KTH - Royal Institute of Technology

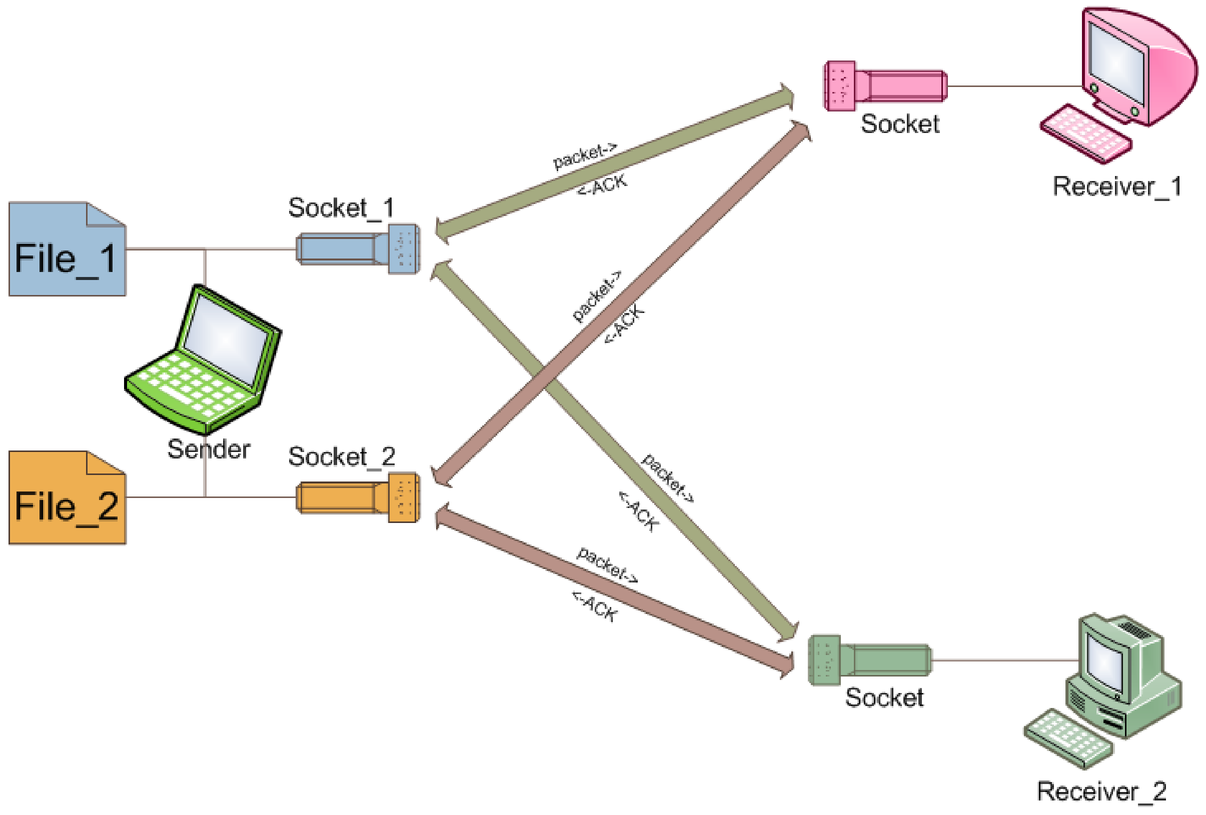
RDUP

IK2213 Network Services and Internet-based Applications

Sike Huang and Shanbo Li

2008-6-8

The Idea



The sender can transfer multiple files to several receivers, as illustrated in the figure above, there are two files on the sender's side, and each file binds to a socket. Since there two receivers, a file is sent via its associated socket to all the receivers respectively.

