academic publishers

INTERNATIONAL JOURNAL OF NETWORKS AND SECURITY (ISSN: 2693-387X)

Volume 05, Issue 01, 2025, pages 57-61
Published Date: - 01-05-2025

Predicting Congestion in High-Speed Low-Latency Networks with Rough Set Theory

Dr. Haruka Fujimoto, PhD
Advanced Networking Research Center, Kyoto University, Kyoto, Japan

Abstract

The relentless demand for faster data rates and reduced end-to-end delays has driven the evolution of communication networks towards technologies like 5G and beyond [3, 5, 6, 11, 18, 19]. High-speed, low-latency applications, ranging from real-time gaming and augmented reality to autonomous systems and industrial automation, are critically dependent on efficient and predictable network performance. Traditional congestion control mechanisms, primarily embodied in variants of the Transmission Control Protocol (TCP), often react to congestion signals (like packet loss or delay) rather than predicting them [1, 4, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 20, 21]. While various TCP variants have been developed for different network conditions [7, 8, 9, 10, 12, 13, 16, 17, 19, 20, 21] and machine learning has been explored for congestion control [5, 6, 18], a proactive approach through accurate prediction is highly desirable for meeting stringent low-latency requirements. Rough Set Theory (RST) [24, 25] is a mathematical framework for dealing with vagueness and uncertainty in data, offering powerful tools for attribute reduction and rule extraction [24, 25]. Although RST has been applied successfully in various prediction and classification tasks in other domains [24, 25], its application to predicting network congestion, particularly from the perspective of network towers supporting high-speed, low-latency traffic, is a novel area. This article proposes a design concept for a congestion prediction model leveraging RST, outlining its methodology, expected benefits in terms of interpretability and efficiency, and potential for deployment on network infrastructure to enable proactive congestion management and enhance high-speed, low-latency communication.

Keywords

Rough Set Theory, Congestion Prediction, High-Speed Communication, Low-Latency Networks, Network Optimization, Traffic Management, Communication Systems, Predictive Modeling, Intelligent Networks, Data Mining.

INTRODUCTION

Modern communication networks are undergoing a significant transformation to meet the escalating demands for high bandwidth and ultra-low latency [3, 5, 6, 11, 18, 19]. The advent of 5G technology and the ongoing research into future wireless systems are specifically targeting applications that require rapid response times and reliable connectivity, such as tactile internet, autonomous vehicles, and remote surgery. In these scenarios, network congestion, which manifests as increased delays, packet loss, and reduced throughput, poses a severe threat to application performance and user experience.

Efficiently managing network congestion is paramount. For decades, the Transmission Control Protocol (TCP) has been the backbone of reliable data transfer on the internet, incorporating various congestion control algorithms [1, 4, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 20, 21]. Algorithms like TCP Reno [8, 9, 10], TCP NewReno [8, 9, 10], TCP CUBIC [12], TCP Vegas [8], TCP BBR [13], and variants like TCP Fast Open [7], TCP CERL+ [16], and Elastic-TCP [20] have been developed or analyzed to improve performance under different network conditions, including broadband wireless [1], multi-path [2, 15], heterogeneous [14], and multi-hop environments [21]. Specific challenges arise in 5G networks, particularly with mmWave deployments [18,

19] and disaster scenarios [5, 18], necessitating new adaptation methods [3, 17]. Recent research has also explored leveraging deep learning and machine learning techniques for developing new TCP congestion control algorithms [5, 6, 18].
However, most traditional and many modern TCP congestion control mechanisms are inherently reactive; they detect congestion after it has already begun and then adjust transmission rates [1, 4, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 20, 21]. For applications where microseconds of delay matter, a reactive approach is often insufficient. Congestion must be predicted before it impacts traffic, allowing for proactive resource allocation or traffic management strategies. This necessitates prediction models deployed at key network points, such as cellular network towers (base stations), which have visibility into local traffic conditions and resource availability.
Developing accurate and efficient prediction models for complex and dynamic wireless network environments is challenging. Rough Set Theory (RST), introduced by Zdzislaw Pawlak, provides a formal framework for knowledge discovery from data containing uncertainty, vagueness, and incompleteness [24, 25]. RST operates by analyzing the relationships between attributes in a dataset to identify dependencies, reduce redundant features (attribute reduction), and extract decision rules [24, 25]. Unlike many machine learning methods, RST does not require prior assumptions about the data distribution and offers interpretable results in the form of IF-THEN rules. RST has demonstrated effectiveness in various applications, including predicting building energy consumption [25] and analyzing digital marketing strategies [24].
Given the need for proactive congestion management in high-speed, low-latency networks and RST's capabilities in handling uncertain data and extracting interpretable rules, this article proposes a design concept for a network congestion prediction model utilizing Rough Set Theory, intended for deployment on network towers. The objective is to outline a methodology that leverages RST's strengths to process network performance data and predict future congestion states, thereby enabling preemptive actions to maintain high speed and low latency.

