**Proses Video Streaming Dengan Protokol SRT: Studio Kasus Menggunakan OBS Studio dan Titan Edge**

Dimas Tri Handika, Andy Haryoko

[dimastrihandika@gmail.com](mailto:dimastrihandika@gmail.com), [andyharyoko@gmail.com](mailto:andyharyoko@gmail.com)

Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Terbuka

E-mail: [*043260075@ecampus.ut.ac.id*](mailto:043260075@ecampus.ut.ac.id)

**ABSTRAK**

Proses video streaming merupakan aspek krusial dalam distribusi konten multimedia secara real-time. Dalam kaitannya, protokol Source Reliable Transport (SRT) telah muncul sebagai solusi yang menjanjikan untuk meningkatkan keandalan dan efisiensi pengiriman data. Penelitian ini bertujuan untuk mengeksplorasi dan menganalisis proses video streaming menggunakan protokol SRT, dengan fokus pada aspek-aspek kritis seperti latensi, stabilitas dan troughput. Penelitian ini membahas implementasi SRT pada proses video streaming dengan fokus pada evaluasi kinerja dan keunggulan protokol tersebut dibandingkan dengan RTMP. Hasil penelitian diharapkan dapat memberikan pandangan yang mendalam tentang potensi SRT dalam mendukung pengalaman pengguna yang lebih baik dalam video streaming.

***Kata kunci:***

**PENDAHULUAN**

Dengan pertumbuhan layanan streaming online dalam beberapa dekade terakhir dan peningkatan permintaan konten video berkualitas tinggi, terutama dalam konteks penyiaran langsung, diperlukan solusi transportasi data yang dapat menangani tantangan seperti kehilangan paket dan variasi jaringan (Bienik, et al.., 2023). Streaming melalui protokol berbasis TCP seperti RTMP memungkinkan di bitrate rendah dan dalam skala kecil, pertumbuhan eksponensial pasar streaming telah menciptakan kebutuhan akan solusi yang lebih andal. Protokol SRT menawarkan pendekatan inovatif dengan menggabungkan keandalan TCP dan efisiensi UDP. Meskipun SRT telah diterapkan dalam berbagai konteks, belum banyak penelitian yang merinci proses implementasi dan dampak penerapannya secara khusus dalam proses streaming video.

Rumusan Masalah:

* Bagaimana kinerja protokol SRT dan RTMP dalam hal keandalan, latensi dan efisiensi pengiriman data?
* Bagaimana konfigurasi latensi pada protokol SRT yang optimal untuk memaksimalkan kinerja video streaming?

Tujuan Penulisan:

Penelitian ini bertujuan untuk menginvestigasi dan menganalisis kinerja protokol SRT dalam konteks proses video streaming. Penulis akan mengevaluasi dampak implementasi SRT terhadap stabilitas, latensi, dan kualitas pengiriman data. Tujuan utama adalah menyajikan wawasan yang mendalam tentang potensi SRT sebagai solusi yang andal untuk kebutuhan video streaming modern.

Manfaat Penulisan:

Penelitian ini diharapkan memberikan kontribusi dalam pemahaman lebih lanjut tentang potensi Protokol SRT dalam meningkatkan keandalan video streaming. Hasil penelitian dapat digunakan oleh pengembang aplikasi, penyedia layanan streaming, dan peneliti lainnya yang tertarik dalam pengoptimalan protokol transport untuk pengiriman data video real-time.

Metode Penulisan:

Penulisan dilakukan melalui studi pustaka untuk memahami dasar teori Protokol SRT sebagai fokus penelitian. Selanjutnya, eksperimen praktis menggunakan Titan Edge sebagai Decoder dan OBS Studio sebagai Encoder akan dilakukan untuk mengukur kinerja Protokol dalam pengiriman data video real-time, dengan memperhatikan variabel-variabel seperti latensi, kestabilan, dan throughput. Analisis statistik akan digunakan untuk membandingkan hasil eksperimen dan menyimpulkan temuan penelitian.

**TINJAUAN PUSTAKA**

**2.1 UDT (UDP Based Data Transfer Protokol)**

Protokol UDT, atau UDP-based Data Transfer Protocol, adalah sebuah protokol transfer data berbasis UDP yang dikembangkan untuk mengatasi beberapa keterbatasan yang dimiliki oleh protokol Transmission Control Protocol (TCP), terutama dalam konteks pengiriman data real-time seperti video streaming. UDT Dirancang untuk aplikasi yang membutuhkan keseimbangan antara kecepatan dan keandalan. Berikut ini adalah pemaparan lebih detail tentang protokol UDT dalam konteks pengiriman data video real-time:

