

# Educational Notes (Tutorial): Interpreting PHY/MAC Sweeps in Wi-Fi using ns-3

## Channel Width, Transmit Power, Rate Control, Goodput, and RTT

This document is written as a tutorial-style companion to a Wi-Fi PHY/MAC sweep experiment in ns-3. The objective is not merely to present plots, but to teach how to *think* about them: how to design the sweep so it is scientifically meaningful, how to compute metrics correctly, and how to interpret results using the right mental models (airtime efficiency, SNR thresholds, retransmissions, and queueing). The experiment varies channel width (20 MHz, 40 MHz, 80 MHz), transmit power (e.g., 10 dBm and 20 dBm), and rate-control strategy (adaptive versus fixed MCS). The core metrics are application goodput and RTT measured by a lightweight probe flow. Whenever this document refers to figures, you can include the project plots (e.g., `fig1_bandwidth_scaling.png`, `fig2_power_sensitivity.png`, `fig3_latency_analysis.png`) in the same directory as this L<sup>A</sup>T<sub>E</sub>X file and compile normally.

A key philosophy underlies the analysis: the most valuable outcome of a sweep is not a single number, but a coherent explanation that connects configuration changes to observable behavior. In Wi-Fi, small differences in link margin and retransmission behavior can create large differences in delivered throughput and delay, especially when traffic load approaches saturation. Thus, the strongest educational value comes from explaining *why* curves take their shapes and *why* some operating points fail entirely.

### 1. Why these parameters matter (and what they teach)

Channel width is the most direct *capacity lever* in the PHY: wider channels offer more subcarriers and, in principle, higher PHY rates. In an idealized model, moving from 20 MHz to 80 MHz can increase peak PHY rate by roughly a factor of four. However, the goodput seen by applications rarely scales perfectly with bandwidth. The main reason is that Wi-Fi consumes airtime for work that does not carry user payload, such as contention backoff, interframe spacing, PHY preambles, MAC headers, acknowledgments, and occasional control traffic. Retransmissions also consume airtime without increasing delivered payload. Therefore, a bandwidth change typically increases capacity, but the fraction of airtime that becomes useful goodput depends on overhead and reliability. This leads to the fundamental teaching point: *bandwidth increases potential capacity, but delivered goodput depends on airtime efficiency*.

Transmit power influences performance mainly by changing the received SNR and thus packet error probability. Many modulation and coding schemes exhibit threshold-like behavior: below a certain SNR, frame errors become frequent and the link may collapse (effectively delivering near-zero payload), while above the threshold the same scheme becomes stable and highly efficient. This is why power sweeps at the cell edge often produce dramatic differences that are not linear. The practical lesson is that an “aggressive” PHY mode can be excellent when feasible but catastrophic when infeasible; a modest power increase can shift the operating point from infeasible to feasible, producing a large step in goodput.

Rate control strategy determines whether the system aims for robustness or peak throughput. A fixed MCS can be close to optimal in a stable channel when chosen appropriately, but it has no mechanism to recover when conditions are worse than expected. Adaptive rate control sacrifices some peak performance in exchange for graceful degradation, selecting safer rates when loss becomes likely. Consequently, it is normal—and educationally important—to observe cases where fixed MCS outperforms adaptive in “good” conditions while adaptive outperforms fixed in marginal or variable conditions. Rather than viewing this as inconsistency, treat it as evidence of a real trade-off.

## 2. Metrics: what you compute, and what each metric really means

Goodput in this tutorial refers to *application-delivered payload rate*. If  $R_x$  is the number of payload bytes received by the sink during the measurement window and  $T_{\text{eff}}$  is the effective interval (simulation time minus warm-up and any final guard time), then

$$G = \frac{8 R_x}{T_{\text{eff}}} \times 10^{-6} \quad [\text{Mbps}]. \quad (1)$$

Goodput is intentionally strict: it discounts all bytes that did not arrive at the application, and it does not “credit” retransmissions or headers. This is why goodput is usually the best metric for comparing configurations in a user-centric way. It automatically includes the hidden costs of contention, overhead, and loss.

