HOKKAIDO UNIVERSITY

Title	超低遅延ネットワークアーキテクチャL4Sを用いた遠隔強化学習の環境構築
Author(s)	奥村, 丈; 三橋, 力麻; 飯田, 勝吉 他
Citation	電子情報通信学会技術研究報告, 125(320), 37-41
Issue Date	2026-01-15
Doc URL	https://hdl.handle.net/2115/100410
Rights	Copyright ©2026 IEICE
Type	conference paper
File Information	NS2025-198.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
NS2025-198(2026-01)

超低遅延ネットワークアーキテクチャL4Sを用いた遠隔強化学習の環境構築

奥村　　丈† 　三橋　力麻†† 　飯田　勝吉††† 　高井　昌彰†††

† 北海道大学 工学部 情報エレクトロニクス学科 情報理工学コース 先端ネットワーク研究室
〒060–0811 札幌市北区北 11 条西 5 丁目
†† , ††† 北海道大学 情報基盤センター 〒060–0811 札幌市北区北 11 条西 5 丁目
E-mail: †okumura.jou.d2@elms.hokudai.ac.jp, ††mitsuhashi@tanega.iic.hokudai.ac.jp,
†††{iida,ytakai}@iic.hokudai.ac.jp

あらまし　近年、自動運転車の制御や自律ドローンの操作などを遠隔で学習・実行するシステムの検討が活発になっている。機械制御に遠隔強化学習を適用したシステムでは、実際の機器とサーバの間でデータを頻繁にやり取りするため、ネットワーク品質が制御性能や安全性に大きく影響する。最近では、ネットワーク遅延を抑制する手法として RFC9330 で標準化された L4S が注目を集めているが、現時点で遠隔強化学習に L4S を適用した報告はほとんど見られていない。そこで本研究では、遠隔強化学習の基礎的評価を目的として、L4S を適用した高品質なネットワーク上に遠隔強化学習の環境構築を行う。

キーワード　遠隔強化学習, 倒立振子, L4S, ns-3, DEFIANCE

Environment Construction of Remote Reinforcement Learning Using L4S Ultra-low Latency Network Architecture

Jo OKUMURA†, Rikima MITSUHASHI††, Katsuyoshi IIDA†††, and Yoshiaki TAKAI†††

† Laboratory of Advanced Information Network, Course of Computer Science and Information Technology,
School of Engineering, Hokkaido University, Kita 11 Nishi 5, Kita-ku, Sapporo, 060–0811, Japan.
†† , ††† Information Initiative Center, Hokkaido University, Kita 11 Nishi 5, Kita-ku, Sapporo, 060–0811, Japan.
E-mail: †okumura.jou.d2@elms.hokudai.ac.jp, ††mitsuhashi@tanega.iic.hokudai.ac.jp,
†††{iida,ytakai}@iic.hokudai.ac.jp

Abstract　In recent years, research on systems that enable remote learning and the execution of tasks, such as autonomous vehicle control and drone operation, has been advancing. Since systems applying remote reinforcement learning frequently exchange data between physical devices and servers, network quality significantly impacts performance and safety. Recently, L4S, standardized as RFC9330, has gained attention as a method for reducing network latency. However, few reports currently exist on applying L4S to remote reinforcement learning. Therefore, in this study, we construct a remote reinforcement learning environment on a high-quality network utilizing L4S for fundamental evaluation of remote reinforcement learning.

Key words　Remote reinforcement learning, cartpole, L4S, ns-3, and DEFIANCE.

