Practical Rate Adaptation for Very High Throughput WLANs

Στολτίδης Αλέξανδρος, Λάντζος Στέργιος, Κουτσούκης Νικόλαος

July 4, 2022

Contents

1 Introduction

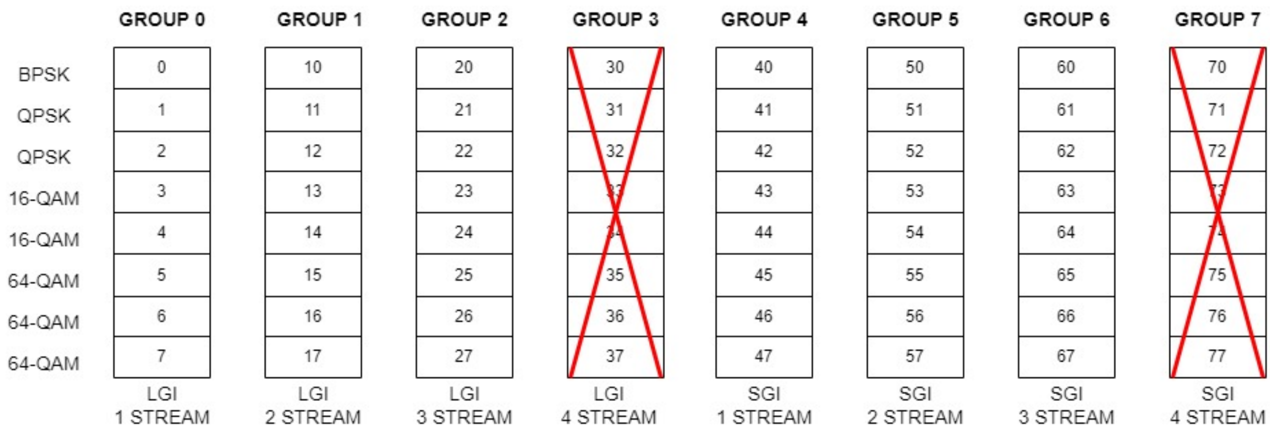
Rate adaptation is the determination of the optimal data transmission rate most appropriate for current wireless channel conditions. It consists of assessing channel conditions and accordingly adjusting the rate. A fundamental problem that wireless channels suffer from are time-varying losses due to mobility and interference, leading to poor and inconsistent throughput performance.

2 Multi Rate Retry Series

The Atheros driver specifies a multiple rate retry series (MRR) which determines the rates that a frame is going to be send with and the maximum amount of tramsmission retries that are going to be attempted when packet losses are detected. A four rate retry series is implemented in ATH9k (r_0/c_0, r_1/c_1, r_2/c_2, r_3/c_3). The four rate retry series indicates that every frame ready to be sent is going to be sent with a specific rate r_i for a maximum transmission attempts c_i. Therefore, the lost packet is transmitted at four different decreased rates until a success or an exceeding of the retry limit. Finally, based on the driver's source code the r_0/c_0 pair is used to send control packets at the slowest possible rate to minimize transmission losses.

3 MCS Groups and Indices

The driver supports up to eight MCS which determine the actual throughput of data. Lower MCS indicates less symbols for transmission and thus more reliability for smaller throughput. On the other hand, higher MCS is more prone to failure but much faster. The driver also supports a set of LGI and SGI streams. The maximum number of streams that our hardware supports is three since only three antennas are available. The paper takes into consideration only two streams but we extended our code to also support three streams. The driver supports up to four streams but we ignore this case since the hardware does not allow such implementations. Additional, SGI groups can be "unlocked" if the access point allows it. We also support these SGI groups with a maximum of three streams.



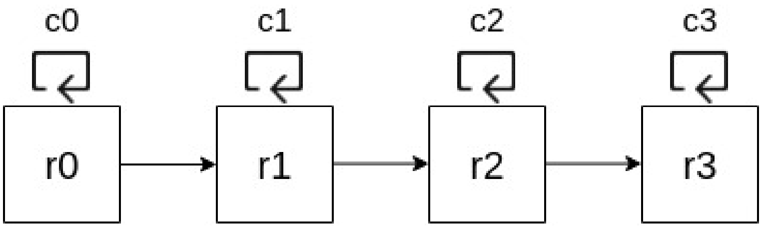
4 Rate Adaptation Algorithm

The rate adaptation algorithm suggested in the given paper aims to maximize throughput while also maintaining system robustness. Contrary to the original Minstrel rate adaptation algorithm, which sends a fraction of data at adjacent rates to evaluate the channel's conditions, the L3S algorithm continuously collects statistics based on the channel's conditions. These statistics can be categorized into two different groups. The short term statistics are responsible to control the responsiveness of the system by detecting fast improving and deteriorating channel conditions. On the other hand, long term statistics are useful for throughput maximization. The inner workings of the L3S algorithm and how these statistics are used is described bellow.