RTT in this experiment is measured by a probe flow that runs while the main traffic (often a high-rate UDP stream) is active. A common mistake is to interpret RTT as propagation delay. Under heavy load, propagation delay is negligible compared with queueing delay. RTT becomes approximately

$$\text{RTT} \approx \underbrace{d_{\text{prop}}}_{\text{tiny}} + \underbrace{d_{\text{MAC}}}_{\text{access/backoff}} + \underbrace{d_{\text{retx}}}_{\text{retries}} + \underbrace{d_{\text{queue}}}_{\text{dominant under load}}. \quad (2)$$

When offered load approaches or exceeds the sustainable service rate, queues build and  $d_{\text{queue}}$  dominates, making RTT a strong indicator of congestion and backlog. In educational terms, RTT is not a pure PHY metric; it is a *system-level* metric shaped by both PHY feasibility and MAC queueing.

## 3. How to interpret bandwidth scaling (and why it is not perfectly linear)

Figure 1 illustrates a typical bandwidth scaling outcome: goodput increases as the channel width grows from 20 MHz to 40 MHz to 80 MHz. This is expected, but the educational goal is to understand the *shape* of the scaling. If the system were payload-dominated and overhead negligible, the goodput ratio from 20 MHz to 80 MHz could approach  $4\times$ . In practice, the ratio is smaller because overhead consumes a portion of airtime that does not shrink when bandwidth increases. An intuitive way to reason about this is to separate airtime into useful payload time and non-payload time. Increasing bandwidth increases the payload-carrying capacity per unit time, but it does not remove contention overhead, ACK overhead, or rate-control probing. Therefore, increasing bandwidth yields diminishing returns when overhead is a significant fraction of airtime.

Distance adds another layer. At short range, link margin is high and retransmissions are relatively rare, so wider channels translate more directly into delivered payload. At longer range (cell-edge conditions), link margin is lower and the system may experience higher retry rates or may select more conservative modes, both of which reduce the fraction of airtime that becomes payload delivery. Thus, the same bandwidth increase can yield a smaller goodput gain at the edge than at short distance. When you see weaker scaling at longer distance, the correct explanation is

not “bandwidth stopped working,” but rather that the system spent more airtime “paying” for reliability and MAC access.

A subtle but important educational observation is that adaptive rate control can underperform a carefully chosen fixed high MCS in some edge cases. This does not mean adaptation is wrong; it means the adaptive algorithm may be conservative (or be tuned for a different mode set) and therefore avoid the highest-throughput modes even when they are feasible most of the time. In a stable, borderline-but-feasible channel, a fixed configuration can sometimes extract slightly higher average goodput, while adaptation provides better robustness across variability.

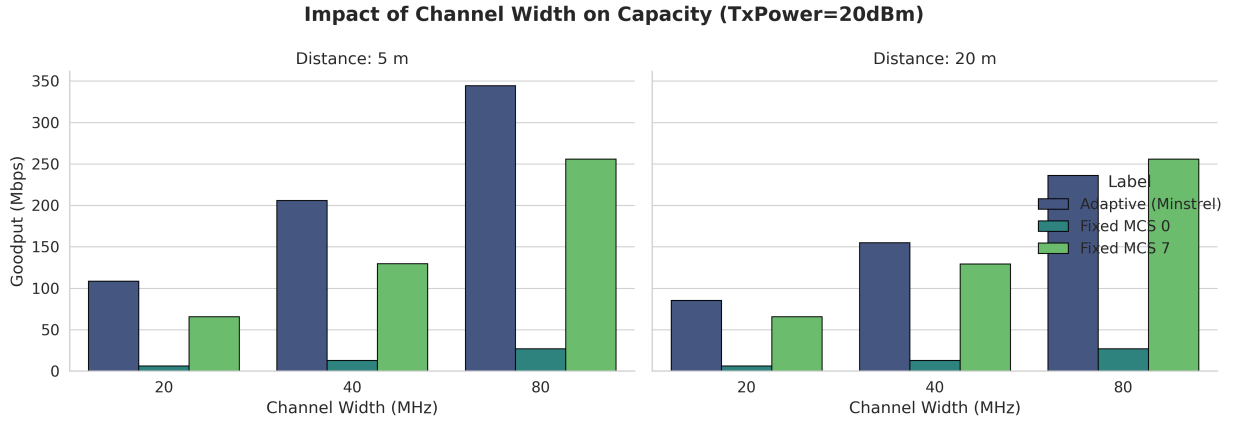


Figure 1: Example plot from the project: application goodput versus channel width for two distances under fixed transmit power. The pedagogical objective is to explain why scaling is strong but not perfectly linear, and why edge conditions often scale less efficiently.

