WIRELESS LAN PROFESSIONALS

# What a Rate Limit Costs Your Wireless LAN

Keith Parsons · October 19, 2026

# What a Rate Limit Costs Your Wireless LAN

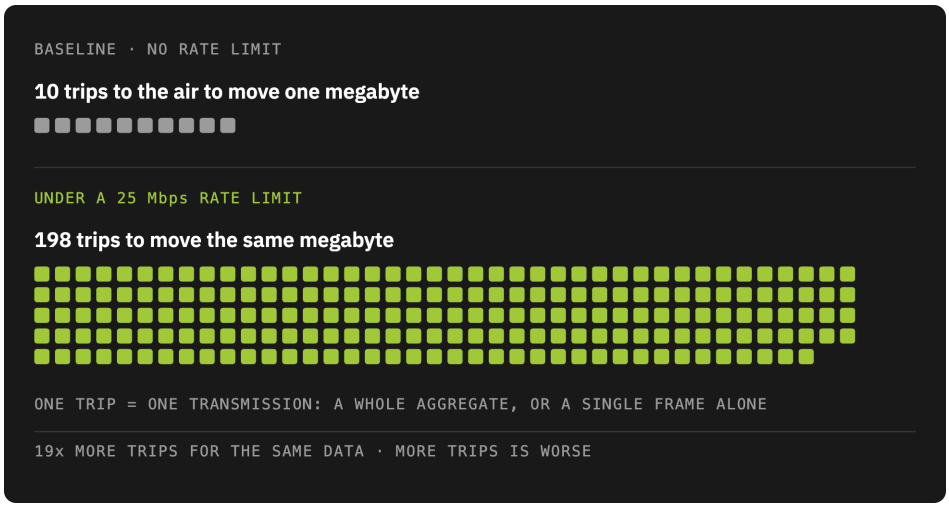
**Keith R. Parsons, CWNE #3** Wireless LAN Professionals, Inc. Measured September 2026. Version 6.1, corrected October 2026: the three corrections are listed at the end.

## Summary

**Stop rate limiting your Wi-Fi.** It is the most common lever in guest and public networks and it is the wrong one, because it rations bits by spending transmit opportunities, and transmit opportunities come out of airtime you do not have enough of.

On the access point I tested, a **100 Mbps** limit on a radio that had been delivering 620 Mbps made it take **11 times as many trips to the air** to move the same megabyte. Tightening from there to 2 Mbps multiplied that by **another ten**. You do not get to tighten your way into trouble. You are in it at the first, most generous setting.

The mechanism is visible in one number. With no limit, **99.7% of the access point's downlink frames went out inside an aggregate**, sharing one trip between many frames. At a 2 Mbps limit, **3% did.** The other 97% each took the medium alone: one contention win, one preamble, one acknowledgment, for one frame.




*Moving one megabyte took 10 trips to the air with no limit and 198 under a 25 Mbps limit.*

No standard dashboard field reports any of this. Worse, every indicator an engineer would normally check says the network got *healthier*. Retry rate falls. Signal, channel and client count do not move. The one number that got dramatically worse is on no screen.

### What this establishes, and what it does not

**Establishes.** That a downlink rate limit collapses frame aggregation on this access point until nearly every frame is sent alone, on both 5 GHz and 6 GHz, at six repetitions per condition. That the effect is visible between two clients inside a single capture, where nothing about the radio environment can differ between them. That control and management overhead airtime per megabit rises with it, measured directly. That the ordinary checks do not reveal it, and the counter that would have is unreliable.

**Does not establish. Total airtime per megabyte**, which needs the PHY data rate of every transmission and I did not record it. What this costs the users of a busy cell. Whether it happens on other vendors' equipment, or on other models from this one. Why the retry rate falls.

**One access point, deliberately.** One radio, one client, one position, one channel per band, six repetitions per condition, so every number here can be set against every other. **Section 7 asks other people to run it on other hardware, which is the only way to generalize any of it.**

## Contents


1. What question this answers
2. Why a rate limit costs anything at all
3. Test setup
4. Results
5. Limitations
6. What to do about it
7. Run it on your own gear
8. Exactly what was run
9. A closing note on WAN limits

# 1. What question this answers

A lot of people want rate limiting, and I understand why. You turn it on to stop one heavy user ruining the network for everybody else, which is a reasonable thing to want. Plenty of networks are running it today because somebody turned it on once and nobody ever asked what it cost. The limit is set in Mbps, checked in Mbps and reported in Mbps, so Mbps is the unit everyone ends up reasoning in.

That is the wrong unit, and using it hides the entire cost.

A wireless LAN does not spend megabits. It spends **airtime**, and airtime is consumed per transmission, not per byte. So the question is not how fast a client is permitted to go. It is how much data the access point (AP) carries each time it takes the medium.

I set out to answer three things:

- How much does a rate limit change the amount of data moved per transmission?
- Does the change show up in anything an engineer would routinely check?
- Does it get better on 6 GHz?

The answers are: by roughly an order of magnitude, no, and not enough to matter.

## The metric

A **trip to the air** is one transmit opportunity, a **TxOP**: one contention win, one preamble and PHY header, one payload, one acknowledgment. **Trips per megabyte** is the metric. The whole of this paper is an argument about TxOPs, so it is worth fixing the word early: a TxOP is the unit a radio actually spends, and it is spent whole whether it carries one frame or forty.

**Counted one way throughout:** every downlink transmission the AP made to a client that carried traffic, divided by the megabytes that client received. A transmission is either an A-MPDU carrying several frames, or **a single frame sent on its own**, and both cost the same fixed overhead.