4.1 Short Term Statistics

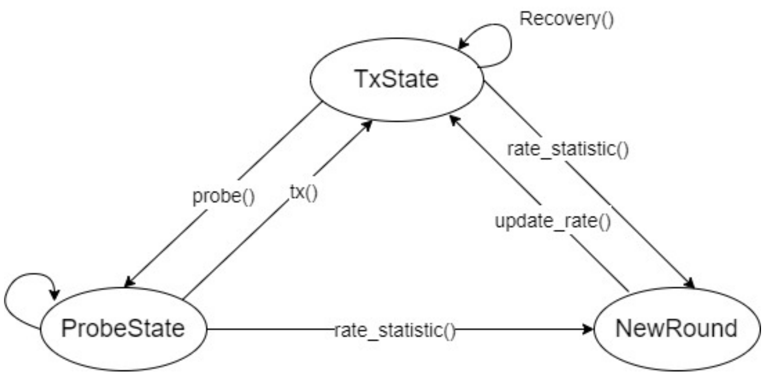
A set of three variables, *consecutive_ retries*, *consecutive_ successes* and *consecutive_failures*, is used to specify channel conditions. Whenever a frame is going to be send the *consecutive_ retries* counter is incremented. This variable denotes the total failed transmission attempts. When a frame is successfully transmitted the *consecutive_ successes* counter is incremented and the *consecutive_failures* variable resets back to zero. Whenever a failed transmission attempt is detected the *consecutive_failures* counter is incremented while the *consecutive_successes* counter resets.

The statistics mentioned above are used to determine whether probing must be accelerated or decelerated. Whenever ten consecutive transmission successes are detected the probing is delayed since the channel is not deteriorating. Whenever two consecutive failures are detected the the sender immediately falls back to its previous rate. Finally, if four or more frames fail to be sent the probing is accelerated since the channel is deteriorating.



4.2 Long Term Statistics

The long term statistics are maintained to adapt the transmission rate, which provides the best throughput, against sustained changes in the link. The potential best rate is determined by estimating the observed throughput performance, whether it has been used in the past transmissions or by the adjustment that enhances the system stability. The L3S algorithm moves from transmission state to probe state. Whenever the algorithm changes state the long term statistics are updated. Whenever a new round is commenced the algorithm is in transmission state and data is sent using the MRR series mentioned above.



4.2.1 Transmission State

Control packets are sent at r_0/c_0 where r_0 is the lowest possible rate to ensure successful transmission with a maximum amount of retries $c_0 = 4$. The data is initially sent at $r_1 = tx\_state$ with $c_1 = 2$. When the tx_rate transmission fails the data is sent at lower rates $r_2 = lower\_rate()$ with $c_2 = 2$ and $r_3 = lower\_rate()$ with $c_3 = 2$. A table of the MRR series for the transmission state is shown bellow.

rix_1/try_1	max_tp_rate [0]	2
rix_2/try_2	max_tp_rate [1]	2
rix_3/try_3	max_tp_rate [2]	2
rix_0/try_0	max_tp_rate [3]	4

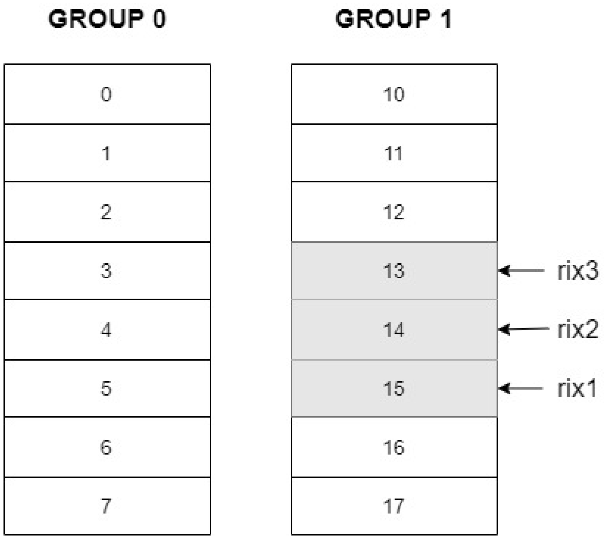
Since the rix_0/try_0 pair is responsible for control packets it is handled by the driver. Thus, we are not going to mentioned it on every MRR table from now on.