1. はじめに

近年、自動運転車の制御や自律ドローン操作など機械制御の高度化が進み、これらの制御アルゴリズムを遠隔で学習・実行するシステムの検討が活発になっている [1], [2]。機械制御に遠隔強化学習を適用したシステムでは、機械側で取得した観測データが中央制御や学習を行うサーバ側に送信され、そこで更新された方策に基づく次の行動が機械側へ返されることになる。つまり、実際の機器とサーバとの間でデータを頻繁にやり取りするため、ネットワークの帯域、遅延、輻輳といった品質が機械制御の性能や安全性に大きく影響する。

従来、ネットワーク遅延を抑制する手法としては、CoDel (Controlled Delay) [3]〜[5] や PIE (Proportional Integral Controller Enhanced) [6] などの AQM 技術が利用されてきたが [7]〜[10]、

This article is a technical report without peer review, and its polished and/or extended version may be published elsewhere.
Copyright ©2026 by IEICE

遠隔強化学習を適用するネットワーク環境では、より大きな性能改善が期待される。こうした中、最近では、RFC9330 として標準化された L4S (Low Latency, Low Loss, Scalable throughput) [11] は、ECN（Explicit Congestion Notification) [12], [13] を活用した柔軟な輻輳検知・回避により、低いキューイング遅延、低い輻輳損失、スケーラブルなスループット制御を実現できる次世代通信アーキテクチャとして注目を集めている。

L4S の適用範囲としては、動画配信サービス、仮想現実 (VR) や拡張現実 (AR)、オンラインゲームなど高品質なネットワークを要求するサービスでの利用が想定されており [14]、自動運転車や産業用ロボットなどを機械制御するための遠隔強化学習と L4S の親和性は高いと考えられるが、現時点で遠隔強化学習に L4S を適用した報告はほとんど見られていない。

そこで本研究では、遠隔強化学習の基礎的評価を目的として、L4S を適用した高品質なネットワークを用意したうえで、機械制御の基本の一つである倒立振り子を対象とした遠隔強化学習の環境構築を行う。具体的には、ネットワークシミュレータである ns-3 [15] と、ns-3 上で遠隔強化学習を実現する DEFIANCE [16] を用いて倒立振子の遠隔制御を実現する。

2. 関 連 研 究

前章で述べたとおり、本研究では、L4S を適用した高品質なネットワークを用意したうえで、倒立振り子を対象とした遠隔強化学習の環境構築を行う。以下では、L4S と倒立振子の遠隔制御についての関連研究を説明する。

2.1 超低遅延ネットワークアーキテクチャ L4S

L4S の中核となるアイデアは、エンドホストとネットワーク機器が協調して遅延の増加を事前に検出し、輻輳発生前に送信レートを微調整することである。その実現手段として、L4S では ECN を用いた輻輳検知シグナルが導入されている。L4S 対応のパケットは ECT(1) としてラベル付けされ、ECN マーキングが生じると、エンドホストはパケットロスを待たずに輻輳を検知し、送信レートを調整する。さらに L4S 非対応の通信を Classic サービスと呼び、Classic のパケットには ECT(0) とラベル付けすることで、ルータが Classic と L4S パケットを区別し、L4S パケットの優先送出を可能とする。

このアプローチにより、従来の輻輳制御が直面していたバッファ滞留によるキューイング遅延を抑制することが可能となり、ネットワーク遅延を数ミリ秒以下のレベルに維持することが可能となる。また、スループットのスケーラビリティを犠牲にすることなく、安定した低遅延通信が提供される。

最近の研究では、Graff ら [17] は Cloud Gaming を対象として、従来のトラフィック制御方式と L4S 方式を比較評価した。実験では、L4S を適用したルータのキューイングに DualPI2 AQM [18] を用いており、classic トラフィック（大容量追求型）」と「scalable トラフィック（低遅延重視型）」を異なるキューに分離することで低遅延重視型のトラフィックに対して低遅延かつ低損失な転送環境を実現した。実験結果では、Cloud Gaming を利用した環境における QoS (遅延、ジッタ、パケット損失) の改善が報告されている。

