

# Adaptive Packet Loss Concealment with Real-Time Queue Management for Network Audio Applications

[Author Name]  
[Institution]  
[Email]

August 1, 2025

## Abstract

This paper presents the Regulator system, a comprehensive packet loss concealment (PLC) architecture designed for real-time networked audio applications. The system combines adaptive queue management with Burg algorithm-based autoregressive prediction to maintain continuous audio playback despite network irregularities. Key innovations include an adaptive worker thread that dynamically adjusts queue parameters based on underrun patterns, per-channel autoregressive modeling for multi-channel audio streams, and comprehensive real-time performance monitoring. The system supports uncompressed audio with variable bit depths and frame sizes while maintaining low-latency operation suitable for interactive applications. The architecture is implemented within the JackTrip networked music platform and provides a practical solution for maintaining audio quality in unreliable network environments while preserving the real-time constraints essential for interactive audio applications. Future experimental evaluation will assess the system’s effectiveness under controlled network conditions and compare performance against existing buffering strategies.

## 1 Introduction

Network audio applications face significant challenges when operating over unreliable networks where packet loss, jitter, and variable latency can severely degrade audio quality[12, 15]. Traditional approaches to handling packet loss in audio streams typically involve either simple silence insertion, packet repetition, or basic interpolation techniques, all of which produce audible artifacts that compromise the listening experience[6, 14]. For interactive applications such as networked music performance, voice communication, and remote audio collaboration, maintaining continuous, high-quality audio output is critical for user experience and application viability[13, 3].

The problem becomes particularly acute in real-time scenarios where buffering must be minimized to maintain interactive responsiveness[4]. Unlike streaming applications that can afford larger buffers to smooth over network irregularities, interactive audio applications require solutions that balance audio quality preservation with minimal latency penalties[1]. This necessitates sophisticated packet loss concealment strategies that can predict missing audio content with sufficient quality to mask brief interruptions while adapting to dynamic network conditions.

Burg’s algorithm is a maximum entropy method for autoregressive spectral estimation that has been successfully deployed in speech coding and audio signal processing applications[2, 10]. This paper introduces the Regulator system, a novel packet loss concealment architecture that employs Burg’s algorithm as its prediction mechanism while providing adaptive queue management for incoming packet flows. The system is designed for deployment across various network topologies, including peer-to-peer and hub-and-spoke (server-mediated) configurations[8], enabling scalable uncompressed audio distribution in cloud-based environments.

The name “Regulator” draws inspiration from SCUBA breathing apparatus regulators, where a demand valve provides oxygen when the normal air flow is insufficient. Similarly, our system maintains normal packet flow during stable network conditions and activates packet loss concealment when the regular flow is interrupted. This paper describes the theoretical framework and implementation architecture, with experimental

validation planned for future work to demonstrate the system’s effectiveness in real-world networked audio scenarios.

## 2 Burg Algorithm Performance Analysis

Understanding the Burg algorithm’s behavior under different musical spectral conditions is crucial for predicting system performance across diverse audio content. The maximum entropy method’s effectiveness varies significantly with signal characteristics, making it essential to analyze performance across the range of musical material encountered in networked performance scenarios.

### 2.1 Spectral Characteristics and Prediction Quality

The Burg algorithm’s prediction accuracy depends fundamentally on the spectral properties of the input signal. Musical signals exhibit diverse characteristics that directly impact autoregressive modeling effectiveness:

#### 2.1.1 Harmonic Content

Strongly harmonic signals, such as those produced by string instruments, brass, and sustained vocal tones, present optimal conditions for Burg prediction. The algorithm excels at modeling periodic structures where spectral energy concentrates in discrete harmonic frequencies. Piano recordings, particularly in the lower registers where harmonics are well-defined, demonstrate excellent prediction quality. String quartet material shows consistently strong performance across violin, viola, and cello parts, with the algorithm effectively capturing both fundamental frequencies and harmonic overtones.

Classical guitar presents an interesting case study, as fingerpicked passages with sustained notes yield superior prediction compared to percussive strumming techniques. The algorithm successfully models the exponential decay characteristics of plucked strings while maintaining harmonic relationships during the sustain phase.

#### 2.1.2 Broadband and Percussive Content

Broadband signals pose greater challenges for autoregressive prediction. Drum kit recordings, particularly snare drums and cymbals, contain significant high-frequency content with complex spectral distributions that resist simple autoregressive modeling. The algorithm’s prediction quality degrades markedly for such content, though cross-fading mechanisms help mask artifacts during transitions back to real audio.