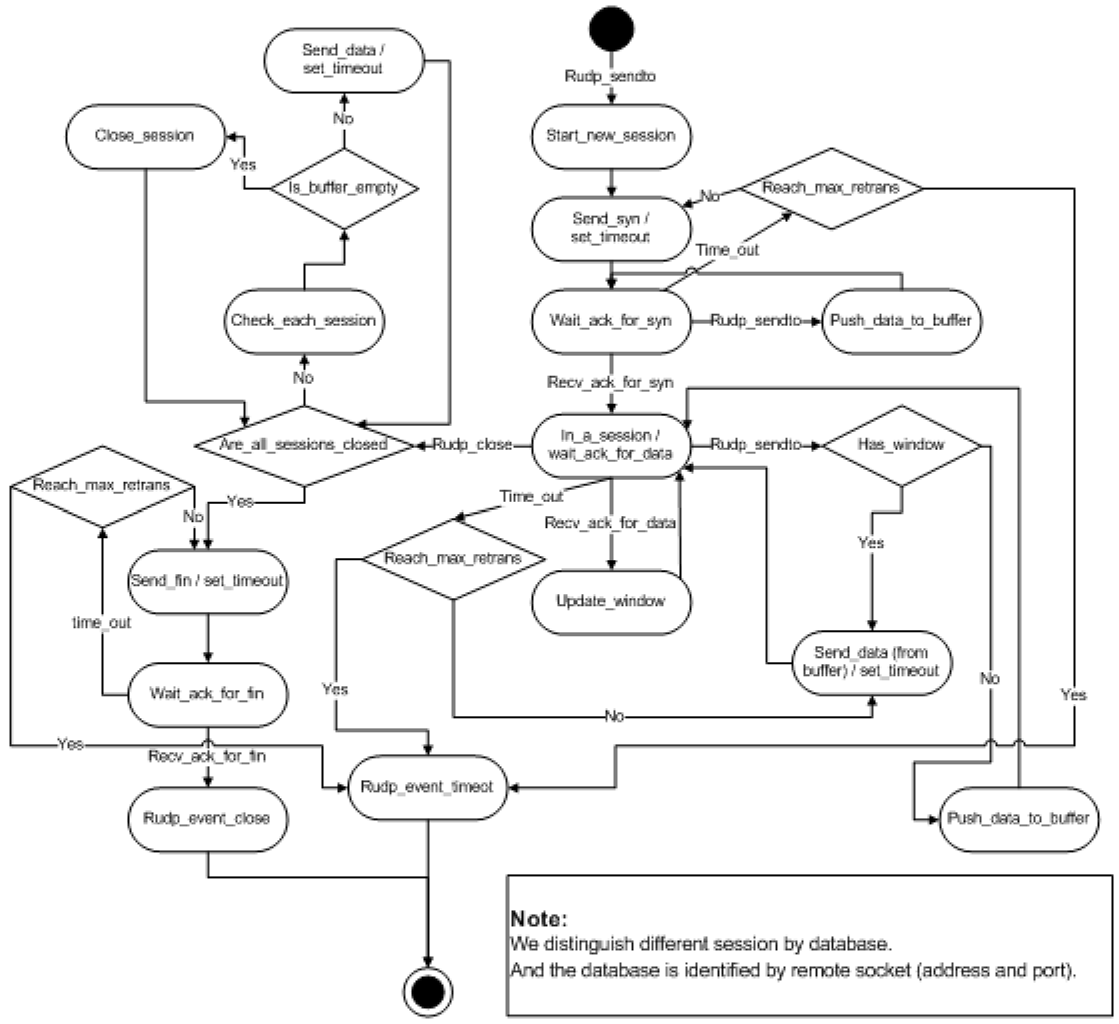
The Data Structure

We define several data structures in ***rudp_api.h***

```
// define the states of a (file transfer) session
typedef enum {
    RSESSION_START, // session has started
    RSESSION_TRANSFER, // transfering
    RSESSION_CLOSED, // session has been closed
    WAIT_FOR_ACK_OF_FIN // waiting for last ACK (for FIN)
} enum_session;

//  a datagram encapsulates the data to be sent/received
struct r_datagram {
    struct rudp_hdr header; // type of data
    char data[RUDP_MAXPKTSIZE]; // data itself
    int len; // length of data
    int has_ack; // whether this data has been ACKed or not
    int has_send; // whether this data has been sent or not
    int retrans_num; // number of retransmission times
    struct r_datagram* next; // pointer to next datagram
    struct sockaddr_in remote_addr; // remote socket
    struct r_socket* rsocket; // local socket
    struct r_database* database; // pointer to the database
};

// a database is used to maintain different sessions
struct r_database {
    struct sockaddr_in *remote; // remote socket
    int is_initialed; // -1 = false; 1 = true;
    int last_recv_seq; // last recived sequence number
    int last_send_seq; // last sent sequence number
    enum_session session_state; // holds the state of session
    struct r_datagram* datagram_buffer; // holds datagrams
    struct r_database* next; // pointer to next database
    struct r_database* pre; // pointer to previous database
};
typedef struct r_database* r_database_t;

// encapsulates a socket
struct r_socket {
    int (*super_rudp_receiver)(struct r_socket*, struct sockaddr_in*, char*, int);
    int (*super_event_handler)(struct r_socket*, rudp_event_t, struct sockaddr_in *);
    int sd; // socket descriptor
    // used by recevier to filter incoming file (identified by remote_socket and seq)
    struct r_database* database;
};
typedef struct r_socket* rudp_socket_t;
```

