GRU-Based Learning for the Identification of Congestion Protocols in TCP Traffic

Paul Bergeron & Sandhya Aneja
Marist Joint Study
Marist University, Poughkeepsie, NY

MARIST UNIVERSITY

Why Identify TCP Congestion Control?

TCP Congestion Control as a feature for network traffic analysis

Network Performance:

- Measure throughput and delay
- Analyze loss and jitter
- Optimize network configuration
- Better capacity planning

Security Applications:

- Device fingerprinting
- Browser fingerprinting
- Web server identification
- Cybersecurity analysis

TCP Congestion Control Protocols Under Study

TCP Reno: Loss-based approach

- ➢ Reduces window by half on packet loss
- ➢ Linear increase (additive increase)
- ➢ Classic TCP congestion control
- ➢ Sawtooth pattern behavior

TCP Cubic: Enhanced loss-based

- ➢ Uses cubic function for window adjustment
- ➢ Aggressive growth when underutilized
- ➢ Smoother convergence near saturation
- ➢ Default in most Linux systems

TCP Vegas: Delay-based approach

- ➢ Compares expected vs actual throughput
- ➢ Uses α and β thresholds
- ➢ Proactive congestion detection
- ➢ Prevents packet loss before it occurs

BBRv1: Model-based approach

- ➢ Explores bandwidth and delay characteristics
- ➢ Exponential increase during startup
- ➢ Keeps window ~3× bandwidth-delay product
- ➢ Optimizes for throughput and latency

Key Observations from Network Traffic

Each protocol follows distinct window adjustment patterns -> Machine Learning is feasible!

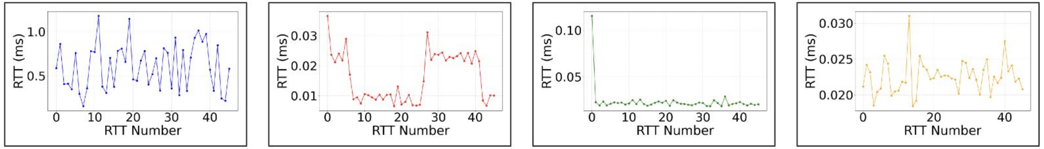
Throughput Ranking: BBR > Cubic > Reno > Vegas

- BBR achieves highest throughput
- Aggressive probing behavior
- Maximum network utilization

Round-Trip Time (RTT): Lower RTT: Vegas, Cubic & Reno

Higher RTT: BBR

- BBR exhibits higher RTT due to larger in-flight data volume
- Cubic achieves minimum RTT in our experiments



(e) RTT - BBR (f) RTT - CUBIC (g) RTT - RENO (h) RTT - VEGAS

Figure 1.1: Size and RTT Variation for BBR, CUBIC, RENO, and VEGAS

Formal Problem Definition

Main Goal: Distinguish behavior our communication flows, where each flow operates under one of a set possible congestion control protocols

Key Hypotheses:

1. **Temporal Pattern Similarity** Flows using the same protocol exhibit similar temporal patterns in their round-trip time sequences
2. **Protocol-Governed Behavior** The number of bytes transmitted during each round trip reflects the underlying protocol behavior
3. **Temporal Modeling Required** Patterns like TCP Reno's sawtooth or BBR's bandwidth ramp-up cannot be identified from single snapshots—they require modeling temporal evolution

Solution Approach: -> GRU (Gated Recurrent Unit) with Attention Mechanism

Choosing the Right Neural Network

"Protocol patterns require temporal evolution modeling. GRUs excel at capturing how congestion windows change over time, the essence of CC identification."

Key Advantages:

✓ Captures temporal dependencies in sequences

✓ Computationally efficient compared to LSTM

✓ Less prone to overfitting on mid-sized datasets

✓ Maintains competitive performance

✓ Integrates well with attention mechanisms

✓ Better for real-time applications

Our Design:

Architecture: 3-layer bidirectional GRU

Hidden size: 512 units

Attention mechanism: Enabled

Dropout rate: 0.4 (regularization)

Sequence length: 60 time steps

Real-World Network Testing Environment

Infrastructure:

🖥️ Server: Virtual machine from ECRL, Marist University

🌐 Network: 1 Gbps bottleneck link

📦 Transfer size: 500 MB per test

📊 Capture tool: Wireshark (pcap format)

🔄 Protocol switching: Automated via SSH

Data Collection Schedule:

⏰ Duration: 15 consecutive days

📅 Frequency: 3 times daily

🕕 Times: 6:00 AM, 12:00 PM, 6:00 PM

🤖 Automation: crontab scheduling

Features Extracted (100ms intervals):

1. Size (bytes transmitted)
2. Max Window Size
3. Throughput (Mbps)
4. Smoothed Throughput
5. Round-Trip Time (RTT in ms)

GRU Model Configuration

Model Architecture:

