Exercises
TCP/IP Networking
With Solutions

Jean-Yves Le Boudec

Fall 2009

3 Module 3: Congestion Control

Exercise 3.2 1. *Assume that a TCP sender, called S, does not implement fast retransmit, but does implement slow start and congestion avoidance. Assume that*

- *segments $n, n+1, n+2, \ldots, n+10$ are transmitted at times $0, 1, 2, \ldots, 10$ deciseconds (ds)*
- *transmission time per segment is 1 ds*
- *round trip time (2-way propagation plus packet transmission, ack processing and transmission) is fixed and equal to 10 ds*
- *segment n is lost*
- *no other segment or ack is lost*
- *there is no misordering of segments or acks by the network*
- *the retransmission timer for segment n is set to 60 ds, starting at the end of the transmission*
- `cwnd` = `twnd` = *64 segments at time 0*
- `offeredWindow` = *70 segments all along the exercise*

By which means and at what time will the loss of segment n be detected ?

Solution:

Loss of segment n is detected by a retransmission timer timeout, at 61 ds.

2. *Immediately after re-transmitting segment n (due to a timeout), S has 3 more segments ready to send (we call them segments $n+11$ to $n+13$). At what time is the ack for segment $n+3$ received (assuming again that there are no losses other than that of segment n) ? For segment $n+13$?*

Solution:

$n+3$ is ACK'ed at 71 ds, $n+13$ is ACK'ed at 91 ds.

Segment $n+3$ is sent at 3 ds and received at 4 ds. At 61 ds, the loss of segment n is detected and a retransmission occurs. Since the RTT is equal to 10 ds, the segment n is ACK'ed at 71 ds along *with segments $n+1$ to $n+10$.*

Since a loss was detected, $cwnd$ is set to 1 and $twnd$ is divided by 2. When the ACK for n is received, $cwnd$ is set to 2.

Hence, segment $n+11$ is sent at 71 ds and segment $n+12$ is sent at 72 ds. $n+11$ is ACK'ed at 81 ds. At this point, $cwnd$ is incremented and segments $n+13$ is sent. It is ACK'ed at 91 ds.

3. *Same question as before assuming now that fast retransmit and fast recovery are implemented and segments $n+11$ to $n+13$ are available for transmission at time 60.*

Solution:
$n + 3$ is ACK'ed at 23 ds.
$n + 13$ is ACK'ed at 72 ds.
Segment $n + 1$ is sent at 1 ds, ACK'ed at 11. Segment $n + 2$ is sent at 2 ds, ACK'ed at 12. Segment $n + 3$ is sent at 3 ds, ACK'ed at 13. Since segment n was lost, the three ACKs received are identical. Hence, segment n is retransmitted at 13 ds and ACK'ed at 23 ds along *with segments $n + 1$ to $n + 10$. Note that with fast recovery, $cwnd$ is only halved. Hence, when segments $n + 11$ to $n + 13$ are ready to be sent at 60 ds, they can be sent in a row at 60 ds, 61 ds and 62 ds. ACK for segment $n + 13$ is received at 72 ds.*