**Frames per transmission** is the same captures counted the other way round: downlink frames divided by transmissions. With no limit it is about 17. Its floor is 1.

**What this metric is not.** It is not a claim that less data arrives. Put a 100 MB file through a rate limited link and all 100 MB arrives. It is also **not a measurement of total airtime. The penalty here is priced in TxOPs, not in clock time.** A TxOP has a fixed cost and a variable one, and the variable part depends on the PHY data rate the AP chose. **I did not record PHY data rate**, so this paper can tell you how many more

times the AP took the medium to move a fixed amount of data, and not how many more microseconds it held it in total. Section 4.3 measures the one airtime component that can be priced without a rate.

## 2. Why a rate limit costs anything at all

Every time an AP sends a downlink transmission, it pays the same fixed cost. It contends for and wins the medium, sends a preamble and a PHY header, delivers its payload, and collects an acknowledgment. **That cost does not shrink when the payload does.**

Frame aggregation exists to spread that cost over as much data as possible. The AP gathers whatever is waiting in the driver queue and sends it under one preamble. A full queue makes a big aggregate and an efficient trip. **An empty queue makes a trip carrying one frame, at the same fixed price as a trip carrying forty.**

My working model: **something upstream of the driver queue paces the frames going into it.** A queue that never fills has nothing to batch. The same bytes then leave in many small transmissions instead of a few large ones.

Note what the model does not require. It does not require the shaper to live inside the access point. On this hardware the gateway and the radio share an enclosure, and section 4.7 shows the limit is enforced on the routed path rather than by the radio. **Pacing that arrives from upstream empties the driver queue just as effectively as pacing applied inside the AP.**

That is a model, not a result. Section 4 is the result and does not depend on it.

# 3. Test setup

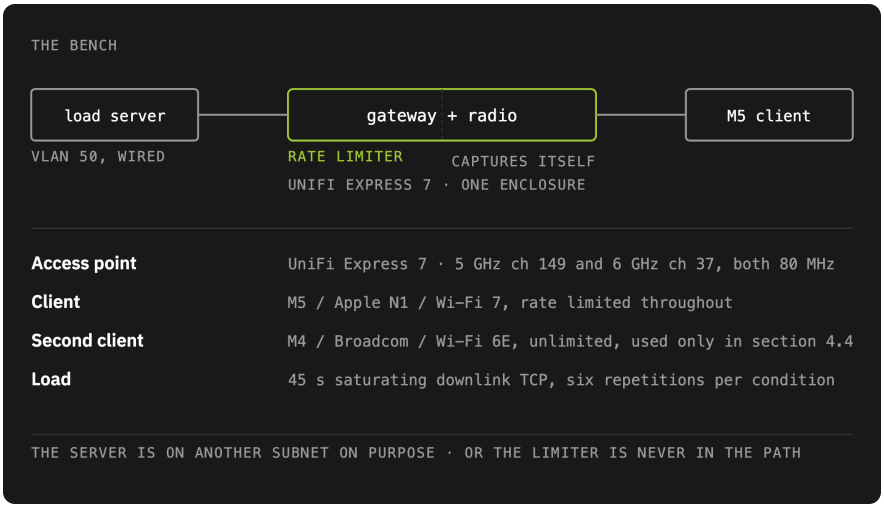
## 3.1 Equipment

| Role | Device | Detail |
|---|---|---|
| Access point | Ubiquiti UniFi Express 7 | Gateway and radio in one enclosure. 5 GHz ch 149 and 6 GHz ch 37, both 80 MHz. 2 spatial streams. |
| Client 1 | Apple MacBook Air, `Mac17,3`, Apple M5 | macOS 26.6.2 (25G83). **Apple N1 Wi-Fi silicon.** 802.11be, Wi-Fi 7. The rate limited client throughout. |
| Client 2 | Apple MacBook Air, `Mac16,12`, Apple M4 | macOS 26.5.2 (25F84). **Broadcom Wi-Fi silicon** ( `0x14E4:0x4388` ). 802.11ax, Wi-Fi 6E. Used only as the second, unlimited client in 4.4. |
| Load server | Wired host, `10.0.50.87`, VLAN 50 | `iperf3 -s` . |

## 3.2 Topology, and where the limiter actually sits

The load server sits on a different subnet from the clients, deliberately.

**On this platform the rate limit is enforced at the gateway, on the routed path, even though the setting is written onto the wireless client's "user group" and reads as though it travels with the client.** Section 4.7 has the measurement. It matters here because the test cannot be built correctly without it: **generate load from a device on the same VLAN and the traffic is switched rather than routed, the limiter is never in the path, and you will measure a rate limited condition that is not rate limited.**



*The bench. The load server sits on a different subnet on purpose, so the rate limiter is genuinely in the path.*

## 3.3 Method

For each condition:

1. Apply the downlink limit to the test client's user group, the platform's per-client limit setting, and wait about 20 seconds for it to take.
2. Start a capture **on the AP itself**, at full channel width, with transmit monitoring enabled.
3. Run 45 seconds of saturating downlink TCP from the wired server. Three idle seconds either side.
4. Stop the capture and retrieve it over SFTP afterwards, never streamed during the run.
5. Remove the limit and repeat for the remaining repetitions.
6. Move to the next condition.

Conditions ran from loosest to tightest, all six repetitions of one completed before the next began. That order was not randomized, which is a real weakness and section 5 says so.

A run is void if the client changed AP or radio, if throughput missed the limit, or if the capture holds no frames from the AP's own BSSID. **Each of those produces a result that looks ordinary**, so the harness tests for them on every run rather than leaving it to whoever reads the output.

