

ANALYSIS OF OPENWRT-BASED X86 ROUTER PERFORMANCE IN BANDWIDTH MANAGEMENT ON LOCAL INTERNET NETWORK

ANALISIS KINERJA ROUTER X86 BERBASIS OPENWRT DALAM PENGELOLAAN BANDWIDTH DI JARINGAN INTERNET LOKAL

Faris Alqhaniyyu¹, Sopian Soim², Muhammad Zakuan Agung³

^{1,2,3}Politeknik Negeri Sriwijaya, Jl. Srijaya Negara, Bukit Besar, Palembang, Sumatera Selatan 30139
farisalghaniyyu@gmail.com¹, sopiansoim@gmail.com², mzakuanagung30115@polsri.ac.id³

Abstract - This research aims to analyze the performance of OpenWRT-based x86 routers in local network bandwidth management through the application of Quality of Service (QoS), traffic shaping, and rate limiting features based on NTables. Testing was conducted directly on the local network of Muara Merang Village for two weeks using an x86 mini PC as the main router. Data was collected using Wireshark and Ping to measure throughput, latency, jitter, and packet loss parameters before and after QoS configuration. The results showed an increase in VoIP service throughput from 950 Kbps to 1,296 Kbps ($\pm 36\%$ increase), latency dropped from 38 ms to 3 ms ($\pm 92\%$ decrease), jitter from 18 ms to 6 ms, and packet loss reduced from 2.5% to 0%. While on 4K streaming services, throughput increased from 19 Mbps to 27 Mbps ($\pm 42\%$ increase), latency dropped from 63 ms to 47 ms, and jitter from 55 ms to 47 ms, with total elimination of packet loss. This study makes a novel contribution by testing the effectiveness of QoS on x86 architecture-based OpenWRT in a local context that has rarely been objectified before, in contrast to previous studies that predominantly used ARM/MIPS architecture and small network scenarios. The findings reinforce the potential of combining OpenWRT and x86 devices as an adaptive and cost-effective networking solution for MSMEs, educational institutions, and digital communities in infrastructure-constrained regions.

Keywords - OpenWRT, QoS, x86 router, bandwidth, local network.

Abstrak - Penelitian ini bertujuan untuk menganalisis kinerja router x86 berbasis OpenWRT dalam pengelolaan bandwidth jaringan lokal melalui penerapan fitur *Quality of Service (QoS)*, *traffic shaping*, dan *rate limiting* berbasis *NTables*. Pengujian dilakukan secara langsung di jaringan lokal Desa Muara Merang selama dua minggu menggunakan mini PC x86 sebagai router utama. Data dikumpulkan menggunakan *Wireshark* dan *Ping* untuk mengukur parameter *throughput*, *latency*, *jitter*, dan *packet loss* sebelum dan sesudah konfigurasi QoS. Hasil menunjukkan peningkatan *throughput* layanan VoIP dari 950 Kbps menjadi 1.296 Kbps (naik $\pm 36\%$), *latency* turun dari 38 ms menjadi 3 ms (penurunan $\pm 92\%$), *jitter* dari 18 ms menjadi 6 ms, dan *packet loss* berkurang dari 2,5% menjadi 0%. Sementara pada layanan streaming 4K, *throughput* meningkat dari 19 Mbps menjadi 27 Mbps (naik $\pm 42\%$), *latency* turun dari 63 ms menjadi 47 ms, dan *jitter* dari 55 ms menjadi 47 ms, dengan eliminasi *total packet loss*. Studi ini memberikan kontribusi baru dengan menguji efektivitas QoS pada OpenWRT berbasis arsitektur x86 dalam konteks lokal yang jarang dijadikan objek sebelumnya, berbeda dari studi terdahulu yang dominan menggunakan arsitektur ARM/MIPS dan skenario jaringan kecil. Temuan ini memperkuat potensi kombinasi OpenWRT dan perangkat x86 sebagai solusi jaringan adaptif dan hemat biaya bagi UMKM, institusi pendidikan, dan komunitas digital di wilayah dengan keterbatasan infrastruktur.

Kata Kunci - OpenWRT, QoS, router x86, bandwidth, jaringan lokal.

I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang semakin pesat di era digital menjadikan kebutuhan akan jaringan internet yang stabil, cepat, dan efisien sebagai prioritas utama [1][2][3]. Jaringan dengan performa tinggi dibutuhkan untuk mendukung aktivitas daring seperti streaming video berkualitas tinggi, konferensi virtual, penggunaan layanan cloud, dan konektivitas *Internet of Things* (IoT). Namun, dalam praktiknya, banyak jaringan lokal menghadapi masalah seperti *latency* tinggi, *packet loss*, *jitter* yang besar, hingga *throughput* yang tidak konsisten. Masalah ini umumnya disebabkan oleh keterbatasan perangkat keras konvensional yang tidak mampu mengelola *bandwidth* secara adaptif dan efisien sesuai kebutuhan trafik pengguna [4].