4.2.2 Probe State

The probing state is divided into two sub-periods. The first period is responsible for probing at adjacent MCS indices with the same amount of spatial streams. The MCS group can change only when the current rate is already at the lowest MCS index. Thus a smaller MCS index indicates a smaller "neighbouring" group. A table of the MRR series used and a figure of probing directions in this sub-period is shown bellow.

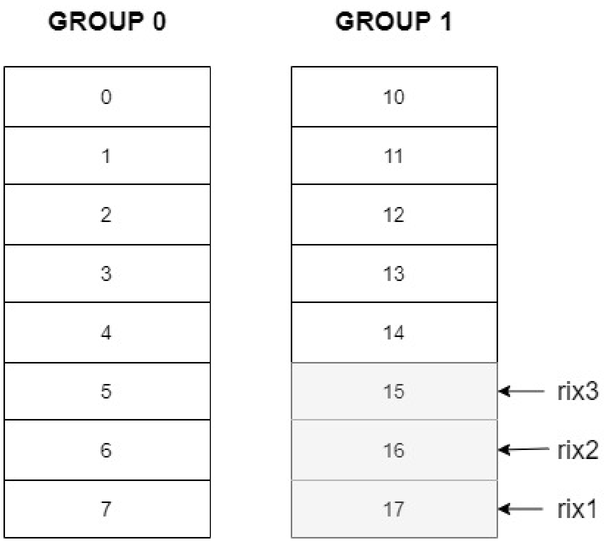
Probing Inside Only the Same Group with MCS Index $\in (0, 7)$

rix_1/try_1	max_tp_rate [0] + 1	2
rix_2/try_2	max_tp_rate [0]	2
rix_3/try_3	max_tp_rate [0] - 1	2



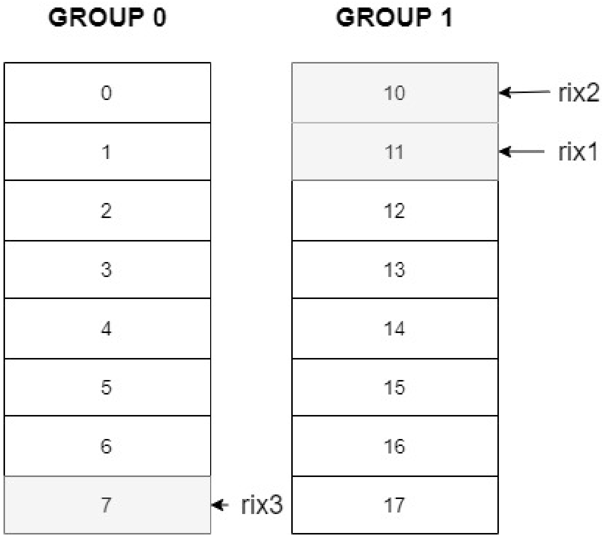
Probing Inside Only the Same Group with MCS Index $= 7$

rix_1/try_1	max_tp_rate [0]	2
rix_2/try_2	max_tp_rate [0] - 1	2
rix_3/try_3	max_tp_rate [0] - 2	2



Probing Inside and Outside the Same Group with MCS Index $= 0$

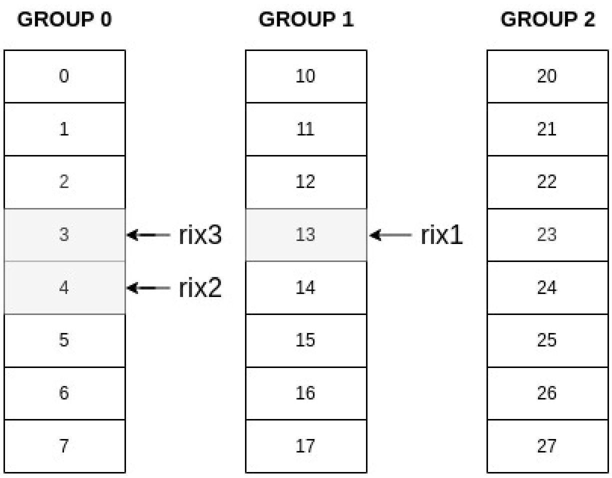
rix_1/try_1	max_tp_rate [0] + 1	2
rix_2/try_2	max_tp_rate [0]	2
rix_3/try_3	max_tp_rate [1]	2



The second probe sub-period is responsible for probing at adjacent MCS groups. When the current rate lies in the leftmost group then the probing direction is to the right group. On the other hand, when the current rate lies in the rightmost group the probing direction is to the right. In the given paper only these two cases are taken into consideration since the total amount of available spatial streams is equal to two. In our implementation we experiment both with two and three streams since the available hardware allows us to. When the current rate lies in an intermediate group then we periodically probe both on the right direction and the left directions. Both MRR series used and MCS group figures are shown below. We show what happens when three streams are available for the group in the middle since both probing directions can be seen.