1. Karakteristik Utama:

* Kontrol Aliran Sendiri: Protokol UDT menyediakan mekanisme pengendalian aliran sendiri (congestion control) untuk mengatasi masalah kehilangan paket dan delay dalam jaringan, tanpa bergantung pada mekanisme pengendalian aliran dari TCP.
* Pengiriman Data Real-Time: UDT dirancang untuk mendukung aplikasi yang membutuhkan pengiriman data real-time atau near real-time, seperti video streaming, video conference, dan aplikasi multimedia interaktif.
* Dukungan untuk Pengiriman Data Besar: UDT mampu mengatasi pengiriman data besar dengan efisien melalui fitur pengaturan ukuran paket yang dapat disesuaikan (adaptive packet size).
* Pengaturan Kualitas Layanan (QoS): Protokol ini memungkinkan aplikasi untuk mengatur parameter kualitas layanan (QoS) sesuai dengan kebutuhan, contohnya seperti latency, throughput, dan keandalan.

2. Mekanisme Pengendalian Aliran:

* Window-based Congestion Control: UDT menggunakan pendekatan pengendalian aliran berbasis jendela (window-based) untuk menyesuaikan laju pengiriman data berdasarkan kondisi jaringan.
* Adaptive Timeout: Protokol ini mengimplementasikan mekanisme timeout yang dapat disesuaikan untuk menanggapi kondisi jaringan yang berubah-ubah.

3. Keamanan dan Enkripsi:

* Keamanan Tambahan: UDT sendiri tidak menyediakan enkripsi data, tetapi keamanan dapat ditingkatkan dengan menggunakan lapisan keamanan tambahan seperti Transport Layer Security (TLS) atau Secure Real-time Transport Protocol (SRTP) jika diperlukan.

4. Implementasi:

* Pustaka dan Perangkat Lunak: Ada beberapa implementasi pustaka UDT yang tersedia untuk berbagai platform, memungkinkan pengembang untuk mengintegrasikan protokol ini ke dalam aplikasi mereka.
* Ketergantungan Terhadap Jaringan: Kinerja UDT sangat tergantung pada kondisi jaringan, dan pengguna perlu memahami karakteristik jaringan tempat aplikasi dijalankan.

Kesimpulan:

Protokol UDT memberikan solusi untuk pengiriman data real-time yang membutuhkan efisiensi dan ketahanan terhadap delay. UDT memungkinkan pengguna untuk mengontrol dan menyesuaikan kualitas layanan sesuai dengan kebutuhan aplikasi mereka. Implementasi yang tepat dan pengaturan parameter yang cermat diperlukan untuk memaksimalkan kinerja UDT dalam konteks pengiriman data video real-time.

**2.2 SRT (Source Reliable Transport)**

Transmisi video streaming dengan latensi rendah melalui jaringan IP yang dapat diandalkan, biasanya di lingkungan lokal, umumnya menggunakan format MPEG-TS [ISO13818-1]. Aliran dapat berupa unicast atau multicast dengan menggunakan protokol UDP/RTP. Menyamakan tingkat latensi di antara lokasi yang berbeda, seperti kota, negara, atau bahkan benua, merupakan suatu tantangan. Penggunaan satelit atau jaringan MPLS khusus [RFC3031] dapat memungkinkan hal ini, tetapi solusi ini tergolong mahal. Meskipun konektivitas Internet publik lebih terjangkau, diperlukan overhead bandwidth yang signifikan untuk mencapai tingkat pemulihan kehilangan paket yang dibutuhkan. Pengenalan transmisi ulang paket selektif (UDP yang dapat diandalkan) membantu mengatasi batasan ini.

SRT, yang berasal dari protokol Transfer Data (UDT) berbasis UDP [GHG04b], merupakan protokol tingkat pengguna yang tetap mempertahankan konsep dan mekanisme dasar sambil mengenalkan sejumlah perbaikan dan peningkatan. Perubahan tersebut mencakup modifikasi pada paket kontrol, peningkatan kontrol aliran untuk mengelola streaming langsung, perbaikan kontrol kemacetan, dan mekanisme untuk mengenkripsi paket.

SRT dirancang sebagai protokol transport untuk menjamin keamanan dan keandalan pengiriman data melalui jaringan yang tidak dapat diprediksi, seperti Internet. SRT sangat ideal untuk streaming video dengan latensi rendah. Protokol ini meningkatkan efisiensi bandwidth jika dibandingkan dengan RTMP, memungkinkan kontribusi bitrate yang lebih tinggi melalui koneksi jarak jauh.

Saat mengirimkan paket dari sumber ke tujuan, SRT secara real-time mendeteksi dan menyesuaikan diri dengan kondisi jaringan antara kedua titik akhir. Hal ini membantu mengatasi fluktuasi jitter dan bandwidth yang mungkin terjadi pada jaringan. Mekanisme pemulihan kesalahan SRT berperan dalam mengurangi kehilangan paket yang umumnya terjadi dalam koneksi Internet.

