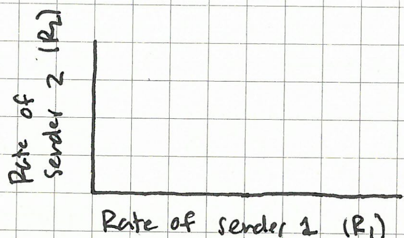
Here's a quick explanation of why an AIMD rule shares the network efficiently and fairly.

We'll concentrate on just two TCP senders. We're going to think about how their sending rates compare by working through some graphs:



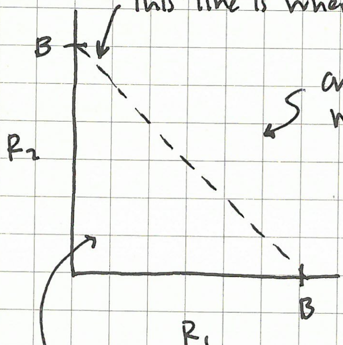

Let's assume they share some bottleneck link (which limits their throughput) that can handle B packets per round-trip-time.

- If $R_1 + R_2 < B$, the network is underutilized
- If $R_1 + R_2 > B$, the network becomes congested; packets will eventually be dropped

(we're assuming no other senders)

How does that play out in our graph?

This line is where $R_1 + R_2 = B$

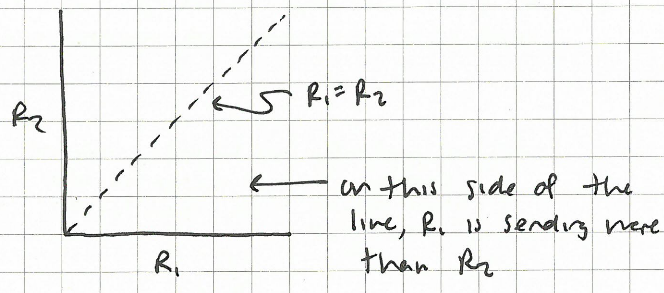



On this side of the line, we see a congested network

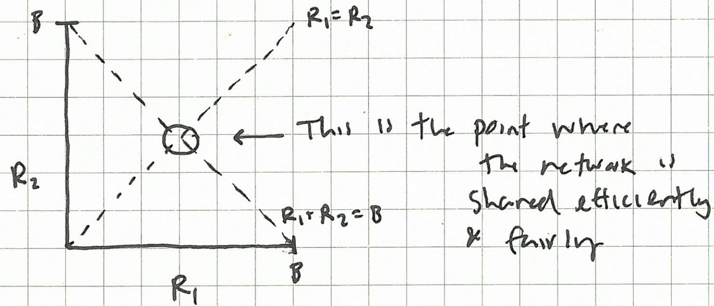
To maximize efficiency, we want $R_1 + R_2 = B$

On this side of the line we have an underutilized network

Maximizing efficiency isn't all we care about. What about fairness? Assuming both senders have infinite offered load, the network is ~~divided~~ shared fairly when $R_1 = R_2$.



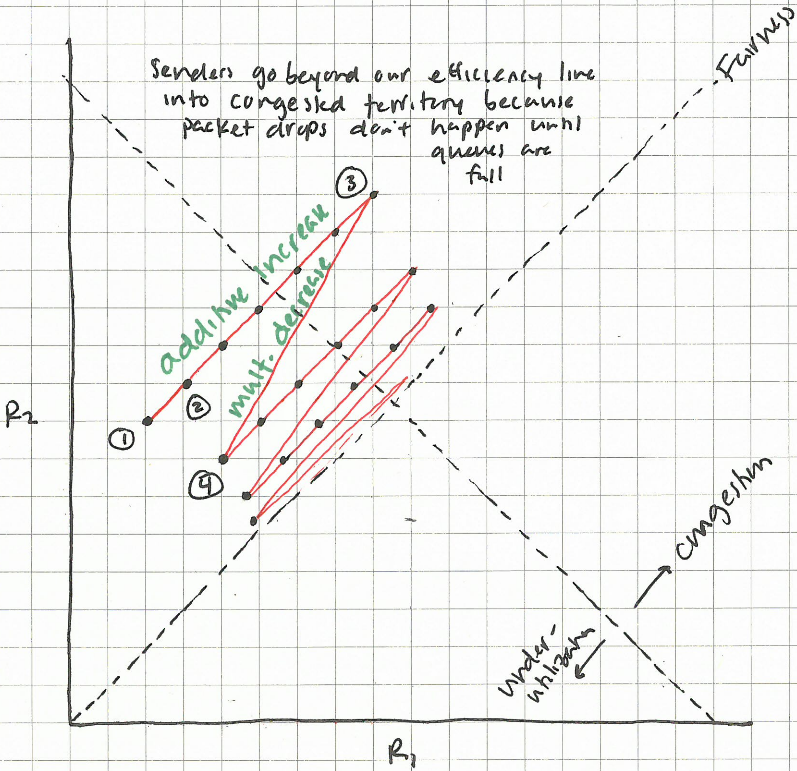

Putting these two graphs together...




That's no surprise — that intersection corresponds to $R_1 = R_2 = B/2$, which is what you should've expected we wanted.

What we need to do now, though, is show that an AIMD rule moves $R_1$ & $R_2$ to this point in the graph, no matter where we start.

① Arbitrarily, let's start here. $R_2$ is larger than $R_1$, but the network is underutilized. No packet drops → both senders will increase their windows additively. We'll end up at ②. ("sending rate" is more or less a shortcut for "window size" here)




This will continue to happen until packets are dropped – so until we are well into the "congested" side of the graph. Let's say ③. After packet loss – which we'll assume happens to both senders – the rates get cut in half (④). This pattern repeats.

As the pattern continues, we get ever closer to oscillating around the "fixed point" of efficiency and fairness, rather than travelling away from it.

LACURTS – 6.033 2017