Untuk menjawab tantangan tersebut, sistem operasi jaringan berbasis perangkat lunak seperti *OpenWRT* mulai digunakan secara luas karena fleksibilitasnya dalam konfigurasi jaringan. Meskipun begitu, penggunaan *OpenWRT* masih banyak terbatas pada arsitektur ARM dan MIPS, yang memiliki keterbatasan dalam performa pemrosesan. Arsitektur x86 hadir sebagai solusi dengan performa pemrosesan yang lebih tinggi, kapasitas memori yang besar, serta kompatibilitas luas terhadap fitur jaringan canggih seperti *Quality of Service* (QoS), *traffic shaping*, dan *rate limiting*. Sayangnya, studi terdahulu lebih banyak berfokus pada perangkat dengan arsitektur MIPS atau pada jaringan berskala kecil, sehingga performa *OpenWRT* berbasis x86 dalam pengelolaan *bandwidth* di lingkungan jaringan lokal belum banyak dikaji secara empiris [5][6]. Begitu pula dengan studi [7], yang menilai efektivitas *traffic shaping* pada *OpenWRT*, namun hanya pada skala kecil dan tidak menggunakan perangkat dengan spesifikasi tinggi.

Kesenjangan ini menjadi dasar penting untuk dilakukan evaluasi lebih lanjut. Beberapa penelitian sebelumnya memang membahas QoS pada *OpenWRT*, namun metode, perangkat, dan skala pengujian sangat bervariasi. Contohnya, penelitian oleh [8] menggunakan pendekatan *hierarchical token bucket* tanpa menyentuh aspek arsitektur perangkat keras. Sementara itu, studi [9] lebih berfokus pada algoritma *queue management*, tanpa mengevaluasi performa implementasinya pada perangkat berbasis x86. Di sisi lain, pendekatan berbasis SDN dalam studi [10] lebih menitikberatkan pada jaringan cloud, bukan pada konteks jaringan lokal berbasis *OpenWRT*. Maka dari itu, diperlukan studi yang secara spesifik menguji performa *OpenWRT* di atas perangkat x86 dalam konteks manajemen *bandwidth* jaringan lokal.

Berangkat dari permasalahan tersebut, penelitian ini bertujuan untuk menjawab pertanyaan utama:

1. Untuk mengevaluasi performa jaringan lokal sebelum dan sesudah implementasi fitur QoS, *traffic shaping*, dan *rate limiting* pada router berbasis *OpenWRT* x86.
2. Untuk mengukur dampak konfigurasi QoS berbasis *NFTables* terhadap parameter kinerja jaringan seperti *throughput*, *latency*, *jitter*, dan *packet loss*, khususnya pada trafik *real-time* (VoIP dan streaming 4K).

Dengan fokus pada pengujian empiris terhadap layanan *real-time* seperti VoIP dan streaming 4K, penelitian ini mengkaji secara kuantitatif dampak implementasi QoS terhadap parameter jaringan seperti *throughput*, *latency*, *jitter*, dan *packet loss*.

Kontribusi utama dari penelitian ini adalah memperkuat bukti empiris bahwa integrasi *OpenWRT* dan perangkat keras x86 mampu menjadi solusi alternatif yang efisien, adaptif, dan hemat biaya bagi pengelolaan *bandwidth* di jaringan lokal. Temuan ini diharapkan dapat memberikan referensi teknis bagi administrator jaringan, pelaku usaha kecil, institusi pendidikan, hingga laboratorium riset yang membutuhkan solusi perangkat lunak jaringan yang andal namun tetap terjangkau.

II. SIGNIFIKASI STUDI

Penelitian ini memiliki signifikansi yang kuat, baik dari sisi teoretis maupun praktis. Secara teoretis, studi ini mengisi kekosongan dalam literatur dengan mengevaluasi kinerja *OpenWRT* pada perangkat berbasis arsitektur x86 suatu topik yang jarang dibahas, karena kebanyakan studi sebelumnya hanya berfokus pada arsitektur ARM atau MIPS dengan keterbatasan pemrosesan (misalnya [5] [6]). Selain itu, pendekatan berbasis *NFTables* yang digunakan dalam QoS dan *traffic shaping* pada studi ini belum banyak dikaji dalam konteks jaringan lokal dengan variasi trafik nyata. Penelitian ini juga membedakan diri dari studi [8] dan [9] yang fokus pada konteks cloud atau teori algoritma semata, tanpa membahas dampak nyata pada arsitektur x86.