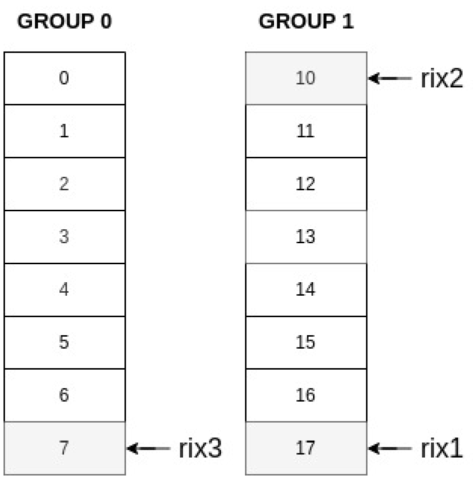
Probe Left with MCS Index $\in [0, 7)$

rix_1/try_1	max_tp_rate [0]	2
rix_2/try_2	max_tp_rate [0] - 9	2
rix_3/try_3	max_tp_rate [0] - 10	2



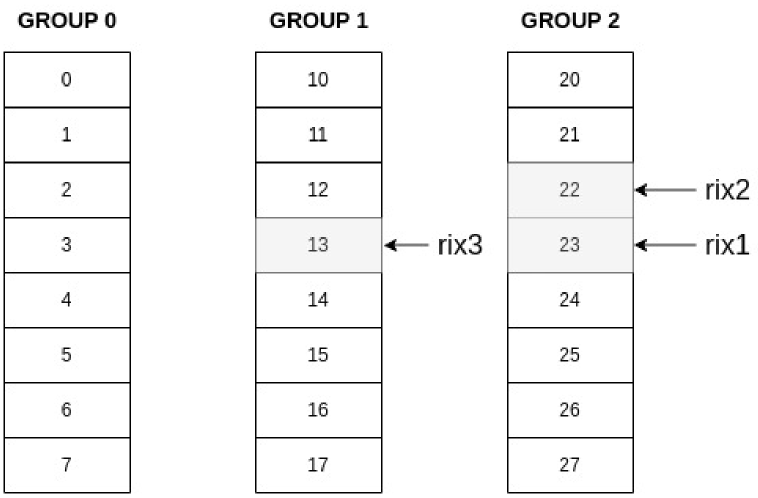
Probe Left with MCS Index $= 7$

rix_1/try_1	max_tp_rate [0]	2
rix_2/try_2	max_tp_rate [0] - 7	2
rix_3/try_3	max_tp_rate [0] - 10	2



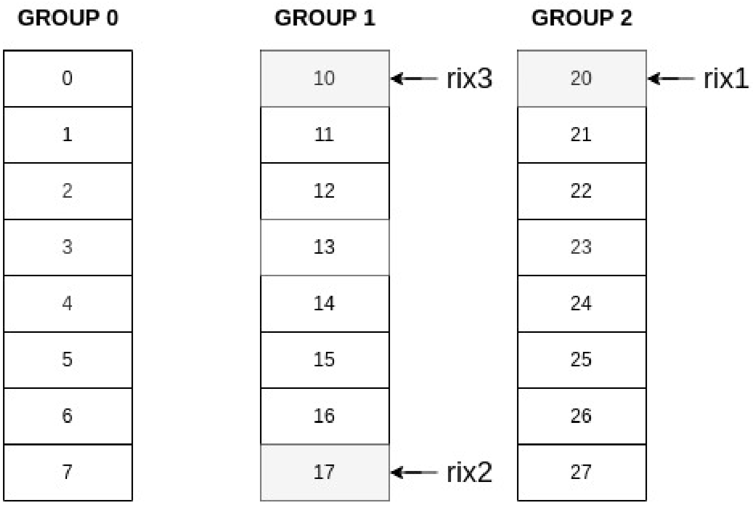
Probe Right with MCS Index $\in [0, 7)$

rix_1/try_1	max_tp_rate [0] $+$ 10	2
rix_2/try_2	max_tp_rate [0] $+$ 9	2
rix_3/try_3	max_tp_rate [0]	2



Probe Right with MCS Index $= 0$

rix_1/try_1	max_tp_rate [0] + 10	2
rix_2/try_2	max_tp_rate [0] + 7	2
rix_3/try_3	max_tp_rate [0]	2

