WHITE PAPER – REPEATIT PACKETHEAL

In packet based transmission systems that are heavily dependent on the Transport Control Protocol (TCP), it is crucial to avoid packet errors on the physical layer. Low layer packet errors that are not handled before the upper TCP/IP layers are affected will cause out of order packet delivery, head of line blocking and reduced TCP window sizes. The result is lower throughput, higher packet delay variation (jitter) and bad network responsiveness. In practice this means end users will have a hard time opening web pages, fetch files and emails and voice based services such as Skype and Lync become shaky.

In a wired IP network the physical layer consists of equipment such as cables, switches, routers and network interface cards (NICs). Such networks often have very low (or zero) physical packet error rates. TCP/IP operation is therefore very efficient. Wired infrastructures also not change much over time. In worst case some site technician unplug the wrong wire or cause an Ethernet loop by misconfiguring a switch, but apart from that you can expect the cables to simply deliver a uniform service level.

Compared to wired backhauling systems, a wireless systems come with benefits such as lower deployment cost, faster time to market and higher flexibility and upgrade capabilities. Though, the wireless backhauling systems are also subject to higher physical layer packet errors, mainly due to that they are exposed to interference and noise from 3rd party equipment mounted in the vicinity. Interference levels also often vary over time as more equipment (sometimes with poor RF characteristics and lack of filters) is mounted on the same sites.

As the TCP/IP operation is heavily dependent on the physical packet error rates, it is of outmost importance that a wireless backhauling system can minimize these errors. These are some of the most common ways of dealing with this issue:

1) Forward Error Correction (FEC) encoders/decoders. This is a way of providing more robust communication on expense of system bandwidth. The basic principle is that each transmitted bit is represented by several bits.
2) Interleaving. The information is spread out (normally in time), so if some consecutive bits are lost it is still possible to repair a data frame.
3) Multiple Input Multiple Output (MIMO) antenna systems. Multiple antennas can be used in a number of different ways:
    - Beamforming. Beams are steered towards the receiver to improve the receiver Signal to Noise Ratio (SNR). This is not useful in point to point backhauling systems where high antenna gain is used and where some range is required.
    - Rx/Tx diversity with some diversity algorithm, for example Selection Diversity or Maximum Ratio Combining. Diversity also aims to improve receiver SNR, and it is also not useful in point to point systems.
    - Spatial multiplexing (one data stream per Rx/Tx antenna pair). As point to point (and often point to multipoint) systems normally are configured in free Line Of Sight (LOS), with directional antennas and limited multipath propagation, only two orthogonal antenna branches are available. These branches are used to enable spatial multiplexing for best data rates, and that also means Beamforming and Diversity becomes irrelevant.
4) Acknowledgement and fast retransmission of data frames. How this is done varies a lot between technologies. In for example Wi-Fi, a data packet is retransmitted eight (8) times before it is considered dropped by the upper layers. In LTE, retransmissions are handled by the Medium Access Control layer (Hybrid Adaptive Repeat Request and Adaptive Repeat Request).

This white paper describes Repeatit's PacketHeal solution. It is the way our Trinity wireless backhauling system handles acknowledgements, retransmissions and minimizes out-of-order delivery of TCP packets (bullet number 4 in the list above). The result is better stability, lower jitter, better performance and happier end users.

What PacketHeal is about

Repeatit's PacketHeal solution consists of these three main elements:

- Automatic adaptive wireless data rate selection for robust performance.
- Optimized acknowledgement handling to hide low-level packet errors for upper layers.
- Flexible (and configurable) data buffering to minimize out-of-order delivery of TCP packets.

Automatic Adaptive Data Rate Handling

Repatit's radio equipment uses various modulation schemes to transmit data. The available modulation rates are PSK (Phase Shift Keying), QPSK (Quadrature Phase-Shift Keying), 16/64QAM (Quadrature Amplitude Modulation) and OFDM (Orthogonal Frequency Division Multiplexing). All these technologies provide different levels of robustness, where there is a relation between modulation and achievable data rate. The used modulation is selected based on user settings, achieved link SNR, noise floor and PacketHeal intelligence.