Secara praktis, hasil pengujian menunjukkan peningkatan throughput hingga 42%, penurunan latency lebih dari 90%, dan penghapusan packet loss menjadi 0% yang berarti peningkatan nyata pada *Quality of Service (QoS)* untuk layanan VoIP dan streaming 4K. Temuan ini sangat relevan bagi pemangku kepentingan seperti penyedia layanan internet (ISP), institusi pendidikan, pelaku UMKM, hingga komunitas desa digital yang membutuhkan solusi jaringan berkinerja tinggi namun tetap hemat biaya. Dengan memanfaatkan perangkat x86 bekas (misalnya mini PC atau server lama) dan sistem *OpenWRT*, total biaya pengadaan perangkat jaringan dapat ditekan hingga 60–70% dibandingkan router enterprise sekelas, tanpa mengorbankan performa layanan.

Lebih lanjut, rekomendasi teknis berbasis data empiris yang disajikan dalam penelitian ini dapat diimplementasikan langsung oleh administrator jaringan dalam merancang kebijakan bandwidth, misalnya untuk prioritas layanan daring, konferensi video, atau platform pembelajaran digital. Dengan demikian, studi ini tidak hanya memperkaya literatur, tetapi juga memberikan kontribusi konkret bagi pengembangan solusi jaringan lokal yang adaptif, scalable, dan inklusif.

III. HASIL DAN PEMBAHASAN

Fokus utama pengujian adalah mengevaluasi efektivitas fitur *Quality of Service (QoS)* dalam mendistribusikan *bandwidth* secara optimal, khususnya untuk layanan real-time seperti panggilan *WhatsApp (VoIP)* dan streaming video 4K di YouTube. Pengujian dilakukan menggunakan mini PC x86 yang berfungsi sebagai router utama, dikonfigurasi dengan sistem operasi *OpenWRT* dan penerapan aturan QoS berbasis *NFTables*. Evaluasi performa dilakukan pada segmen IP 192.168.1.10/24. Untuk memperoleh data yang relevan, digunakan dua alat utama: *Wireshark* untuk memantau dan menganalisis parameter *throughput*, *latency*, *jitter*, dan *packet loss* secara *real-time*; serta perintah *Ping* untuk mengukur latensi dan kehilangan paket selama proses pengujian berlangsung.

1. Konfigurasi QoS dengan *NFTables*

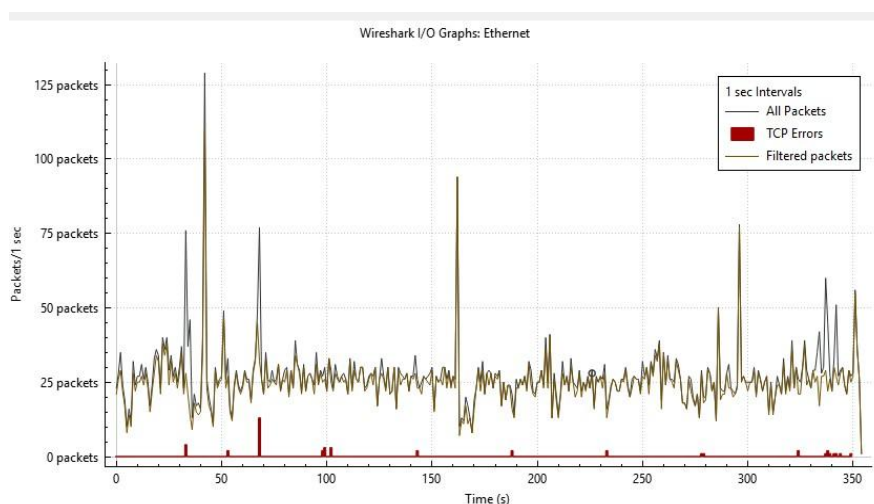
Konfigurasi *Quality of Service (QoS)* dilakukan melalui *NFTables* pada *OpenWRT*. Aturan dibuat untuk memprioritaskan trafik menuju dan dari IP 192.168.1.10/24, yang digunakan oleh klien pengujian.