- ➢ Type: Bidirectional GRU
- ➢ Layers: 3 layers
- ➢ Hidden size: 512 units per layer
- ➢ Dropout: 0.4 between layers
- ➢ Sequence length: 60 time steps
- ➢ Attention mechanism: Integrated

Training Configuration:

- ➢ Loss function: Cross-entropy
- ➢ Optimizer: Adam (learning rate = 0.000075)
- ➢ Scheduler: ReduceLROnPlateau (factor=0.5, patience=5)
- ➢ Epochs: 30
- ➢ Batch size: 8

Data Split: Training: 70% | Validation: 10% | Test: 20%

Performance Metrics: → Classification accuracy (%) → Cross-entropy loss

Accuracy Achieved

Dataset Distribution:

Protocol	Samples	Percentage
TCP Vegas	3,221	38.6%
TCP Reno	1,802	21.6%
TCP Cubic	1,777	21.3%
BBRv1	1,629	19.5%

Total Samples: 8,429

Network Conditions:

- Bandwidth: 1 Gbps
- Delay: 0.09 – 0.10 ms
- Environment: Campus network (real-world, competitive)

Data Handling:

✓ Dataset balanced by standardizing to minimum size
✓ Attention mechanism handles protocol context transitions
✓ Training and validation loss converged smoothly

Key Result: 🎯 Test Accuracy: 97.04%

How We Compare to Existing Research

Comparison Table:

Method	Approach	Network	Accuracy	Limitation
TBIT (Pahdye & Floyd, 2001)	Heuristic rules	Active probing	Rule-based	Requires server cooperation
CAAI (Yang et al., 2014)	Active probing	30,000 servers	Varies	Limited to active probing
DeepCCI (Sander et al., 2019)	CNN + LSTM	2-50 Mbps, 0-50ms delay	99%	Controlled environment
Our Work (2025)	**GRU + Attention**	**1 Gbps, 0.09ms delay**	**97.04%**	**Real campus network**

Our Advantages:

✓ Faster neural network architecture (GRU vs CNN+LSTM)

✓ More complex and competitive network environment

✓ Comparable high accuracy

✓ Works with encrypted traffic (metadata only)

✓ Passive identification (no active probing needed)

Research Contributions

Contribution #1: High Accuracy 97.04% accuracy in identifying congestion control algorithms using an RNN-based GRU model

- Real-world campus network testing
- Competitive 1 Gbps environment
- Multiple daily conditions

Contribution #2: Feature Identification Identified Congestion Control as a representative feature that encapsulates:

- Packet size patterns
- Maximum window size behavior
- Throughput characteristics
- Smoothed throughput trends
- Round-trip time variations

Contribution #3: Network Characterization Identified key characteristics of Marist Campus network:

- Bottleneck link: 1 Gbps at Hancock Building
- Maximum throughput: Achieved by BBRv1
- Minimum RTT: Achieved by TCP Cubic

Limitations & Future Work

Current Limitations:

Environment-Specific Characteristics

- ➢ Different networks exhibit distinct behaviors
- ➢ Data centers: Stringent delay requirements
- ➢ WiFi/Cellular: Variable throughput and RTT
- ➢ May affect accuracy in different contexts

Protocol Coverage

- ➢ Limited to four protocols in current study
- ➢ Many newer protocols emerging
- ➢ Need broader protocol evaluation

Future Research Directions:

Expand Testing Environments
- ➢ Satellite
- ➢ Wireless networks (WiFi, 5G)
- ➢ Wide-area networks
- ➢ Different bandwidth conditions

Additional Protocols
- ➢ BBRv2, BBRv3, QCC
- ➢ QUIC congestion control
- ➢ HTTP/2, HTTP/3
- ➢ Newer emerging protocols

Real-Time Implementation
- ➢ Live traffic identification
- ➢ Streaming classification
- ➢ Low-latency inference

Cross-Environment Validation
- ➢ Transfer learning across networks
- ➢ Robustness testing
- ➢ Generalization studies

Conclusion & Impact

Main Achievement: *97.04% accuracy in identifying TCP congestion control protocols on a competitive 1 Gbps campus network*

Research Summary:

Method: GRU-based neural network with attention mechanism

- 3-layer bidirectional architecture
- Temporal pattern recognition
- Efficient and effective

Protocols Identified: TCP Reno, TCP Cubic, TCP Vegas, and BBRv1

- Distinct behavioral patterns
- Consistent classification
- Real-world testing

Key Advantages:

✓ Works with encrypted traffic using only metadata

✓ Passive identification (no active probing)

✓ Applicable to diverse use cases

Impact Areas:

- Network management and optimization
- Security and device fingerprinting
- Performance analysis and troubleshooting
- Traffic classification for QoS

Questions?

Thank you for your attention!

Contact Information:

Authors: Paul Bergeron, Paul.Bergeron1@marist.edu

Dr. Sandhya Aneja, sandhya.aneja@marist.edu

Institution: School of Computer Science and Mathematics
Marist University Poughkeepsie, NY, USA

Research Area: Network Traffic Analysis | Machine Learning | Congestion Control