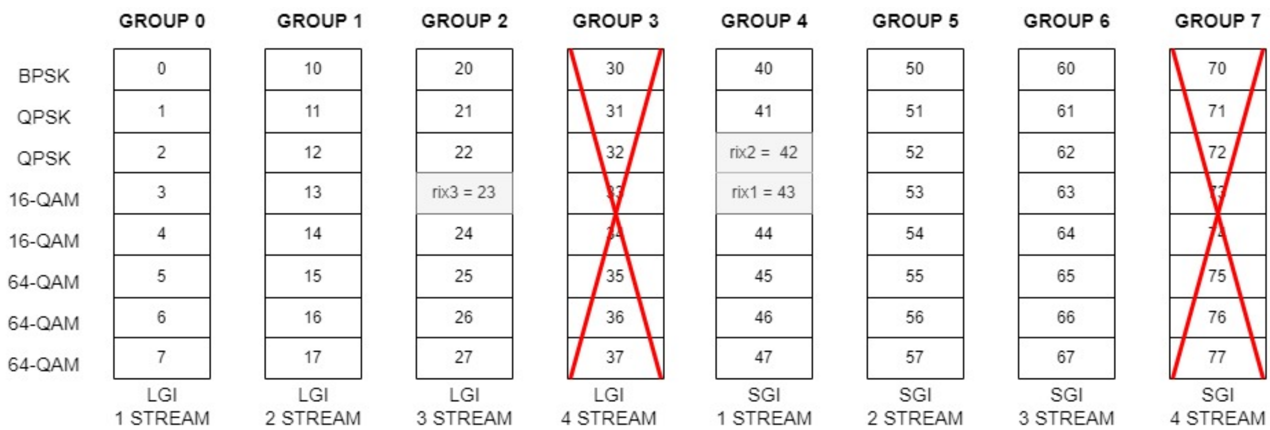


5 Additional SGI Groups

When probing with additional SGI groups we must avoid probing at rates that need four spatial streams since the available hardware does not support such amount of streams. Whenever probing right we must now check if we are at the second group and and we must avoid probing at rates supported at group three. Thus, we go directly to group four. Accordingly when on group four and we probe left the third group must be avoided. Thus, we directly switch to the second group. The MRR series and a figure of which groups are selected for the most general case is presented bellow.

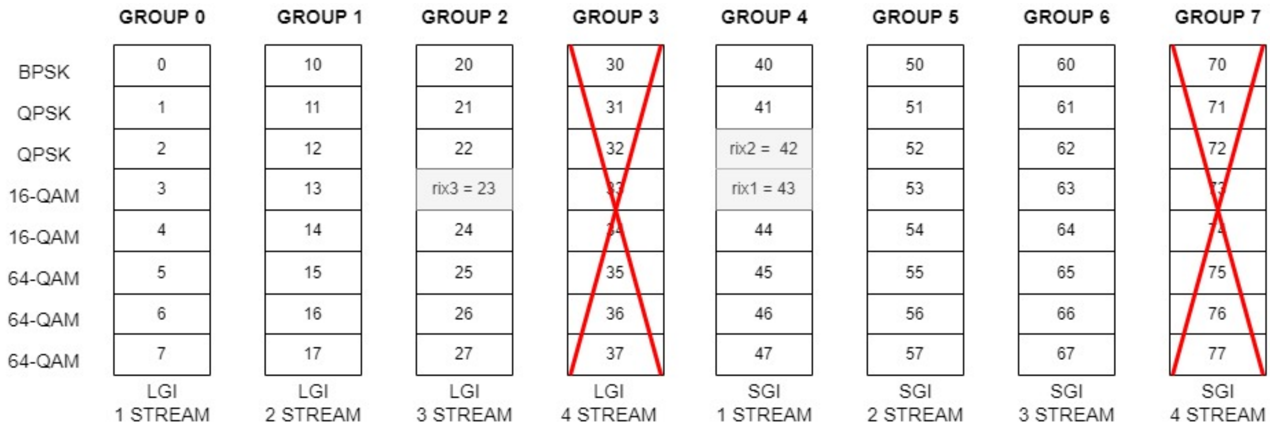
Probe Right with MCS Index $\in (0, 7]$

rix_1/try_1	max_tp_rate [0] + 20	2
rix_2/try_2	max_tp_rate [0] + 19	2
rix_3/try_3	max_tp_rate [0]	2