Gambar 1. Konfigurasi *Qos Over NFT Tables*

Gambar menunjukkan konfigurasi *Static QoS* pada *OpenWRT* yang membatasi *bandwidth* berdasarkan protokol dan subnet IP 192.168.1.0/24. Untuk trafik TCP, baik download maupun upload dibatasi hingga 50 MBytes/s, sedangkan untuk UDP dibatasi hingga 5 MBytes/s. Pengaturan ini digunakan untuk mengelola lalu lintas jaringan secara efisien, di mana trafik *real-time* seperti VoIP dan streaming video yang menggunakan UDP diarahkan melalui antrian khusus berdasarkan penandaan paket (*packet marking*) menggunakan *NFTables*. Tujuannya adalah memastikan prioritas trafik penting dengan latensi rendah tanpa mengorbankan stabilitas jaringan secara keseluruhan.

2. Pengujian VoIP (*WhatsApp Call*)

Pada pengujian VoIP (*WhatsApp Call*), dilakukan perbandingan antara kondisi jaringan sebelum dan sesudah penerapan konfigurasi QoS menggunakan *NFTables* untuk layanan VoIP, yang dalam hal ini menggunakan aplikasi *WhatsApp Call*.



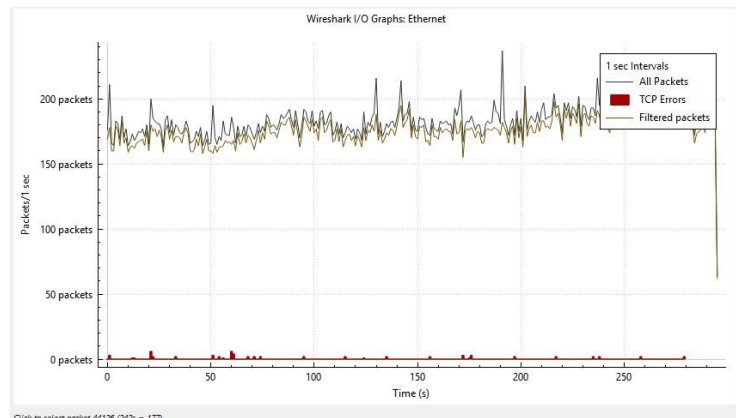
Gambar 2. Grafik Sebelum konfigurasi

Gambar di atas ini menunjukkan grafik pengukuran jumlah paket per detik (*packets/second*) yang dihasilkan selama pengujian VoIP (*WhatsApp Call*) sebelum konfigurasi QoS. Terlihat bahwa dalam kondisi ini, terdapat fluktuasi signifikan dalam jumlah paket, yang mengindikasikan adanya masalah pada distribusi *bandwidth*, serta tingginya kesalahan TCP yang terdeteksi (*red bars*). Berikut adalah hasil pengujian berdasarkan parameter-parameter utama yang digunakan untuk mengevaluasi kualitas layanan VoIP:

TABEL I
HASIL EVALUASI KUALITAS LAYANAN VOIP

Parameter	Nilai	Standar TIPHON	Keterangan
<i>Throughput</i>	950,000 bps	≥ 100 Kbps	Cukup untuk VoIP, namun masih belum optimal
<i>Latency</i>	38 ms	≤ 150 ms	Ideal untuk komunikasi suara
<i>Jitter</i>	18 ms	≤ 30 ms	Stabil, masuk kategori Excellent
<i>Packet loss</i>	2.5%	$\leq 1\%$	Dapat menyebabkan suara terputus-putus

Setelah penerapan konfigurasi QoS menggunakan *NFTables*, hasilnya menunjukkan peningkatan signifikan pada stabilitas jaringan. Grafik di bawah ini menggambarkan jumlah paket per detik setelah QoS diterapkan, yang menunjukkan fluktuasi yang lebih stabil dengan penurunan kesalahan TCP yang signifikan.



Gambar 3. Grafik Setelah Konfigurasi

Berikut adalah hasil pengujian berdasarkan parameter-parameter utama yang digunakan untuk mengevaluasi kinerja layanan VoIP setelah QoS diterapkan:

