Efficient Multipath Selection for IoT Video Transmission

Fabiano Bhering
Centro Federal de Educação Tecnológica (CEFET-MG)
Universidade Federal Fluminense (UFF), Brazil
fabianobhering@cefetmg.br

Diego Passos
Universidade Federal Fluminense (UFF), Brazil
ISEL, Instituto Superior de Engenharia, Portugal
dpassos@ic.uff.br

Célio Albuquerque
Universidade Federal Fluminense (UFF), Brazil
celio@ic.uff.br

Katia Obraczka
Univesity of California, Santa Cruz (UCSC), USA
katia@soe.ucsc.edu

***Abstract*—In IoT video applications, such as public transportation and traffic monitoring, video is often transmitted over a multi-hop wireless network and has stringent Quality of Service requirements. While multipath routing has been proposed as an effective strategy for supporting this demand, it should select paths that meet the requirements of the driving video application in a timely manner. In this paper, we introduce FITPATH, an efficient and adaptable path selection scheme for multipath routing that can accommodate scenarios where multiple video sources can simultaneously transmit heterogeneous video flows, i.e., flows of different bitrates. FITPATH uses a novel heuristic-based iterative optimization approach that estimates the conditions of the underlying network in real time while accounting for the different bitrate requirements of the application video flows. We show FITPATH outperforms various existing path selection mechanisms both in terms of user QoE and network performance. We also evaluate FITPATH's convergence to demonstrate that it can be used in practice to quickly generate solutions with adequate video quality while incrementally improving QoE as the algorithm iterates.**

I. Introduction

In the last few years, we have witnessed the rapid proliferation of IoT deployments and applications. A number of them, including public transportation, traffic monitoring, parking and surveillance systems in smart cities [1], [2], capture and transmit video over a multi-hop wireless network.

Like other services that transmit video streams, IoT video applications have stringent QoS requirements in terms of throughput, delay and packet loss [3]. Mutipath routing, a routing strategy that uses multiple paths between the source and destination, has been proposed as an effective way to meet Quality of Service (QoS) requirements of video transmission by providing adequate bandwidth and delay [4] as well as reliability and resilience [3].

In order to fully leverage its ability to find and transmit flows through multiple paths, multipath routing must be able to, in a timely manner, select paths that meet the requirements of the driving video application (e.g., multiple video sources transmitting flows at different bitrates [3]), while accounting for the current conditions of the underlying network. Applications with multiple video sources, for example, require more complex path selection approaches to satisfy user Quality of Experience (QoE), because the probability of paths sharing nodes and links can be significantly higher. Additionally, the performance of a single wireless link can vary according to the link-layer bitrate and its SNR (Signal to Noise Ratio). Uncontrolled transmissions of multiple flows can also cause congestion, as well as flow interference, medium access contention, and collisions [5]. Depending on the number of simultaneous flows, the likelihood of losing or delaying video frames that are particularly loss- and delay-intolerant may increase as a consequence of flow interference, mainly due to collision and queuing.

In this paper, we introduce FITPATH, an efficient and adaptable path selection scheme for multipath routing that can accommodate scenarios where multiple video sources can simultaneously transmit heterogeneous video flows, i.e., flows of different bitrates. FITPATH uses a novel heuristic-based iterative optimization approach that estimates the conditions of the underlying network in real time while accounting for the different bitrate requirements of the application video flows. We conduct an extensive comparative performance study considering application scenarios that mimic realistic IoT deployments and show FITPATH outperforms existing path selection mechanisms, delivering superior end-user QoE and achieving higher network performance. We evaluate FITPATH's convergence characteristics and show that it is able to, relatively quickly, generate path selection solutions that yield high QoE. We also demonstrate that FITPATH can be used in practice by video transmission sources to quickly generate feasible intermediate solutions that result in adequate video quality, while incrementally improving QoE as the algorithm iterates. Practical deployment options for FITPATH are also discussed.

II. Related Work

A number of multipath selection mechanisms for wireless video transmission have been proposed [1], [3]. Table I summarizes the current state-of-the-art in multipath routing path selection approaches and classifies them according to a number

978-1-6654-8627-9/22/$31.00

TABLE I: Multipath selection state-of-the-art approaches.