Probe Left with MCS Index $\in [0, 7)$

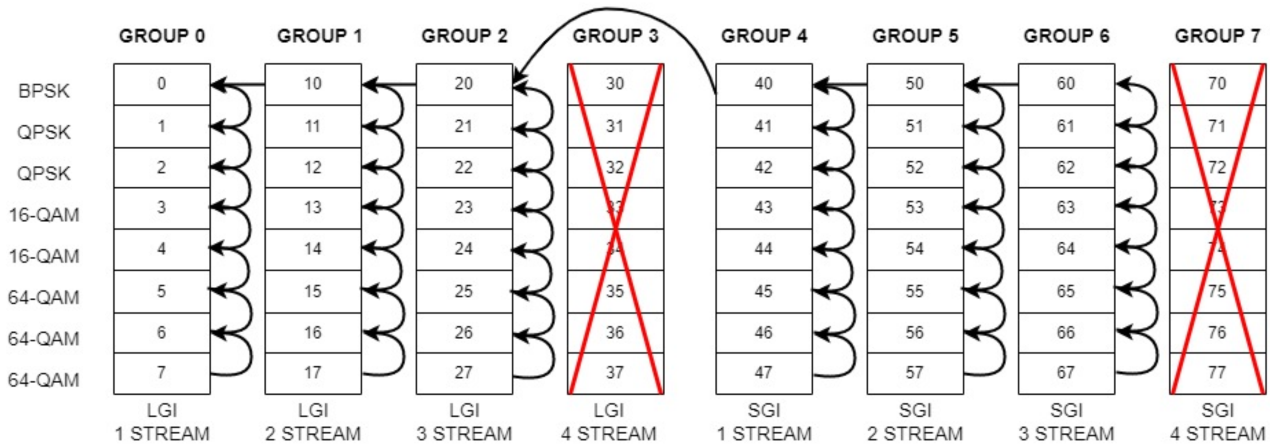
rix_1/try_1	max_tp_rate [0]	2
rix_2/try_2	max_tp_rate [0] - 19	2
rix_3/try_3	max_tp_rate [0] - 20	2



6 Recovery

Whenever the channel was not evaluated correctly and frames are lost we must immediately fall back to the previous MCS rate. When additional SGI groups are activated and the recovery function is called we must be careful to avoid group three since our hardware does not support three spatial streams. When the MCS index is zero and recovery is called the one less stream is used with the highest modulation possible. An image of the recovery process is presented bellow.