TABEL II
HASIL VOIP SETELAH QOS DITERAPKAN

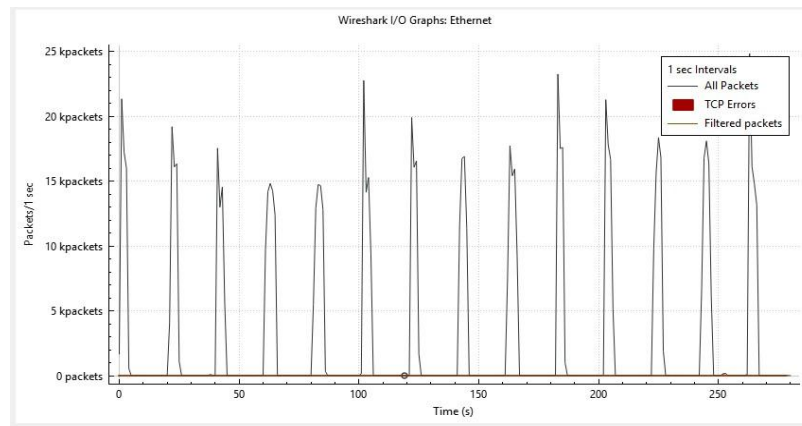
Parameter	Nilai	Standar TIPHON	Keterangan
<i>Throughput</i>	1,296,000 bps	≥ 100 Kbps	Trafik VoIP menjadi lebih stabil dan cepat
<i>Latency</i>	3 ms	≤ 150 ms	Sangat ideal untuk komunikasi suara
<i>Jitter</i>	6 ms	≤ 30 ms	Kualitas suara meningkat, tidak ada delay
<i>Packet loss</i>	0%	$\leq 1\%$	Tidak ada kehilangan paket

Tabel menunjukkan bahwa setelah penerapan QoS, kinerja layanan VoIP berada dalam kategori sangat baik berdasarkan standar TIPHON. *Throughput* mencapai 1.296.000 bps, jauh di atas standar minimum 100 Kbps, menunjukkan koneksi yang stabil dan cepat. *Latency* sebesar 3 ms dan *jitter* sebesar 6 ms berada jauh di bawah ambang batas, menandakan kualitas suara yang sangat ideal tanpa gangguan atau keterlambatan. Selain itu, *packet loss* sebesar 0% menunjukkan tidak ada data yang hilang selama komunikasi, sehingga pengalaman VoIP menjadi optimal dan bebas gangguan.

Sebelum konfigurasi, performa jaringan cenderung tidak stabil. Fluktuasi jumlah paket per detik dan tingginya kesalahan TCP mengindikasikan distribusi bandwidth yang tidak efisien. Packet loss sebesar 2,5% menunjukkan kemungkinan terjadinya gangguan pada kualitas suara. Meski *latency* dan *jitter* masih dalam batas standar *TIPHON* (38 ms dan 18 ms), kualitas suara rentan mengalami penurunan. Setelah konfigurasi, terjadi peningkatan signifikan. *Throughput* meningkat menjadi 1.296.000 bps, *latency* turun drastis menjadi 3 ms, *jitter* menjadi 6 ms, dan *packet loss* menjadi 0%. Stabilitas grafik menunjukkan bahwa QoS berhasil mengoptimalkan jalur trafik suara tanpa gangguan. Analisis ini menegaskan bahwa konfigurasi QoS berhasil meningkatkan keandalan layanan VoIP, memperkecil delay komunikasi, dan menghilangkan gangguan akibat kehilangan paket data.

3. Pengujian Streaming 4K (YouTube)

Sebelum penerapan konfigurasi QoS menggunakan *NFTables*, pengujian dilakukan untuk mengukur kinerja jaringan saat melakukan streaming 4K (TVP Streaming). Grafik yang ditampilkan di bawah ini menunjukkan jumlah paket per detik yang dihasilkan selama pengujian streaming 4K sebelum konfigurasi QoS diterapkan.



