

# Performance and Fairness Analysis of Multi-STA Wi-Fi Transport (P5)

## Abstract

This report analyzes the measured *goodput* and *fairness* of a multi-station (multi-STA) Wi-Fi scenario for two transports (UDP and TCP) and three network sizes ( $N \in \{2, 5, 10\}$ ). The primary goal is to interpret how aggregate throughput scales with contention, how the distribution of per-STA throughput changes with  $N$ , and why fairness degrades differently under UDP and TCP. The analysis is driven by the experiment summary (aggregate goodput and Jain index) and the accompanying per-STA goodput bar charts. The key findings are: (i) TCP aggregate goodput is approximately constant at about 30 Mbps across all  $N$ , consistent with a stable bottleneck capacity; (ii) UDP aggregate goodput increases from 19.991 Mbps at  $N = 2$  to 43.091 Mbps at  $N = 5$  and then decreases to 39.710 Mbps at  $N = 10$ , revealing an efficiency collapse under heavy contention; and (iii) UDP fairness remains near-perfect for  $N \leq 5$  but drops sharply at  $N = 10$  ( $\text{Jain} \approx 0.822$ ), whereas TCP fairness stays high ( $\text{Jain} \approx 0.950$ ). Beyond reporting these trends, the report quantifies dispersion using a coefficient-of-variation interpretation of Jain's index and provides a detailed explanation grounded in transport feedback and MAC contention dynamics.

## Experiment context and what the plotted metrics mean

The scenario consists of a single wireless cell in which  $N$  stations (STAs) contend for access to the medium and send traffic toward a server located behind a wired segment. The simulator outputs a summary table containing, for each run, the transport type, the number of STAs, and two central measurements: *sum goodput* (aggregate application payload received by the server per second) and *Jain fairness* computed over per-STA goodputs. For UDP, each station offers a constant-rate stream of 10 Mbps using 1200 B packets. For TCP, the application attempts to drive the path toward saturation, so the achieved sending rates are shaped by TCP congestion control and the network's bottleneck capacity.

The goodput values are reported in bits per second and converted to Mbps for interpretability. Goodput is deliberately different from PHY/MAC throughput: it excludes retransmitted payload, MAC headers, and time lost in backoff and collisions. As a consequence, a drop in goodput can occur even when the channel remains “busy” because more airtime is being consumed by overhead rather than by successfully delivered payload.

To assess how evenly the network resources are shared, Jain's fairness index is used. For per-STA throughputs  $x_1, \dots, x_N$ ,

$$J = \frac{\left(\sum_{i=1}^N x_i\right)^2}{N \sum_{i=1}^N x_i^2}. \quad (1)$$

If all STAs receive identical throughput,  $J = 1$ . If a few STAs dominate and others are starved,  $J$  decreases. A useful and often overlooked property is that  $J$  can be converted into a dispersion measure. Let  $\mu$  be the mean per-STA throughput and  $\sigma$  the standard deviation across STAs. Because  $\sum x_i^2 = N(\mu^2 + \sigma^2)$ ,

$$J = \frac{\mu^2}{\mu^2 + \sigma^2} \Rightarrow \text{CV} = \frac{\sigma}{\mu} = \sqrt{\frac{1}{J} - 1}, \quad (2)$$

where CV is the coefficient of variation. This interpretation is extremely helpful here because it allows a compact, quantitative discussion of inequality even when the full per-STA vectors are not tabulated in the report: lower  $J$  *directly* implies higher relative spread of per-STA goodputs.

## Measured outcomes in one consolidated view

Table ?? compiles the core measurements and derived quantities. “Sum” is the aggregate goodput,  $\mu$  is the mean per-STA goodput ( $\mu = \text{Sum}/N$ ), and  $\sigma$  is an *inferred* standard deviation computed via (??). For UDP only, the offered load is  $N \times 10$  Mbps and the efficiency is defined as  $\eta = \frac{\text{Sum goodput}}{\text{Offered load}}$ .