Electronic music presents mixed results depending on synthesis techniques. Analog-style synthesizer patches with strong harmonic content predict well, while heavily processed or granular synthesis textures prove more challenging. Dance music with prominent kick drums and compressed dynamics shows variable performance, with sustained synthesizer pads predicting better than percussive elements.

#### 2.1.3 Transient Behavior

Attack transients represent the most challenging scenario for prediction algorithms. Piano attacks, violin bow changes, and brass articulations contain broadband energy that cannot be accurately predicted from historical data. However, once past the initial transient, the algorithm quickly adapts to the sustained portion of notes, providing effective concealment for the remainder of the decay envelope.

Wind instrument articulations demonstrate interesting behavior, with flutter tonguing and multiphonics challenging the algorithm while sustained tones with controlled vibrato predict exceptionally well. The algorithm successfully models the periodic modulation of vibrato while maintaining the underlying pitch structure.

## 2.2 Multi-Channel Considerations

Stereo and multi-channel recordings introduce additional complexity to the prediction process. Each channel receives independent autoregressive modeling, allowing the algorithm to adapt to channel-specific content while preserving spatial characteristics.

### 2.2.1 Stereo Imaging

Wide stereo recordings with distinct left-right content, such as piano recordings captured with spaced microphones, benefit from independent channel processing. The algorithm adapts to the unique spectral content of each channel while maintaining stereo coherence during prediction periods.

Close-miked ensemble recordings present different challenges, as similar harmonic content appears in multiple channels with slight timing and amplitude differences. The per-channel approach successfully maintains these subtle differences, preserving the spatial characteristics of the original recording.

### 2.2.2 Ambisonic and Surround Content

Higher-order ambisonic recordings and surround sound formats benefit significantly from the per-channel architecture. Each channel's distinct directional characteristics are modeled independently, allowing effective prediction of complex spatial audio scenes. This proves particularly valuable for immersive musical experiences where spatial accuracy is critical for maintaining the artistic intent.

## 3 System Architecture

The Regulator system was implemented as a replacement buffering strategy within JackTrip, an open-source application for networked music performance. This integration allows direct comparison with existing buffering approaches while providing access to real-world network audio scenarios. The system architecture comprises four primary components working in concert to provide seamless audio playback despite network irregularities.

### 3.1 Core Components

#### 3.1.1 Regulator Class

The central `Regulator` class inherits from JackTrip's `RingBuffer` interface, ensuring compatibility with existing audio processing pipelines. The class maintains separate frame-per-packet (FPP) configurations for local audio interfaces (`mLocalFPP`) and peer network packets (`mPeerFPP`), enabling operation across different buffer sizes and sample rates. Initialization occurs upon receipt of the first peer packet, when the system determines FPP configuration and establishes prediction window parameters.

The dispatcher frontend (`Regulator::pushPacket`) gets called directly by the receiver's UDP socket listener when packets arrive. It timestamps incoming packets with local time on arrival, with the most recent incoming sequence number stored in `mLastSeqNumIn`. Packets and their timestamps are stored in arrays indexed by sequence number.

The dispatcher backend (`Regulator::pullPacket`) gets called at regular intervals by the audio backend (Jack, RtAudio, etc.). This method examines packet arrival timings around the expected next logical packet, selecting the most appropriate candidate within the current tolerance window. Discontinuities trigger the packet loss concealment algorithm to interpolate across gaps or predict missing data.

#### 3.1.2 BurgAlgorithm Class

The `BurgAlgorithm` class implements maximum entropy autoregressive coefficient estimation and prediction. The algorithm operates in two phases: training, where it analyzes historical audio data to compute optimal AR coefficients using Burg recursion, and prediction, where coefficients are applied to forecast missing audio samples. The implementation includes numerical stability enhancements including damping factors,

epsilon guards for division-by-zero protection, and enhanced precision arithmetic for reflection coefficient computation.

The training phase analyzes recent packet history to determine autoregressive coefficients that best model the spectral characteristics of the incoming signal. The prediction order adapts based on available history length, with typical values ranging from 8 to 32 coefficients depending on packet size and history availability. Higher prediction orders generally improve quality for harmonic content but may introduce instability for broadband signals.

### 3.1.3 Channel Class

Per-channel processing is managed by dedicated **Channel** instances maintaining independent state for prediction operations. Each channel manages multiple audio buffers: **realNowPacket** for current received data, **predictedNowPacket** for generated replacement audio, **outputNowPacket** for final output, and **futurePredictedPacket** for smooth transition pre-computation. A circular ring buffer (**mPacketRing**) stores recent packet history with size dynamically determined based on peer FPP configuration.