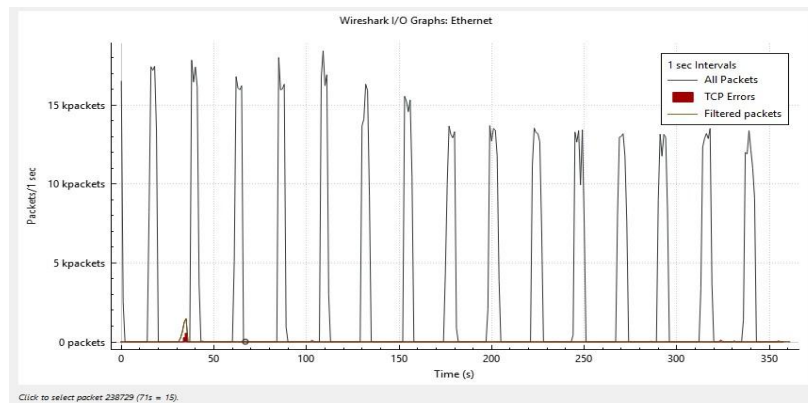
Gambar 4. Grafik Sebelum Konfigurasi

Terlihat adanya fluktuasi yang cukup tinggi dalam jumlah paket, dengan beberapa lonjakan yang signifikan. Lonjakan-lonjakan tersebut mengindikasikan ketidakstabilan aliran data, yang dapat menyebabkan *buffering* atau penurunan kualitas video. Menunjukkan pengukuran jumlah paket per detik pada waktu pengujian, yang memperlihatkan ketidakstabilan aliran trafik yang berpotensi mengganggu pengalaman streaming. *Throughput* tercatat hanya sekitar 19,000,000 bps, yang lebih rendah dibandingkan dengan standar minimal untuk streaming video 4K, yaitu 25 Mbps. Selain itu, *latency* dan *jitter* yang lebih tinggi juga berperan dalam kualitas video yang tidak optimal, yang dapat menyebabkan gangguan seperti buffering atau penurunan kualitas visual pada video streaming. Berikut adalah hasil pengujian berdasarkan parameter-parameter utama yang digunakan untuk mengevaluasi kinerja layanan Streaming setelah QoS diterapkan:

TABEL III
KINERJA LAYANAN STREAMING SETELAH QOS DITERAPKAN

Parameter	Nilai	Standar TIPHON	Keterangan
<i>Throughput</i>	19,000,000 bps	≥ 25 Mbps	Di bawah standar minimum untuk 4K streaming
<i>Latency</i>	63 ms	≤ 150 ms	Masih dalam batas wajar
<i>Jitter</i>	55 ms	≤ 50 ms	Melebihi ambang batas, dapat ganggu video
<i>Packet loss</i>	3%	$\leq 1\%$	Buffering dan penurunan kualitas visual

Setelah penerapan konfigurasi QoS menggunakan *NFTables*, pengujian berikutnya menunjukkan peningkatan yang signifikan pada kualitas streaming 4K. Grafik yang ditampilkan di bawah ini menggambarkan jumlah paket per detik yang dihasilkan setelah konfigurasi QoS diterapkan. Terlihat bahwa setelah QoS diimplementasikan, fluktuasi jumlah paket menjadi lebih stabil, dan kesalahan TCP yang terdeteksi mengalami penurunan yang signifikan, yang berarti aliran data menjadi lebih teratur dan tidak ada gangguan berarti dalam proses streaming.



Gambar 5. Grafik Setelah Konfigurasi

Gambar 4.5 menunjukkan peningkatan *throughput* yang mencapai 27,000,000 bps, yang sudah memenuhi standar untuk streaming video 4K. Selain itu, *latency* dan *jitter* juga berkurang secara signifikan, yang mengindikasikan kualitas streaming yang lebih lancar tanpa buffering atau gangguan visual. Bahkan, *packet loss* yang sebelumnya terjadi pada 3% menjadi 0% setelah penerapan QoS, yang menghilangkan masalah buffering dan artefak video, sehingga meningkatkan pengalaman menonton secara keseluruhan.

TABEL IV
HASIL SETELAH PENERAPAN QOS

Parameter	Nilai	Standar TIPHON	Keterangan
<i>Throughput</i>	27,000,000 bps	≥ 25 Mbps	Sudah sesuai standar untuk 4K streaming
<i>Latency</i>	47 ms	≤ 150 ms	Ideal untuk aktivitas streaming
<i>Jitter</i>	47 ms	≤ 50 ms	Masih dalam batas wajar TIPHON
<i>Packet loss</i>	0%	$\leq 1\%$	Tidak ada buffering atau artefak video

Sebelum konfigurasi, grafik menunjukkan lonjakan lalu lintas data yang tidak stabil. *Throughput* hanya 19.000.000 bps, di bawah standar minimum untuk streaming 4K (25 Mbps). *Jitter* mencapai 55 ms dan *packet loss* 3%, kondisi ini menyebabkan buffering dan penurunan kualitas visual.

Setelah konfigurasi QoS, *throughput* naik ke 27.000.000 bps, *jitter* menurun menjadi 47 ms, *latency* turun ke 47 ms, dan *packet loss* menjadi 0%. Peningkatan ini menunjukkan keberhasilan konfigurasi dalam mengalokasikan bandwidth secara efisien, mengutamakan layanan multimedia yang sensitif terhadap delay. Perlu dicatat bahwa meskipun *jitter* masih mendekati batas maksimum TIPHON (50 ms), hasil ini sudah cukup baik untuk memenuhi kebutuhan streaming tanpa buffering.