Untuk mencapai streaming dengan latensi rendah, SRT perlu mengatasi tantangan waktu yang timbul saat aliran data melintasi Internet publik. SRT dilengkapi dengan mekanisme yang memastikan agar latensi end-to-end tetap konsisten, mengoptimalkan sinyal di sisi penerima, dan mengurangi kebutuhan buffering.

Seperti TCP, SRT mengadopsi model listener/caller, memungkinkan arus data dua arah tanpa bergantung pada inisiasi koneksi. Protokol ini mendukung multiplexing internal, memungkinkan beberapa koneksi SRT untuk berbagi port UDP yang sama, dengan kontrol akses yang memudahkan identifikasi caller di sisi listener

**2.3 Konsep Dasar Video Streaming**

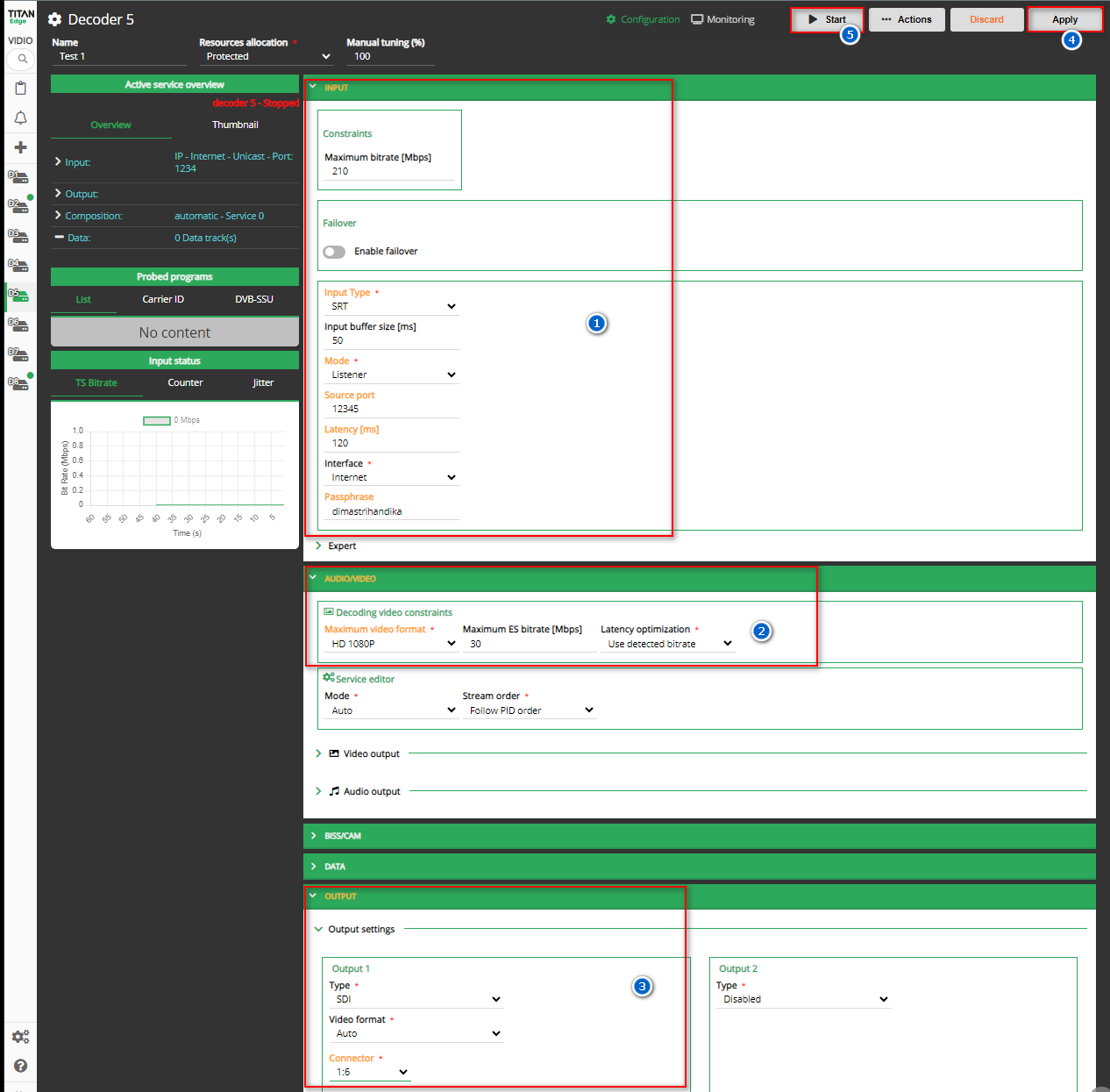
Streaming adalah sebuah teknologi untuk memaninkan file video atau audio secara langsung ataupun dengan pre-recorder dari sebuah mesin server