Mechanism	Route Discovery	Type of Algorithm	Routing Metric	Awareness			
				Video Coding	Multiple Sources	Multi-rate Flows	Flow Interference
ERVT [6]	Reactive	Disjoint-Path	Hop count	✓			
CLMR [7]	Reactive	Disjoint-Path	Delay, packet loss and energy	✓			
RTVP [8]	Hybrid	Position-Based	Position, delay, packet loss and energy	✓			
Q-MMTP [9]	Reactive	Position-Based	Position, delay and energy consumption				
QSOpt [10]	Proactive	Heuristic-Based	QoE model based on packet loss		✓		✓
ILS-MDC [11]	Proactive	Heuristic-based	Throughput estimates	✓	✓		✓
FITPATH	Proactive	Heuristic-based	Throughput and delay estimates	✓	✓	✓	✓

of fundamental design choices. We note that most current path selection mechanisms handle a single video source and ignore flow interference [3]. Some existing approaches use routing metrics, such as QoS and hop count, which are not necessarily the best options to select optimal paths for all flows. They analyze links and paths individually increasing the probability that two or more paths share a link, which causes flow interference. Disjoint path routing is an alternative adopted in several proposals to avoid flow interference. It provides better performance and reliability, but disjoint paths may not be available for all flows [11]. Heuristic-based approaches that take into account flow interference from multiple sources typically ignore the effects of video coding which may result in flows with different bitrates [10].

FITPATH addresses the above issues by using an efficient algorithm to: (1) find a set of candidate paths and (2) given the requirements of the flows to be transmitted by the different video sources (e.g., number of flows, flow bitrates), rank the candidate paths based on their throughput and end-to-end delay.

III. FITPATH

In this paper, we focus on the problem of choosing paths for video flows in a multihop wireless network. As illustrated in Figure 1, we consider a known set S of network nodes that act as video sources. Each source $s \in S$ generates a known number f_s of video flows, according to the video encoder used, transmitted to one or more sink nodes.

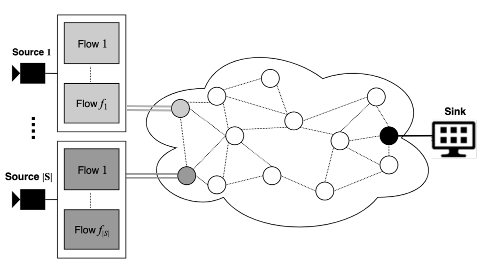

Fig. 1: Multi-source, multi-path video streaming scenario.

In some applications, video sources generate multiple flows with different video compression techniques [2]. For instance, traditional video compression techniques adopted in multipath video transmission, such as Layered Coding (LC) and Multiple Description Coding (MDC), generate flows with different bitrates. Thus, the multipath selection mechanism must account for the flow's bitrate to balance offered load and reduce packet losses and delays. Hence, it needs to be aware of the application's characteristics to maximize the QoE.

A. FITPATH Overview

FITPATH is a multipath selection mechanism based on the Iterated Local Search (ILS) metaheuristic [12] for selecting candidate paths for each video flow generated from multiple sources. Figure 2 shows FITPATH's main components. As input, FITPATH uses information about the network topology and about the video flows from the different video sources (e.g., number of flows and flow bitrate).

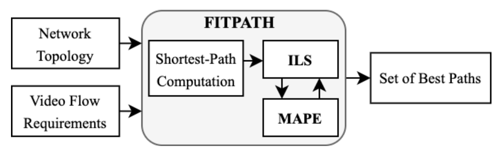

Fig. 2: FITPATH overview

FITPATH starts by performing an all-pairs least-cost (or shortest) path computation to generate a set of candidate paths P, which is used by ILS as input. The set P of candidate paths are generated based on Yen's algorithm [13] by choosing, for each flow, a list of K least cost paths between the source and the sink using the ETX (Expected Transmission Count) metric [14]. ILS applies its iterated search algorithm and evaluates each solution using the Multimedia-Aware Performance Estimator (MAPE) [15]. MAPE, which is described in more detail in Section III-C, estimates the network performance of candidate paths, i.e., throughput, delay and loss, which FITPATH's ILS algorithm uses to select the best candidate path for each flow.

B. Iterated Local Search

ILS is considered one of the best heuristic-based approaches to solve a variety of combinatorial optimization problems of the NP-complete class [12], including routing problems [16]. Additionally, ILS is well suited to address the problem of multipath selection because its iterative nature continuously seeks to improve a set of solutions.

