

# Joshi Reliable Transfer Protocol (JRTP)

## Specification

### Introduction

The Joshi Reliable Transfer Protocol (JRTP) is a protocol used for reliable file transfer over the noisy landscape of Mars. It is inspired from the Transmission Control Protocol described in RFC 793 (<https://tools.ietf.org/html/rfc793>) It takes advantage of the special assumed configurations and conditions on Mars to considerably reduce the size of the packets sent and the number of packets sent all together. At its base, it assumes that the Routing Information Protocol has been applied to learn the path from a source to a destination and whenever a file is sent, the routes are already learnt.

### Philosophy

The philosophy of JRTP is focused on creating a small packet size and minimum control packets to enable a reliable transmission. It uses a relatively simple scheme for communication to enable this: at any time, the sender will send a packet of several bytes and wait for an acknowledgement. Sending on a packet by packet basis makes the algorithm simple and reduces the need for several packets flooding the system.

### Functional Specification

This section will describe how JRTP is actually implemented.

#### Header Format

In JRTP, there are only three types of packets (Further details of the packets are specified later)

#### **SYN**

0	1	2	3	4
FLAGS	LENGTH			
DESTINATION IP			SOURCE..	
..IP	PAYLOAD...			

## ACK

0	1	2	3
FLAGS	DESTINATION IP		
SOURCE IP			ACK..
.. NUMBER			

## NORMAL PACKET

0	1	2	3
FLAGS	DESTINATION IP		
SOURCE IP			SEQ..
.. NUMBER			PAYLOAD...

Below is a description of each of these fields:

- **FLAGS (1B)** : This byte represents the type of packet this is. (1 : ACK, 2 : SYN, 4 : NORMAL PACKET)
- **LENGTH (4B)** : The total size of the packet to be transferred (Only sent with SYNs)
- **DESTINATION IP (3B)** : The destination IP of where the packet is destined to. Since the range of IPs of the rovers will always be 10.x.x.x, we only need 3 bytes to represent the packet
- **SOURCE IP (3B)** : The source IP of where the packet is coming from. Since the range of IPs of the rovers will always be 10.x.x.x, we only need 3 bytes to represent the packet
- **SEQUENCE NUMBER (4B)** : The sequence number of a packet carrying a payload. (ACK packets don't have a sequence number)
- **ACK NUMBER (4B)** : The acknowledgement for a particular sequence packet. This doesn't carry any payload.
- **PAYLOAD ( 0 - 5000B)** : The payload being sent over the JRTP.

### Example transfer:

Let's consider the scenario where A (10.7.0.1) wants to send a packet to B (10.11.0.1). The following steps will take place:

1. **A** : Send a SYN packet of the format:
  - a. FLAGS : 2
  - b. LENGTH : size of file
  - c. DESTINATION IP: 11.0.1
  - d. SOURCE IP : 7.0.1
  - e. PAYLOAD : the size of the payload which fits inside the window (0 to 5000 bytes)
2. **B** : Receives the packet, sets a counter for the total size and sends an ACK of the format:
  - a. FLAGS : 1
  - b. DESTINATION IP: 7.0.1
  - c. SOURCE IP : 11.0.1
  - d. ACK NUMBER : 1 (since the first sequence number is assumed to be 0)
3. **A** : Receives the ACK and sends a normal packet with the remaining payload of the format (like SYN but without the LENGTH field)
  - a. FLAGS : 4
  - b. DESTINATION IP: 11.0.1
  - c. SOURCE IP : 7.0.1
  - d. SEQUENCE NUMBER : the sequence of the packet being sent
  - e. PAYLOAD : the size of the payload which fits inside the window (0 to 5000 bytes)

This goes on till the whole file is transferred. The receiver knows when to stop since it keeps track of the total size and the data received so far.

### Schemes for ensuring reliability

Since the packets may be dropped while transmitting, J RTP implement the following schemes to ensure reliability:

- **ACKs**: Every sent packet needs to be ACKed
- **Timeouts**: When a packet is sent out, a timer is started. If an ACK is received before the timer runs out, the next packet (if any) is sent and the timer is reset. Else, the earlier packet is retransmitted.
- **Checksums**: Since the underlying protocol is UDP, there is an implicit guarantee of a non corrupted packet because of UDP's checksum.
- **Sequence Numbers**: To avoid reading duplicate packets or delayed packets, a sequence number count is kept.

## Evaluation

For evaluating the performance of JRTP, a diamond topology was set up, i.e.,  
A connected to B and C  
D connected to B and C.

The file to be transferred is bible.txt which is 4.3 mb file available here :  
<https://github.com/mxw/grmr/blob/master/src/finaltests/bible.txt>

Initially, no packets are dropped and file is transferred. Later, a 10% drop is simulated at node D.

The performance is tabulated in the table below:

% drops on D	Time taken	Throughput
0%	7s	0.61 MB/s
10%	1m 43s	0.041 MB/s

## Conclusion

JRTP is a reliable transfer protocol which can transfer a file over a series of Rovers even in a noisy environment using lesser and smaller packets than TCP.