**HASIL DAN PEMBAHASAN**

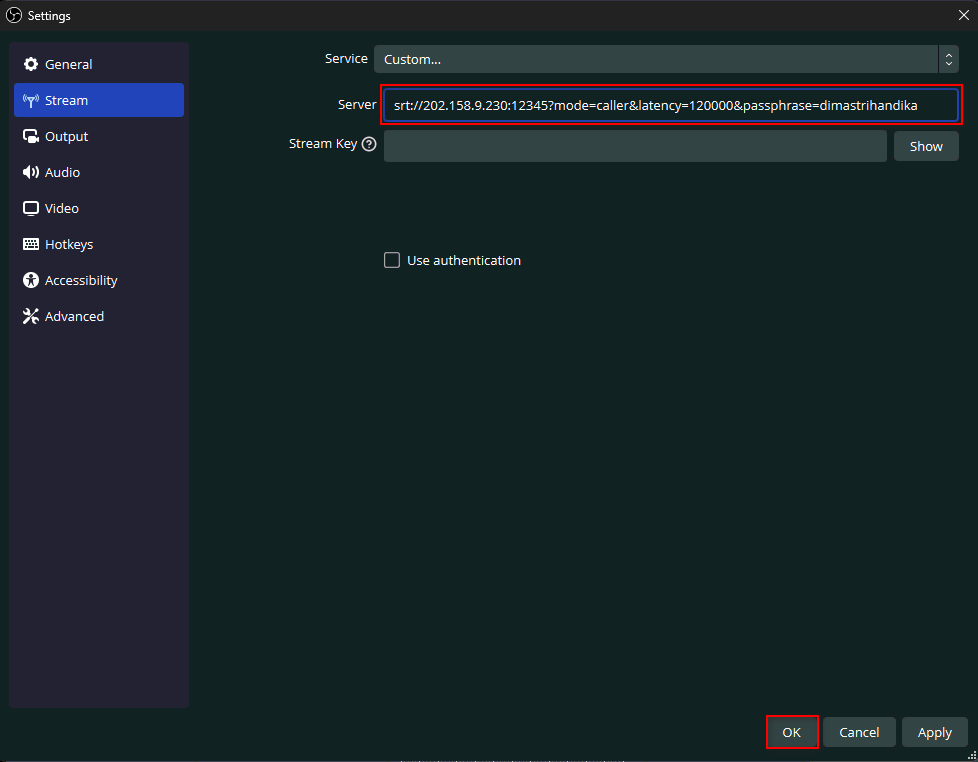
**3.1 Latensi:**

Menurut Arnao dan Buttimer (2022), Kecepatan transmisi video dapat bervariasi berdasarkan pada berbagai faktor, misalnya, kemacetan jaringan dan/atau kualitas komunikasi. Variasi ini bisa menyebabkan paket video dalam jaringan protokol internet (IP) diterima dengan penundaan yang bervariasi antara satu atau lebih paket video.

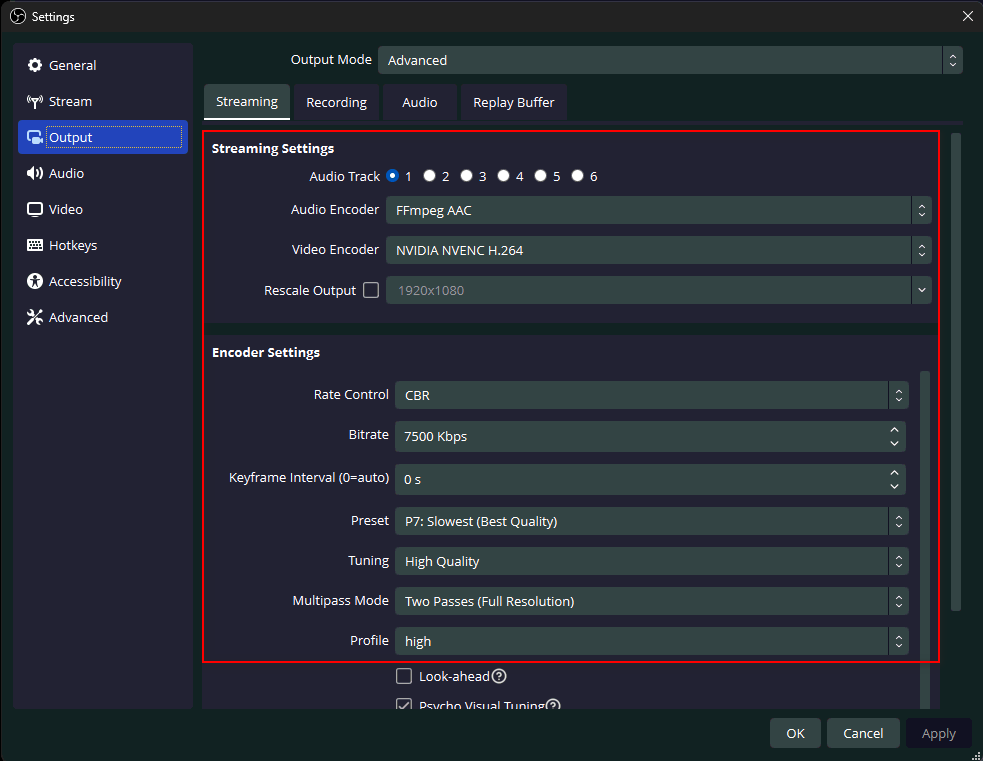
Pada bagian ini, penulis menganalisis hasil pengujian parameter latensi untuk menjaga kualitas pengiriman data. Penulis akan mengevaluasi keunggulan SRT dalam mengatasi delay dalam proses video streaming. Pengujian latensi dilakukan menggunakan aplikasi OBS Studio SRT Caller dan Titan Edge sebagai SRT Listener.



