As the packet passes through a router, the Sustainable Rate Measurement function may lower the value of the **Rate** field if it has allocated a lower rate to the traffic flow than the current value of the **Rate** field. If it alters the **Rate** field, the **Bottle_Node** field is set to identify the router. When the packet reaches its destination, the **Rate** field holds the lowest sustainable rate allocated by the routers along the packet's path, and the **Bottle_Node** field identifies the node that set this lowest rate.

In the direction of flow from destination to source (e.g in acknowledgment packets), I propose that one field be used:

Return_Rate: The value of the last **Rate** field which reached the destination.

This field is not altered as the packet travels back to the source. Thus, when the source receives the packet, it obtains a measure of the lowest bit rate available to it for the flow of traffic. It can then adjust its rate to not exceed this value, so as to minimise the congestion that the traffic flow causes to the network.

Note that acknowledgment packets carry not only the **Return_Rate** field but also the three fields **Desired_Rate**, **Rate** and **Bottle_Node**. This is because data packets and acknowledgment packets form *two* traffic flows, and they require different amounts of bandwidth and in different directions, as shown in Figure 5. Thus data packets carry the 'return' field to supply the destination with details of its acknowledgment traffic flow.

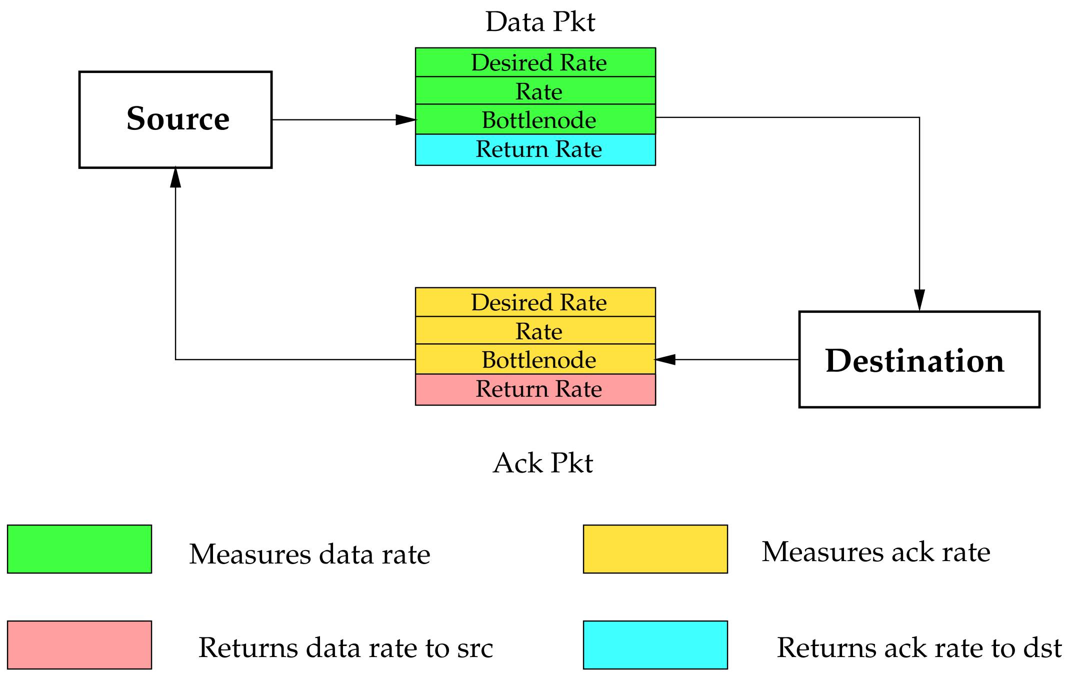

Figure 5: A Bi-directional Traffic Flow

4.5 Source Flow Control and Packet Admission Functions

In order to best utilise the **Return_Rate** values from the network components, and in order to admit packets as evenly spaced as possible into the network, the source should use a packet admission scheme. I propose a variant of Leaky Bucket [Turner 86]. The bucket has infinite buffer capacity, and admits packets into the network so that, at packet admission time, the flow's bit rate is exactly the **Return_Rate**. The inter-packet delay, shown in Figure 6, is calculated using the equation