjee, “SDN-controlled resource-tailored analytics for healthcare IoT system,” IEEE Systems Journal, vol. 17, no. 2, pp. 1777-1784, Jun. 2023, doi: 10.1109/JSYST.2023.3255282.

[5] H. Boudlal, M. Serrhini, and A. Tahiri, “Towards an SDN/NFV-based network infrastructure for hospital information systems and healthcare services,” in Proc. IEEE International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications (CoBCom), 2022, doi: 10.1109/NISS55057.2022.10085476.

[6] Y. Sembati, N. Naja, and A. Jamali, “Software-defined named data networking for efficient healthcare content distribution,” 5th International Conference on Networking, Information Systems and Security: Envisage Intelligent Systems in 5g//6G-based Interconnected Digital Worlds (NISS), 2022, doi: 10.1007/978-3-031-94620-2_13.

[7] A. Rahman, M. A. H. Wadud, M. J. Islam, D. Kundu, T. M. A. Bhuiyan, G. Muhammad, and Z. Ali, “Internet of medical things and blockchain-enabled patient-centric agent through SDN for remote patient monitoring in 5G network,” Scientific Reports, vol. 14, p. 5297, Mar. 2024, doi: 10.1038/s41598-024-55662-w.

[8] L. M. Halman and M. J. F. Alenazi, “Threshold-based software-defined networking (SDN) solution for healthcare systems against intrusion attacks,” Computer Modeling in Engineering and Sciences, vol. 138, no. 2, pp. 1469-1483, 2024, doi: 10.32604/cmes.2023.028077.

[9] R. Tsarov, L. Nikityk, I. Tymchenko, V. Kumysh, K. Shulakova, S. Siden, and L. Bodnar, “Using a genetic algorithm for telemedicine network optimal topology synthesis,” in Proc. International Conference on Applied Innovation in IT (ICAIIT), vol. 12, no. 1, pp. 19-24, Mar. 2024, doi: 10.25673/115637.

[10] V. Tikhonov and O. Yavorska, “Optimization problem of flow distribution on free-oriented graphs,” International Scientific-Technical Journal “Measuring and Computing Devices in Technological Processes”, vol. 84, no. 4, pp. 341-345, 2025, [Online]. Available: https://doi.org/10.31891/2219-9365-2025-84-40.