Considers four methods for sending packets across a 1 Mbps wireless link.

1. No coding. A packet is sent without any error correcting codes (ECC) or CRC.

2. Lower rate. A packet is sent at 1/2 the rate.

3. ECC with interleaving. A packet is sent with a rate 1/2 code applied to the header and data. After encoding the data is sent through an interleaver matrix. The interleaver writes the packet in rows that are 20 bits long and reads out the columns.

4. CRC. A packet is sent with a 4 byte CRC applied to the header and data. Every packet received has an ACK sent back. The ACK is equivalent to a packet with 0 data payload (note the 4 byte CRC is applied to the ACK header). The ACK is sent 10 µsec after the Data packet is received.

A packet consists of 20 bytes of synchronization and 20 bytes of header, followed by the data payload. Errors in the synchronization bits are not significant. The ECC corrects up to 3 errors in any block of 20 bits otherwise errors are not corrected. If errors are random (or if they are relatively spread out), the probability of the ECC failing to correct all the errors in a B byte packet is:

\[ P_{PE} \approx 2000 \cdot B \cdot P_{BER}^4 \]

where \( P_{BER} \) is the channel bit error rate (see eq. 7.126 in Rappaport if you want to know more). Note \( B \) is the size of the packet after ECC. Without the ECC, the probability of an error is the probability that no bit is corrupted.

If the data or ACK packet has an error, then the exchange fails and the exchange is retried.

If the errors come in bursts, then the errors are specified by a burst length \( L_B \) and a burst gap \( L_G \). For a packet of time duration \( \tau \), the probability of a burst appearing during that packet is \( P_B = 1 - e^{-\tau/L_G} \) (This is the probability that there is no event given a Poisson process of intensity \( 1/L_G \)). If a burst occurs, it causes \( L_B/2 \) errors (remember a noisy channel has half the bits in error). For simplicity, let the probability of a burst occurring be independent from packet to packet.
Every data packet sent (or data/ACK exchange) is followed by a \( \tau_w = 50 \mu \text{sec} \) wait period whether the packet is successfully received or not (the reason will become apparent in later discussions). The time to send a packet without acknowledgements is given by
\[
\tau_s = \tau_{ins} + \tau_w
\]
where \( \tau_{ins} \) is the packet insertion time.

The time to send one data/ACK exchange is
\[
\tau_e = \tau_{ins} + \tau_{prop} + \tau_{wa} + \tau_{prop} + \tau_{insA} + \tau_w
\]
where \( \tau_{prop} \) is the propagation delay which is 1\( \mu \text{sec} \) here, \( \tau_{wa} \) is the wait time before sending an ACK, and \( \tau_{insA} \) is the insertion time of an ACK.

The throughput, \( T \), is calculated as the data payload per packet divided by \( \tau_{cyc} \), the complete cycle time to send one data packet. \( \tau_{cyc} = \tau_s \) for a non-acknowledgement scheme and \( \tau_{cyc} = \tau_e N_E \) in a data/ACK scheme.

The goodput is \( G = T(1 - P_{PE}) \). \( P_{PE} \) for the CRC case is the probability that at the end of an exchange a packet is accepted because the CRC failed to detect the error.

We consider several combinations of packet size, performance metric, and channel model. There are two packet data payload sizes: 10, and 1000 bytes. There are three performance measures: probability a packet is accepted when it is in error, \( P_{PE} \); the goodput, \( G \); and the delay, \( \tau_{cyc} \). There are four channels: the first is an additive white Gaussian noise channel (AWGN) which has random errors with probability \( P_{BER} = 10^{-3} \); the second is a bursty Rayleigh fading channel with \( L_B = 2 \mu \text{s} \) and \( L_G = 1 \text{ms} \); the third is a bursty Rayleigh fading channel with \( L_B = 1 \text{ms} \) and \( L_G = 500 \text{ms} \), the last is an error free channel. All the channels with errors have the same average error rate, \( P_{BER} = 10^{-3} \). This is a high error rate but it represents the channel at the fringe of coverage.

For the case of reducing the channel rate by half, see the handout showing the BER as a function of \( E_b/N_0 \) (aka SNR). Estimate the reduction in average BER of adding 3dB to the SNR for the AWGN and Rayleigh channels. In the AWGN case reduce \( P_{BER} \) appropriately. In the Rayleigh case reduce \( L_B \) appropriately.

The solution to this problem should consist of 3 tables, one for each performance measure. One such table is illustrated below. For each combination of channel and packet size, highlight the coding scheme that is most effective. While there are a lot of values to compute in this problem, it can be done with a small C, java, perl or python program (or matlab, R...).

<table>
<thead>
<tr>
<th>Frame Payload (Bytes)</th>
<th>Coding Scheme</th>
<th>Channel 1 (Random)</th>
<th>Channel 2 (Short Burst)</th>
<th>Channel 3 (Long Burst)</th>
<th>Channel 4 (Error Free)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>no ECC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>no ECC 1/2 rate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ECC with inter</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CRC with ACK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>no ECC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>no ECC 1/2 rate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ECC with inter</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CRC with ACK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Based on your analysis answer the following questions.

1. Under which channel model did goodput increase as the packet size increased?
2. Comment on the relative merits of each of the methods for sending a packet.
3. Which method worked best with each channel model?
4. Which method would most likely support voice traffic across the different channels?
5. In which channel cases would ECC without interleaving be effective or ineffective?