#### 4. Power sensitivity at the edge: threshold behavior and “cliff” effects

Figure 2 captures one of the most educational phenomena in wireless networks: *threshold behavior*. At the cell edge, increasing transmit power by 10 dB can produce a disproportionately large improvement in delivered goodput, especially for high MCS. The reason is that packet error probability can change sharply around the decoding threshold for a given MCS. If the link operates below the threshold, frames fail frequently; retransmissions consume airtime but do not deliver payload, and goodput can collapse to near zero. Once the link crosses into the feasible region, the same MCS can suddenly become stable, enabling high throughput.

This is also where the interaction with bandwidth becomes instructive. Wider bandwidth increases potential capacity, but it can also make edge operation more sensitive because noise power is integrated over a wider band. In simplified terms, if the transmitter power is fixed, a wider band can reduce power spectral density and shift feasibility boundaries for high MCS. The takeaway is not that wide bandwidth is “bad,” but that it is a *high-capacity configuration that demands sufficient margin*. If margin is insufficient, a robust mode or an adaptive controller may be the only way to maintain delivery.

Adaptive control typically does not collapse because it can retreat to more robust modes as conditions worsen. This yields a smooth performance curve as power decreases: the system sacrifices spectral efficiency to remain operational. In contrast, fixed high MCS can be excellent above threshold and effectively dead below threshold. When analyzing such results, it is important to describe the feasibility boundary explicitly. A point with goodput  $\approx 0$  should be treated as a *failed operating point*, not merely a low-performing one. This distinction becomes even more important for latency interpretation, because RTT is only meaningful when packets are actually delivered.

