

# RateControlProtocol

## Rate Control Protocol (RCP)

Developed by Nandita Dukkkipati and Nick McKeown in Stanford University, RCP aims to emulate processor sharing(PS) over a broad range of operating conditions. TCP's congestion control algorithm, and most of the other proposed alternatives such as ExplicitControlProtocol, try to emulate processor sharing by giving each competing flow an equal share of a bottleneck link. They emulate PS well in a static scenario when all flows are long-lived, but in scenarios where flows are short-lived, arrive randomly and have a finite amount of data to send, as is the case in today's Internet, they do not perform as well.

In RCP a router assigns a single rate to all flows that pass through it. The router does not keep flow-state nor does it do per-packet calculations. The flow rate is picked by routers based on the current queue occupancy and the aggregate input traffic rate.

## The Algorithm

The basic RCP algorithm is as follows:

1. Every router maintains a single fair-share rate,  $R(t)$ , that it offers to all flows. It updates  $R(t)$  approximately once every  $RTT$ .
2. Every packet header carries a rate field,  $R_p$ . When transmitted by the source,  $R_p = \text{infinity}$ . When a router receives a packet, if  $R(t)$  at the router is smaller than  $R_p$ , then  $R_p \leftarrow R(t)$ ; otherwise it's unchanged. The destination copies  $R_p$  into the acknowledgement packets, so as to notify the source. The packet header also carries an  $RTT$  field,  $RTT_p$ ; where  $RTT_p$  is the source's current estimate of the RTT for the flow. When a router receives a packet it uses  $RTT_p$  to update its moving average of the  $RTT$  of the flows passing through it,  $d0$ .
3. The source transmits at a rate  $R_p$ , which corresponds to the smallest offered rate along the path

## Papers

[Processor Sharing Flows in the Internet](#). N. Dukkkipati and N. McKeown  
Stanford University High Performance Networking Group Technical Report TR04-HPNG-061604, June 2004

– Main.OrlaMcGann - 11 Oct 2005