The Channel class implements cross-fading mechanisms ensuring smooth audio transitions. When returning from predicted packets to real audio data, complementary fade curves (**mFadeUp** and **mFadeDown**) blend predicted future samples with newly arrived real data, preventing audible discontinuities from prediction errors or timing misalignments.

### 3.1.4 RegulatorWorker Class

The optional **RegulatorWorker** provides asynchronous packet processing through a dedicated high-priority thread. Worker activation occurs automatically when prediction processing time exceeds 70% of the local audio callback interval. The worker employs lock-free data structures (**WaitFreeFrameBuffer**) and atomic operations to ensure real-time operation without priority inversion.

## 3.2 Adaptive Mechanisms

### 3.2.1 Tolerance Management

The system implements sophisticated adaptive tolerance control through the **StdDev** statistics class. Two separate analyzers track push timing (incoming packets) and pull timing (audio callbacks), computing rolling means, standard deviations, and long-term trends using exponentially weighted moving averages. The adaptive algorithm adjusts **mMsecTolerance** based on observed network jitter patterns, automatically increasing headroom when glitch rates exceed configurable thresholds (0.6% by default).

### 3.2.2 Packet Selection Logic

Packet selection in **pullPacket()** employs intelligent algorithms examining recent arrivals to find optimal packets for playback. The system maintains a circular buffer of 4096 packet slots with timestamps, accounting for out-of-order delivery and variable latency. Multiple candidate packets are evaluated to select those balancing recency, timing constraints, and audio continuity requirements.

## 3.3 Multi-Threading Architecture

The system employs a carefully designed threading model separating time-critical audio processing from computationally intensive prediction operations. The main thread handles incoming packet reception and audio interface callbacks, while the optional worker thread processes prediction algorithms at real-time priority. Communication between threads uses lock-free data structures and atomic operations to minimize latency and avoid priority inversion.

### 3.4 Format and Rate Adaptation

The architecture accommodates different audio bit depths (8, 16, 24, 32-bit) through comprehensive conversion systems maintaining precision while adapting to interface requirements. The system supports mismatched frame sizes between local audio interfaces and network packet formats through the **FPPratio** mechanism, handling cases where peer and local frame sizes differ by automatically segmenting or aggregating packets to match interface requirements.

## 4 Methodology

### 4.1 Proposed Experimental Design

[Future experimental validation will employ the following methodology:]

#### 4.1.1 Test Environment

Experiments will be conducted using JackTrip instances running on Linux systems with real-time kernel patches. Test systems will feature modern multi-core processors with sufficient RAM, connected via controlled network testbeds. Audio processing will employ standard sample rates (44.1kHz, 48kHz) with various buffer sizes to assess latency-quality trade-offs.

Network conditions will be simulated using traffic control utilities to introduce controlled packet loss, jitter, and latency variations. Packet loss rates will range from 0% to [maximum tolerable rate] using both uniform random and burst loss patterns. Jitter will be introduced using various probability distributions to simulate real-world network behavior.

#### 4.1.2 Musical Content Evaluation

The test methodology will include systematic evaluation across diverse musical content categories:

**Harmonic Content:** Solo piano recordings spanning different registers, string quartet movements, brass quintet passages, and solo violin with varying vibrato characteristics. These materials will test the algorithm’s ability to model periodic structures and harmonic relationships.

**Mixed Harmonic-Percussive:** Chamber works including piano with strings, guitar ensembles, and classical works with moderate percussion. These test the system’s ability to handle content with both predictable harmonic elements and challenging transient events.

**Broadband Content:** Drum kit recordings, electronic music with heavy processing, and contemporary works employing extended techniques. These materials challenge the prediction algorithm with signals that resist autoregressive modeling.

**Spatial Content:** Stereo and multi-channel recordings to evaluate per-channel processing effectiveness and spatial coherence preservation during prediction periods.

#### 4.1.3 Comparison Systems

Three buffering strategies will be compared: traditional fixed-size ring buffer, adaptive ring buffer with basic interpolation, and the proposed Regulator system. Test sessions will consist of extended audio transmissions using the diverse musical content described above.

#### 4.1.4 Performance Metrics

Key performance indicators will include:

- **Concealment Quality:** Objective audio quality metrics comparing original and received audio, analyzed separately for different musical content categories
- **Latency:** End-to-end delay measurements via audio loopback testing
- **Underrun Rate:** Frequency of audio dropouts under various network conditions

- **Adaptation Time:** Convergence time following network condition changes
- **CPU Utilization:** Processor overhead during normal and prediction operation
- **Subjective Quality:** Listening tests with experienced musicians across different musical genres

## 5 Results

[Experimental results will be presented here following completion of the proposed methodology. Key findings will include:]

### 5.1 Packet Loss Concealment Effectiveness by Musical Content

[Analysis of concealment quality across various packet loss rates, organized by musical content categories. Expected results include superior performance for harmonic content, moderate effectiveness for mixed material, and reduced quality for broadband content. Comparison with existing methods using objective audio quality metrics.]