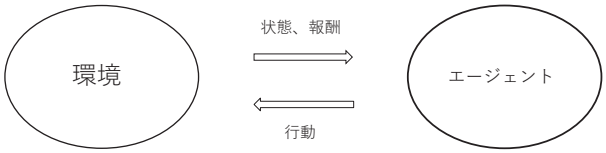


図 1　シングルエージェント強化学習

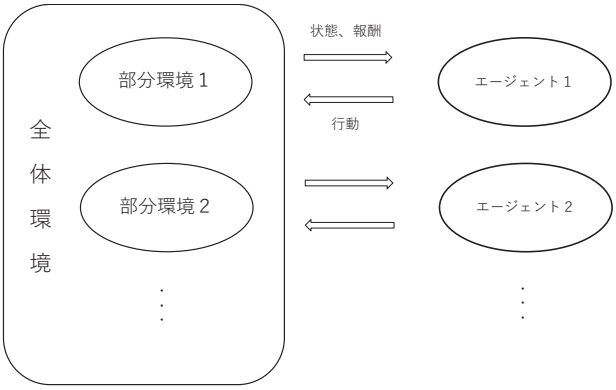


図 2　マルチエージェント強化学習

2.2 倒立振子の遠隔制御

ネットワーク越しに倒立振り子を制御する試みについては、多くの先行研究がある。例えば、Stanoev ら [19] は、無線通信を用いたクローズドループ制御を用いて、二輪型倒立振り子を遠隔で安定させた成果を報告している。また、Music ら [20] は、無線を介したネットワーク環境において、通信遅延や遅延の変動など不確実性を考慮した制御性能と安定性を評価している。

しかしながら、これらの先行研究はいくつかの制限や課題があることに注意する必要がある。既存の遠隔機械制御は、主にモデル不確実性や物理パラメータの変動に対処するためのものであって、通信ネットワークにおける遅延やパケット損失の影響を除去することや低減することを目的としたものではない。本研究では、倒立振子の遠隔制御で利用されるネットワークに対して L4S を適用することで、低いキューイング遅延、低い輻輳損失、スケーラブルなスループット制御を実現し、倒立振り子の遠隔制御に関する性能向上を目指す。

3. 提案手法：遠隔強化学習への L4S の適用

前章で述べたとおり、本研究では、倒立振子を対象とした遠隔強化学習の環境構築を行う。以下では、遠隔強化学習へ L4S を適用するための提案手法について述べる。

3.1 マルチエージェントによる強化学習

強化学習（Reinforcement Learning）は、エージェントが環境との試行錯誤を通じて、将来にわたる報酬の累積を最大化する行動方策を学習する機械学習の一分野である。教師データを用いて正解を学習する教師あり学習とは異なり、強化学習ではエージェント自身が行動の結果として得られる報酬を基に、望ましい行動を自律的に獲得する点に特徴がある。

図 1 に一般的な強化学習である単一エージェントの強化学

習環境を示す。図に示すとおり、強化学習ではエージェントは環境の状態を観測し、その状態に基づいて行動を選択する。行動の結果として環境の状態は遷移し、同時に報酬が与えられる。エージェントの目的は、これらの報酬の累積値が最大となるような行動方策を学習することである。

本研究では、柔軟性や今後の拡張を考え、マルチエージェントによる強化学習環境を実現する。図 2 にマルチエージェント強化学習環境を示す。図にある通り、各エージェントは対応する制御対象の状態観測を入力として行動を決定し、得られた報酬に基づいて方策を更新する。この分散化により、マルチエージェント環境における学習データの収集効率が向上し、学習時間の短縮を図る。

さらに本研究では強化学習の枠組みを実現する方法として、方策勾配法に基づくアルゴリズムである PPO (Proximal Policy Optimization) を採用する。PPO は方策の更新幅を制限することで学習の安定性を向上させる手法であり、複雑な制御問題に対しても比較的安定した学習が可能であることが知られている。

