Differential Phase Shift Keying

—

Krishna Subramani - 150020008
Darshan Tank - 150020012
Satyam Mohla - 150010028

Motivation - Avoid Phase Ambiguity

- All types of PSK systems, including QPSK, suffer from **phase ambiguity**. This is because the phase of the modulated signal, with reference to the modulator carrier, conveys the data.
- To detect the phase of the received signal, the demodulator compares its phase with a stable, recovered carrier that is generated by a carrier recovery circuit in the demodulator. Since all available phases are usually present in the received signal, the carrier recovery circuit can lock onto any one of these phases to regenerate the carrier, even though only one of them allows correct demodulation of the data. This phenomenon is known as phase ambiguity.

Use of a Differential Encoding-Decoding Scheme

- Instead of using the data to set the phase of the PSK signal at the modulator, it can be used to change the phase by a specific amount. The demodulator then detects the changes in phase, rather than the absolute phase.
- Since this technique depends on the difference between successive phases, it is called **Differential Phase-Shift Keying (DPSK)**.
- Although the demodulator in a coherent DPSK system arbitrarily locks onto one of the phases in the received signal, this phase ambiguity is unimportant since only the phase differences are important.

Flow of our Scheme

Input Image

Python

Pre Processing:
1)Add a unique pilot sequence at the start and end
2)Convert to uint8 dump(readable by grc)

Post Processing:
1)Parse the output dump for the pilot sequences, and extract the relevant data