Gambar 1 Konfigurasi Input dan Output Channel pada Titan Edge

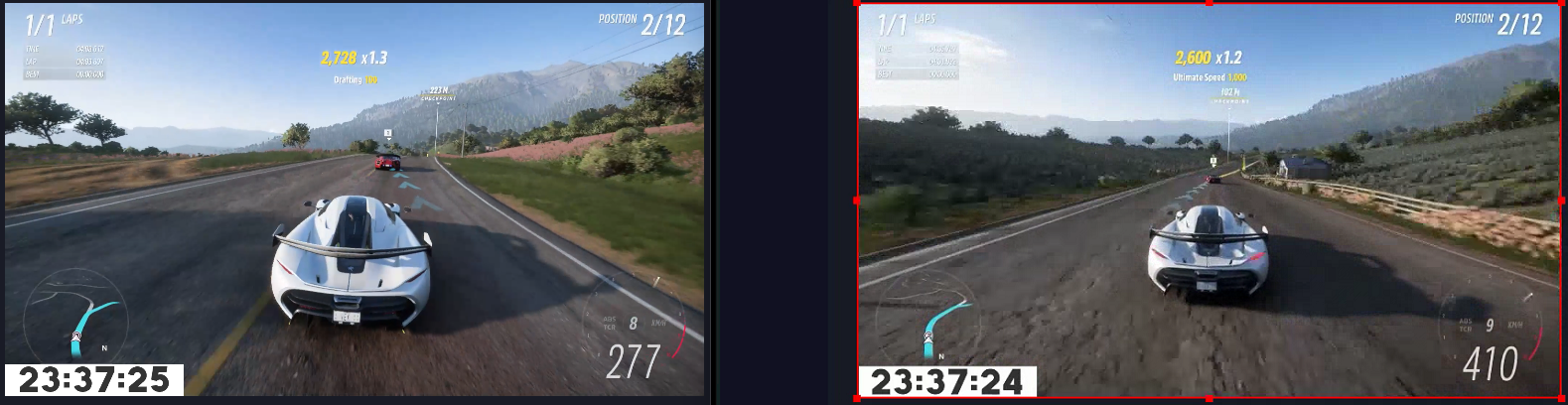


Gambar 2 Konfigurasi Parameter SRT pada Aplikasi OBS Studio

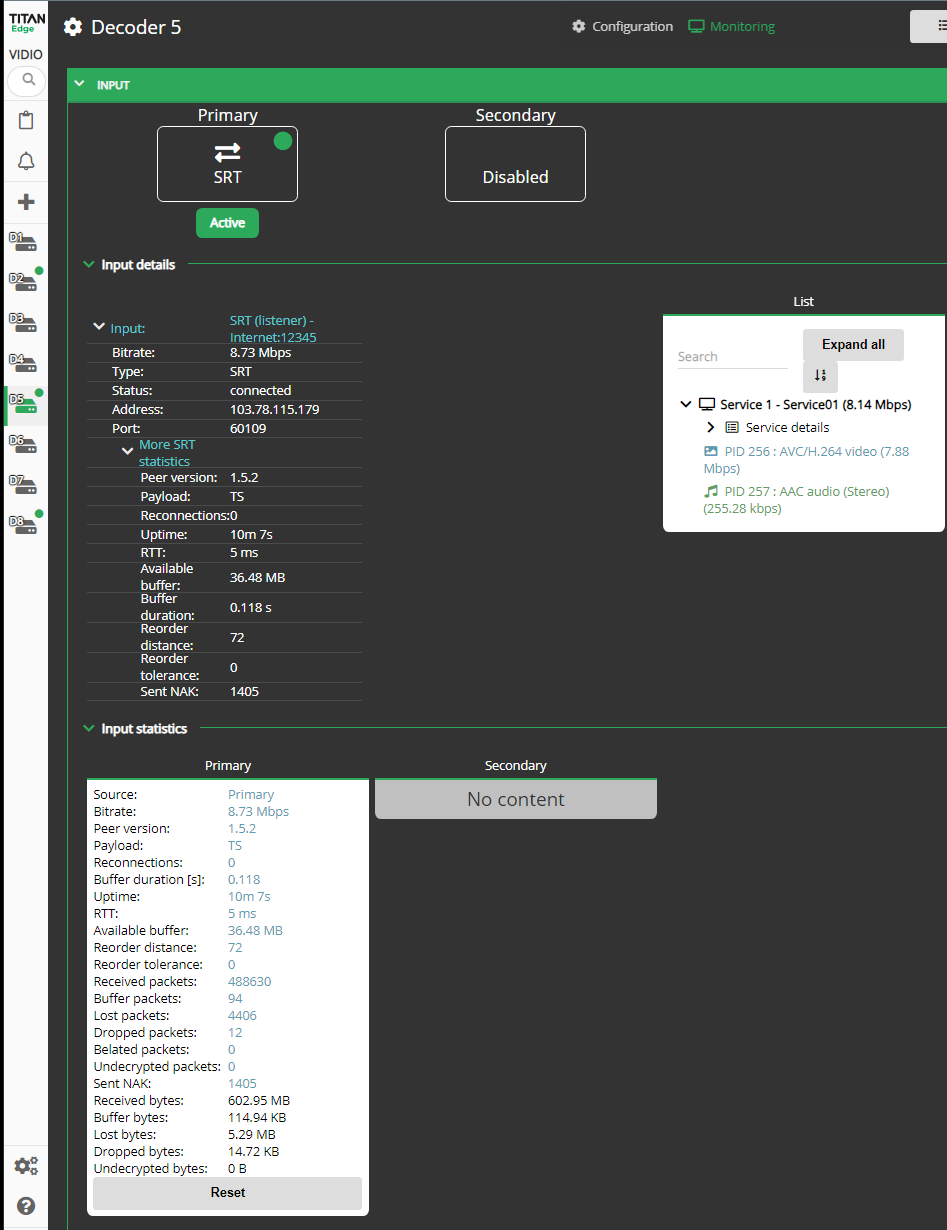