遠隔強化学習の環境では通信遅延やジッタにより、観測された状態と実際に行動が適用されるタイミングとの間にずれが生じる可能性があるため、方策の急激な変化が学習性能の低下を招く恐れがある。PPO は保守的な方策更新を行う設計であり、通信条件が理想的でない環境においても比較的頑健な学習挙動を示すと期待されるため、本研究では遠隔強化学習のアルゴリズムとして PPO を採用した。

3.2 L4S によるネットワークの低遅延化

マルチエージェント強化学習においては、各エージェントが観測した状態に基づいて行動を決定し、その結果が環境に反映されるまでの時間的遅延が学習および制御性能に大きな影響を与える。特に、遠隔制御環境では、通信遅延やジッタにより、エージェントが取得する状態情報が実際の環境状態と乖離する可能性がある。

この課題に対して、本研究では低遅延通信を実現するためのネットワークアーキテクチャとして L4S を用いる。L4S を適用したネットワークでは、L4S トラフィックと従来のトラフィックを論理的に分離し、低遅延を要求するトラフィックに対して優先的かつ低遅延なキュー管理を行う。これによりキューイング遅延を数ミリ秒以下に抑えることが可能となる。

本研究では、マルチエージェント強化学習における状態、報酬、行動のデータをやり取りする通信経路に L4S を適用する。これにより遠隔強化学習に用いられるデータが滞りなく送受信されることで、エージェントが効率的に行動を決定できるようになることが期待される。

4. 評価環境の構築

前章で述べたとおり、本研究ではマルチエージェントによる強化学習と L4S によるネットワークの低遅延化を実現した評価環境を構築する。以下では、DEFIANCE を用いた環境・設定とネットワーク構成について説明する。

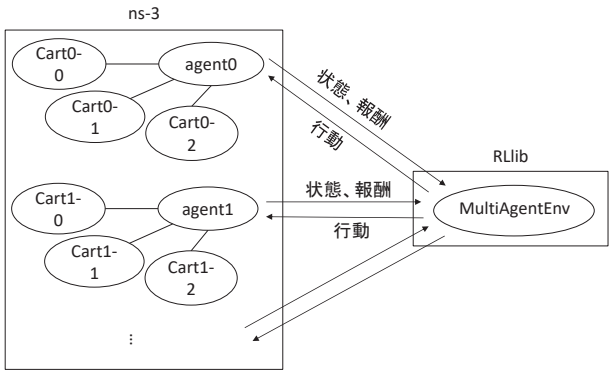


図 3　DEFIANCE におけるマルチエージェント強化学習環境

4.1 DEFIANCE: ns-3 におけるマルチエージェント強化学習フレームワーク

本研究では、評価環境の構築にあたり、ns-3 上でマルチエージェントによる強化学習を提供する DEFIANCE [16] を用いる。DEFIANCE はネットワークシミュレーション環境上で強化学習を可能とするフレームワークである。

従来、ns-3 上で強化学習を評価する際には、ns3-gym [21] や ns3-ai [22] などのモジュールが用いられてきた。しかし、これらの手法では、強化学習エージェントの分散配置や、複数エージェントによる協調・競合を伴う学習を十分に扱うことが困難であった。特に、強化学習の各構成要素（観測、行動、報酬、エージェント）がネットワーク上の異なるノードに配置されるような実運用を想定した評価は容易ではなかった。

DEFIANCE はこの課題に対し、強化学習プロセスを構成する要素を観測アプリケーション、エージェントアプリケーション、行動アプリケーション、報酬アプリケーション の 4 種類の RL アプリケーションとして明確に分離し、それぞれを ns-3 の任意のノードに配置可能とする設計を採用している。このモジュール化により、強化学習アルゴリズム自体を変更することなくエージェントの配置場所や通信経路のみを変更した実験が可能となる。また、DEFIANCE は Ray/RLlib [23] が提供するマルチエージェントインタフェースを実装しており、複数エージェントが同一環境内で学習・推論を行う構成を扱うことができる。図 3 に DEFIANCE におけるマルチエージェント強化学習環境を示す。DEFIANCE は ns-3 内に配置した複数のエージェントの強化学習に関する計算を、ns-3 の外にある Rlib MultiAgentEnv にて実行するためのフレームワークである。