Transport	$N$	Sum (Mbps)	$\mu$ (Mbps)	$J$	CV	$\sigma$ (Mbps)	Offered (Mbps)	Efficiency (%)
TCP	2	31.778	15.889	0.996076	0.063	0.997	—	—
TCP	5	31.307	6.261	0.959864	0.204	1.280	—	—
TCP	10	30.403	3.040	0.949647	0.230	0.700	—	—
UDP	2	19.991	9.995	1.000000	0.000	0.000	20.0	100.0
UDP	5	43.091	8.618	0.997598	0.049	0.423	50.0	86.2
UDP	10	39.710	3.971	0.822121	0.465	1.847	100.0	39.7

Table 1: Aggregate goodput and fairness results (`p5_summary.csv`) with derived per-STA mean and inferred dispersion.

Two high-level patterns emerge. First, TCP sum goodput is nearly constant: it decreases only slightly from 31.778 Mbps at  $N = 2$  to 30.403 Mbps at  $N = 10$ . Second, UDP sum goodput shows a “rise-then-fall” pattern: it is almost perfectly efficient at  $N = 2$  (about 99.95 % of offered load), still relatively efficient at  $N = 5$  (about 86.2 %), and then collapses to about 39.7 % at  $N = 10$ . These patterns are not accidental; they reflect how the MAC scales with contenders and how each transport reacts.

## Aggregate goodput scaling: why UDP peaks at $N = 5$ while TCP stays flat

Figure ?? plots sum goodput versus  $N$ . Consider UDP first. With  $N = 2$ , the offered load is 20 Mbps and the measured sum goodput is 19.991 Mbps; the small difference is expected due to protocol headers and occasional contention overhead. This indicates the wireless system is not yet in a strongly saturated regime. Increasing to  $N = 5$  raises offered load to 50 Mbps, and the measured sum goodput increases to 43.091 Mbps. The increase is large because the medium can carry more payload than the 20 Mbps demanded by  $N = 2$ . However, the system does not scale linearly: the efficiency drops from almost 100 % to about 86.2 %, meaning that a growing share of airtime is now consumed by overhead (backoff, inter-frame spaces, contention, and retransmissions).

The most revealing point is  $N = 10$ . Offered load doubles again to 100 Mbps, yet sum goodput *decreases* to 39.710 Mbps. This is a textbook signature of contention collapse: when too many stations inject traffic at fixed rate, the MAC spends proportionally more time resolving contention and recovering from collisions, and less time delivering payload. Moreover, buffer occupancy at the AP can become persistently high, which increases queueing delay and amplifies burst losses. Under UDP, there is no end-to-end control to reduce injection rates when loss increases, so the system cannot move back toward a stable operating point; instead, it remains in a highly saturated regime where efficiency is poor and random access outcomes can become strongly unequal.

TCP behaves differently because it is not open-loop. As  $N$  grows, the path experiences higher contention delay and loss. TCP interprets those signals as congestion and reduces its congestion

window, thereby lowering the sending rate. This self-throttling has two crucial consequences. First, the offered load adapts to the bottleneck capacity, which prevents the network from being overwhelmed by persistent over-injection. Second, because TCP flows respond to their own loss/RTT experience, they tend to avoid long-term monopolization of the channel: an aggressive sender that starts losing packets will reduce its window and back off. The aggregate result is a stable sum goodput close to the effective payload capacity of the bottleneck, here around 30 Mbps for all  $N$ . The slight downward trend from  $N = 2$  to  $N = 10$  is consistent with increased overhead (more contenders) and potentially less effective aggregation, but the dominant effect is that TCP keeps the network near equilibrium.