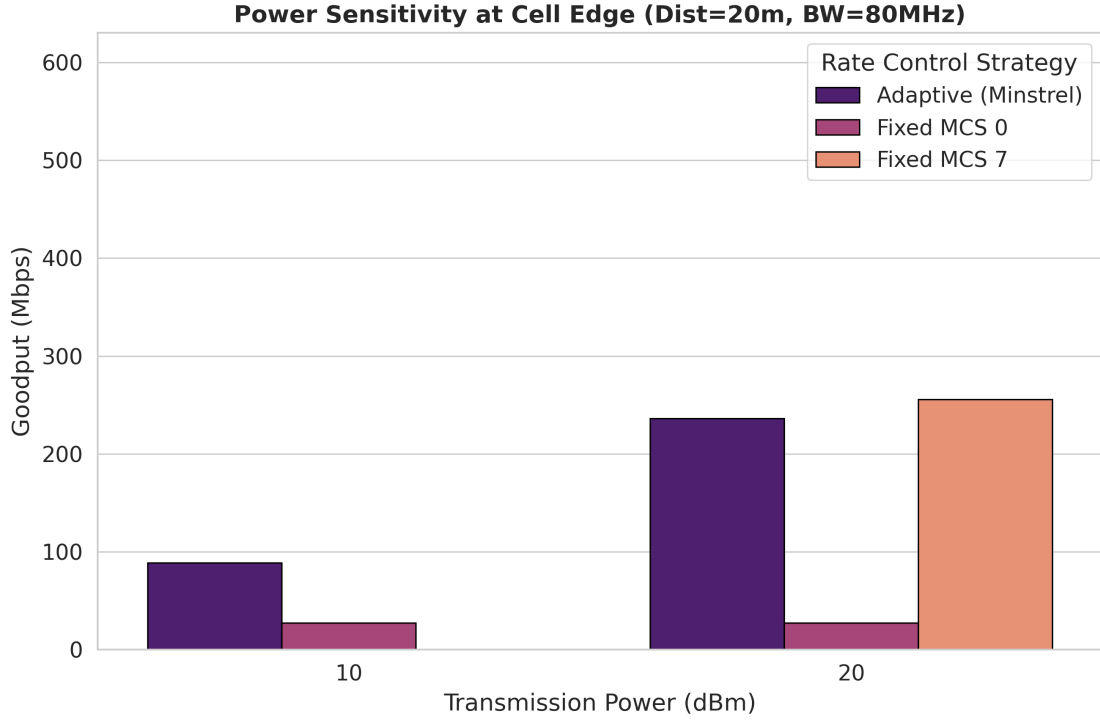


Figure 2: Example plot from the project: goodput at the cell edge for two transmit powers under wide bandwidth. The key educational message is threshold behavior: high MCS may collapse below an SNR boundary and recover sharply above it.

## 5. RTT under heavy load: queueing as the dominant mechanism

Figure 3 presents mean RTT at the cell edge as a function of channel width. The first teaching point is that RTT values can be large (tens to hundreds of milliseconds) without implying a bug. Under heavy offered load, the dominant factor is queueing: when the sender injects packets faster than the link can service them, backlog grows and packets wait. RTT then becomes a measure of backlog and MAC access, not a measure of physical distance. This is analogous to “bufferbloat” in real networks, where large buffers can produce large delays during congestion.

The second teaching point is that throughput and RTT are tightly coupled under saturation. If a configuration provides a higher service rate (higher goodput), it tends to reduce backlog and thus reduce RTT. Therefore, improvements in RTT with increased bandwidth can be explained by increased service capacity. This explains why a robust but inefficient mode (like MCS0) often exhibits the largest RTT values: it delivers low goodput relative to the offered load, guaranteeing persistent backlog. As bandwidth increases, MCS0 still remains inefficient, but its service rate increases enough to reduce backlog somewhat, lowering RTT.

For adaptive and high MCS configurations, RTT can be much lower when the link is feasible. However, in marginal regimes where a high MCS becomes infeasible (for example, low transmit power combined with large bandwidth at long distance), RTT may become undefined because probe packets are not delivered. In datasets, this often appears as missing RTT samples or sentinel values (e.g.,  $-1$ ). The correct educational treatment is to label such points as “no samples / link infeasible” rather than averaging them into latency curves. From an engineering perspective, “very high RTT” and “no delivery” are different failure modes: the former indicates congestion with delivery, the latter indicates feasibility failure.