さらに、DEFIANCE では強化学習におけるアプリケーション間の通信を抽象化するチャネルインタフェースが導入されており、TCP/IP を用いたネットワーク通信だけでなく、遅延を明示的に設定した通信も扱うことができる。この設計により、通信遅延やジッタが強化学習の性能に与える影響を体系的に評価することが可能となる。

以上の特徴から、DEFIANCE は「マルチエージェント強化学習」と「L4S を適用可能なネットワーク構成」を同時に実現可能といえる。本研究では、DEFIANCE が提供するフレームワークを拡張し、L4S による低遅延通信を用いた遠隔強化学習

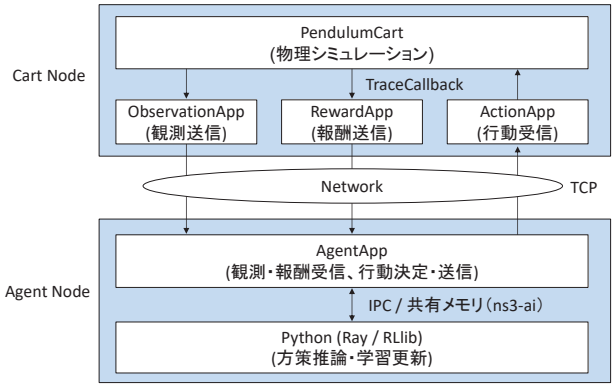


図 4　ノード間データフロー

の環境を構築する。

4.2　DEFIANCE を用いた環境構築

DEFIANCE では、推論エージェントの各機能をノードと呼び、ノード間でネットワーク通信することで全体として強化学習を実現する。本研究では、DEFIANCE を用いて倒立振子制御シナリオを構築する。倒立振子制御シナリオでは、制御対象となるカートノードおよび推論を行うエージェントノードを配置する。図 4 に今回 DEFIANCE を用いて開発したシミュレーション環境における、ノード間のデータフローを示す。なお、DEFIANCE には倒立振子制御を実現するサンプルコードとして `balance1-scenario.cc` と `balance2-scenario.cc` が付属されている。本研究では、マルチエージェントで倒立振子を実現する `balance2-scenario.cc` をベースコードに採用し、L4S の実現などの拡張を行った。

図 4 において、カートノードには、以下の 4 つのコンポーネントアプリケーションが配置する。1 つ目の PendulumCart は、倒立振子の物理シミュレーションを担当するクラスであり、振子の角度・角速度、カートの位置・速度といった状態量を管理する。シミュレーション時間の進行に伴い、物理モデルに基づいて状態を更新する。2 つ目の ObservationApp は、PendulumCart から状態情報を取得し、観測データとしてエージェントノードに送信する。3 つ目の RewardApp は、PendulumCart の状態に基づいて報酬を計算し、その結果をエージェントノードに送信する。4 つ目の ActionApp は、エージェントノードから受信した行動（カートへの加速度指令）を PendulumCart に適用する。

強化学習は以下の手順で実行される。まず CartNode の PendulumCart により物理状態が更新され、その結果が ObservationApp および RewardApp に通知される。次に ObservationApp は観測データを、RewardApp は報酬データをそれぞれエージェントノードの AgentApp に送信する。そして AgentApp は受信した観測データおよび報酬データを Python (Ray/RLlib) に渡し、学習済み方策に基づいて行動を決定する。その後、決定された行動データはカートノードの ActionApp に送信される。ActionApp は受信した行動を PendulumCart に適用する。これらを一連の流れとして何度も繰り返すことにより、期待累積報酬が最大化される。

