INVESTIGATING THE IMPACT OF **ACK AGGREGATION** ON TCP PERFORMANCE USING NS-3

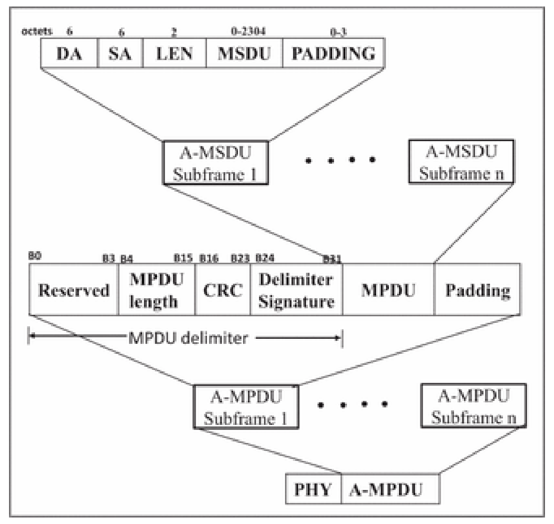
EVALUATION OF TRANSPORT AND MAC-LAYER AGGREGATION TECHNIQUES

AUTHOR

Hanna Heinczinger
Email: h.heinczinger-1@student.tudelft.nl

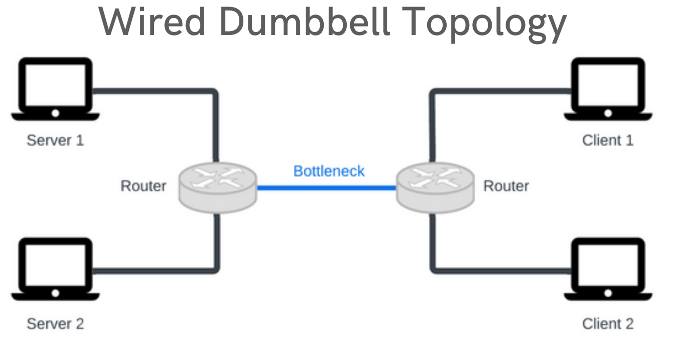
INTRODUCTION

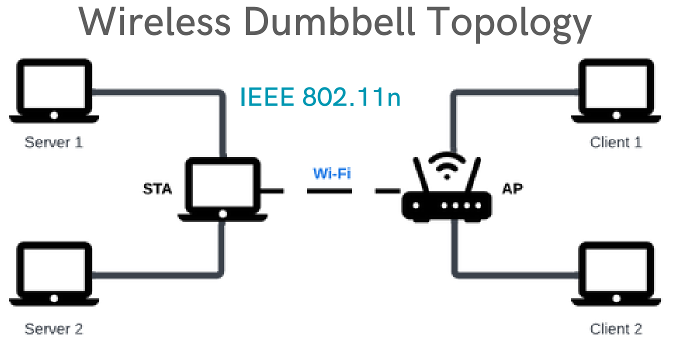
- **TCP**[1] is a core protocol of the Internet, enabling reliable communication across diverse networks.
- Despite its robustness, TCP performance can degrade in real-world conditions like wireless and asymmetric links [2].
- In such environments, **ACK** packets may be delayed, lost, or create unnecessary overhead [3][4].
- ACK suppression/aggregation techniques aim to reduce protocol overhead but can distort the timing of TCP feedback
- To study this, we evaluate how TCP behaves under two aggregation methods:
  - Router-based ACK aggregation, using a programmable queue to delay and suppress ACKs.
  - MAC-layer aggregation, using IEEE 802.11n features:
    - **A-MSDU**: Aggregates multiple payloads under one MAC header [8].
    - **A-MPDU**: Bundles full MAC frames with separate headers [12].
- These mechanisms alter TCP's feedback loop, and our study compares their impact on four modern congestion control algorithms in both wired and wireless topologies.



METHODOLOGY

Simulations are conducted using ns-3, a discrete-event network simulator well-suited for protocol-level TCP behavior analysis.