## 3.4 Why the capture is taken at the AP

An AP capturing its own transmissions knows what it sent, how often, and how it batched it. Every measurement here comes from one.

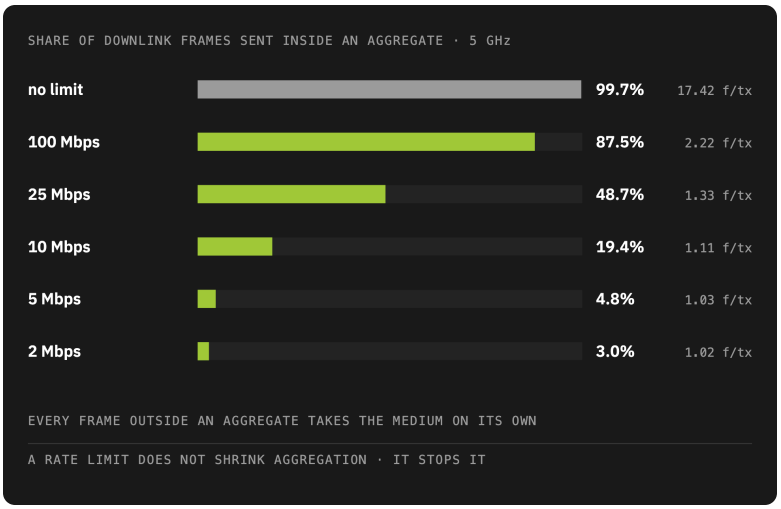
**I also captured over the air, and it under-reported the effect.** A monitor-mode adapter beside the clients saw a much smaller collapse than the AP did, and how much smaller depended strongly on how far the sniffer sat from the transmitter. **That is a real finding and it is not this paper's**, because settling it needs a distance sweep, which is a separate investigation. For now: **capture at the AP. If you can only capture over the air, validate your setup against a known result before you trust a negative from it.**

# 4. Results

All six repetitions per condition unless noted, one access point, one client, both bands measured inside one night.

## 4.1 Aggregation does not shrink. It stops.

With no limit, almost every downlink frame traveled inside an aggregate, sharing one trip to the air with sixteen others. As the limit tightened, that stopped happening, and by 2 Mbps almost every frame was making the trip alone.



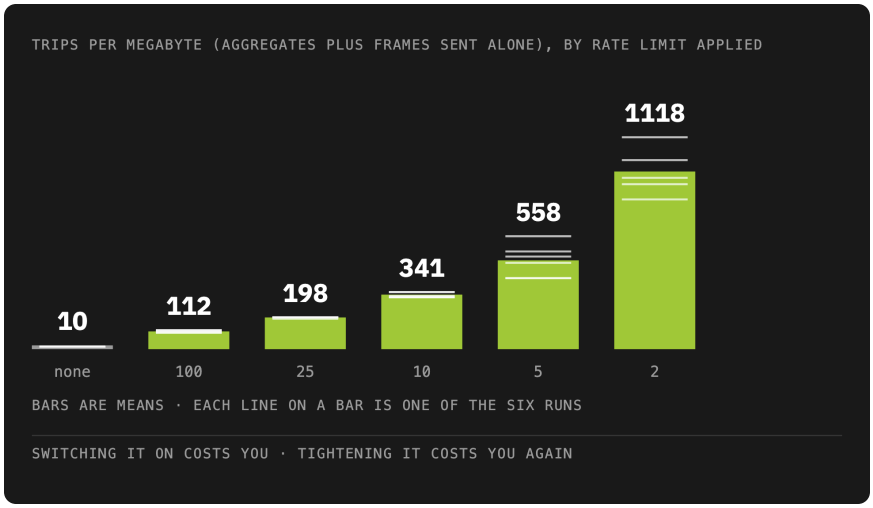

*The share of downlink frames sent inside an aggregate falls from 99.7% with no limit to 3.0% at a 2 Mbps limit.*

**One frame per transmission is the floor, and it is a physical one.** The access point cannot take the medium less than once to send a frame. At a 2 Mbps limit it is doing exactly that, over and over: contend, preamble, one frame, acknowledgment, repeat.

A limit set in megabits per second did not make the radio send smaller aggregates. It made the radio stop aggregating.

## 4.2 The cost per megabyte, and where it arrives

Count every transmission and divide by the data delivered.




*Trips per megabyte rise from 10 with no limit to 1118 at a 2 Mbps limit.*

**Most engineers would call 100 Mbps a generous setting.** It was applied to a radio measured at 620 Mbps. With no limit the access point needed about **ten trips** to move a megabyte. Under that generous setting it needed **a hundred and twelve**.

Tightening recovers none of it. At 2 Mbps the same megabyte took **more than eleven hundred trips**.

***Switching the feature on costs you, and tightening it costs you again. There is no setting at which this becomes cheap.***

## 4.3 What that costs in airtime, and what it does not

Everything above counts transmissions. This section prices as much of them in real time as the captures allow.