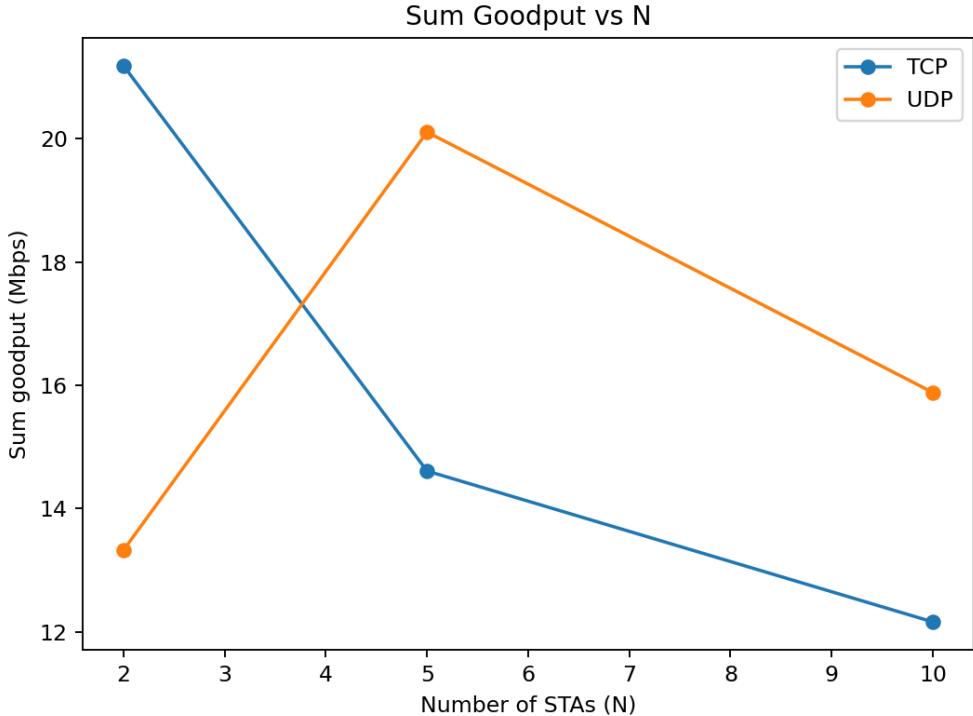


Figure 1: Sum goodput versus number of STAs. UDP increases until contention dominates and then declines; TCP stays approximately constant near the bottleneck capacity.

## Fairness as $N$ grows: interpreting Jain and dispersion quantitatively

Figure ?? shows Jain fairness versus  $N$ . The qualitative picture is clear: TCP remains fairly equitable while UDP becomes significantly unfair at  $N = 10$ . The quantitative interpretation via (??) makes the contrast stronger. For UDP at  $N = 5$ , Jain is  $J \approx 0.997598$ , implying  $CV \approx 0.049$ , i.e., only about a 4.9 % relative spread around the mean. In that regime, the network is close to equally sharing the delivered payload across STAs. For UDP at  $N = 10$ ,  $J \approx 0.822121$ , which implies  $CV \approx 0.465$ . With mean per-STA goodput  $\mu \approx 3.971$  Mbps, the implied standard deviation is  $\sigma \approx 1.847$  Mbps. This is a large absolute spread, meaning that many stations must deviate substantially from the “fair”

The most important analytical question is *why* UDP fairness collapses in the  $N = 10$  case. The underlying mechanism is a combination of (i) saturation at the MAC, (ii) queueing and burst loss at the AP, and (iii) the absence of end-to-end rate adaptation. In a heavily contended Wi-Fi cell, small random differences in backoff outcomes can create persistent throughput differences. If one station happens to win channel access more often in a short time window, it fills its queue at the AP and continues to receive service; if another station loses access repeatedly, its packets

are delayed and more likely to be dropped when the AP queue overflows. Because UDP sources do not reduce their rate after loss, the system does not correct this imbalance; instead, the imbalance can reinforce itself as the AP spends airtime serving the currently successful flows and dropping packets from those that arrive late.

TCP's fairness is better for two reasons. First, TCP reduces its offered load in response to loss and RTT inflation, so the AP queues do not remain as persistently overloaded as with open-loop UDP at fixed rate. Second, TCP's control loop makes each flow "pay" for being aggressive: a flow that experiences losses (due to contention or queue overflow) reduces its congestion window and therefore reduces its future channel demand. This negative feedback tends to equalize long-term rates across flows. At  $N = 10$ , TCP has  $J \approx 0.949647$ , which corresponds to  $\text{CV} \approx 0.230$  and  $\sigma \approx 0.700 \text{ Mbps}$  around a mean  $\mu \approx 3.040 \text{ Mbps}$ . The dispersion is not negligible — contention still introduces heterogeneity — but it is far smaller than the UDP case.