The figure below illustrates how the achieved SNR in the receiver is dependent on the system range (the black line). Some conclusions can be drawn from this figure:

1) As the SNR drops, the link has to use lower and lower modulation and more and more coding in order to maintain low packet error rates.
2) Packet error could be a result of bad link budget (low SNR) *or* interference (the red marked source in the figure). Note: Radio equipment that only uses packet errors as input to rate selection will not be able to select appropriate rates.
3) Lower modulation and more coding naturally means that it takes longer time to transmit the same amount of data over the air. Not only does this result in lower system throughput, but the longer packets also become more vulnerable to interference. Yet another reason why it is important that the radio equipment knows its surrounding environment.

As mentioned above, many wireless systems use rate adaptation algorithms that are unaware of interference sources and instead use packet errors as indication of which modulation scheme should be used. If a link that has excellent SNR experiences a number of packet errors, a common approach is to use a more robust modulation in order to get the data transmitted. As the SNR was good in the first place, this only results in that the retransmitted packet is exposed to even higher risk of being interfered.

The figure below illustrates that PacketHeal bases the selected rates on both the SNR level and by analyzing the actual cause of packet loss.

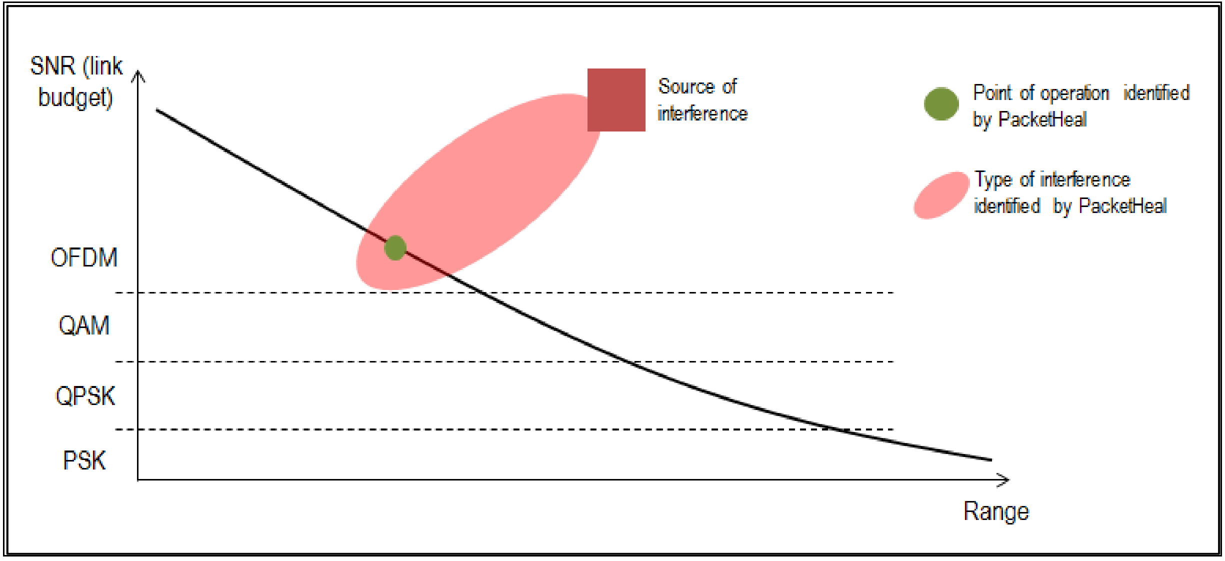
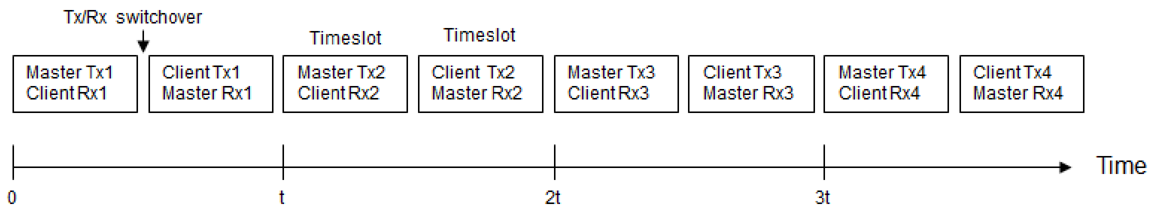


