

Appendix B: MATLAB Code

Parsing Script:

```
function [event, time, from, to, ptype, psize, flow_id, src, dst, seqnum, pid] = ns2_parse1(tcp)
%Initialize Variables
event = ''
time = 0;
from = 0;
to = 0;
ptype = '';
psize = 0;
flow_id = 0;
src = 0;
dst = 0;
seqnum = 0;
pid = 0;
fid = fopen(tcp, 'r');
tline = fgetl(fid);
while ischar(tline)
s = regexp(tline, ' ', 'split');
event = [event; s{1}];
time = [time; str2double(s{2})];
from = [from; str2double(s{3})];
to = [to; str2double(s{4})];
ptype = [ptype; s{5}];
psize = [psize; str2double(s{6})];
flow_id = [flow_id; str2double(s{8})];
src = [src; str2double(s{9})];
dst = [dst; str2double(s{10})];
seqnum = [seqnum; str2double(s{11})];
pid = [pid; str2double(s{12})];
tline = fgetl(fid);
end
i = size(time,1);
%Remove the first entries to get rid of the initialized values (only
%necessary for non-character arrays)
time = time(2:i);
from = from(2:i);
to = to(2:i);
psize = psize(2:i);
flow_id = flow_id(2:i);
src = src(2:i);
dst = dst(2:i);
seqnum = seqnum(2:i);
```

```
pid = pid(2:i);  
fclose(fid);  
end
```

Measure and Plot Script:

```
load('file.mat')  
int = 1;  
flow_id = 1;  
i1 = 1;  
j = 1;  
if(flow_id == 1)  
Rx = [9 4];  
elseif(flow_id == 2)  
Rx = [8 0];  
end  
bytes_rec = 0;  
C = [cell2mat(RTP_Event)'];  
while (max(RTP_Int_Time) > int*j)  
i2 = find(RTP_Int_Time >= int*j, 1, 'first');  
for i = i1:i2  
if (C(i) == 'r' && RTP_Tx_Node(i) == Rx(1) && RTP_Rx_Node(i) == Rx(2) && RTP_Fid(i) == flow_id)  
bytes_rec = RTP_Psize(i) + bytes_rec;  
end  
end  
if(i1 == 1)  
throughput = (bytes_rec*8)/(int*1e3);  
t = RTP_Int_Time(i2);  
else  
throughput = [throughput, (bytes_rec*8)/(int*1e3)];  
t = [t RTP_Int_Time(i2)];  
end  
bytes_rec = 0;  
i1 = i2 + 1;  
j = j + 1;  
end  
plot(t, throughput)  
xlabel('Time (s)')  
ylabel('Throughput (kbit/s)')  
if(flow_id == 1)  
title('RTP: Node 0 --> Node 4 Throughput');  
elseif(flow_id == 2)  
title('RTP: Node 4 --> Node 0 Throughput');  
end  
int = 1;
```

```
flow_id = 1;
i1 = 1;
j = 1;
lost = 0;
while (max(RTP_Int_Time) > int*j)
    i2 = find(RTP_Int_Time >= int*j, 1, 'first');
    for i = i1:i2
        if (C(i) == 'd' && RTP_Fid(i) == flow_id)
            lost = lost + 1;
        end
    end
    if (i1 == 1)
        lost_p = lost;
        lost_p_total = lost;
        t = RTP_Int_Time(i2);
    else
        lost_p = [lost_p lost];
        lost_p_total = [lost_p_total sum(lost_p)];
        t = [t RTP_Int_Time(i2)];
    end
    lost = 0;
    i1 = i2 + 1;
    j = j + 1;
end
figure
subplot(1,2,1); plot(t,lost_p);
if (flow_id == 1)
    subplot(1,2,1); title('RTP: Node 0 --> Node 4 Instantaneous Packet Loss');
elseif (flow_id == 2)
    subplot(1,2,1); title('RTP: Node 4 --> Node 0 Instantaneous Packet Loss');
end
subplot(1,2,1); xlabel('Time (s)')
subplot(1,2,1); ylabel('Lost Packets')
subplot(1,2,2); plot(t,lost_p_total, 'r');
if (flow_id == 1)
    subplot(1,2,2); title('RTP: Node 0 --> Node 4 Cumulative Packet Loss');
elseif (flow_id == 2)
    subplot(1,2,2); title('RTP: Node 4 --> Node 0 Cumulative Packet Loss');
end
subplot(1,2,2); xlabel('Time (s)')
subplot(1,2,2); ylabel('Lost Packets')
int = 4;
flow_id = 2;
i1 = 1;
j = 1;
t1 = 0;
t2 = 0;
```

```

flag = 1;
delay_int = 0;
if (flow_id == 1)
Tx = [0 8];
Rx = [9 4];
elseif (flow_id == 2)
Tx = [4 9];
Rx = [8 0];
end
while (max(RTP_Int_Time) > int*j)
i2 = find(RTP_Int_Time >= int*j, 1, 'first');
for i = i1:i2
flag = 1;
if (C(i) == '+' && RTP_Fid(i) == flow_id && RTP_Tx_Node(i) == Tx(1) && RTP_Rx_Node(i) == Rx(2))
t1 = RTP_Int_Time(i);
packet_id = RTP_Pan_Id(i);
for k = i:i2
if (C(k) == 'r' && RTP_Fid(k) == flow_id && RTP_Tx_Node(k) == Rx(1) && RTP_Rx_Node(k) == Rx(2) &&
RTP_Pan_Id(k) == packet_id)
t2 = RTP_Int_Time(k);
delay_int = [delay_int, t2 - t1];
break
end
end
end
end
delay_int = delay_int(2:size(delay_int,2));
if (i1 == 1)
delay = mean(delay_int);
t = RTP_Int_Time(i2);
else
delay = [delay mean(delay_int)];
t = [t RTP_Int_Time(i2)];
end
delay_int = 0;
i1 = i2 + 1;
j = j + 1;
end
figure
plot(t, delay);
xlabel('Time (s)')
ylabel('Delay (s)')
if (flow_id == 1)
title('RTP:Node 0 --> Node 4 End-to-End Delay');
elseif (flow_id == 2)
title('RTP: Node 4 --> Node 0 End-to-End Delay');
end