Model Design Methodology (Adapted Methods Section)

The proposed design for a congestion prediction model using Rough Set Theory involves several key steps, focusing on data utilization, feature processing, and the application of RST for knowledge discovery. The model is envisioned to operate locally on network towers, processing real-time or near-real-time data streams to provide timely predictions.

1. Problem Formalization and Data Definition: The problem is defined as predicting the likelihood or level of congestion in a specific cell or sector served by a network tower within a defined future time window (e.g., the next few seconds or minutes). This prediction is based on historical and current network performance data collected at the tower. Potential congestion states could be classified (e.g., Low Congestion, Moderate Congestion, High Congestion).

Data needed includes various network performance indicators and context information available at the base station. This might include:

o Traffic Load: Aggregate uplink/downlink throughput, active user count, type of traffic (e.g., real-time, best-effort).
o Resource Utilization: CPU load on network processing units, memory usage, radio resource block allocation, power amplifier status.
o Queueing Status: Buffer occupancy levels for different traffic queues, packet queueing delays.
o Packet Dynamics: Packet drop rates, retransmission rates (though TCP handles some of this, tower metrics reflect it), RTT measurements (if available at the tower edge).
o Environmental/Contextual Data: Time of day, day of week, specific events causing sudden load changes (though this is harder to predict locally).

Datasets like the one available [30], focusing on 5G network performance metrics, would be invaluable for training and evaluating such a model, providing realistic data reflecting network behavior under varying loads.

2. Feature Engineering and Discretization: Raw network data streams need to be processed into discrete features suitable for RST. This involves:

o Aggregation: Aggregating raw metrics over small time intervals to create meaningful samples (e.g., average buffer occupancy over 1 second, packet drop count per 100 packets).
o Normalization/Scaling: While RST doesn't assume data distribution, scaling might be helpful for some preprocessing steps or if combined with other methods.
o Discretization: RST operates on discrete values. Continuous network metrics (like buffer occupancy percentage, throughput) must be converted into a finite set of intervals or categories (e.g., Buffer_Low, Buffer_Medium, Buffer_High; Throughput_0-10Mbps, Throughput_10-50Mbps, etc.). Defining appropriate discretization bins is a critical step influencing model performance.

The chosen features form the set of conditional attributes. The predicted congestion state forms the decision attribute.

3. Application of Rough Set Theory (RST): With the data formalized as an information system with conditional and decision attributes, RST is applied [24, 25]:

o Indiscernibility Relation: RST groups objects (data samples) based on their attribute values. An indiscernibility relation is defined for sets of attributes, grouping samples that are indistinguishable based on those attributes.
o Set Approximation: For each value of the decision attribute (e.g., "High Congestion"), RST calculates the lower and upper approximations [24, 25]. The lower approximation contains samples that are definitely in the set based on the conditional attributes. The upper approximation contains samples that possibly belong to the set. The boundary region consists of samples

that cannot be definitively classified. RST analysis of the boundary region helps understand the uncertainty.

o Attribute Reduction (Finding Reducts): A core strength of RST is identifying reducts [24, 25]. A reduct is a minimal subset of conditional attributes that preserves the same indiscernibility relation (and thus the same classification capability) as the full set of attributes. Finding reducts helps identify the most relevant network metrics for congestion prediction, reducing the dimensionality of the problem and potentially improving model efficiency and robustness [24, 25]. There might be multiple reducts.

o Rule Extraction: From the reduced set of attributes (or the full set), RST derives a set of decision rules in the form of "IF (conditions on conditional attributes) THEN (decision attribute value)" [24, 25]. These rules represent the relationships learned from the data about what network conditions lead to specific congestion states.

4. Model Utilization: The RST-derived knowledge (either the reducts used to train a subsequent simpler classifier or the extracted decision rules themselves) forms the basis of the congestion prediction model. For real-time prediction on the tower, incoming network data samples are processed (aggregated, discretized) and then evaluated against the derived RST model or rules to predict the congestion state in the upcoming time window.