TCP variants

- BBRv3
- CUBIC
- NewReno
- Vegas

Aggregation methods:

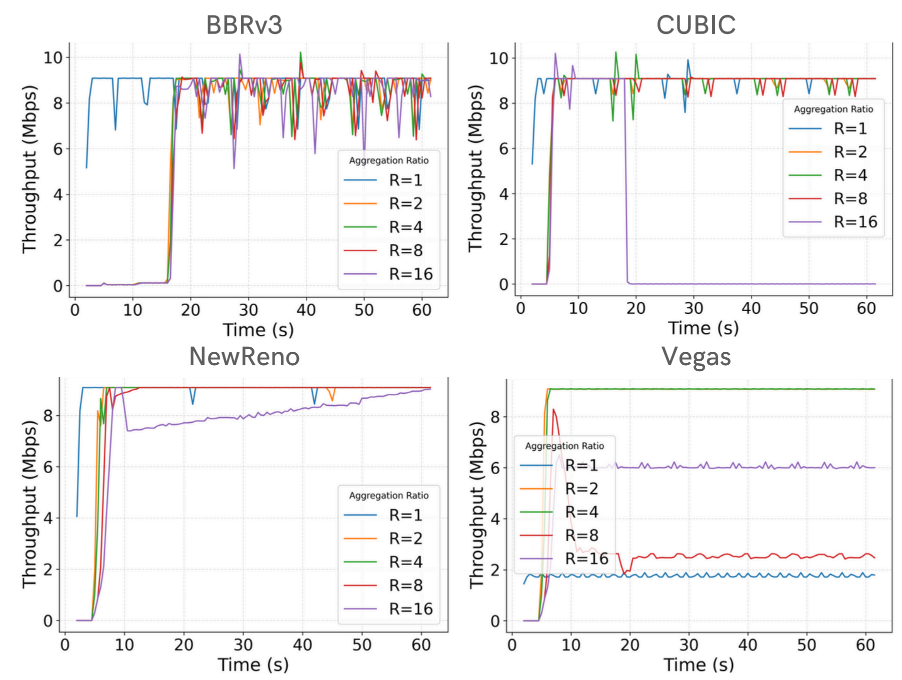
- Custom ACK aggregation queue (set aggregation ratio/timeout)
- A-MSDU/A-MPDU

METRICS

- Throughput
- Congestion Window Evolution
- Retransmissions
- Jain's Fairness Index

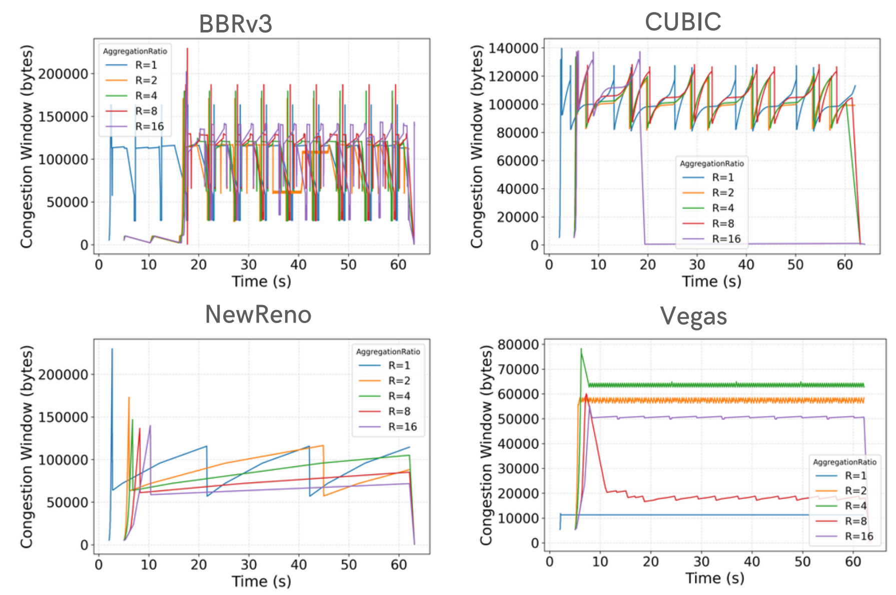
RESULTS

TRANSPORT-LAYER AGGREGATION



THROUGHPUT

- **BBRv3**: Throughput drops with any aggregation due to disrupted startup.
- **CUBIC**: Stable under moderate aggregation; collapses at 16:1.
- **NewReno**: Similar to CUBIC, but recovers after a dip at 16:1.
- **Vegas**: Unpredictable; performs better than baseline with moderate aggregation.

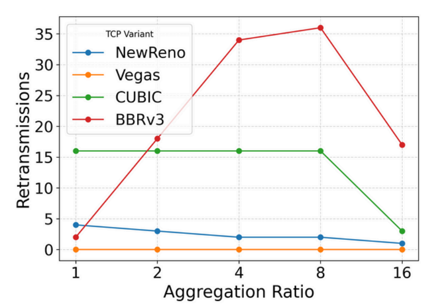


CONGESTION WINDOW

- **BBRv3**: Frequent oscillations due to RTT/bandwidth probing.
- **CUBIC**: Typical growth-drop cycles; slower ramp-up with more aggregation.
- **NewReno**: Clear sawtooth pattern; higher aggregation delays cwnd growth.
- **Vegas**: cwnd quickly stabilizes and stays flat; aggregation has minimal effect.