Algorithm 1 shows FITPATH's ILS computation which consists of the following steps: (i) $initSolution(P)$ initializes p with the first candidate solution in the list of candidate solutions P generated by an all-pairs least-cost path algorithm. P's calculation is based on the graph $G(N, L, F, R)$ representing the underlying network topology, where N is the set of nodes

$\{0, 1, \ldots, n\}$, L is the set of *links* $\{0, 1, \ldots, l\}$, F is a list of flows associated with their sources and R is a list of bitrates corresponding to each flow. As detailed in Section III-D, G can be obtained in practice using different approaches, such as proactive link-state routing protocols (e.g., OLSR [17]), or from an SDN controller in SDN-based networks [18]. Note that the initial solution p contains a single path per source, through which all its flows are transmitted; (ii) $MAPE$ [15] is invoked to calculate the cost of the initial solution p and its cost is used to try to find better solutions. Section III-C provides a detailed description of MAPE; (iii) $LocalSearch$ performs local searches to refine p seeking a better solution; (iv) $Perturbation$ generates intermediate solutions p' and p'' by applying perturbations to p; and (v) $AcceptanceCriterion$ evaluates and determines if a new solution can be accepted according to the objective function.

Algorithm 1: Iterated Local Search (ILS)

```
Input  : P - List of candidate solutions;
Output: p - set of best paths;
begin
    p ← initSolution(P);
    Γ(p), Δ(p) ← MAPE(p);
    Γ_best ← Γ(p); Δ_best ← Δ(p);
    p ← LocalSearch(p, P, Γ_best, Δ_best);
    repeat
        p' ← Perturbation(p, P, H);
        p'' ← LocalSearch(p', P, Γ_best, Δ_best);
        p, Γ_best, Δ_best ←
          AcceptanceCriterion(p, p'', H);
    until termination condition met;
end
```

$LocalSearch$, $Perturbation$ and $AcceptanceCriterion$ iterate to traverse potentially good alternative solutions with the goal of accepting a solution that improves network performance while meeting the requirements of the video flows. Based on MAPE's performance estimates, the ILS algorithm calculates the objective function given by Equations 1 and 2 which evaluate two performance metrics, namely: (1) throughput gap $\Gamma(c)$ of a candidate solution c is the sum of the percentage difference between the estimated throughput $\overline{Th_f}$ and bitrate λ_f for all flows f; and (2) end-to-end delay $\Delta(c)$ which is the estimated delay $\overline{\Delta_f}$ averaged over all flows f of a candidate solution c. We choose to use percentages for the throughput gap because flows may have different bitrates.

$$\Gamma(c) = \sum_{f \in F} \frac{\lambda_f - \overline{Th_f}}{\overline{Th_f}} \quad (1) \qquad \Delta(c) = \frac{\sum_{f \in F} \overline{\Delta_f}}{|F|} \quad (2)$$

FITPATH ILS' termination condition establishes limits on the number of iterations without significant improvement to the solution and on the algorithm's total execution time (often based on the application response time requirements). It can be used to control the balance between *intensification*, i.e., performing local search, and *diversification*, i.e., exploring perturbations in the solution space of that search.

Local Search: At each iteration, $LocalSearch$ evaluates solutions that are "neighbors" of the current best solution p. Here, two feasible solutions are considered neighbors if they differ by exactly one path. We use a simple local search technique to minimize computation and storage overhead. As shown in Algorithm 2, $LocalSearch$ uses paths of the next solution in P to generate and evaluate all possible neighbor candidates of the current solution. Before estimating the network performance of each neighbor solution using MAPE, solutions are pruned to reduce computation resource usage and execution time. This pruning uses the *minimum throughput gap* $\Gamma_{min}(c)$, which is computed using Equation 1, but replacing $\overline{Th_f}$ with the inverse of the ETX of flow f's path.

Solutions that are not pruned are evaluated by MAPE, which returns estimates for the throughput and delay of each flow. A new candidate solution is considered better than the current best solution if it either has a lower throughput gap or if it results in the same throughput, but in lower delay.

Algorithm 2: Local Search

```
Input  : p, P, Γ_best, Δ_best;
Output: p;
begin
    p' ← nextSolution(P);
    for new candidate c ∈ neighbors(p, p') do
        if Γ_min(c) ≤ Γ_best then
            Γ(c), Δ(c) ← MAPE(c);
            if Γ(c) < Γ_best then
                Γ_best ← Γ(c); Δ_best ← Δ(c); p ← c;
            if Γ(c) = Γ_best then
                if Δ(c) < Δ_best then
                    p ← c;
    end
end
```

Perturbation: To avoid converging too soon and getting "stuck" in local optima, FITPATH applies perturbations to the current best solution. Our current strategy is to generate a new candidate solution by randomly replacing one path for each source in the best currently known solution p. It uses paths in P and the history solution list H which may lead to visiting new portions of the solution space not previously visited by the initial all-pairs least cost path ($initSolution$) or nearest neighbor ($LocalSearch$).

