CS378 Lab-5 Report

Saksham Rathi (22B1003)
Department of Computer Science, IIT Bombay

Nandan Manjunath (22B0920)
Department of Computer Science, IIT Bombay

Part1: Theory

Let us prove this by induction.

Case1 : number of links(k) = 1

Packet 1 is ahead and reaches $t_1 = P/C_1$ where C_1 is the link's bandwidth. Packet 2 waits until Packet 1 is completely sent and then is sent. It takes time $P/C_1 + P/C_1$, which is $t_2 = 2 * P/C_1$.

In this case, $P/(t_2 - t_1)$ is C_1, which is the correct bottleneck bandwidth. Therefore, the base case is true.

Case2 : True for number of links(k) then true for links(k+1)

Until it covers k links, let the time they take be t_1 and t_2, and bottleneck bandwidth be C which is $P/(t_2 - t_1)$ (induction assumption). This bandwidth C is the minimum among all links k. Let $k+1^{th}$ link bandwidth be D.

Case2.1: If D is greater than or equal to C

Now new t_1' will become $t_1 + P/D$ and as $D >= C$, t_1' will be less than t_2 therefore, by the time complete Packet 2 crosses link k, complete Packet 1 will cross $k+1^{th}$ link also. So, Packet 2 doesn't wait, and the new t_2' will be $t_2 + P/D$. In this case, $P/(t_2' - t_1')$ is C, which matches the bottleneck bandwidth.

Case2.2: Id D is less than C

Then, the bottleneck bandwidth will become D. The new t_1' is $t_1 + P/D$. and as $D < C$, t_1' will be more than t_2 therefore Packet2 has to wait until Packet 1 is completely sent across $k+1^{th}$ link. This waiting time is $t_1' - t_2$, and then it is sent. The new t_2' is t_2+waiting time $+P/D$ which is $t_2 + t_1' - t_2 + P/D$. In this case, $P/(t_2' - t_1')$ is D, which matches the bottleneck bandwidth.

By induction, we proved that bottleneck bandwidth can be calculated as $P/(t_2 - t_1)$