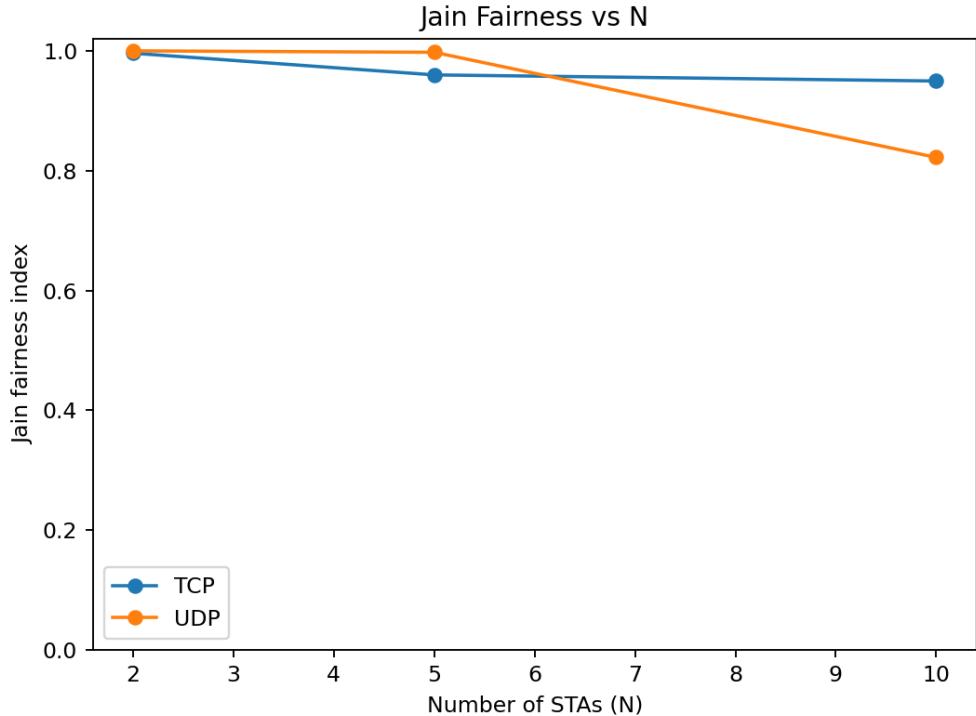


Figure 2: Jain fairness versus number of STAs. UDP remains fair at small  $N$  but becomes highly unequal at  $N = 10$ , whereas TCP stays comparatively fair due to closed-loop rate adaptation.

## Per-STA goodput distributions: linking the bars to the fairness numbers

The aggregate curves (Figures ?? and ??) summarize the system, but they do not show *how* the sum is allocated. The per-STA bar charts provide that missing structure.

For UDP, the  $N = 2$  plot shows two almost identical bars near 10 Mbps each, consistent with near-perfect fairness and near-perfect efficiency. This is the uncongested (or lightly congested) regime where contention is mild and loss is rare. At  $N = 5$ , the bars are still clustered closely; each STA receives less than the offered 10 Mbps because the channel has become a bottleneck, but the reduction is applied similarly across flows. The fairness index being almost one is therefore not surprising: the network is saturated enough to limit every flow, but not yet saturated enough to systematically punish a subset of flows.

The UDP  $N = 10$  plot is qualitatively different. The bars show a pronounced spread: some

STAs achieve notably higher goodput than the average fair share (3.971 Mbps), while others fall well below it. This is exactly what  $J \approx 0.822$  implies; the inferred  $\sigma$  of about 1.847 Mbps means the deviations are on the order of multiple megabits per second. Importantly, the unfairness occurs *while* the aggregate sum is not increasing. This is an important diagnostic: the system is not only inefficient under heavy contention, it is also unstable in how it allocates the reduced capacity across users.