Acceptance Criterion: FITPATH compares the costs of solutions p and p'' considering the same criterion used in $LocalSearch$ and accepts the best one. Whenever the best solution is accepted, Γ_{best} is updated and the previous solution is added to the solution history list H that is consulted upon future perturbations.

C. Network Performance Estimation

MAPE [15] is used to estimate the performance of the candidate paths selected by FITPATH's ILS computation. It is a deterministic simulator that estimates the long-term average performance for all flows considering their bitrate as well as flow interference.

Unlike stochastic simulators that study network behavior over a predefined period of time, MAPE estimates the performance of the network, e.g., throughput, packet loss, and end-to-end delay at steady-state, i.e., after a finite number of iterations when the network state ceases to change. When compared to other deterministic estimators, MAPE is able to attain more accurate performance estimates due to its ability to account for flows with specific bitrates. This is consistent with real-world multimedia application scenarios. MAPE's performance, including its estimation accuracy, is studied in [15].

MAPE receives as input the set of paths to be evaluated and the bitrate of each flow. It then uses this information to deterministically simulate the network dynamics by generating simulated packets for each flow and triggering relevant network events, such as wireless medium access transmission, queue management packets being added, removed and discarded from buffers. As a result, MAPE is able to take into account flow interference that happens due to buffer overflow, link-layer transmission losses, and medium access contention. MAPE continues to simulate the network until it detects that steady state has been reached, i.e., when the same sequence of events repeats itself. At that point, MAPE computes each flow's throughput and delay estimates, which are used to calculate the ILS objective function according to Equations 1 and 2.

D. Deployment Considerations

FITPATH targets IEEE 802.11-based wireless multihop networks. In particular, it focuses on IoT applications such as Smart Cities and surveillance systems which require transmission of video streams. In these scenarios, nodes are typically stationary, powered by continuous energy sources and equipped with sufficient storage and computing power. As such, frequent topology changes caused by node mobility, energy depletion, etc. are not expected to play a significant role. In scenarios where topology changes may occur more frequently, information about network conditions can be obtained using different mechanisms depending on the underlying control plane architecture as discussed below.

FITPATH can be deployed under decentralized network control, e.g., multi-hop wireless ad-hoc networks using distributed routing protocols. In these scenarios, each node maintains a complete view of the network topology utilizing topology state monitoring and dissemination mechanisms similar to those implemented by proactive link-state routing, e.g., OLSR. [17]. At each video source, FITPATH makes routing decisions based on the current network state obtained from the topology monitoring module. It finds a set of paths that video sources can use to send their video flows.

FITPATH can also be deployed under centralized network control *a la* SDN (Software-Defined Networking) [18]. The SDN controller makes routing decisions based on its global view of the network and informed by FITPATH, while network nodes perform data forwarding based on the routing rules they receive from the controller. Note that in centralized network control environments, FITPATH will be used to assist the controller in its routing decisions based on current network topology knowledge and video traffic requirement information. The controller then updates the forwarding nodes' routing tables to forward flows according to the selected paths.

While FITPATH's convergence time depends on the network topology and application scenarios, since the solution space scales with the number of sources and flows, video transmission can start and use FITPATH's output after the first few iterations; FITPATH can continue to execute in the background, looking for better solutions.

IV. Performance Evaluation

We evaluate the performance of FITPATH using the ns-3 network simulator. We consider application scenarios that mimic typical surveillance deployments in urban regions where video sources transmit flows simultaneously to a single monitoring center [19].

The simulated network is deployed in a 300 m $\times$ 300 m area over which we randomly distribute 30, 45 or 60 nodes using IEEE 802.11g radios operating at 18 Mb/s, and Cost231 [20] as the propagation model. Despite random, node placement guarantees a minimum distance of 5 m.