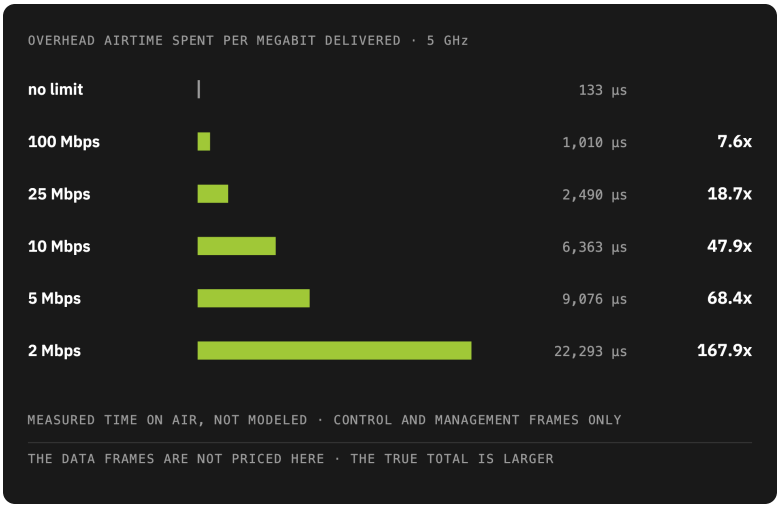
On 5 GHz, control and management frames go out at the basic rates rather than a negotiated data rate, so **their** time on air can be computed directly without knowing what PHY rate anything else used.




*With no limit the medium spent 82 ms of every second on overhead while delivering 620 Mbps. Under a 100 Mbps limit it spent 96 ms while delivering 95 Mbps.*

**The limit bought more overhead and less data at the same time.** A generous setting made the medium spend longer on traffic nobody asked for, while carrying a fraction of what it had been carrying.

Per megabit delivered, which is the comparison that survives the different data volumes:



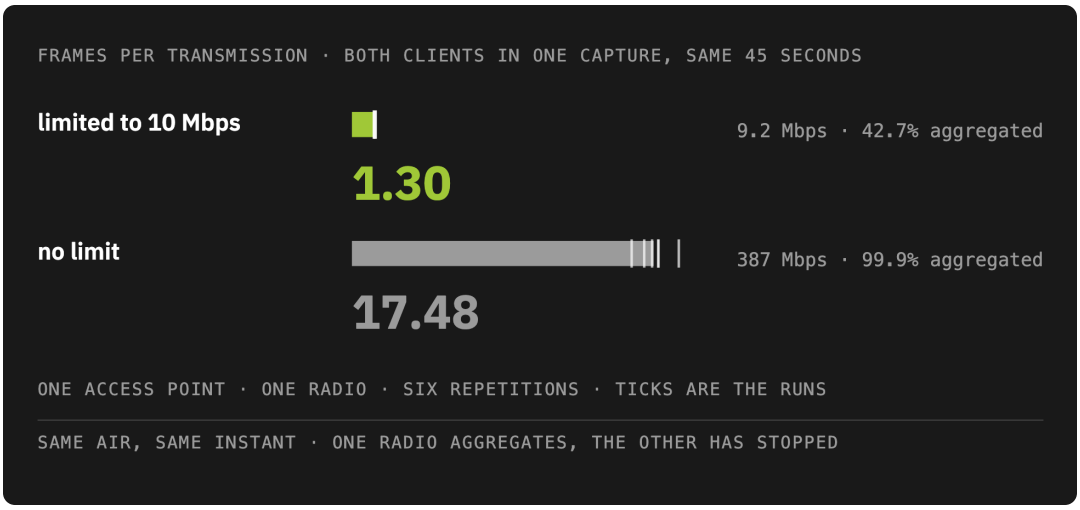

*Overhead airtime per megabit rises from 133 microseconds with no limit to 22,293 at a 2 Mbps limit, a factor of 168.*

**This is real airtime, and it is not the whole bill.** It prices the control and management frames, the ones a capture can time without knowing any data rate. Pricing the data frames needs the rate each of them went out at, which I did not record, so the true total is larger than anything in this section and this paper does not establish it.

## 4.4 Two clients, one radio, one instant

Everything above compares runs taken at different times. This removes the environmental half of that objection.

Both clients ran against the same access point at the same moment, 45 seconds, six times. The M5 was limited to 10 Mbps, the M4 was not. Both appear in **the same capture file**, so time of day, neighboring traffic, retry rate, position and channel conditions are identical by construction.




*In one capture the unlimited client had 99.9% of its frames aggregated and the limited client 42.7%.*

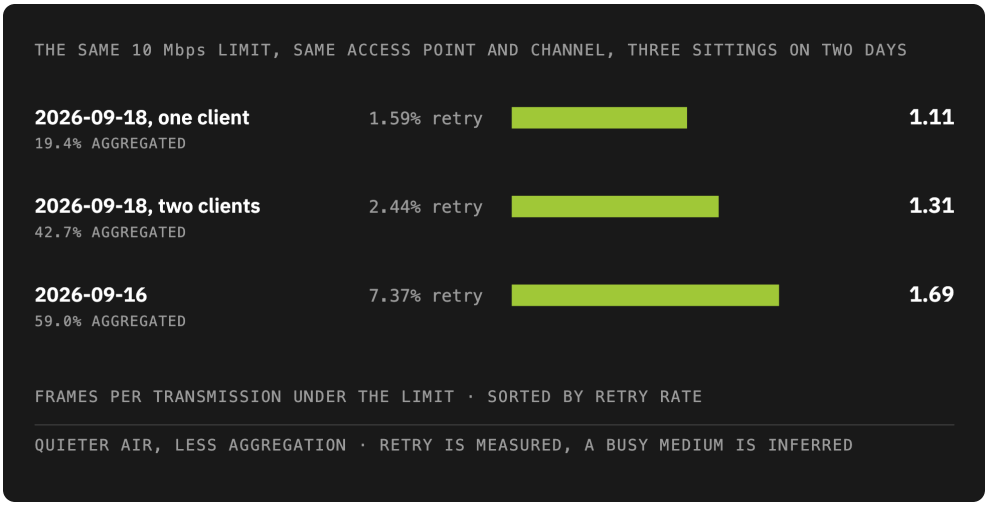
No cross-run comparison can claim a shared environment and this one can: nothing about the air explains a gap measured inside one capture. **What it does not remove is the clients.** The limited one is Apple N1 at Wi-Fi 7, the unlimited one Broadcom at Wi-Fi 6E, on different macOS builds. Two unrelated radios will not batch identically even with no limit anywhere, so **the rate limit is not the only difference between those rows, and an identical pair would be a better test.** I do not own one.