強化学習の手順において、カートノードとエージェントノードの間はネットワークを経由するため、ボトルネックリンク

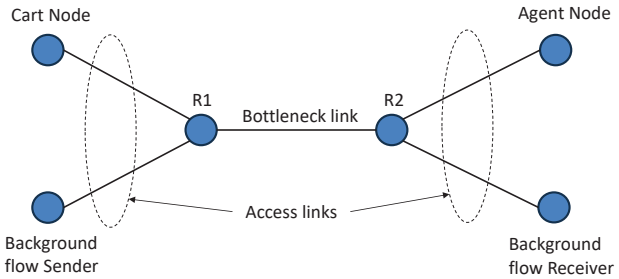


図 5　ネットワークトポロジー

に輻輳が発生した場合、観測データや行動データの到着遅延が生じる。この遅延は制御性能および学習安定性に直接影響を与えるため、低遅延通信を実現する L4S アーキテクチャの有効性を評価する上で重要な要素となる。

4.3　ネットワーク構成

DEFIANCE 付属の `balance2-scenario.cc` の初期設定では、カートノードとエージェントノードのネットワークが LTE（Long Term Evolution）で構成されている。本研究ではこれを Point-to-Point（P2P）に置き換えて、図 5 に示すネットワークトポロジーを構成する。本トポロジでは、ボトルネックリンクを配置し、ボトルネックルータ上で L4S を適用する。また、すべての通信フローはボトルネックリンクを通過するため、背景フローが強化学習の性能に与える影響の調査を可能とする。

なお、図 4 で示した通り、カートノードとエージェントノード間の通信のために 3 つの TCP 接続を設定する。観測と報酬に関するものは、カートノードからエージェントノードに向かう方向（右向き）の通信を行い、行動に関するものは逆にエージェントノードからカートノードの方向（左向き）で通信する。図 5 における背景フローはカートノードからエージェントノードに向かう方向（右向き）の通信であるため、背景フローは観測と報酬の TCP 接続に対する妨害トラヒックとなる。

具体的な ns-3 上での実装については、EnvironmentCreator クラスの SetupPendulumScenario メソッドを再設計する。まず、PointToPointHelper を用いて各リンクを構成し、ボトルネックリンクには帯域幅および遅延パラメータを可変設定可能とする。次に、競合する背景トラフィックを追加できるように、ルーターとのリンクを作成する。その後、TrafficControlHelper を用いて、ボトルネックリンクに FqCoDelQueueDisc をインストールする。FqCodelQueueDisc は L4S の提供を可能とする AQM のうち、ns-3 で幅広く用いられる Flow Queue Controlled Delay (FqCodel) AQM [24] を提供するものである。FqCodel の設定としては、L4S モード有効時には、CeThreshold を 1ms に設定し、低遅延トラフィックに対する即時 ECN マーキングを実現する。

さらに、TcpSocketBase の UseEcn 属性を有効化する。TCP 輻輳制御については、TcpL4Protocol の SocketType 属性を通じて Data Center TCP (DCTCP) [25] を選択可能とする。DCTCP は、ECN マーキングの頻度に応じて送信ウィンドウを段階的に調整する輻輳制御アルゴリズムであり、L4S アーキテクチャにおける低遅延フローの実現に不可欠である。ns-3 の TcpDctcp 実装では UseEct0 属性により ECT マークを制御でき、この属性を

false に設定することで ECT(1) を使用する L4S フローとして動作する。本研究では強化学習フローには DCTCP と ECT(1) を適用し、背景フローの TCP 輻輳制御に NewReno を適用することで、FqCoDel の L4S モードによるフロー分離を実現する。さらに、DEFIANCE フレームワークにおける強化学習通信のソケット作成部分（SocketChannelInterface クラス）を拡張し、TcpL4Protocol の CreateSocket メソッドを通じて DCTCP 輻輳制御を個別に適用できるようにする。これにより、強化学習フローのみに L4S 対応の低遅延 TCP 設定を適用し、背景フローには従来の NewReno を使用するというフロー種別に応じた輻輳制御の使い分けが可能とする。