While the core is RST, the design could potentially integrate other machine learning concepts for performance tuning. For instance, optimization algorithms (like Gazelle Optimization [28, 29] or others [26, 27]) could potentially be applied to optimize the selection of discretization bins or the parameters of a classifier trained on RST-reduced features. However, the primary predictive power comes from the RST analysis of feature dependencies and rule extraction [24, 25]. Other ML approaches like deep learning [5, 6, 18] could be considered as alternative or supplementary models, but the core proposal focuses on RST's unique benefits.

Expected Outcomes and Advantages (Adapted Results Section)

Applying the proposed RST-based methodology to network congestion prediction on towers is expected to yield several outcomes and offer distinct advantages, particularly relevant for high-speed, low-latency communication.

1. Identification of Key Predictors: The attribute reduction process in RST [24, 25] is expected to identify a minimal subset of network performance indicators that are most predictive of future congestion. For instance, metrics like queue depth on specific interfaces or the rate of buffer drops might be identified as highly relevant, while others like overall CPU load (if not a bottleneck) might be deemed less significant for prediction in certain scenarios. This provides valuable insight into the underlying dynamics causing congestion.

2. Generation of Interpretable Prediction Rules: RST is designed to extract IF-THEN decision rules [24, 25]. The output would be a set of rules linking specific combinations of network states (e.g., IF average queue depth is 'High' AND active user count is 'Increasing' THEN predict 'High Congestion' in next 5 seconds). These rules are human-readable and understandable, which is a significant advantage over black-box models like deep neural networks often used in network tasks [5, 6, 18, 22, 23]. This interpretability can aid network operators in understanding the reasons behind predictions and refining network policies.

3. Potential for Real-Time Efficiency: By reducing the number of features considered for prediction [24, 25], the computational load for real-time inference on the network tower can potentially be decreased. This is critical for meeting the processing speed requirements imposed by low-latency communication needs.

4. Handling Data Imperfection: RST's inherent ability to handle data with vagueness and uncertainty [24, 25] is well-suited to the noisy and dynamic nature of wireless network measurements. It can potentially derive meaningful rules even from incomplete or imprecise datasets.

5. Enabling Proactive Management: The primary expected benefit is the shift from reactive congestion control (characteristic of many TCP variants [1, 4, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 20, 21]) to proactive prediction. By predicting congestion before it occurs, network towers can trigger preemptive actions, such as adjusting scheduling algorithms, temporarily limiting new connections, or signaling to upstream network elements, thereby preventing performance degradation and maintaining low latency.

The application of machine learning for network management is a growing field [5, 6, 18, 22, 23, 26, 27, 28, 29], and while various models exist, RST offers a unique balance of predictive power and interpretability that could be particularly valuable for deployment in critical network infrastructure like cellular towers.

Discussion

The proposal to utilize Rough Set Theory for congestion prediction on network towers represents a potential advancement in managing the complex and dynamic environments of high-speed, low-latency communication. The critical need for predictive capabilities stems directly from the limitations of purely reactive control mechanisms, including many sophisticated TCP variants [1, 4, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 20, 21]. While machine learning has been applied to network problems [5, 6, 18, 22, 23, 26, 27, 28, 29], RST offers distinct advantages in interpretability [24, 25], which is highly desirable for practical network operations and debugging.

The strength of RST lies in its ability to analyze data without requiring a probabilistic framework, directly identifying the essential attributes and the relationships between them that determine the classification or prediction outcome [24, 25]. This is particularly relevant in a network setting where data relationships might be non-linear and influenced by many interacting factors. The resultant rules provide transparent insights into the conditions leading to congestion [24, 25].

However, transitioning this design concept into a functional system involves significant challenges:

1. Data Acquisition and Labeling: Obtaining high-quality, labeled data representing various congestion states from operational network towers is difficult. Public datasets like [30] provide valuable resources, but may need specific labeling for congestion states relevant to prediction. Defining "congestion" itself can be subjective and context-dependent (e.g., acceptable delay for one application is unacceptable for another).
2. Discretization Strategy: The performance of RST is sensitive to how continuous attributes are discretized [24, 25]. Developing effective, potentially dynamic, discretization methods for continuously changing network metrics is crucial.
3. Computational Feasibility: While RST can reduce features, the process of computing indiscernibility relations, reducts, and rules can be computationally intensive, especially with large datasets. Efficient algorithms and implementations are necessary for deployment on resource-constrained tower hardware, where real-time performance is paramount.
4. Handling Temporal Dynamics: Network congestion is a time-series phenomenon. The current RST framework is primarily static. Adapting RST or integrating it with time-series analysis techniques or models (e.g., using historical sequences as features) is needed to capture temporal dependencies accurately.
5. Integration with Network Control Plane: The predicted congestion state must be translated into actionable instructions for the network tower's scheduler, radio resource manager, or traffic shaper. This requires seamless integration between the prediction model and the network's control plane, potentially informing mechanisms like traffic shaping or prioritizing low-latency flows. The prediction could also potentially be fed back to advanced TCP variants [5, 6, 18, 19] or other transport layer protocols for more intelligent rate adaptation.
6. Model Maintenance: Network conditions and traffic patterns evolve. The RST model or rules will require periodic retraining or dynamic updating based on new data to maintain accuracy [mention dataset [30] again as a source for such ongoing training].