What holds it up is that the same M5, on this radio, measures **about 17 frames per transmission unlimited and just over one at a 10 Mbps limit** in section 4.1. It is not sitting near one because it is an M5.

**The cell average across both clients was about 11 frames per transmission**, because the unlimited client dominates the total. An engineer reading a per-radio summary would see a healthy number and conclude the limit cost nothing. How badly pooling misleads depends on how many clients are limited and how hard the others are working, which I did not test across a range. **Report by receiver address, not by an average mixed across limited and unlimited clients**, because a pooled figure cannot tell you which situation you are in.

## 4.5 The same limit costs different amounts in different sittings

The same 10 Mbps limit, same access point, same client, same channel, measured in three separate sittings on two days, did not give the same answer. The 2026-09-16 sitting loaded the client with a 50 MB file download rather than iperf3.




*Across three sittings on the same hardware, a lower retry rate goes with less aggregation under the same 10 Mbps limit.*

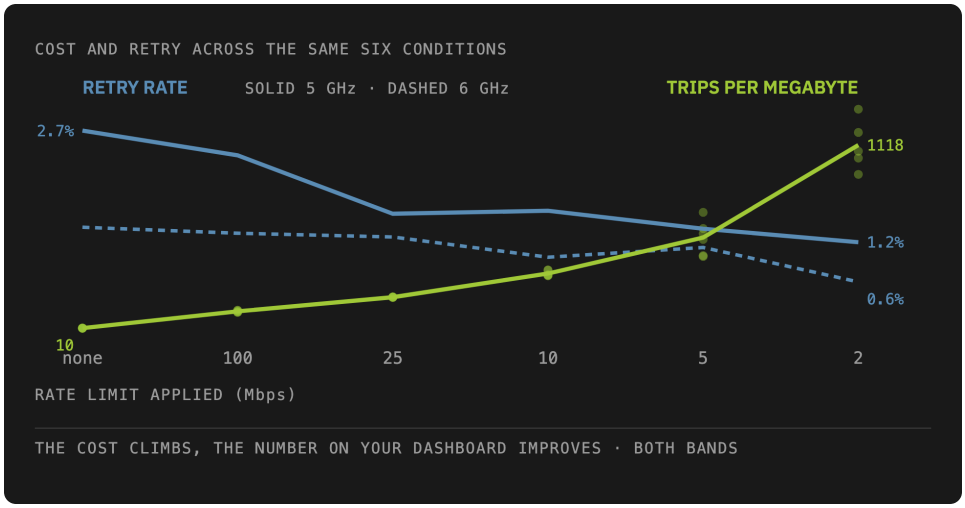
**Retry rate is what I measured. A busier medium is my reading of it**, and I did not instrument channel utilization or interference separately to check. The likely mechanism is that contention delays the next transmit opportunity, so the queue has longer to fill and the access point finds more to batch when it finally gets the medium. Queue depth was not measured either.

**One paired comparison beats a correlation across sittings.** On 2026-09-18 the same client under the same limit measured **1.1 alone and 1.3 with a second client loading the radio**, about four hours apart on the same day. Contention moved it the way that mechanism predicts.

**What follows for anyone reproducing this: report your retry rate.** Without it these results cannot be lined up against each other, and any two of my own sittings have to differ by more than half again before the difference means anything. It also suggests a rate limit costs most on a clean, well-designed network, where the queue has least chance to fill between transmissions. **That last part is the mechanism talking, not a measurement.**

## 4.6 Nothing looks worse, and the retry rate improves

As the limit tightened and trips per megabyte climbed, **the retry rate fell**, on both bands. Roughly halved, on a quiet bench where it was low to begin with.




*Cost per megabyte and retry rate across the same six conditions, moving in opposite directions.*

### Why it falls, which I do not know

A tempting explanation is that the AP sends at lower data rates under a limit and that lower rates are individually more reliable. **This work cannot support that.** I never measured the data rate each transmission went out at, and the captures cannot supply it: the per-frame radio information they carry records no data rate of any kind for the limited client's downlink. A capture that does not hold the number cannot settle the question in either direction.

The arithmetic says this much. Under a tight limit the AP makes far fewer transmissions overall. But at 100 Mbps it makes **more** transmissions than the unlimited case and the retry rate still falls, so "fewer attempts" does not explain the whole curve.

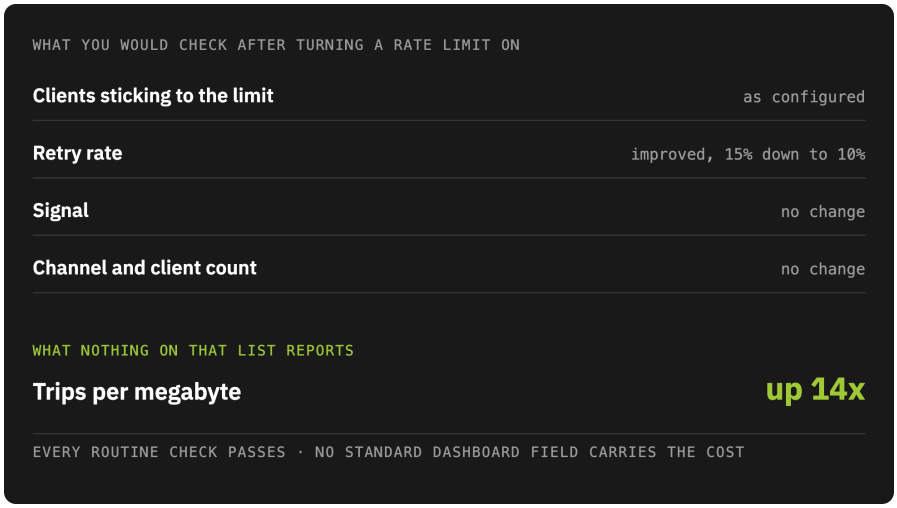
### Where this disagrees with other people's field results