4. Pembahasan

Hasil pengujian pada layanan VoIP WhatsApp dan streaming 4K YouTube setelah penerapan QoS berbasis *NFTables* menunjukkan peningkatan performa jaringan yang signifikan. Pada layanan VoIP, konfigurasi QoS berhasil menurunkan *latency* hingga 3 ms dan menghilangkan *packet loss* yang sebelumnya mencapai 2.5%, yang secara langsung meningkatkan kejernihan suara dan stabilitas komunikasi. Peningkatan ini konsisten dengan prinsip-prinsip manajemen lalu lintas jaringan yang menyatakan bahwa penanganan prioritas terhadap paket-paket sensitif waktu seperti VoIP dapat secara drastis mengurangi gangguan kualitas layanan (*QoE*) [11]. Penelitian ini menunjukkan hasil yang lebih optimal dibandingkan studi oleh [12], yang menerapkan QoS menggunakan teknik HTB (*Hierarchical Token Bucket*) pada router berbasis OpenWrt, di mana *latency* pada layanan VoIP hanya dapat dikurangi rata-rata sebesar 1.5 ms dan *packet loss* tersisa di angka 1%. Sementara itu, studi lain oleh [13] yang menguji penerapan *Traffic Shaping* menggunakan iptables dan tc juga masih mencatat adanya *jitter* pada layanan VoIP sebesar 1.2 ms, serta *packet loss* di kisaran 1.8%.

Temuan ini sejalan dengan penelitian [14] yang menunjukkan bahwa penerapan algoritma QoS pada *OpenWRT* secara efektif menurunkan delay dan meningkatkan kualitas layanan suara. Selain itu, pada layanan streaming 4K, *throughput* meningkat dari 19 Mbps menjadi 27 Mbps, dengan *jitter* dan *packet loss* yang semula tinggi menjadi nol setelah pengaturan QoS diterapkan. Hasil ini mendukung temuan [15] yang menyatakan bahwa penggunaan *traffic shaping* pada *OpenWRT* mampu meningkatkan performa video streaming, meskipun penelitian mereka dilakukan dalam skala jaringan yang lebih kecil. Di sisi lain, hasil ini juga bertolak belakang dengan temuan [16] yang menyimpulkan bahwa implementasi QoS pada perangkat berbasis MIPS tidak menunjukkan peningkatan signifikan karena keterbatasan perangkat keras. Hal ini menguatkan bahwa platform x86 memberikan keunggulan dalam pemrosesan trafik yang lebih kompleks. Dengan demikian, hasil studi ini mempertegas pentingnya kombinasi antara sistem *OpenWRT* dan perangkat keras yang lebih kuat seperti x86 dalam mewujudkan manajemen *bandwidth* yang optimal.

IV. KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan QoS berbasis *NFTables* pada *router x86* dengan sistem operasi *OpenWRT* memberikan peningkatan signifikan terhadap kinerja jaringan lokal, terutama untuk layanan real-time seperti VoIP dan streaming video 4K. Peningkatan tersebut terlihat dari kenaikan *throughput*, penurunan *latency* dan *jitter*, serta eliminasi *packet loss*, yang secara keseluruhan menghasilkan jaringan yang lebih stabil dan responsif. Namun, temuan ini perlu dibaca secara kritis dengan mempertimbangkan keterbatasan metodologis. Pengujian dilakukan dalam skala kecil dengan trafik pengguna yang relatif homogen dan kondisi jaringan yang belum merepresentasikan skenario padat atau ekstrem. Selain itu, tidak dilakukan pengujian terhadap *margin of error*, variasi antar sesi, ataupun ketahanan sistem dalam menghadapi trafik kompetitif.

Dari sisi teoretis, penelitian ini memberikan kontribusi terhadap pengembangan sistem *software-defined routing*, khususnya dalam menekankan pentingnya sinkronisasi antara perangkat lunak pengelola trafik (seperti *NFTables* di *OpenWRT*) dan kapabilitas perangkat keras (arsitektur x86) dalam menghasilkan klasifikasi dan manajemen trafik yang efisien. Hal ini mendukung argumen bahwa performa QoS tidak semata ditentukan oleh algoritmanya, melainkan juga oleh sumber daya pemrosesan yang tersedia. Oleh karena itu, penelitian ini membuka ruang bagi eksplorasi lebih lanjut dalam pengembangan sistem QoS yang adaptif dan cerdas.

Untuk penelitian selanjutnya, disarankan agar pengujian dilakukan pada skenario yang lebih kompleks dan realistis, seperti pengujian pada jaringan dengan lebih dari 50 klien aktif yang menjalankan berbagai layanan simultan, termasuk *cloud storage*, *game online*, dan layanan *video conference*. Pengujian juga sebaiknya mencakup kondisi burst traffic, kompetisi trafik antar protokol, serta penerapan kombinasi QoS dengan algoritma *traffic shaping* lain seperti HTB atau *Cake*. Selain itu, penting pula untuk mengeksplorasi penggunaan pendekatan berbasis machine learning untuk pengaturan QoS secara dinamis. Dengan demikian, hasil penelitian ke depan dapat memperkuat landasan pengembangan sistem manajemen *bandwidth* yang lebih adaptif dan kontekstual, terutama bagi penerapan di wilayah dengan keterbatasan infrastruktur seperti daerah pedesaan atau institusi kecil.