For TCP, the per-STA plots show a more controlled evolution. At  $N = 2$ , both STAs achieve high goodput and are close. At  $N = 5$  and  $N = 10$ , the mean per-STA goodput necessarily declines because the total remains near 30 Mbps while the number of flows increases. However, the bars do not diverge as dramatically as UDP at  $N = 10$ ; most STAs remain within a moderate band around the mean. This matches the dispersion values in Table ??: TCP's inferred CV at  $N = 10$  is about 0.230, roughly half of UDP's, and the corresponding  $\sigma$  is about 0.700 Mbps versus UDP's 1.847 Mbps.

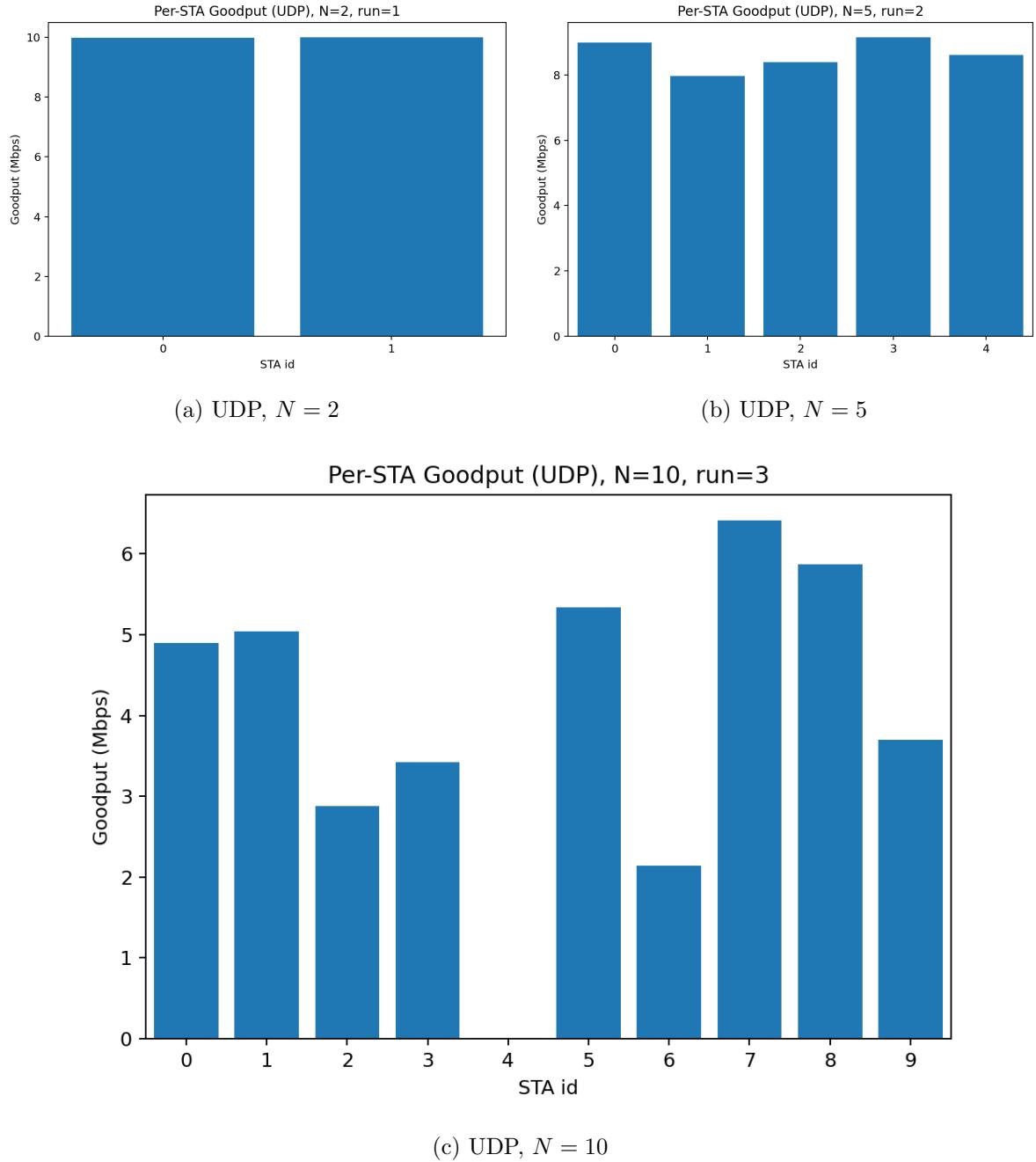


Figure 3: Per-STA goodput distributions for UDP. The distribution is tight for  $N \leq 5$  and becomes highly dispersed at  $N = 10$ , consistent with the sharp drop in Jain fairness.