**Jono Thompson's measurements show retries rising under a limit, which is the reverse of mine.** In his WiCo Fest 2026 deck he reports MAC retries at **3.84% under a 2 Mbps limit against 2.52% unlimited**, and TCP retries at **9.27% against 0.18%**. Both go up. **Peter Mackenzie reports the same direction from his own field work.**

**I cannot reconcile that from my data.** Every sweep I hold shows the change going the other way. Section 4.5 establishes that the absolute retry level on this bench moves from one sitting to the next, which is a reason to distrust the magnitude of any single measurement including mine, **but a moving baseline is not evidence that the limit-induced change can flip sign.** It might. My data does not show it.

So this is an open disagreement rather than a resolved one. For anyone reproducing: **report retry for every condition, limited and unlimited, and report the change rather than the level.** Nobody can line these results up without it.

### What an engineer would actually check




*Every routine check after enabling a rate limit passes, and the one that moves improves. Trips per megabyte is on no dashboard.*

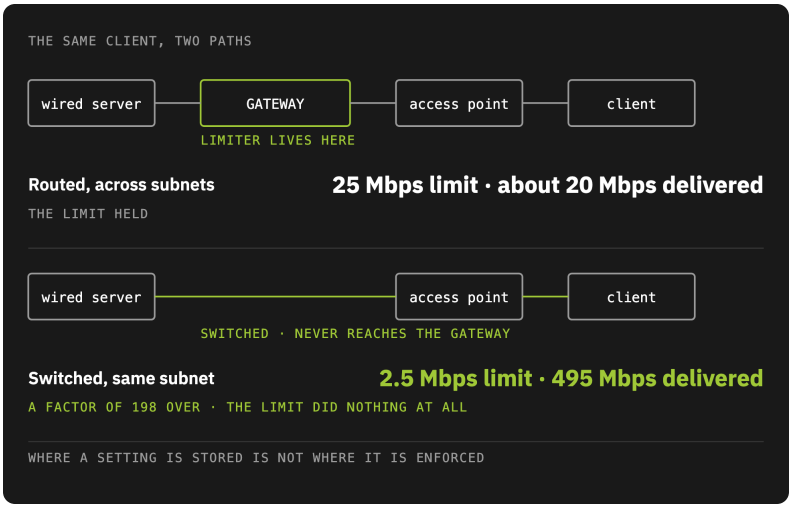
Nothing on that list looks worse. The one number that moves, improves.

You might reach for channel utilization, the one field that could have shown this. On the AP I tested it sat flat much of the time, holding the same value for more than 50 samples in a row. **On this bench that is what a quiet channel looks like, not a stuck counter.** One client, mostly downstream iPerf, almost nothing contending for the medium: there was very little utilization to report. **The number was uninformative here, which is not the same as wrong.** On a busy cell it may well be the field that shows this, and that is untested by this work.

This is where an investigation usually stops. Every box is ticked, so the change worked. **Go and measure trips per megabyte yourself, even when nothing tells you to look.**

## 4.7 The limit was not enforced where its name suggests

The downlink limit is written onto the client's **user group**, which reads as though it travels with the wireless client. It does not. **It is enforced at the gateway, on the routed path.** The radio is not doing the limiting and the client certainly is not.




*Two paths for the same client. Routed through the gateway, a 25 Mbps limit held the client to about 20 Mbps. Switched on the same subnet, a 2.5 Mbps limit delivered 495 Mbps.*

On the switched path I set the limit deliberately low, **2.5 Mbps**, and the client still pulled **495 Mbps**: a factor of 198 over its setting. The limited samples came back marginally faster than the unlimited ones, which is what noise looks like on a link nothing is limiting.

**Put the load generator on the same subnet as the client and you will build a rate-limited test that is not rate limited, with nothing in the output to tell you.**

There is no wireless-side alternative here either. The controller exposes no rate-limit field on the WLAN itself: the bandwidth-profile endpoint errors, the WLAN object carries no rate-limit key, and both traffic-rule endpoints return nothing. **Rate limiting on this platform is a gateway function wearing a client-side name.**

**One open question I cannot answer from these captures:** the 25 Mbps setting delivered about 20 Mbps, and I do not know whether that shortfall is ordinary overhead or the airtime cost in this paper holding throughput below the number that was configured.

## 4.8 6 GHz is not a way out

The obvious question is whether a newer band behaves differently. Same access point, same client, same night, both bands at 80 MHz, six repetitions each.




*At matched width, both bands collapse to about one frame per transmission. 5 GHz gives up 17x and 6 GHz 16x.*

**6 GHz gives up slightly less, and that is the extent of the good news.** Both bands land on the same floor, because the floor is physical rather than a property of the band. 6 GHz holds its aggregation a little better through the middle of the range: at a 10 Mbps limit it keeps 35% of frames aggregated against 5 GHz's 19%. By 2 Mbps that advantage is gone.