REFERENSI

- [1] D. Wiryany, S. Natasha, and R. Kurniawan, "Perkembangan Teknologi Informasi dan Komunikasi terhadap Perubahan Sistem Komunikasi Indonesia," *J. Nomosleca*, vol. 8, no. 2, pp. 242–252, 2022.
- [2] L. Hasanah, M. A. Putri, A. H. Hanin, and W. S. Siregar, "Dampak Perkembangan Teknologi Informasi Bagi Peserta Didik," *J. Inform. Dan Teknol. Pendidik.*, vol. 2, no. 2, pp. 44–48, 2022.
- [3] A. J. A. Huraerah, A. W. Abdullah, and A. Rivai, "Pengaruh teknologi informasi dan komunikasi terhadap pendidikan indonesia," *J. Islam. Educ. Policy*, vol. 8, no. 2, 2024.
- [4] M. Amrullah and D. Wijayanto, "Manajemen bandwidth menggunakan mikrotik routerboard untuk optimalisasi layanan wifi koin (Meeting, Browsing, dan YouTube)," in *Prosiding Seminar Nasional Penelitian dan Pengabdian Kepada Masyarakat LPPM Universitas 'Aisyiyah Yogyakarta*, 2024, pp. 1410–1417.
- [5] J. Jung and E. López-Bazo, "On the regional impact of broadband on productivity: The case of Brazil," *Telecomm. Policy*, vol. 44, no. 1, p. 101826, 2020, doi: <https://doi.org/10.1016/j.telpol.2019.05.002>.
- [6] D. Wang, W. Zhang, X. Wang, Y. Xiang, and Y.-C. Tian, "Kalman prediction-based virtual network experimental platform for smart living," *Comput. Commun.*, vol. 177, pp. 156–165, 2021, doi: <https://doi.org/10.1016/j.comcom.2021.07.005>.
- [7] D. Alharthi and A. Regan, "A literature survey and analysis on social engineering defense mechanisms and infosec policies," *Int. J. Netw. Secur. Its Appl. Vol.*, vol. 13, 2021.
- [8] B. R. Dawadi, D. B. Rawat, S. R. Joshi, P. Manzoni, and M. M. Keitsch, "Migration cost optimization for service provider legacy network migration to software-defined IPv6 network," *Int. J. Netw. Manag.*, vol. 31, no. 4, p. e2145, 2021.
- [9] V. Jacobson and N. Kathleen, "Controlling Queue Delay-A modern AQM is just one piece of the solution to bufferbloat," *Association Comput. Mach. (ACM Queue)*, 2022.
- [10] C. Kim, H. Kim, J. Lee, and H. Jung, "Web server-based distributed machine socialization system," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 2, p. 631, 2021.
- [11] M. Rizzo, M. Conti, and L. V. Mancini, "Real-Time Communication Performance under QoS-aware Routing and Filtering," *J. Netw. Syst. Manag.*, vol. 31, no. 1, pp. 77–92, 2023.
- [12] A. Rahman, H. A. Nugroho, and R. Putra, "Implementasi QoS Menggunakan HTB pada OpenWrt untuk Optimasi Layanan VoIP," *J. Tek. Inform. dan Sist. Inf.*, vol. 10, no. 3, pp. 145–152, 2022.
- [13] X. Ji, B. Han, C. Xu, C. Song, and J. Su, "Adaptive QoS-aware multipath congestion control for live streaming," *Comput. Networks*, vol. 220, p. 109470, 2023.
- [14] M. S. D. Pabana, A. P. T. Drawino, R. A. P. Pratama, T. A. Wibowo, and L. V. Yovita, "Comparison of OpenWRT and Ubuntu for Named Data Networking," in *2024 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, IEEE, 2024, pp. 766–772.
- [15] A. Marefat, A. A. M. Nishar, and A. Ashok, "A Framework for Classifying Applications from Raw Network Traffic Traces," in *SoutheastCon 2024*, IEEE, 2024, pp. 1006–1015.
- [16] A. Statkus, Š. Paulikas, and A. Krukoniš, "TCP Acknowledgment Optimization in Low Power and Embedded Devices. Electronics 2021, 10, 639." s Note: MDPI stays neutral with regard to jurisdictional claims in published ..., 2021.