Gambar 3 Konfigurasi Preset Encoding pada OBS Studio

Hasil Pengujian:



Gambar 4 Perbandingan Delay antara Pengirim dan Penerima



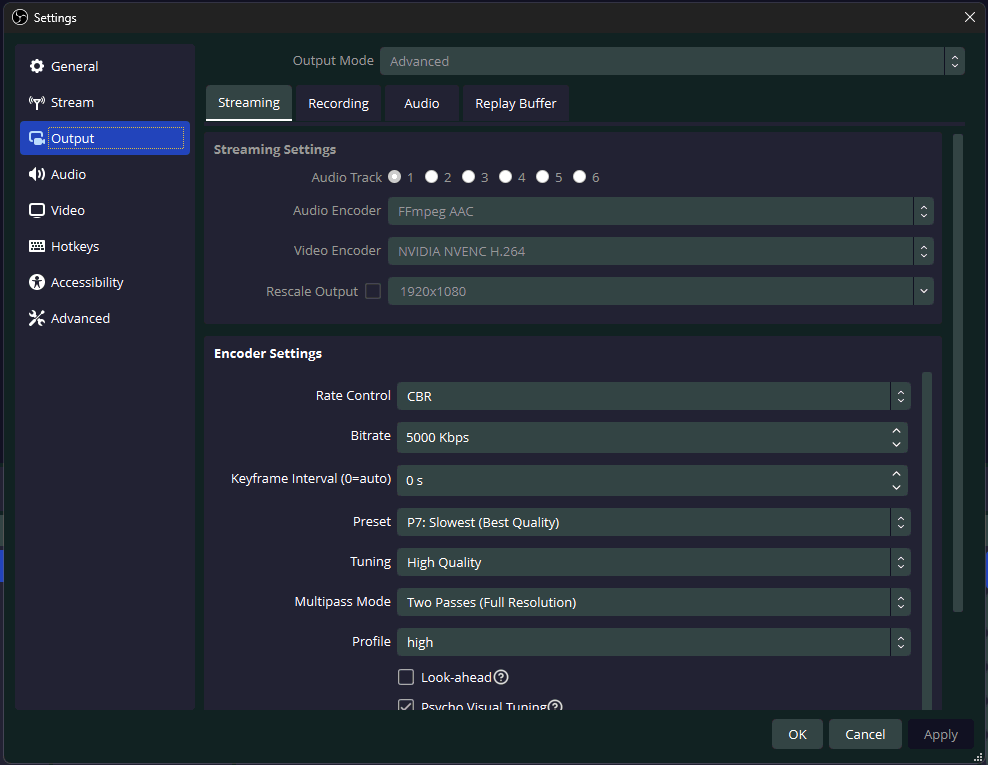
Gambar 5 Statistik Penerimaan di Titan Edge



