Nicholas Rypkema (rypkema@mit.edu)

MIT Course 2.680 – Lab #11

Part 1

2.

a) The primary advantages of arithmetic source encoding include: flexibility (given any model with sequence of event probabilities, arithmetic coding can be used); optimality (optimal in theory and very nearly optimal in practice).

b) The primary disadvantages of arithmetic source encoding include: speed (tends to be slow to encode); does not produce a prefix code (disallowing parallel processing); poor error resistance .

c) This encoding scheme makes sense for marine robotics because most messages are repetitive (i.e. symbol probabilities are well known), we desire optimality (the high cost of underwater communications means we wish to compress as much information into a single message as possible), and speed is not much of an issue (since we cannot send many messages at once anyway) .

3.

a) i) In a way, Google Protobuf messages and C++ classes are directly analagous - a message is defined by a set of internal variables, from which a set of getter, setter, and checker methods are automatically produced; this is much the same as a C++ class, where we can define internal variables and class methods.
ii) set_temperature(21.435);

Part 2

1.

a) The size of the DCCL message is 6 bytes; the bit rate of the link is 25.6bps=3.2Bps; therefore the minimum interval to send 6 bytes is 6/3.2=1.875~=2 seconds; the maximum queue needed at this interval is theoretically 10/2=5 (messages in queue sent every 10 seconds), since for 5 messages the size of the queue is 5x6=30 bytes and the link is set to 32 bytes/10 seconds, but a max queue of 6 was observed in practice.

b) the latency increases as messages build up in the queue in an unbounded fashion, until the queue is filled. Messages are being created and requested to be sent at a faster rate than they can actually be sent. This can be overcome by either reducing the rate at which messages are requested to be sent, increasing the data rate in some way, or creating a very large queue (if a known number of messages are going to be sent). The latency was observed to be linearly increasing with a best-fit equation of approximately L(N)=2*N+2, starting from N=0 (packet 0).

c) A FILO/LIFO message queue would be well suited for positional/navigation information, hazard information, or messages that contain the operational state of the vehicle, as these are good examples where the latest message to be sent should be prioritized (sent first), as this type of information is important for both vehicle operational safety and monitoring; messages that have some form of temporal or spatial pattern may be better suited to a FIFO message queue, as a shoreside/topside observer would be able to make more sense of this type of data when it is sent in the order in which it is observed by the vehicle (data that has some form of serialized pattern).

2.

Assuming uniform distribution within each range, and assuming that the probabilities of sending messages in each range being P(small) for range [0,128) and P(large) for the range (128,32768], then P(large)=1-P(small) from the law of total probabilities (i.e. p_0=P(large)).

Part 3

1.

The following messages were received by the vehicle when the described commands were sent:

```
$CARXP,0*44
$CADQF,253,2*57
$CAMUA,1,2,1e8a*79
$CACST,172323,0000,1,5709,31,0387,0205,250,00,00,01,01,0001,002,0,2,1,0,-01,-1,-1,-1,00,253,0.0,-1,25120,4000*49
```

With the raw mini-packet NMEA-0183 message being:

```
$CAMUA,1,2,1e8a*79
```

2.

The length of the mini-packet as observed on the oscilloscope was approximately 1 second.

3.

Besides sending short vehicle commands, a 13-bit packet can also be used to relay scientific (CTD etc.) or navigational data (position, speed etc.) to and from the vehicle and shoreside; data from vehicle to the shoreside can be used for observation, while data from shoreside to the vehicle can be used to inform the vehicle's internal models for a scientific mission.

4.

a) The hopping sequence used is hop 0, hop 1, hop 2, hop 3, hop 4, hop 5, hop 6, and then repeated twice more from hop 0 (i.e. hop 0 to 6, hop 0 to 6, hop 0 to 6).
b) The binary symbol sequence represented by figure 3 is 1111100 1000101 1100110.
c) The length (in seconds) of our mini-packet sent earlier is approximately 0.5 seconds, given that the mini-packet sequence was observed to be about 1 second long, and the modem doubles the length of the physically sent message for redundancy purposes.
d) i) The bandwidth used by this modulation scheme is 0.416 x 10^4 Hz = 4160 Hz.
ii) If no hopping was performed, we would only need 1 hop/bin, and each bin has a bandwidth of 320 Hz.
iii) The purpose of hopping is to help prevent overlap/blending of sent data (of the same frequency range), and to allow us to fix bit errors that occur during transmission (lost or flipped bits can be corrected).

Part 4

1.

a) Compared to the default DCCL encoder, my arithmetic encoder generally outperforms (requires fewer bits) it in most cases (for the random sequence tests, my encoder usually uses around 15-20 bits, as compared to 20 bits for the default encoder).
b) My arithmetic encoder does not always outperform the default encoder; in the worst cases, where the temperature sequence is composed entirely of temperatures in the range [11, 19] my arithmetic encoder requires 29 bits; in the best cases, where the temperature sequence is composed entirely of 10 degree Celsius temperatures, my arithemtic encoder drastically outperforms the default encoder, requiring only 3 bits.
c) The expected number of bits is 10 – a 100% improvement of the arithmetic encoder over the default encoder (20 bits).

2. The performance of this arithmetic encoder can be further improved by incorporating probabilities

relating to the movement of the AUV – for example, if the AUV is diving/surfacing at a certain rate, the probabilities of certain temperatures being recorded drastically increases depending on the current depth of the AUV.