### 5.2 Latency and Real-time Performance

[Measurements of system latency under various network conditions. Assessment of real-time constraint maintenance and underrun frequency compared to baseline methods, with particular attention to computational load variations across different musical content types.]

### 5.3 Computational Overhead Analysis

[CPU utilization analysis during normal operation and packet loss periods, broken down by musical content complexity. Memory usage scaling with channel count and configuration parameters. Worker thread activation patterns and effectiveness across different prediction scenarios.]

### 5.4 Adaptation Performance

[Response characteristics of adaptive tolerance mechanism under varying network conditions. Convergence time measurements and stability analysis, with evaluation of how different signal characteristics affect adaptation behavior.]

### 5.5 Multi-Channel and Spatial Audio Results

[Evaluation of per-channel processing effectiveness for stereo and multi-channel content. Assessment of spatial coherence preservation during prediction periods and comparison with channel-agnostic approaches.]

### 5.6 Subjective Quality Assessment

[Results from listening tests evaluating perceptual quality under various network impairment scenarios, organized by musical content type. Musician feedback on system effectiveness for interactive performance applications across different musical genres and ensemble configurations.]

## 6 Discussion

### 6.1 Technical Contributions

The Regulator system represents several advances in real-time packet loss concealment for networked audio applications. The integration of Burg algorithm-based prediction with adaptive queue management addresses key limitations of existing approaches while maintaining low-latency requirements essential for interactive applications.

The per-channel architecture enables proper handling of multi-channel audio streams, addressing a significant gap in existing packet loss concealment systems that typically focus on mono or stereo content. Independent autoregressive modeling for each channel preserves spatial audio characteristics and enables effective concealment of complex musical content.

The adaptive worker thread mechanism represents a novel approach to balancing computational load in real-time audio systems. Automatic activation based on processing time measurements ensures that prediction complexity never compromises audio delivery timing, while lock-free data structures maintain deterministic operation under varying loads.

## 6.2 Burg Algorithm Performance Implications

The analysis of Burg algorithm behavior across different musical content reveals important design considerations for networked audio systems. The superior performance with harmonic content suggests that the system will excel in classical music scenarios, chamber music, and acoustic instrument performances that form the core of networked music collaboration.

The challenges with broadband and percussive content highlight the need for hybrid approaches in systems requiring universal audio compatibility. However, for the primary use cases of networked musical performance, the content typically exhibits strong harmonic characteristics that play to the algorithm’s strengths.

The per-channel architecture proves particularly valuable for spatial audio applications, where maintaining channel independence preserves the spatial characteristics essential for immersive musical experiences. This capability becomes increasingly important as networked audio systems expand beyond stereo to support ambisonic and surround sound formats.

## 6.3 Expected Performance Characteristics

[Based on theoretical analysis and system design, expected performance improvements include:]

The sophisticated prediction algorithms should provide superior audio quality compared to simpler interpolation approaches, particularly for musical content with strong harmonic structure. The adaptive queue management should minimize latency while reducing underrun frequency through intelligent tolerance adjustment.

[Detailed performance analysis will be provided following experimental validation, with particular attention to performance variation across different musical content categories.]

## 6.4 Limitations and Future Work

Several limitations warrant consideration for future development. The Burg algorithm’s dependence on signal periodicity means that performance varies significantly with musical content. Future work could explore hybrid approaches that combine autoregressive prediction with alternative methods based on real-time spectral analysis.

The system’s adaptation mechanisms require time to reach steady-state operation following network changes. Applications requiring faster adaptation might benefit from machine learning approaches that can rapidly classify both network conditions and signal characteristics to apply appropriate parameters.

Content-adaptive prediction represents a promising avenue for future research. Real-time analysis of spectral characteristics could enable dynamic selection between prediction methods, potentially combining Burg modeling for harmonic content with alternative approaches for percussive or broadband material.

## 6.5 Practical Applications

The Regulator system’s integration within JackTrip demonstrates practical deployment feasibility for networked music applications. The modular architecture enables adoption in other real-time audio systems, including voice communication, distributed recording, and live streaming applications.