Figure 1 - PacketHeal Adaptive Data Rate Handling

Optimized Acknowledgement Handling

Once the physical radio layer operates on an optimal modulation and coding scheme, it is time to look at retransmissions of corrupted frames. As Repeatit's Trinity system uses a Time-Division Duplex protocol, transmitted data frames cannot be acknowledged directly like in for example Wi-Fi. Instead, a number of data frames are transmitted in one (Tx) window and then all those frames are acknowledged in an ACK bitmap sequence positioned in a following (Rx) window.

The upper part of the figure below shows the Tx/Rx window raster. Packets that are sent in Master Tx1 window are acknowledged in a later Master Rx window. The lower part of the figure is a snapshot of an ACK bitmap array that carries information about which data frames should be retransmitted.



512	513	514	515	516	517	518	519	520	521
NOK	OK	NOK	OK	NOK	OK	NOK	NOK	NOK	NOK

Figure 2 - Blabla

Flexible And Configurable Data Buffering

Maybe the most important and central part of Repeatit's PacketHeal feature is the ability to buffer data frames for in-order transmission over the wired backhaul. In the figure below, a Trinity Base Station transmits many data packets in time slot A1 and A2. In this example, one data frame (sequence number 2) in aggregate A1 is lost during transmission. The Client reports this to the Base Station which retransmits the Data Frame. Meanwhile, the Client buffers all following Data Frames (frame 3, 4, 5 etc.) until frame 2 arrives. This ensures in-order deliver from the wireless link and as the result the TCP transmission window can be maintained.

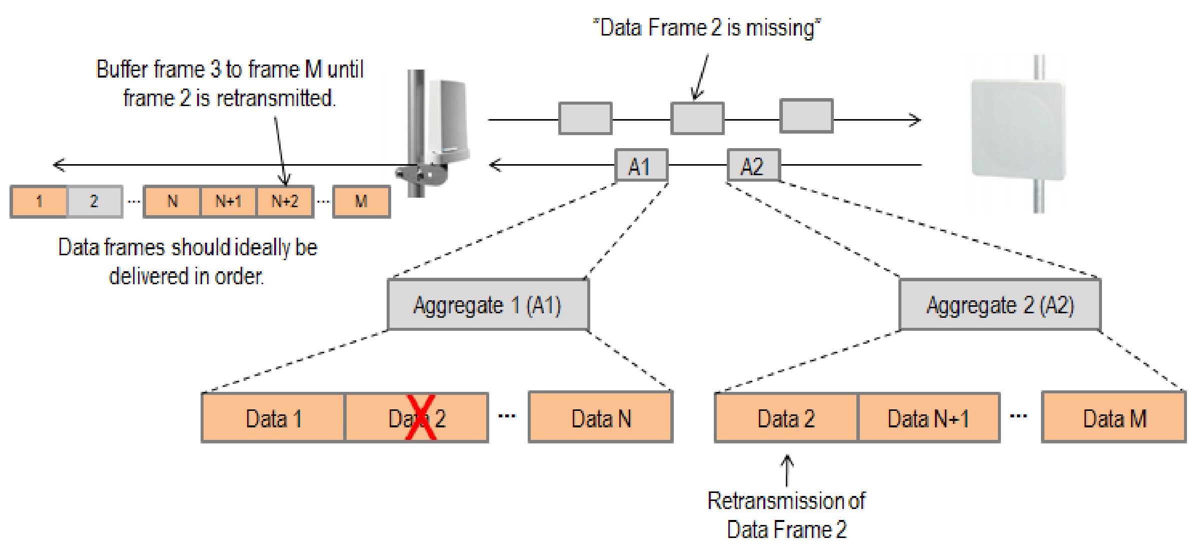


Figure 3 - Wireless operation when 50/50 bandwidth allocation is assigned in uplink/downlink direction. Data frame 2 is lost, and the client is buffering consecutive data frames (3, 4, ...) until frame 2 has been successfully received.