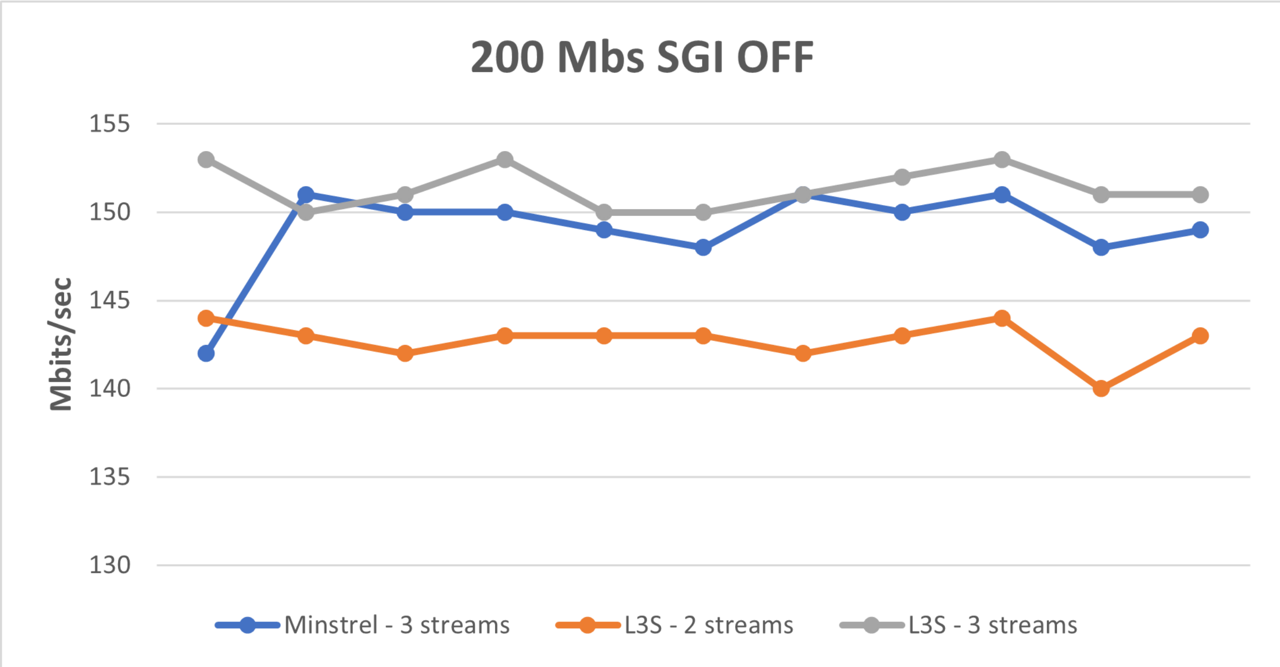
Recovery

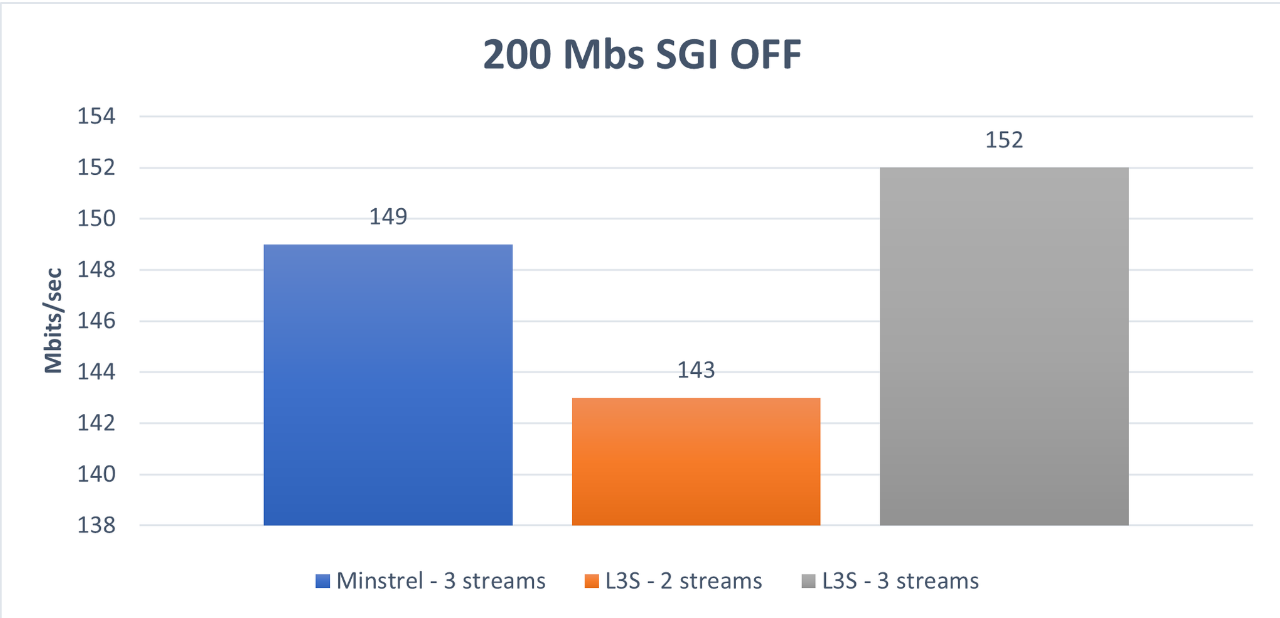


7 Benchmarks

We tested our code and compared it to the original, Minstrel, implementation. All tests were performed on the same channel with IEEE80211n. The code was tested for both two and three spatial streams. After the we observed that our implementation with three spatial streams outperforms the original algorithm we implemented additional SGI group. Finally, we compared the Minstrel algorithm with the L3S with an SHORT-GI-20. All benchmarks are displayed both bellow and on the submission files.