We use a variety of 30 fps video sequences that contain scenes of different levels of motion — classified as *low*, *medium* or *high* — in an attempt to reproduce video traffic diversity in complex environments, such as smart cities. The Evalvid [21] framework was used to generate realistic video traffic for the simulations. We use Variable Bit Rate (VBR) encoding and perform experiments with flows with average rates varying from 256 kb/s to 2 Mb/s. All simulations have 4 video sources and each source generates 2 flows, with a total offered load of 4 Mb/s. The video quality is evaluated using the *Structural Similarity Index Measure* (SSIM) and the *Peak Signal-to-Noise Ratio* (PSNR). Both metrics have been widely used to measure QoE [22].

Our evaluation considers both network performance and quality of experience. We evaluate FITPATH by comparing its performance against the mechanisms described in Section II, namely: RTVP, CLMR, ERVT, Q-MMTP, QSOpt and ILS-MDC. Results presented in Sections IV-A and IV-B are averaged over a total of 270 runs (3 different node densities, 3 different scene motion levels and 30 runs for each combination) and are shown along with their respective 95% confidence intervals [1].

A. FITPATH Performance

Table II summarizes FITPATH's average performance as well as the performance of the other existing multipath path selection mechanisms. These experiments were executed for all scenarios (different numbers of nodes and coding rates).

[1] In some graphs, confidence intervals are not shown because they are too small.

As expected, the network performance of each mechanism — throughput, delay and frame loss — correlates well with their video quality performance. Overall, FITPATH performs better in terms of per-flow throughput and video frame loss, resulting in superior video quality as reflected in the PSNR and SSIM metrics.

TABLE II: Average performance of the different multipath selection mechanisms. Ranges denote the 95% confidence intervals.

Mechanism	Per-flow Throughput (kb/s)	End-to-End Delay (ms)	Video Frame Loss (%)	PSNR (dB)	SSIM
FITPATH	[420,427]	[98,100]	[29,30]	[30,31]	[0.85,0.86]
ERVT	[406,413]	[91,93]	[33,34]	[29,30]	[0.80,0.81]
CLMR	[373,380]	[121,123]	[39,40]	[26,27]	[0.74,0.75]
RTVP	[360,366]	[102,104]	[40,41]	[26,27]	[0.75,0.76]
Q-MMTP	[267,270]	[206,208]	[58,59]	[18,19]	[0.54,0.55]
QSOpt	[409,417]	[105,107]	[34,35]	[28,29]	[0.81,0.82]
ILS-MDC	[392,398]	[109,111]	[39,40]	[27,28]	[0.77,0.78]

Figure 3 shows the mean network utilization for videos with different scene motion levels. As expected, network utilization increases for all mechanisms as video motion levels increase due to transmission bursts associated with changes in scenes. However, FITPATH is the least affected and yields lower network utilization for all levels of video motion, indicating it chooses more efficient paths overall and thus incurs less network load overall. Since network utilization influences video quality, proposals that better distribute the load among paths are able to deliver superior QoE. Although ERVT and QSOpt present similar results in terms of video quality as shown in Table II, they tend to incur in higher network overhead by, respectively, transmitting duplicate packets to improve error resilience and selecting paths that do not meet specific flow bitrates.

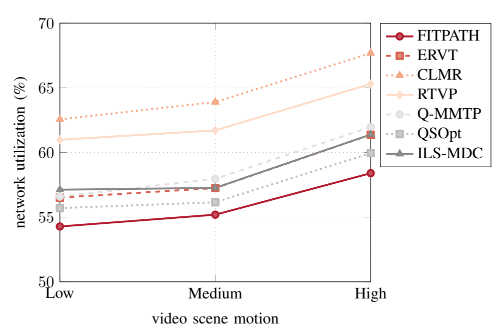

Fig. 3: Average network utilization for videos with different scene motion levels.

Table III summarizes SSIM results for all multipath selection mechanisms studied in scenarios with 60 nodes and medium-motion scenes — different numbers of nodes, loads and motion levels show similar trends. We again observe that FITPATH obtains higher video quality throughout the experiment. Additionally, FITPATH's SSIM's variance is always low, indicating that it can consistently find good solutions. While QSOpt and ERVT present results that, on average, were relatively close to FITPATH's, their variances are much larger, suggesting they more often result in inferior solutions. FITPATH's efficiency is even more evident when we look at the lowest SSIM obtained by each proposal: for low-motion videos, for example, FITPATH's lowest SSIM is 0.82, while QSOpt and ERVT both have at least one instance with an SSIM as low as 0.52. FITPATH SSIM's low variance also demonstrates that FITPATH can identify good paths in a wide range of scenarios.