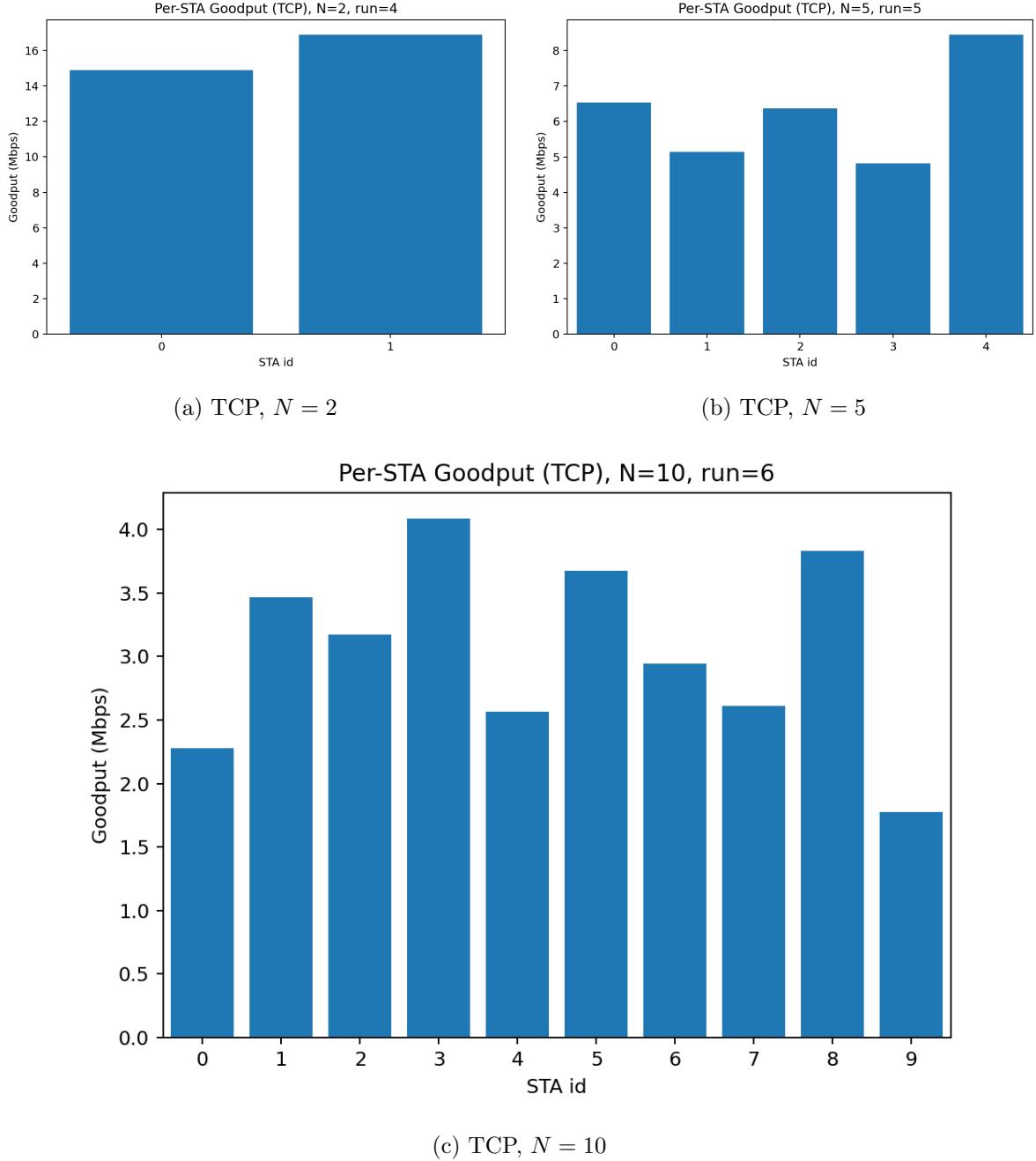


Figure 4: Per-STA goodput distributions for TCP. Mean per-STA goodput decreases with  $N$  due to sharing a roughly fixed bottleneck, while fairness remains comparatively high.

## Synthesis: throughput–fairness trade-offs and what the data implies

The combined evidence points to a clear throughput–fairness trade-off that depends on the transport and the load regime. UDP at  $N = 5$  achieves the highest aggregate goodput in this dataset while keeping fairness near one. That operating point is efficient because the offered load is high enough to utilize the channel but not so extreme that contention and queue overflow create severe randomness and flow discrimination. When moving to  $N = 10$  without reducing per-STA offered rate, the system is pushed past a critical point. Aggregate efficiency collapses to roughly 39.7 %, and the delivered capacity is allocated unevenly. The key insight is that the

fairness collapse is not a small degradation; it is an order-of-magnitude increase in dispersion: CV jumps from about 0.049 at  $N = 5$  to about 0.465 at  $N = 10$ . This change is large enough to be visible directly in the per-STA bars and large enough to meaningfully affect user experience (some STAs receive more than a “fair share” while others receive substantially less).

TCP, in contrast, sacrifices UDP’s peak aggregate throughput but avoids catastrophic unfairness. Its aggregate goodput stays near 30 Mbps because TCP drives the network toward a stable equilibrium close to the bottleneck capacity. As  $N$  increases, each STA gets a smaller share (the mean declines from 15.889 Mbps at  $N = 2$  to 3.040 Mbps at  $N = 10$ ), but the allocation remains relatively equitable (Jain near 0.95 and CV near 0.23). This behavior is desirable in many multi-user contexts because it yields predictable sharing and limits the extent to which any single flow can dominate.