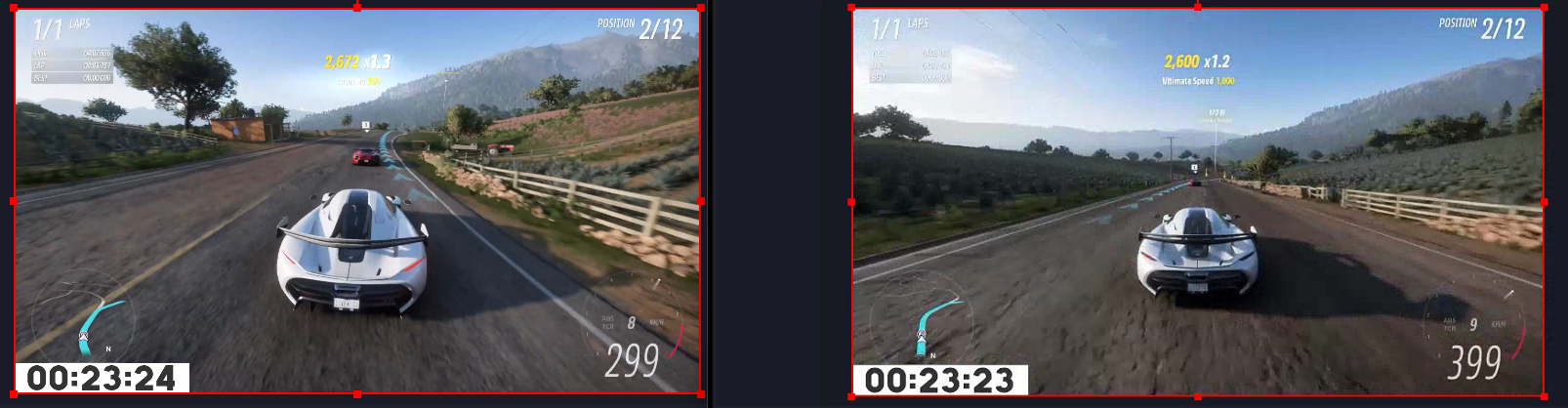
Gambar 6 Implikasi dari Dropped Network

Berdasarkan hasil pengujian diatas, delay yang di dapat hanya 1 detik antara pengirim dan penerima, namun terlihat adanya dropped network yang dapat disebabkan oleh beberapa faktor, salah satunya kemacetan dalam jaringan sehingga paket gagal di kirim ulang. Dampak dari kasus ini bisa di lihat pada Gambar 6 yaitu adanya artefak yang artinya terdapat frame yang hilang akibat dropped network tersebut.