It is important to reiterate that while the provided references establish the context of high-speed, low-latency networks and TCP challenges [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21], demonstrate the use of ML/DL in network or related fields [5, 6, 18, 22, 23, 26, 27, 28, 29], and show RST's capabilities in other prediction tasks [24, 25], none of these references describe or validate a system that combines RST specifically for network congestion prediction on network towers. This article outlines a conceptual design based on synthesizing insights from these distinct areas of research.

Future work should focus on implementing the proposed RST-based model and evaluating its performance using real-world or realistic simulated network data [30]. Comparative studies against other machine learning techniques [5, 6, 18, 22, 23, 26, 27, 28, 29] will be necessary to validate the advantages of the RST approach in this specific domain. Research into optimizing RST algorithms for speed, developing robust discretization strategies for network data, and effectively integrating the prediction output with real-time network control mechanisms are also critical next steps.

conclusion

The increasing demands of high-speed, low-latency communication necessitate a move beyond reactive congestion control towards proactive prediction. This article has proposed a design concept for a network congestion prediction model intended for deployment on network towers, leveraging the strengths of Rough Set Theory [24, 25]. RST offers a powerful framework for identifying key predictive network metrics through attribute reduction and extracting interpretable decision rules, overcoming some limitations of black-box models [24, 25]. While the provided references highlight the network context [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21], ML techniques [5, 6, 18, 22, 23, 26, 27, 28, 29], and RST in other domains [24, 25], the specific combination for tower-based congestion prediction is a novel conceptual design. Challenges related to data handling [30], computational efficiency, temporal dynamics, and integration with network infrastructure must be addressed through future research and implementation efforts. Realizing this approach could lead to more intelligent and responsive networks capable of maintaining the stringent performance requirements of future high-speed, low-latency applications.

REFERENCES

1. L. Li, P. Sharma, M. Gheisari and A. Sharma, “Research on TCP performance model and transport agent architecture in broadband wireless network”, Scalable Computing: Practice and Experience, 2021, vol. 22, no. 2, pp. 193-201.
2. M. Prakash and A. Abdrabou, “On the fidelity of ns-3 simulations of wireless multipath tcp connections”, Sensors, 2020, vol. 20, no. 24, pp. 7289.
3. M. H. Hanin, M. Amnai and Y. Fakhri, “New adaptation method based on cross layer and TCP over protocols to improve QoS in mobile ad hoc network”, International Journal of Electrical & Computer Engineering, 2021, vol. 11, no. 3, pp. 2088-8708.
4. M. N. Girish and A. Pande, “Improving TCP performance in wireless networks”, 2020.
5. S. P. Kuppusamy and M. Subramaniam, “Deep learning-based TCP congestion control algorithm for disaster 5G environment”, 2023.
6. M. R. Kanagarathinam, S. Singh, I. Sandeep, H. Kim, M. K. Maheshwari, J. Hwang, A. Roy and N. Saxena, “NexGen D-TCP: Next generation dynamic TCP congestion control algorithm”, IEEE Access, 2020, vol. 8, pp. 164482-164496.
7. E. Sy, T. Mueller, C. Burkert, H. Federrath and M. Fischer, “Enhanced performance and privacy for TLS over TCP fast open”, arXiv preprint arXiv:1905.03518. 2019.
8. T. Chowdhury and M. J. Alam, “Performance evaluation of TCP Vegas over TCP Reno and TCP NewReno over TCP Reno”, JOIV: International Journal on Informatics Visualization, 2019, vol. 3, no. 3, pp. 275-282.

9. A. Hernawan, “Comparative Performance Testing of the Impact of ACK Loss in TCP Tahoe, TCP Reno, and TCP New Reno on the ns-2 Simulator”, JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING, 2023, vol. 7, no. 1, pp. 91-101.

10. V. U. Devi and A. Thilaka, “A Comprehensive Analysis and Comparison of TCP Reno and TCP New Reno”.