```

```

int = 4;
flow_id = 1;
i1 = 1;
j = 1;
t1 = 0;
t2 = 0;
flag = 1;
delay_int = 0;
if (flow_id == 1)
    Tx = [0 8];
    Rx = [9 4];
elseif (flow_id == 2)
    Tx = [4 9];
    Rx = [8 0];
end
while (max(RTP_Int_Time) > int*j)
    i2 = find(RTP_Int_Time >= int*j, 1, 'first');
    for i = i1:i2
        flag = 1;
        if (C(i) == '+' && RTP_Fid(i) == flow_id && RTP_Tx_Node(i) == Tx(1) && RTP_Rx_Node(i) == Tx(2))
            t1 = RTP_Int_Time(i);
            packet_id = RTP_Pan_Id(i);
        for k = i1:i2
            if (C(k) == 'r' && RTP_Fid(k) == flow_id && RTP_Tx_Node(k) == Rx(1) && RTP_Rx_Node(k) == Rx(2) &&
                RTP_Pan_Id(k) == packet_id)
                t2 = RTP_Int_Time(k);
                delay_int = [delay_int, t2 - t1];
            break
        end
    end
    end
    end
    delay_int = delay_int(2:size(delay_int,2))
    if (i1 == 1)
        jitter = mean(abs(diff(delay_int)));
        t = RTP_Int_Time(i2);
    else
        t = RTP_Int_Time(i2);
        jitter = [jitter, mean(abs(diff(delay_int)))]
        t = [t, mean(abs(diff(delay_int)))]
    end
    i1 = i2 + 1;
    j = j + 1;
end
figure
plot(jitter);
xlabel('Time (s)')

```

```
ylabel('jitter(s)')
if(flow_id == 1)
title('RTP: Node 0 --> Node 4 jitter');
elseif(flow_id == 2)
title('RTP: Node 4 --> Node 0 jitter');
end
```