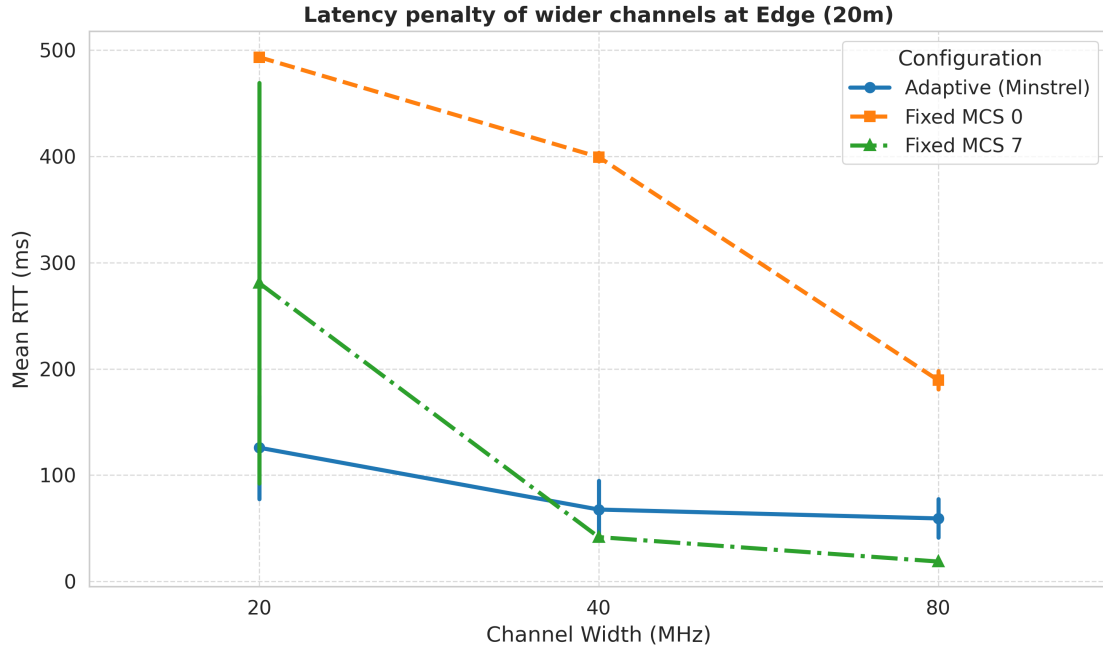


Figure 3: Example plot from the project: mean RTT versus channel width at the cell edge. Under heavy offered load, RTT is dominated by queueing and tracks service capacity; infeasible points should be treated as missing, not as large delay.

## 6. Turning plots into strong technical explanations (a reusable method)

A useful educational habit is to convert every plot into a mechanism-based narrative. A simple method is to write an explanation that (i) states what is held constant and what varies, (ii) describes the visible trend, (iii) identifies the mechanism, and (iv) checks internal consistency by connecting multiple metrics. For example, when bandwidth increases, goodput usually increases; if the offered load is high, RTT should usually decrease because backlog shrinks. If you observe higher goodput but also higher RTT, that suggests an additional mechanism (e.g., aggressive retries, burstiness, buffer sizing) is counteracting the expected queueing improvement. This cross-metric consistency check is one of the best ways to avoid superficial interpretations.

Another reusable principle is to report feasibility explicitly. When a point shows near-zero goodput for a high MCS at low power, interpret it as operating below the SNR threshold. That is not merely a “low throughput” case; it is a regime where the configuration is not viable. This framing makes the analysis more rigorous and aligns with how engineers discuss wireless systems: performance cliffs matter, and adaptation exists largely to avoid them.

## 7. Common interpretation pitfalls (and how to avoid them)

One frequent mistake is to claim a numeric realism percentage (e.g., “90% accurate”) without calibration against real measurements. Without a measured ground truth and an error metric, a percent claim is not defensible. A much stronger and more correct statement is that the simulation reproduces expected qualitative behavior and provides meaningful *relative comparisons* between configurations. If one wants a numeric accuracy claim, one must run a real experiment under comparable conditions and compute an error measure such as mean absolute percentage error.

A second mistake is treating RTT as propagation delay. Under heavy offered load, RTT is dominated by queueing and MAC access. A third mistake is hiding missing data: if RTT samples are

missing because the link is infeasible, that should be described explicitly. These practices improve scientific credibility and prevent misleading conclusions.

## 8. Optional extension for deeper learning

If the offered load is very high (hundreds of Mbps), the experiment emphasizes saturation and queueing, which is excellent for studying capacity–latency coupling. A valuable extension is to repeat the sweep at a moderate offered load (e.g., 50 Mbps to 100 Mbps). Under moderate load, queues remain small and RTT approaches a baseline dominated by MAC access rather than backlog. Comparing the low-load and high-load regimes teaches a fundamental networking lesson: *latency is load-dependent*, and capacity improvements matter most when the system is near saturation.

## 9. Summary of the key lessons

This sweep teaches three core engineering lessons that generalize beyond ns-3. First, wider channels provide large throughput gains, but goodput scaling is limited by overhead and reliability, especially at the edge. Second, transmit power strongly affects feasibility and can create cliff effects for high MCS; small power changes can produce large throughput differences near thresholds. Third, under heavy offered load, RTT is primarily a queueing metric, tightly coupled to service capacity; higher delivered throughput tends to reduce backlog and thus reduce RTT, while infeasible operating points should be treated as failures rather than as high-delay cases.

If you include your project plots in this document, it becomes a complete educational handout: it not only shows *what* happened in the sweep, but also explains the mechanisms and gives readers a repeatable reasoning method for future wireless experiments.