For anyone planning a Wi-Fi 6E or Wi-Fi 7 deployment: moving to 6 GHz buys you a great deal, and it does not buy an exemption from this.

**One caution, and it is a hypothesis rather than something I measured.** Always compare bands at matched channel width. A wider channel may already be batching less before any limit is applied, and a limit cannot take away what was not there, so an unmatched comparison could make a wide channel look as though it resists the effect.

## 5. Limitations

**Total airtime per megabyte was not measured, and it is the headline number this paper does not have.** Trips per megabyte is a count. Overhead airtime per megabit is real time but covers only control and management frames. Pricing the data frames needs the data rate each one went out at, and the captures record none for the limited client's downlink. **107 times the trips is not 107 times the airtime.** Do not read it that way, and do not let anyone quote it that way.

**I do not know whether this limiter shapes or drops, and it changes what the retry numbers mean.** A shaper delays packets and the cost stays visible at Layer 2, where these captures are taken. A policer drops them, and the retransmission happens in TCP, which an access point capture cannot see at all. **If this limiter drops, there is a second cost here that my method is structurally blind to**, and the retry rate in 4.6 is measuring half of the story.

**I measured the waste, not the damage.** Nothing here puts a number on what this costs the users of a busy cell. On a radio with idle time to spare, the waste comes out of capacity nobody was using that second. Quantifying it needs a capacity run: apply the limit to every client on a loaded radio and compare what the cell delivers against what the limits should have permitted.

**One access point and one model.** Everything here is a UniFi Express 7. Whether this appears on other UniFi models, let alone other vendors, is not established by this work. Section 7 is the ask.

**Run order was not randomized.** Conditions ran from loosest to tightest with repetitions consecutive, so the limit value marched in one direction across each sitting and anything else that drifted the same way cannot be separated from it. The effect is large enough that I do not believe drift explains it, and I cannot prove that here.

**Six repetitions per condition, with two exceptions**, both named in the appendix: the 2 Mbps condition on 5 GHz and the 100 Mbps condition on 6 GHz are five, because one run each voided.

**Results from different sittings are not interchangeable.** On identical hardware the same configuration varies by about half again between sittings, which is what 4.5 measures.

**"A busier medium" in 4.5 is an inference from retry rate, not a separate measurement.** Channel utilization and interference were not independently instrumented. **The platform's utilization counter sat flat through these runs, and 4.6 reads that as a genuinely quiet channel rather than a broken field, so it is untested here rather than discredited.** Whether it would reveal this cost on a contended cell is an open question this work does not answer.

**The two-client comparison uses two unrelated radios.** Apple N1 at Wi-Fi 7 against Broadcom at Wi-Fi 6E, on different macOS builds. It removes every environmental difference and none of the client differences.

**How badly pooling misleads was measured at one point, not across a range**, and cannot be interpolated from two clients because shared traffic such as beacons does not multiply per client.

**Radio features were left at platform defaults and I did not record what those were.** OFDMA, MU-MIMO and airtime fairness all shape how an AP batches and schedules.

## 6. What to do about it

**My recommendation is to stop rate limiting Wi-Fi. This is how far the evidence behind it reaches.**

Established, on one access point across two bands at six repetitions: a downlink rate limit stops the radio aggregating, drives frames per transmission to the physical floor of one, multiplies trips per megabyte by up to 107, multiplies control and management overhead airtime per megabit by up to 168, and shows up on no routine check. Not established: the total airtime bill, what a busy cell loses, and whether any other hardware behaves this way.

**So the strong form is scoped to platforms that behave like this one.** If you run a UniFi Express 7, this is about your gear. If you run anything else, section 7 is a twenty-minute test, and I would rather you ran it than took my word.

**A rate limit controls megabits. An access point never runs out of megabits.** It runs out of airtime, and a feature that makes the radio take the medium once per frame is spending airtime on overhead instead of on somebody's data. Where the cost is an order of magnitude and no dashboard reports it, the burden belongs on keeping the feature rather than on removing it.

**What I cannot tell you is how much it hurts.** On a radio with idle time, waste is waste and nobody notices. On a saturated one it has to come out of something. **I did not measure that. The argument here is that spending a hundred times the transmit opportunities a transfer requires is wrong on its own terms.**

**And the decision is often not yours to make.** Rate limiting usually arrives as a business call rather than an engineering one, the same way a captive portal does. Make the case, on the record. If you are overruled anyway, what follows is what protects the network afterwards.

### If you are going to do it anyway

- **Use the loosest limit that meets your goal, and do not tighten it.** The cost is already there at a generous setting and it keeps climbing all the way down.
- **Do not trust the dashboard to tell you how it went.** It will tell you the opposite.
- **Measure the share of frames that are still being aggregated**, by receiver address rather than by a cell average that mixes limited and unlimited clients together. It is the clearest single indicator and it is in any capture taken at the AP.

## 7. Run it on your own gear

Everything above is one access point. The response to that is to run it somewhere else.

The method needs no special hardware beyond an AP that can capture its own transmissions, and produces one number that compares directly against the figures here.

Follow section 3.3. Then report:

- **The share of downlink frames sent inside an aggregate, per condition.** That is the clearest number in this paper and the easiest to compute.
- **Frames per transmission, counting lone frames as transmissions.** If your figure bottoms out near 2 rather than near 1, you are dropping the un-aggregated frames.
- **Your retry rate for every condition, as a change rather than a level.** Section 4.6 is an open disagreement with other people's field results and nobody can line them up without it.
- **Where your platform enforces the limit**, tested with traffic rather than read from the configuration.
- **PHY data rates under the limit, if your capture carries them.** Mine did not, and it is the biggest hole in this work.
- **What your radios have OFDMA, MU-MIMO and airtime fairness set to.**

If you have time, repeat with two clients on one radio and only one limited, **both from a single capture, reported separately by receiver address.** Identical client hardware if you can manage it, which is the one thing I could not.