TABLE III: SSIM for medium-motion levels with 60 nodes.

Mechanism	SSIM	σ^2	max	min
FITPATH	0.90	0.05	0.93	0.79
ERVT	0.85	0.25	0.93	0.61
CLMR	0.80	3.22	0.93	0.25
RTVP	0.84	0.64	0.91	0.43
Q-MMTP	0.62	10.20	0.90	0.00
QSOpt	0.87	0.32	0.93	0.69
ILS-MDC	0.85	0.31	0.92	0.70

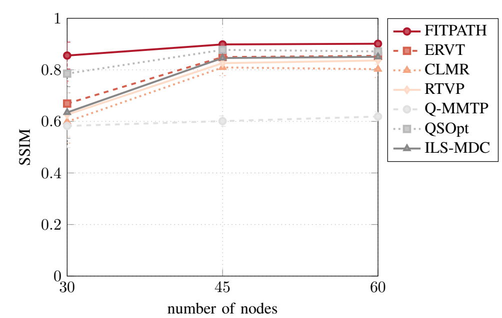

Fig. 4: SSIM for different network densities and medium scene motion levels.

In order to study the impact of network density on the performance of the different multipath selection algorithms, Figure 4 shows the average SSIM as a function of the number of nodes — for a fixed area of 300m $\times$ 300m — under medium motion level. We also ran experiments with various densities, offered loads and video motion levels, and observed similar performance trends. Due to space limitations, we do not include these results here. In general, SSIM tends to improve for all mechanisms as the number of nodes increases because of higher path diversity. Position-based approaches, in particular, leverage high network density but might not accommodate multiple video sources, which explains the challenges faced by RTVP and Q-MMTP in finding paths for all flows, resulting in lower performance. FITPATH and QSOpt, on the other hand, are relatively immune to the effects of network density since their approaches are based on the current conditions of the underlying network, regardless of the position of the nodes.

B. FITPATH Computational Overhead

We evaluate FITPATH's computational overhead by measuring its convergence time, i.e., the time it takes for FITPATH to select "adequate" paths and compare it against ERVT and QSOpt, the two top performing path selection mechanisms according to the results in Section IV-A.

To study how long it takes for FITPATH to converge and how convergence impacts video quality, we analyze how the SSIM of the solutions found by FITPATH evolves over its execution period — limited to 60 seconds. This kind of analysis can also help define FITPATH's ideal termination criterion. Figure 5 shows SSIM mean of the best solution found by FITPATH at different execution moments for all simulations with 60 nodes. We observe relatively little improvement is achieved after 40 seconds regardless of video motion level. Furthermore, after 5 seconds, FITPATH already found solutions whose average SSIM is similar to the one obtained by QSOpt, as shown by the reference blue lines in the graph. Note that QSOpt was the second best performer in all previous experiments. Moreover, QSOpt itself takes between 5 to 10 seconds to obtain a solution for this 60-node scenario — its execution time increases with the number of nodes and flows. However, differently from FITPATH, QSOpt does not provide intermediate solutions while executing. Thus, another advantage of FITPATH is that it can use early solutions to start video transmission, and paths can be updated as new solutions are found.

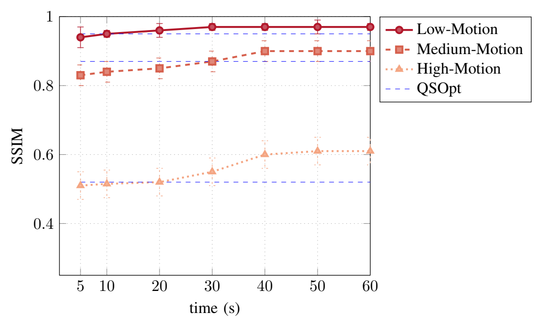

Fig. 5: FITPATH convergence: SSIM over time for scenarios with 60 nodes.

V. Conclusion

This paper introduced FITPATH, an efficient and adaptive multipath selection scheme that can accommodate environments where multiple video sources may simultaneously transmit flows of different bitrates. FITPATH uses a novel heuristic-based iterative optimization approach that estimates in real time the conditions of the underlying network while accounting for the different bitrate requirements of the application video flows. Experimental results in realistic IoT scenarios show that FITPATH outperforms various existing path selection mechanisms both in terms of user QoE and network performance. Additionally, its convergence is suitable for practical usage as it quickly generates feasible solutions with adequate performance while incrementally improving QoE with time. As future work, we will evaluate different objective functions for the ILS algorithm and more elaborate local search algorithms. We also plan to support adaptive video coding and test FITPATH alongside a proactive link-state routing protocol under dynamic topology conditions.