RETRANSMISSIONS

- **Vegas**: Zero at all levels; avoids loss by design.
- **NewReno**: Low and stable; conservative and reliable.
- **CUBIC**: Retransmissions increase due to aggressive probing.
- **BBRv3**: Initially low, but increases with aggregation.



FAIRNESS

- At low aggregation, bandwidth is shared fairly across all TCP variants (except for Vegas).
- Loss-based algorithms (CUBIC, NewReno) tend to dominate, while BBRv3 and Vegas often lose out due to sensitivity to sparse ACKs.

Algorithms	1:1	2:1	4:1	8:1	16:1	Generally Favoured Variant
BBRv3 vs. CUBIC	0.998	0.936	0.500	0.500	0.501	CUBIC
BBRv3 vs. NewReno	0.997	0.972	0.500	0.500	0.507	NewReno
BBRv3 vs. Vegas	0.557	0.559	0.500	0.653	0.846	BBRv3
CUBIC vs. NewReno	0.995	0.979	0,500	0.500	0.849	NewReno
CUBIC vs. Vegas	0.535	0.516	0.505	0.594	0.828	CUBIC
NewReno vs. Vegas	0.545	0.539	0.968	0.501	0.503	Vegas

MAC-LAYER AGGREGATION

Algorithm	baseline	A-MPDU + A-MSDU	A-MPDU	A-MSDU
BBRv3	11.90	41.43	40.24	37.57
CUBIC	11.17	41.19	40.33	37.23
NewReno	11.15	41.32	40.19	37.26
Vegas	11.94	17.34	16.52	16.68

THROUGHPUT

- **No aggregation**: All TCP variants show low throughput due to MAC overhead.
- **A-MSDU only**: ~3× throughput boost; reduces overhead by bundling data pre-header.
- **A-MPDU only**: Even better gains; subframes are independently ACKed and retransmitted.
- **Both enabled**: Highest throughput (~41 Mbps); combines efficiency and reliability.
- **Vegas**: Consistently lower; prefers A-MSDU due to smoother RTTs, struggles with A-MPDU's burstiness.

Varying A-MPDU

Algorithm	64KB	32KB	16KB	8KB	4KB
BBRv3	40.24	40.17	38.71	34.53	28.20
CUBIC	40.33	40.33	38.48	34.28	27.42
NewReno	40.19	40.11	38.46	38.46	27.22
Vegas	16.52	16.52	15.58	16.20	17.50

- Throughput improves with larger sizes for BBRv3, CUBIC, and NewReno, but gains flatten at 64 KB.
- Vegas degrades at first due to delay sensitivity, then stabilizes.
- Bigger A-MPDUs reduce MAC overhead but may hurt delay-based algorithms.

CONCLUSION

- Future work should explore fairness under MAC-layer aggregation and incorporate more realistic wireless conditions including mobility, interference, and Block ACK behaviour.
- MAC-layer aggregation (A-MSDU, A-MPDU) significantly boosts throughput by reducing protocol overhead, with two-level aggregation offering the best performance.
- In contrast, transport-layer aggregation (e.g., ACK suppression) is less effective, causing throughput drops, fairness loss, and more retransmissions, especially for algorithms needing timely feedback like BBRv3 and Cubic.
- NewReno is surprisingly robust across both scenarios, showing stable throughput and fairness. Vegas, however, is unstable under both, due to its RTT sensitivity and conservative response to burstiness.

RELATED LITERATURE

[1] Wesley Eddy. Transmission Control Protocol (TCP). RFC 9293, August 2022.
[2] H. Balakrishnan, V. Padmanabhan, and R. Katz, The effects of asymmetry on TCP performance, ACM Mobile Networks and Applications, 1999.
[3] Carlo Augusto Grazia, Natale Patriciello, Toke Hoiland-Jorgensen, Martin Klapez, and Maurizio Casoni. Aggregating without bloating: Hard times for tcp on wi-fi. IEEE/ACM Transactions on Networking, 30(5), 2022.
[4] M. Abrahamsson, TCP ACK Suppression, IETF AQM Mailing List, 2015.

TU Delft