GNU Radio

Differential Encoding

Channel

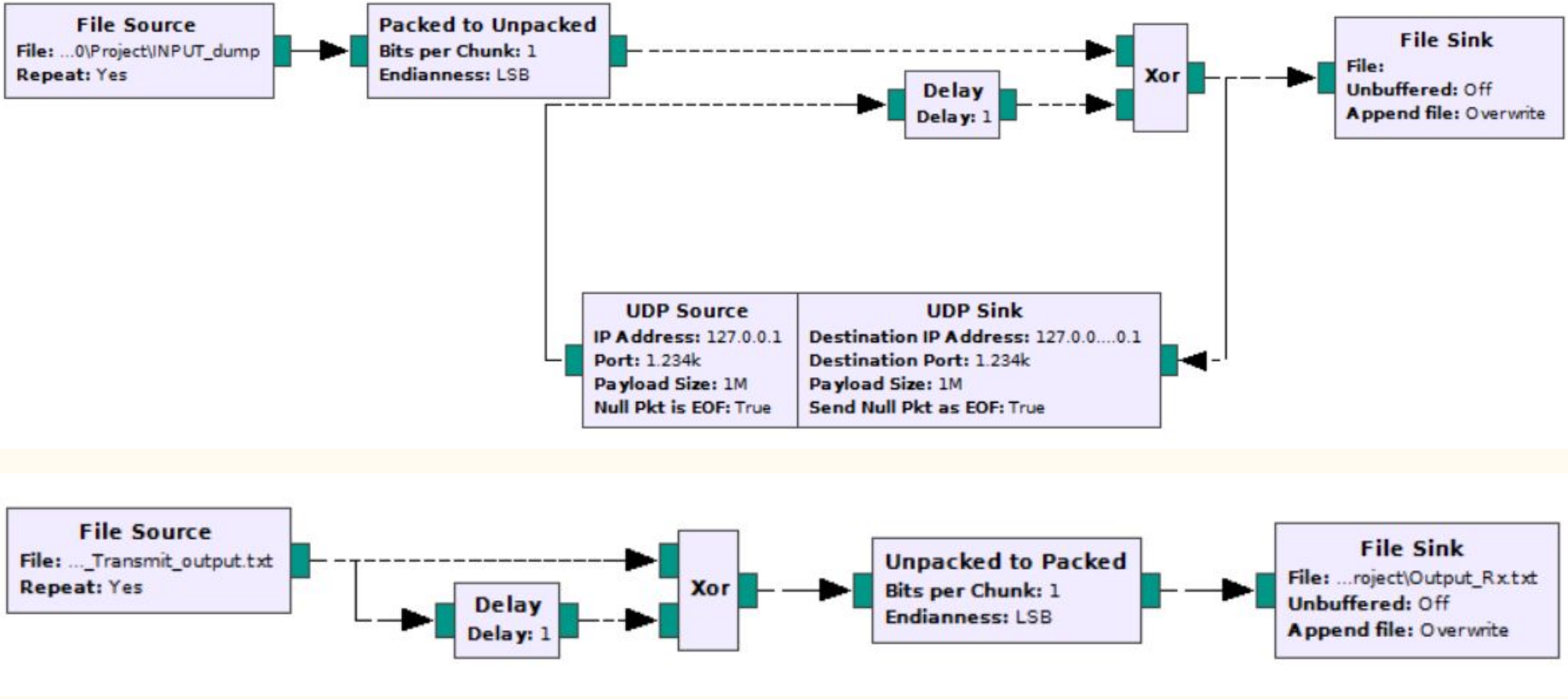
Differential Decoding

Output Image

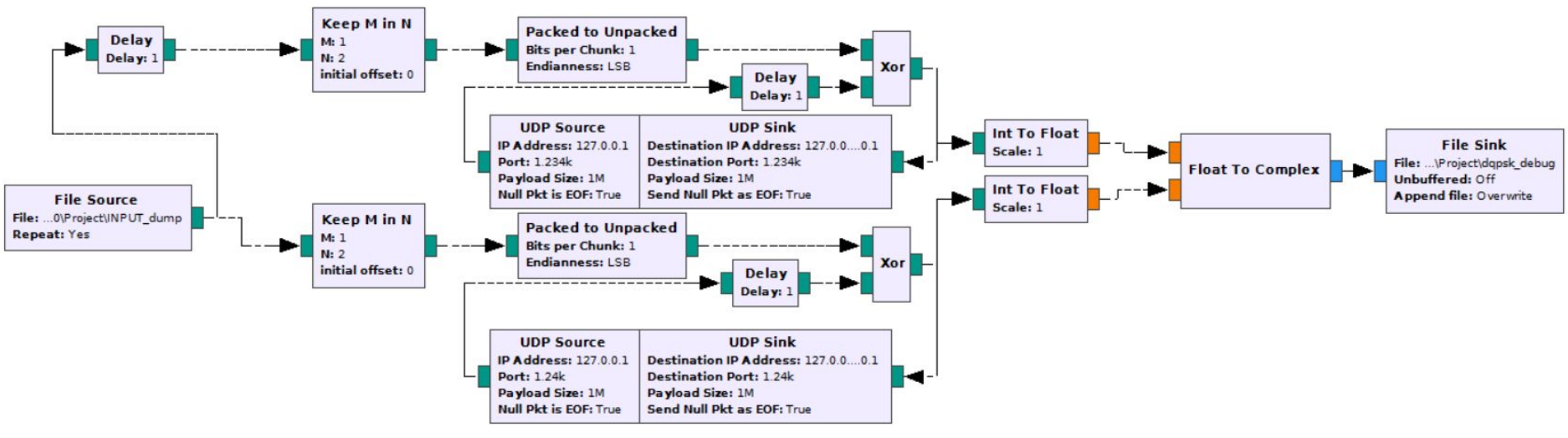
Python Pre/Post Processing

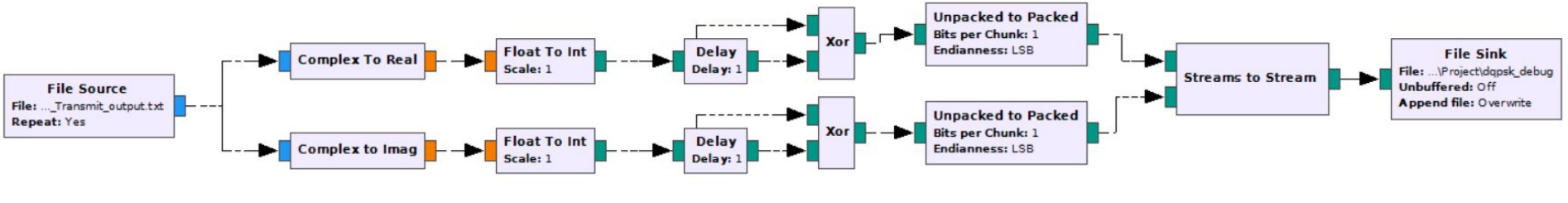
- The image was read, and converted to a binary dump for transmission.
- Unique Pilot sequences were added at the beginning and the end of the above file for detection at the receivers end. The pilot sequences were generated pseudorandomly, and are known to both the transmitter and receiver.
- After the file was received , a script was written to parse the obtained dump for the above two sequences, and the relevant image data was extracted.

Differential BPSK Modulation/Demodulation



Differential QPSK Modulation/Demodulation





Hardware Implementation using USRP/RTL-SDR

1. Using the Digital Modulation/Demodulation techniques learned in the Lab, we tried implementing the above mentioned schemes on USRP. However, we faced a number of challenges which we have highlighted below :
    a. The Blocks used in GNU radio have not been documented properly, so, we cannot fine tune the parameters to our data. We have used the same value of filter coefficients as in the lab course, but we found the received constellation diagram to be very noisy. We tried varying the parameters and observing the constellation, but there was data corruption/loss due to the imperfect constellation diagram received.
    b. There was a sampling rate issue with the hardware when being used with the Polyphase Arbitrary Resampler. This was the error ‘USRP Overrun - samples from USRP dropped’.

Challenges Faced

- In Differential QPSK, there might be an issue while combining the two data streams at the receiver's end.
- Since GNU Radio deals in integer bytes as the least unit(i.e. All operations occur with 4 byte input and output), the data has to be unpacked to a 4 bit number with either a '1' or '0' at the end. This unnecessarily increases the length of the file substantially.
- To implement feedback in GNU Radio, we have to use a UDP source-sink pair. The data rate is limited by the software, hence, data transmission is extremely slow(Took almost 20-25 minutes to transfer a 256kB grayscale image after overhead)