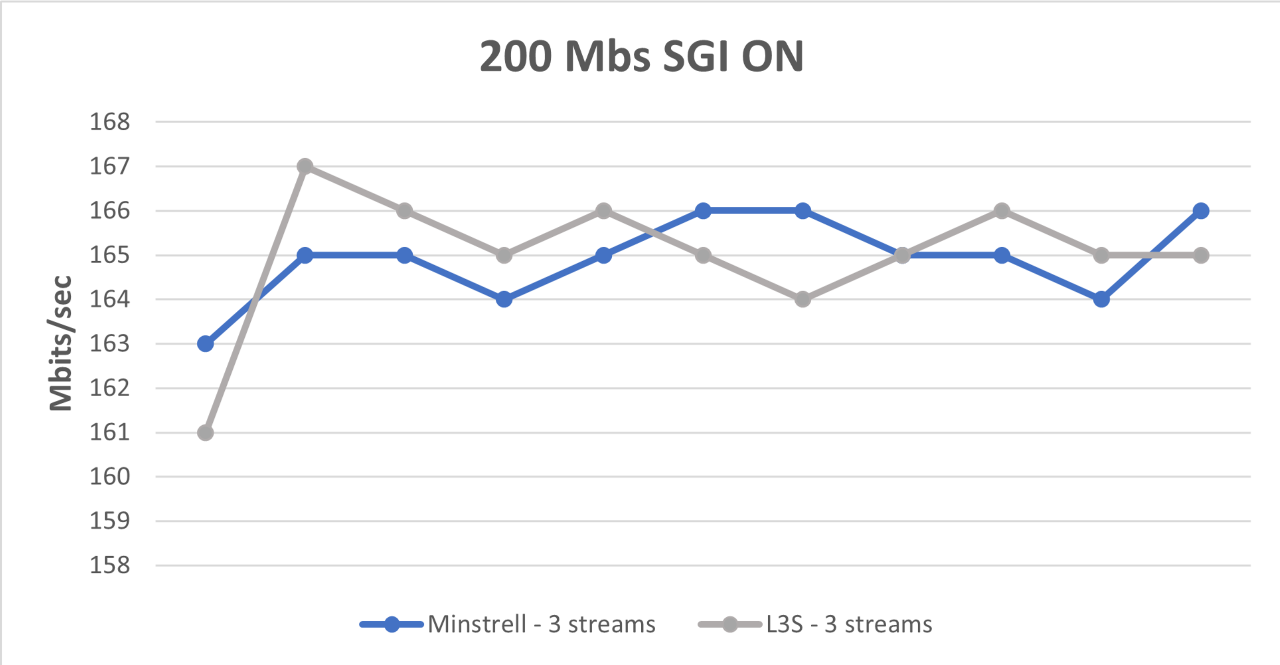
7.1 SGI-20 Disabled

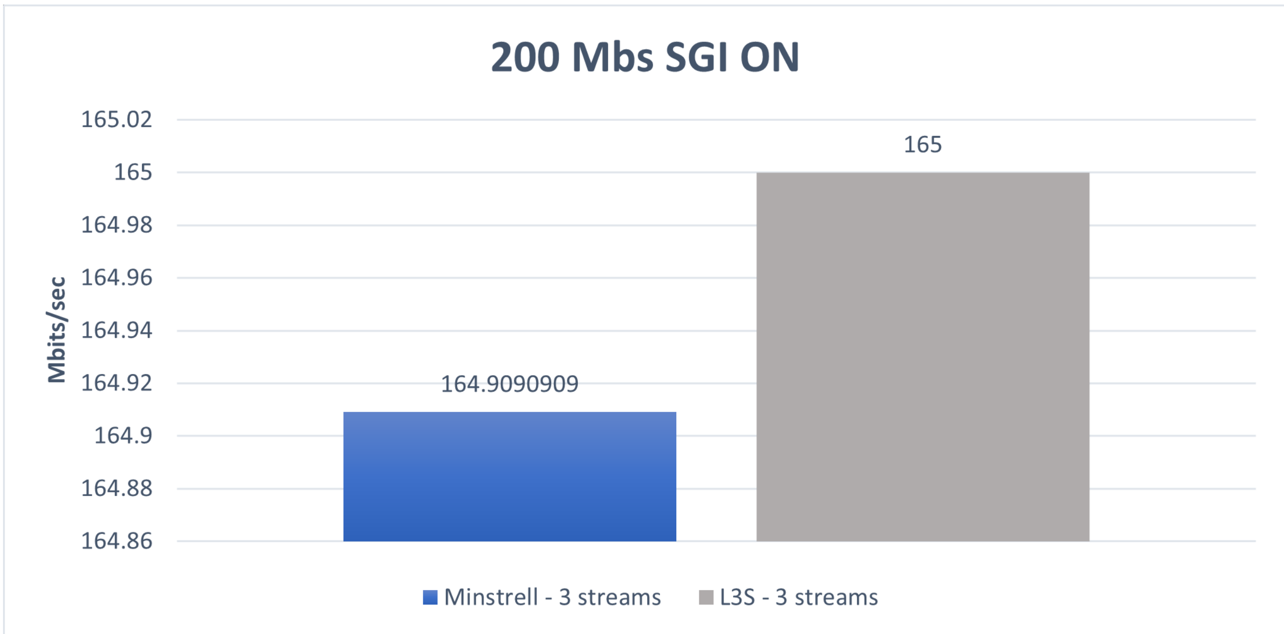




As you can see the original Minstrel algorithm outperforms our two stream implementation since the original algorithm uses by default all the available spatial streams that it can handle. A more fair comparison is the Minstrel algorithm versus the L3S with three spatial streams. In this case the L3S truly outperforms the Minstrel algorithm.

7.2 SGI-20 Enabled





As we can see with additional SGI groups the L3S outperforms the Minstrel by a small margin. This can be attributed to the fact that the channel was not degrading and thus the L3S does not presents its actual potential which is to optimize for deteriorating channel conditions.