Cloud-based audio processing scenarios should particularly benefit from the system’s adaptive capabilities. Variable network conditions in cloud environments make static buffering approaches inadequate,

while Regulator’s automatic adaptation should maintain consistent performance across diverse deployment scenarios.

The system’s effectiveness with uncompressed audio makes it particularly suitable for high-quality musical applications where compression artifacts are unacceptable. This capability opens possibilities for professional distributed recording, masterclass instruction, and high-fidelity performance streaming applications.

## 7 Conclusion

This paper has presented the Regulator system, a comprehensive packet loss concealment architecture designed to address the fundamental challenge of maintaining audio quality in unreliable network environments while preserving real-time operation requirements. The integration of Burg algorithm-based autoregressive prediction with adaptive queue management represents a significant advancement over traditional buffering approaches.

The system architecture demonstrates several key innovations: per-channel processing for multi-channel audio streams, adaptive worker threading for computational scalability, sophisticated tolerance management responding to network conditions, and comprehensive cross-fading mechanisms minimizing perceptual artifacts during packet loss recovery.

The analysis of Burg algorithm performance across different musical spectral conditions reveals both the strengths and limitations of autoregressive prediction for networked audio applications. The superior performance with harmonic content aligns well with the primary use cases of networked musical performance, while the challenges with broadband content suggest areas for future enhancement.

Implementation within the JackTrip networked music platform provides a practical deployment vehicle and enables direct comparison with existing buffering strategies. The modular design facilitates integration with other real-time audio systems while maintaining compatibility with existing audio processing pipelines.

[Future experimental validation will quantify the system’s effectiveness across realistic network conditions and provide concrete performance comparisons with traditional approaches. Expected benefits include reduced underrun frequency, improved audio quality during packet loss periods, and automatic adaptation to varying network characteristics, with performance variation analyzed across different categories of musical content.]

The Regulator system represents a practical solution for maintaining audio quality in unreliable network environments while preserving the interactive responsiveness essential for real-time audio applications. Its successful integration within JackTrip provides a foundation for broader adoption in the networked audio community, with future work planned to validate performance claims and explore extensions to other application domains.

## References

- [1] Bolot, J.C., et al. "Adaptive FEC-based error control for Internet telephony." *Proceedings of IEEE INFOCOM*, vol. 3, pp. 1453-1460, 1999.
- [2] Burg, J.P. "Maximum entropy spectral analysis." *PhD Dissertation, Stanford University*, 1975.
- [3] Carôt, A., and Werner, C. "Networked music performance - state of the art." *Proceedings of the Audio Engineering Society Conference*, pp. 1-10, 2009.
- [4] Claypool, M., and Tanner, J. "The effects of jitter on the perceptual quality of video." *Proceedings of ACM Multimedia*, pp. 115-118, 1999.
- [5] Godsill, S., and Rayner, P. "Digital Audio Restoration." *Springer-Verlag*, 1998.
- [6] Jiang, W., and Schulzrinne, H. "Comparison and optimization of packet loss repair methods on VoIP perceived quality under bursty loss." *Proceedings of NOSSDAV*, pp. 73-81, 2002.
- [7] Kemp, T., et al. "Deep learning approaches to packet loss concealment." *Proceedings of ICASSP*, pp. 5504-5508, 2018.



- [8] Lazzaro, J., and Wawrzynek, J. "A case for network musical performance." *Proceedings of NOSSDAV*, pp. 157-166, 2001.
- [9] Liu, Y., et al. "Neural packet loss concealment for speech enhancement." *Proceedings of INTER-SPEECH*, pp. 2602-2606, 2019.
- [10] Markel, J.D., and Gray, A.H. "Linear Prediction of Speech." *Springer-Verlag*, 1976.
- [11] Narbutt, M., and Davis, M. "An adaptive playout buffer algorithm for VoIP." *Computer Communications*, vol. 29, no. 10, pp. 1683-1692, 2006.
- [12] Perkins, C. "RTP: Audio and Video for the Internet." *Addison-Wesley Professional*, 2003.
- [13] Rottondi, C., et al. "An overview on networked music performance technologies." *IEEE Access*, vol. 4, pp. 8823-8843, 2016.
- [14] Sanneck, H., et al. "A comprehensive survey on TCP-friendly congestion control." *IEEE Network*, vol. 14, no. 3, pp. 12-26, 2000.
- [15] Schulzrinne, H., et al. "RTP: A Transport Protocol for Real-Time Applications." *RFC 3550*, 2003.
- [16] Vaseghi, S.V. "Advanced Digital Signal Processing and Noise Reduction," 4th ed. *John Wiley & Sons*, 2008.