References

[1] F. Bhering, D. Passos, L. S. Ochi, K. Obraczka, and C. Albuquerque, "Wireless multipath video transmission: when IoT video applications meet networking—a survey," *Multimedia Systems*, pp. 1–20, 2022.

[2] S. A. Alvi, B. Afzal, G. A. Shah, L. Atzori, and W. Mahmood, "Internet of multimedia things: Vision and challenges," *Ad Hoc Networks*, vol. 33, pp. 87–111, 2015.

[3] M. Z. Hasan, H. Al-Rizzo, and F. Al-Turjman, "A Survey on Multipath Routing Protocols for QoS Assurances in Real-Time Wireless Multimedia Sensor Networks," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1424–1456, 2017.

[4] H. Shen and G. Bai, "Routing in wireless multimedia sensor networks: A survey and challenges ahead," *Journal of Network and Computer Applications*, vol. 71, pp. 30–49, 2016.

[5] C.-H. Ke, K.-W. Lin, and C.-A. Huang, "Inter-flow fairness support and enhanced video quality delivery over multi-hop wireless networks," *Computers & Electrical Engineering*, vol. 56, pp. 193–204, 2016.

[6] S. Ghahremani and M. Ghanbari, "Error resilient video transmission in ad hoc networks using layered and multiple description coding," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 9033–9049, 2017.

[7] M. Abazeed, N. Faisal, and A. Ali, "Cross-layer multipath routing scheme for wireless multimedia sensor network," *Wireless Networks*, vol. 6, 2018.

[8] A. A. Ahmed, "A real-time routing protocol with adaptive traffic shaping for multimedia streaming over next-generation of Wireless Multimedia Sensor Networks," *Pervasive and Mobile Computing*, vol. 40, pp. 495–511, 2017.

[9] M. S. Hossain, X. You, W. Xiao, J. Lu, and E. Song, "QoS-oriented multimedia transmission using multipath routing," *Future Generation Computer Systems*, vol. 99, pp. 226–234, 2019.

[10] P. T. A. Quang, K. Piamrat, K. D. Singh, and C. Viho, "Video streaming over ad hoc networks: A qoe-based optimal routing solution," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1533–1546, 2016.

[11] F. Bhering, C. Albuquerque, D. Passos, and L. S. Ochi, "Multipath selection mechanism for wireless video-surveillance systems," *9th Latin American Network Operations and Management Symposium, LANOMS*, 2019.

[12] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search: Framework and applications," in *Handbook of metaheuristics*. Springer, 2010, pp. 363–397.

[13] J. Y. Yen, "An algorithm for finding shortest routes from all source nodes to a given destination in general networks," *Quarterly of Applied Mathematics*, vol. 27, no. 4, pp. 526–530, 1970.

[14] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003, pp. 134–146.

[15] F. Bhering, D. Passos, C. Albuquerque, and K. Obraczka, "Network performance estimator with applications to route selection for iot multimedia applications," *arXiv preprint arXiv:2203.15126*, 2022.

[16] P. H. V. Penna, A. Subramanian, L. S. Ochi, T. Vidal, and C. Prins, "A hybrid heuristic for a broad class of vehicle routing problems with heterogeneous fleet," *Annals of Operations Research*, pp. 1–70, 2017.

[17] T. Clausen and P. Jacquet, "Rfc3626: Optimized link state routing protocol (olsr)," 2003.

[18] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[19] P. Natarajan, P. K. Atrey, and M. Kankanhalli, "Multi-camera coordination and control in surveillance systems: A survey," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 11, no. 4, p. 57, 2015.

[20] M. Stoffers and G. Riley, "Comparing the ns-3 propagation models," in *2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2012, pp. 61–67.

[21] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid–a framework for video transmission and quality evaluation," in *International conference on modelling techniques and tools for computer performance evaluation*. Springer, 2003, pp. 255–272.

[22] P. Orosz, T. Skopkó, Z. Nagy, P. Varga, and L. Gyimóthi, "A case study on correlating video QoS and QoE," in *2014 IEEE Network Operations and Management Symposium (NOMS)*. IEEE, 2014, pp. 1–5.