Send me what you find, including results that disagree with mine. Especially those.

## 8. Appendix: exactly what was run

Measured September 16 to 18, 2026, on a UniFi Express 7 with an Apple M5 MacBook Air as the rate limited client.

| Section | Band and channel | Conditions | Repetitions |
|---|---|---|---|
| 4.1, 4.2, 4.3, 4.6 | 5 GHz ch 149, 80 MHz | none, 100, 25, 10, 5, 2 Mbps | 6, except 2 Mbps which is 5 |
| 4.4 | 5 GHz ch 149, 80 MHz | 10 Mbps on the M5, none on the M4 | 6 |
| 4.5 | 5 GHz ch 149, 80 MHz | 10 Mbps, three separate sittings | 6, 6, 3 by row |
| 4.6, 4.8 | 6 GHz ch 37, 80 MHz | none, 100, 25, 10, 5, 2 Mbps | 6, except 100 Mbps which is 5 |
| 4.7 | 5 GHz ch 149 | routed 25 Mbps, switched 2.5 Mbps | 3 routed, 1 switched |

**Load, generated on the client, pulling from a wired server on another subnet:**

```
iperf3 -c 10.0.50.87 -p 5201 -t 45 -P 4 -R -J
```

`-R` makes it downlink. `-P 4` is four parallel streams. The server sits on a different VLAN so the traffic crosses the routed hop where the limiter lives.

**Capture, on the access point itself, over SSH:**

```
iw phy <phy> interface add moncap0 type monitor
cfg80211tool <radio> tx_monitor 1
tcpdump -i moncap0 -s 200 -w /tmp/airtime-capture.pcap
```

`tx_monitor` is the flag that makes this work. Without it the capture contains no downlink at all and still looks like a healthy capture.

**`-s 200` records only the first 200 bytes of each frame and discards the rest.** That is enough to keep every header this analysis reads, plus each frame's original length, and it keeps the files small enough to copy off the access point afterwards.

**Analysis. The field name matters and the obvious one is wrong.**

```
tshark -r capture.pcap -T fields \
       -e radiotap.ampdu.reference -e wlan.fc.type -e wlan.fc.retry \
       -e wlan.fc.fromds -e wlan.fc.tods -e wlan.ta -e wlan.ra
```

**It is `radiotap.ampdu.reference`, not `ampdu.reference`.** The latter is not a field and Wireshark rejects the command outright with `Some fields aren't valid`.

Take downlink data frames, meaning `wlan.fc.type==2 && wlan.fc.fromds==1 && wlan.fc.tods==0`. Group the ones that carry an A-MPDU reference; each distinct reference is one transmission. **Every frame with no reference is also one transmission, on its own.** Transmissions is the sum of the two. Frames per transmission is downlink frames divided by that sum, and trips per megabyte is that sum divided by the megabytes the client received.

**Count only the grouped frames and your figure cannot fall below about 2**, because a two-frame aggregate is the smallest group there is. You would read that as aggregation bottoming out at two frames per transmission. The floor is one.

**If you want to close the biggest gap in this work, capture the data rates.** The per-frame radio information in these captures carries no `radiotap.mcs`, `radiotap.vht` or `radiotap.he` fields on the limited client's downlink, which is why total airtime is unresolved here. A capture setup that records them would settle it.

The full run-by-run data and the scripts that produced it are available on request.

## A closing note on WAN limits

The commonest objection to all of this is not really about Wi-Fi. It goes: the site has a 100 Mbps WAN, nothing shapes traffic on the edge router, and rate limiting clients on the wireless is the only lever anybody has.

That is a real situation, and it is a WAN problem wearing a Wi-Fi costume.

The constraint lives on a wire. Solving it on the air means paying in airtime, the one resource on that site nobody can buy more of, to fix something that costs nothing to fix where it actually is. Most edge routers can shape egress. If you are reaching for a Wi-Fi rate limit because yours cannot, that is worth evaluating before you spend the airtime.

I mention it only because it comes up every time, and because the answer is not a defense of rate limiting on Wi-Fi. It is a reason to go and fix the router.

## Acknowledgments

**Jono Thompson** presented the prior work this paper measures itself against, *"Please form an orderly queue at the bar and you will be served"*, WiCo Fest, September 2026. He reached the same recommendation from different measurements, and section 4.6 sets his retry findings beside mine where they disagree. **Peter Mackenzie** contributed field observations on aggregation and retries under rate limits. Neither has endorsed this paper and any remaining errors are mine.

### Corrections in version 6.1 (October 2026)

A cold check of the companion talk against the raw data found three errors in version 6. None changes a conclusion.

1. **Section 4.4:** the two-client cell averaged about 11 frames per transmission, not 16. The 16 counted only frames inside an aggregate, which section 8 says not to do. Recounted from the six captures: 10.2 to 12.0, mean 11.4.
2. **Section 4.5:** the three sittings ran on two days (16 and 18 September), not three evenings, and the two on 18 September were about four hours apart, not one. The 16 September sitting used a file download rather than iperf3.
3. **Summary of what is established:** no limit to 2 Mbps multiplies trips per megabyte by 107 (10.4 to 1,118, the means of the runs), not 108. The earlier figure rounded the baseline to 10.4 before dividing.

*Corrections and disagreement are welcome and useful.* *keith@wlanpros.com*