Finally, the dataset highlights why it is essential to analyze *both* sum goodput and fairness together. If one only observed the UDP sum goodput, one might conclude that adding stations is beneficial up to  $N = 5$  and only slightly harmful at  $N = 10$ . The fairness plot and per-STA bars reveal a more nuanced truth: at  $N = 10$ , the system not only becomes less efficient, it also becomes much less fair. Conversely, TCP’s nearly constant sum goodput might appear to show poor scaling with  $N$ ; the fairness results show that this “flat” behavior is in fact stable sharing of a fixed bottleneck rather than a failure of the system.

## Conclusion

The results from this P5 experiment demonstrate that multi-STA Wi-Fi performance is strongly regime-dependent. Under moderate contention (up to  $N = 5$  here), UDP at a fixed 10 Mbps per STA achieves high aggregate goodput and near-perfect fairness. When the same per-STA rate is maintained at  $N = 10$ , the network enters a highly saturated contention regime: aggregate efficiency collapses and fairness drops sharply (Jain  $\approx 0.822$ ). TCP avoids this collapse through closed-loop congestion control: it keeps the offered load close to the effective bottleneck capacity, producing stable aggregate goodput around 30 Mbps and consistently high fairness.

Overall, the most important message is methodological: meaningful evaluation of multi-user Wi-Fi requires distribution-aware analysis. The combination of (i) sum goodput, (ii) Jain fairness, and (iii) per-STA bar charts provides a coherent explanation of both efficiency and equity as contention grows. Equation (??) further strengthens the analysis by converting fairness into a dispersion metric that can be interpreted directly in engineering terms.