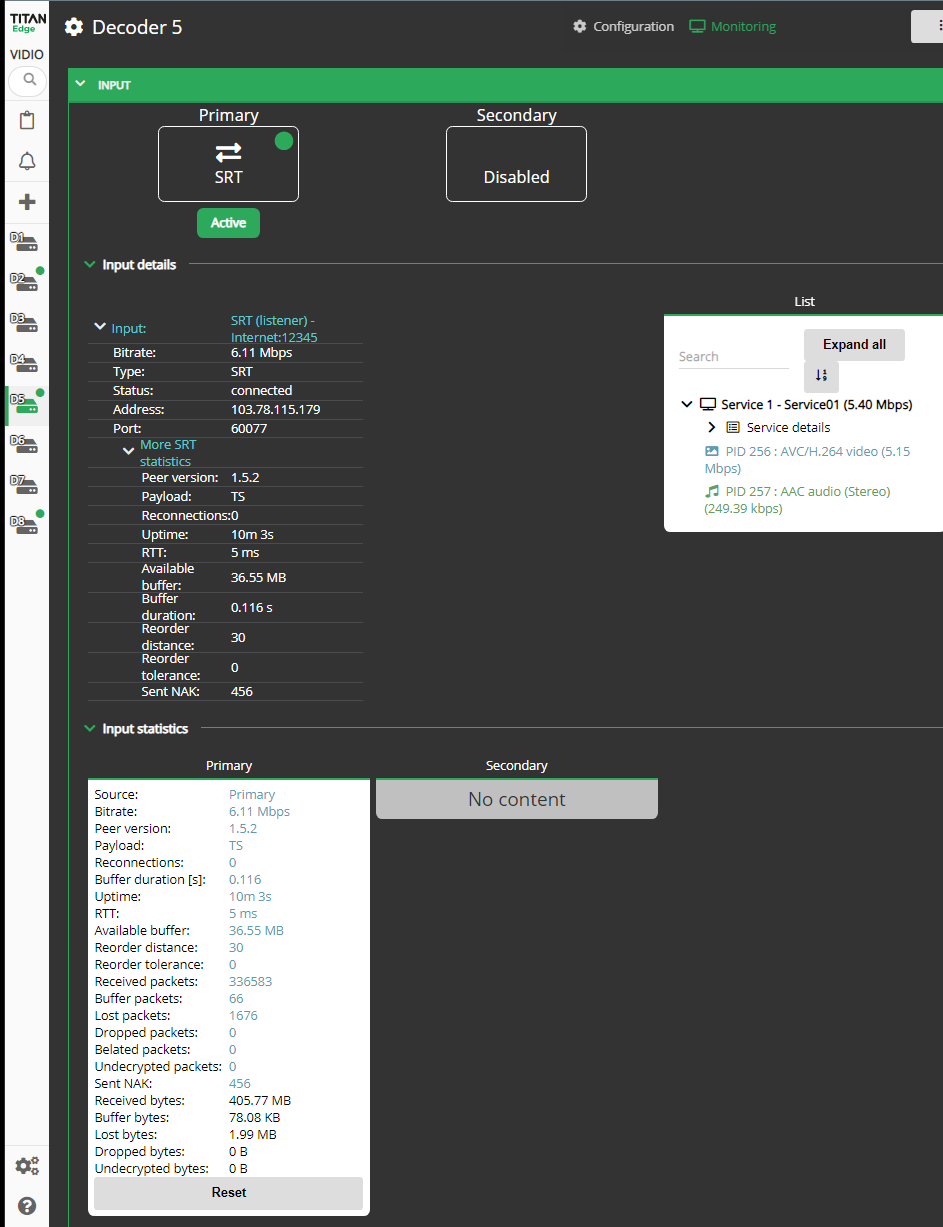
Mempertahankan Delay namun Menurunkan Kualitas Gambar:



Gambar 7 Menurunkan Bitrate dari 7500 Kbps menjadi 5000 Kbps



Gambar 8 Perbedaan Delay antara Pengirim dan Penerima



Gambar 9 Statistik Penerimaan di Titan Edge

Setelah dilakukan penurunan Bitrate dari 7500 Kbps menjadi 5000 Kbps terlihat tidak ada lagi dropped network dan delay antara pengirim dan penerima tetap terjaga di 1 detik.

Referensi:

ISO, "Information technology - Generic coding of moving pictures and associated audio information: Systems", ISO/ IEC 13818-1, September 2021.

Roimela, K, & You, Y (2020). Video streaming method. *US Patent 10,600,153*, Google Patents, <https://patents.google.com/patent/US10600153B2/en>

Ranganathan, G, & Periyaeluvan, RE (2023). Secure media streaming communication via user datagram protocol. *US Patent 11,611,542*, Google Patents, <https://patents.google.com/patent/US11611542B2/en>

Maolin, Z (2022). Method for retransmitting lost network packet based on transport stream format and user datagram protocol. *US Patent 11,489,902*, Google Patents, <https://patents.google.com/patent/US11489902B2/en>

Steffen, J, Salem, AS, Leicht, H, Mertens, M, & Khalil, M (2021). Technologies for managing TCP/IP packet delivery. *US Patent 11,012,367*, Google Patents, <https://patents.google.com/patent/US11012367B2/en>

Chang, W, & Sonwalkar, S (2020). Live video streaming services. *US Patent 10,721,499*, Google Patents, <https://patents.google.com/patent/US10721499B2/en>

Purbo, OW (2018). Internet-TCP/IP: Konsep &Implementasi. *Yogyakarta: Andi*, lms.onnocenter.or.id, <https://lms.onnocenter.or.id/pustaka/REVIEW-BUKU/2018-Internet-TCPIP%20Konsep%20Dan%20Implementasi.pdf>

Herr, DA, Kassimis, C, & Stevens, JW (2018). Protocol selection for transmission control protocol/internet protocol (TCP/IP). *US Patent 9,954,979*, Google Patents, <https://patents.google.com/patent/US9954979B2/en>

Smirnov, V (2018). Maximizing bandwidth utilization in networks with high latencies and packet drops using transmission control protocol. *US Patent 9,882,831*, Google Patents, <https://patents.google.com/patent/US9882831B2/en>

Bienik, J, Uhrina, M, Sevcik, L, & Holesova, A (2023). Impact of Packet Loss Rate on Quality of Compressed High Resolution Videos. *Sensors*, mdpi.com, <https://www.mdpi.com/1424-8220/23/5/2744>