以上をまとめると、構築した環境により L4S による低遅延化がマルチエージェント強化学習の学習効率および制御性能に与える影響の定量的な評価が可能となる。

5. おわりに

機械制御に遠隔強化学習を適用したシステムでは、実際の機器とサーバの間でデータを頻繁にやり取りするため、ネットワーク品質が制御性能や安全性に大きく影響する。本研究では、遠隔強化学習の基礎的評価を目的として、L4S を適用した高品質なネットワーク上に遠隔強化学習の環境を構築した。具体的には、ns-3 上でマルチエージェントによる強化学習を実現する DEFIANCE 上で、倒立振り子の遠隔制御のためのシミュレーション環境を設計・実装した。カートノードとエージェントノードの間にボトルネックリンクを明示的に配置することで、L4S をネットワークに適用するとともに、輻輳発生時に強化学習の性能に与える影響の調査を可能とした。

今後は、この環境を用いて遠隔強化学習の性能評価を進める予定である。具体的には、L4S 適用時と非適用時における通信遅延の比較、およびその遅延差が強化学習の収束速度や制御精度に与える影響の定量的評価を行う。また、ボトルネック帯域や背景トラフィック量といったネットワークパラメータを変化させ、遠隔強化学習に適したネットワーク設定の探索を予定している。

文　献

[1] A. Mahmood, L. Beltramelli, S. Fakhrul Abedin, S. Zeb, N.I. Mowla, S.A. Hassan, E. Sisinni, and M. Gidlund, “Industrial IoT in 5G-and-beyond networks: Vision, architecture, and design trends,” IEEE Transactions on Industrial Informatics, vol.18, no.6, pp.4122–4137, Sept. 2022.

[2] I. Behnke and H. Austad, “Real-time performance of industrial IoT communication technologies: A review,” IEEE Internet of Things Journal, vol.11, no.5, pp.7399–7410, March 2024.

[3] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar, “Controlled Delay Active Queue Management,” IETF RFC 8289, Jan. 2018. https://www.rfc-editor.org/info/rfc8289

[4] K. Nichols and V. Jacobson, “Controlling queue delay,” ACM Queue, vol.10, no.5, pp.1–15, May 2012.

[5] 妹尾龍馬，平澤崇佳，森弘樹，中務諭士，高橋賢，“超低遅延サービスの提供を見据えた遅延抑制技術 CoDel の適用性評価,” 信学技報，vol.119，no.460，pp.129–134，March 2020.

[6] R. Pan, P. Natarajan, F. Baker, and G. White, “Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem,” IETF RFC 8033, Feb. 2017. https://www.rfc-editor.org/info/rfc8033

[7] A.A. Mahawish and H.J. Hassan, “Survey on: A variety of AQM algorithm schemas and intelligent techniques developed for congestion control,” Indonesian Journal of Electrical Engineering and Computer Science, vol.23, no.3, pp.1419–1431, Sept. 2021.

[8] J. Luo, J. Jin, and F. Shan, “Standardization of low-latency TCP with explicit congestion notification: A survey,” IEEE Internet Computing, vol.21, no.1, pp.48–55, Jan. -Feb. 2017.

[9] F. Baker and G. Fairhurst, “IETF Recommendations Regarding Active Queue Management,” IETF RFC 7567, July 2015. https://www.rfc-editor.org/info/rfc7567

[10] R. Al-Saadi, G. Armitage, J. But, and P. Branch, “A survey of delay-based and hybrid TCP congestion control algorithms,” IEEE Commun. Surveys & Tutorials, vol.21, no.4, pp.3609–3638, Nov. 2019.