State Machine

Sender



Note:
We distinguish different session by database.
And the database is identified by remote socket (address and port).

Figure 1 Sender

Refer to **vs_send_statechart.png**

Receiver

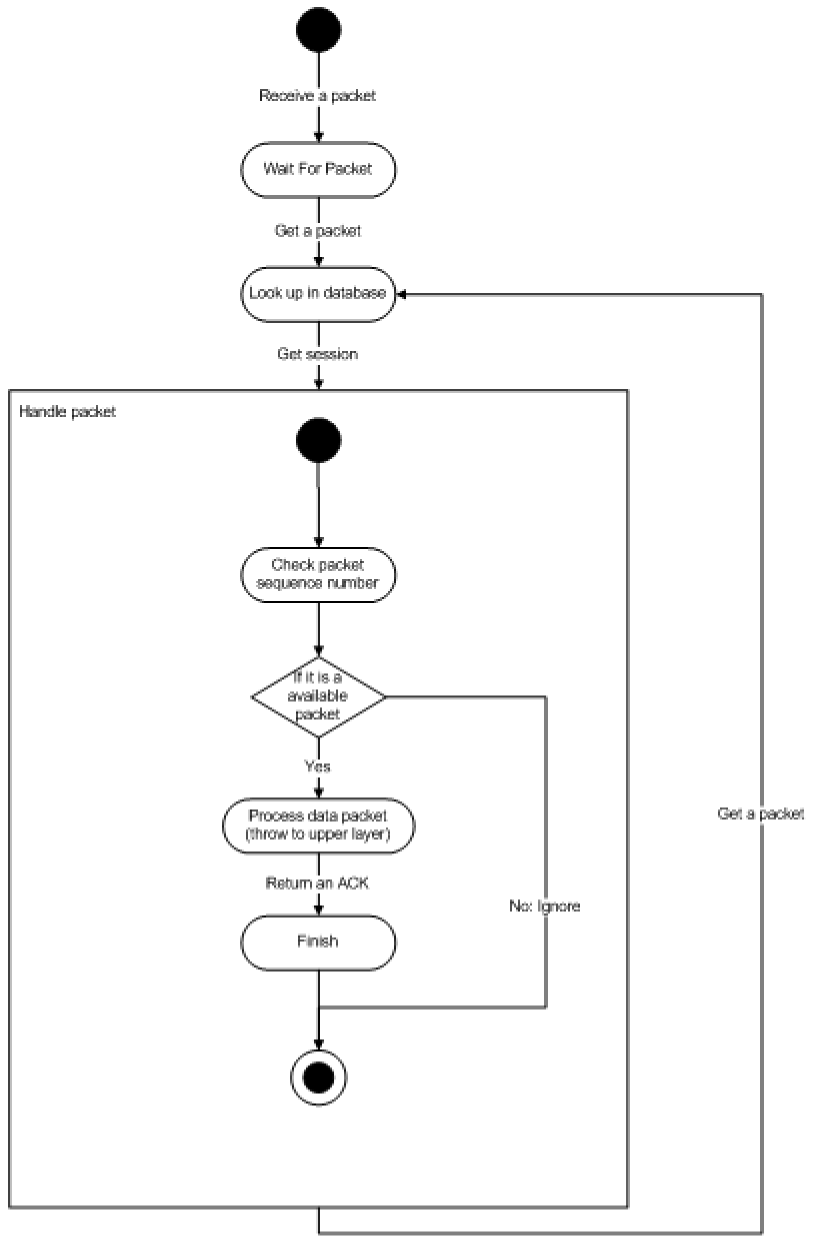


Figure 2 Receiver

Refer to **vs_recv_statechart.png**