[11] B. Briscoe, K.D. Schepper, M. Bagnulo, and G. White, “Low Latency, Low Loss, and Scalable Throughput (L4S) Internet Service: Architecture,” IETF RFC 9330, Jan. 2023. https://www.rfc-editor.org/info/rfc9330

[12] S. Floyd, D.K.K. Ramakrishnan, and D.L. Black, “The Addition of Explicit Congestion Notification (ECN) to IP,” IETF RFC 3168, Sept. 2001. https://www.rfc-editor.org/info/rfc3168

[13] K.D. Schepper and B. Briscoe, “The Explicit Congestion Notification (ECN) Protocol for Low Latency, Low Loss, and Scalable Throughput (L4S),” IETF RFC 9331, Jan. 2023. https://www.rfc-editor.org/info/rfc9331

[14] G. Pan, S. Xu, and P. Jiang, “Optimizing 5G-advanced networks for time-critical applications: The role of L4S,” IEEE Wireless Communications, vol.32, no.2, pp.230–237, April 2025.

[15] “The network simulator ns-3,” https://www.nsnam.org/. Accessed: 2025/12/19.

[16] A.-B. Burmeister, J. Dittrich, A.M. Fennig, T. Rothe, and R. Singh, “DEFIANCE: Simulate reinforcement learning deployments in realistic network scenarios,” Proc. ACM Int’l Conf. on ns-3 (ICNS3’25), pp.54–61, Aug. 2025.

[17] P. Graff, X. Marchal, T. Cholez, B. Mathieu, S. Tuffin, and O. Festor, “Improving cloud gaming traffic QoS: A comparison between class-based queuing policy and L4S,” Proc. IEEE Network Traffic Measurement and Analysis Conference (TMA), pp.1–10, May 2024.

[18] K.D. Schepper, B. Briscoe, and G. White, “Dual-Queue Coupled Active Queue Management (AQM) for Low Latency, Low Loss, and Scalable Throughput (L4S),” IETF RFC 9332, Jan. 2023. https://www.rfc-editor.org/info/rfc9332

[19] A. Stanoev, A. Aijaz, A. Portelli, and M. Baddeley, “Demo: Closed-loop control over wireless - remotely balancing an inverted pendulum on wheels,” Proc. ACM Int’l Conf. Embedded Wireless Systems and Networks (EWSN’20), pp.195–197, Feb. 2020.

[20] Z. Music, F. Molinari, S. Gallenmüller, O. Ayan, S. Zoppi, W. Kellerer, G. Carle, T. Seel, and J. Raisch, “Design of a networked controller for a two-wheeled inverted pendulum robot,” Proc. IFAC Workshop on Distributed Estimation and Control in Networked Systems (NECSYS’19), vol.52, pp.169–174, Sept. 2019.

[21] P. Gaw lowicz and A. Zubow, “ns-3 meets OpenAI Gym: The playground for machine learning in networking research,” Proc. Int’l ACM Conf. Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM’19), pp.113–120, Nov. 2019.

[22] H. Yin, P. Liu, K. Liu, L. Cao, L. Zhang, Y. Gao, and X. Hei, “ns3-ai: Fostering artificial intelligence algorithms for networking research,” Proc. ACM Workshop on ns-3, pp.57–64, June 2020.

[23] The Ray team, “RLlib: Industry-grade, scalable reinforcement learning,” https://docs.ray.io/en/master/rllib/index.html. Accessed: 2025/12/23.

[24] T. Høiland-Jørgensen, P. McKenney, D. Taht, J. Gettys, and E. Dumazet, “The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm,” IETF RFC 8290, Jan. 2018. https://www.rfc-editor.org/info/rfc8290

[25] S. Bensley, D. Thaler, P. Balasubramanian, L. Eggert, and G. Judd, “Data Center TCP (DCTCP): TCP Congestion Control for Data Centers,” IETF RFC 8257, Oct. 2017. https://www.rfc-editor.org/info/rfc8257