Exercise 3.4 *Consider the scenario in Figure 1.*

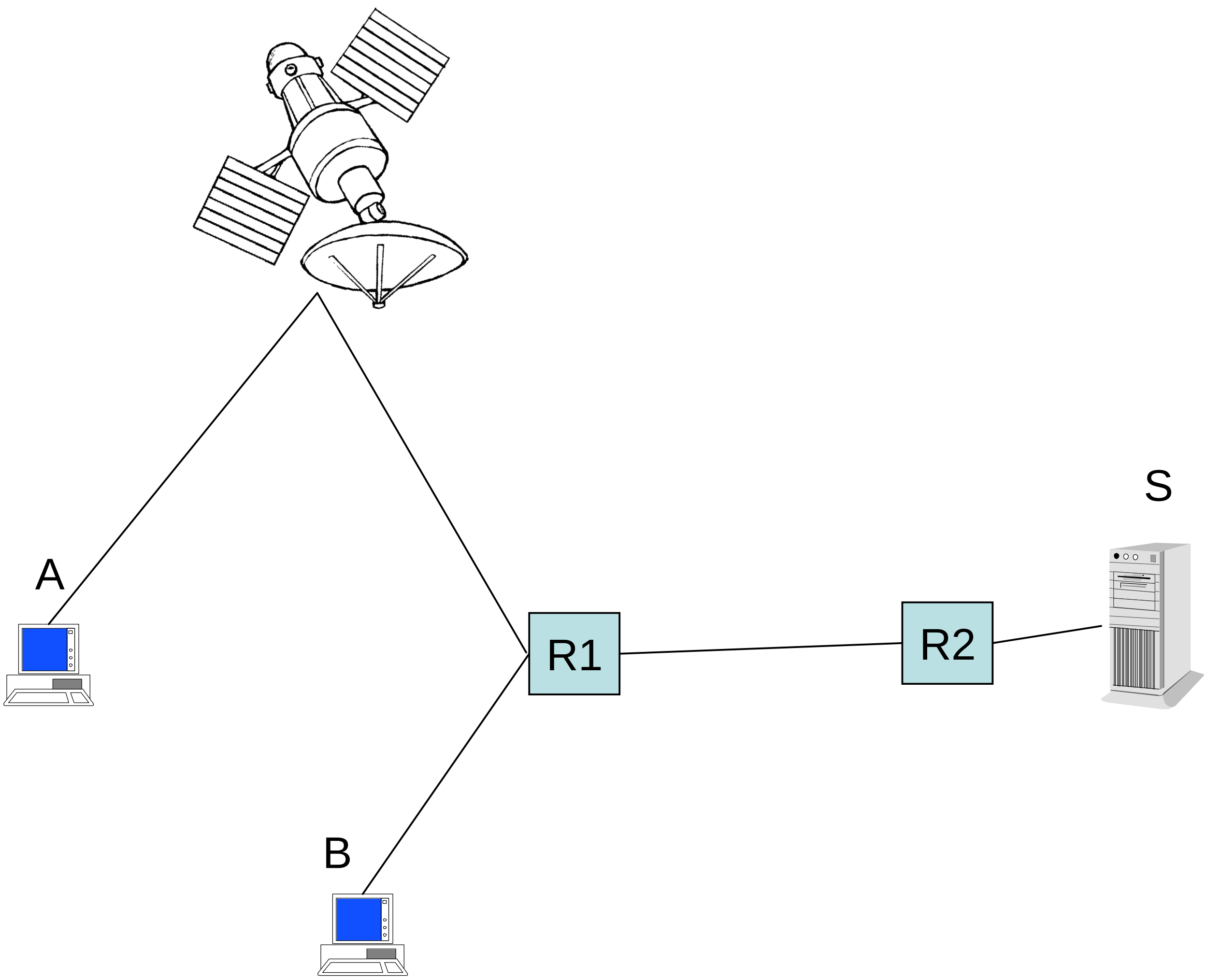

Figure 1: Network for exercise 3.4, step 1.

A and B each transfer a very large file over a TCP connection with server S. The link rates are:

- *between A and R1: 54 Mb/s in each direction*

- *between B and R1: 100 Mb/s in each direction*
- *between R1 and R2: 6 Mb/s in each direction*
- *between R2 and S: 1000 Mb/s in each direction*

Assume there is no other traffic than these two TCP connections. The RTT for A is 1000 ms, for B it is 200 ms.

1. *What would be the throughputs of A and B if they were max-min fair ? if they were proportionally fair ?*

   Solution:
   In both cases, by computing throughputs we have that each source gets 3 Mb/s.

2. *Assume we can neglect all losses except on the link R1-R2. Also assume that the two TCP connections can, together, fully utilize the link R1-R2 (i.e. the sum of their throughputs is 6 Mb/s). What are the throughputs achieved by A and B ?*

   Solution:
   By the loss-throughput formula, the throughputs are proportional to the round trip times, and their sum is 6Mb/s, thus they are: for A: 1 Mb/s; for B: 5 Mb/s.

3. *Assume now that the satellite link has a high loss ratio, due to transmission errors (FEC is disabled). The loss rate for the connection from A is now assumed to be fixed and equal to 0.01. We assume the RTTs stay the same, and, as before, that the two TCP connections can, together, fully utilize the link R1-R2 (i.e. the sum of their throughputs is 6 Mb/s). What are now the throughputs achieved by A and B ?*

   Solution:
   Take $L = 1500$ *bytes and* $C = 1.22$. *For A:* $\frac{L*C}{Ta*sqrt(0.01)} = 146$ *kbit/s; For B:* $6Mbit/s - 146$ *kbit/s=* 5854 *kbit/s.*

4. *Assume we change the configuration and introduce a transport layer gateway H, as shown on Figure 2.*

   There is now one TCP connection from A to H, a second one from H to S (and one from B to S as before). The RTT between H and S is 200 ms, between A and H is 800 ms and between B and S 200 ms as before. We assume FEC is enabled so that there is no loss between A and H. What is the end to end throughput of A and B ? Same question if FEC is disabled, so that the loss rate between A and H is again 0.01.

   Is it useful to use a transport layer gateway in this scenario ?

   Solution:
   FEC enabled: then the maximum throughput for the source at A will be the same as for source B. Indeed, on link A-H there is only one TCP connection, that can exploit the whole bandwidth. Also the connection H-S is identical to B-S, so that they will get the same bandwidth share, 3Mbit/s.
   FEC disabled:
   for A: $\frac{L*C}{0.8*sqrt(0.01)} Ra = (L/800)(C/sqrt(0.01)) = 183$ *kbit/s;*
   for B: $6Mbit/s - 183$ *kbit/s=* 5817 *kbit/s.*

   It is not very useful, as it guarantees a share of the bandwidth to the link with larger round trip delay which is only slightly higher.

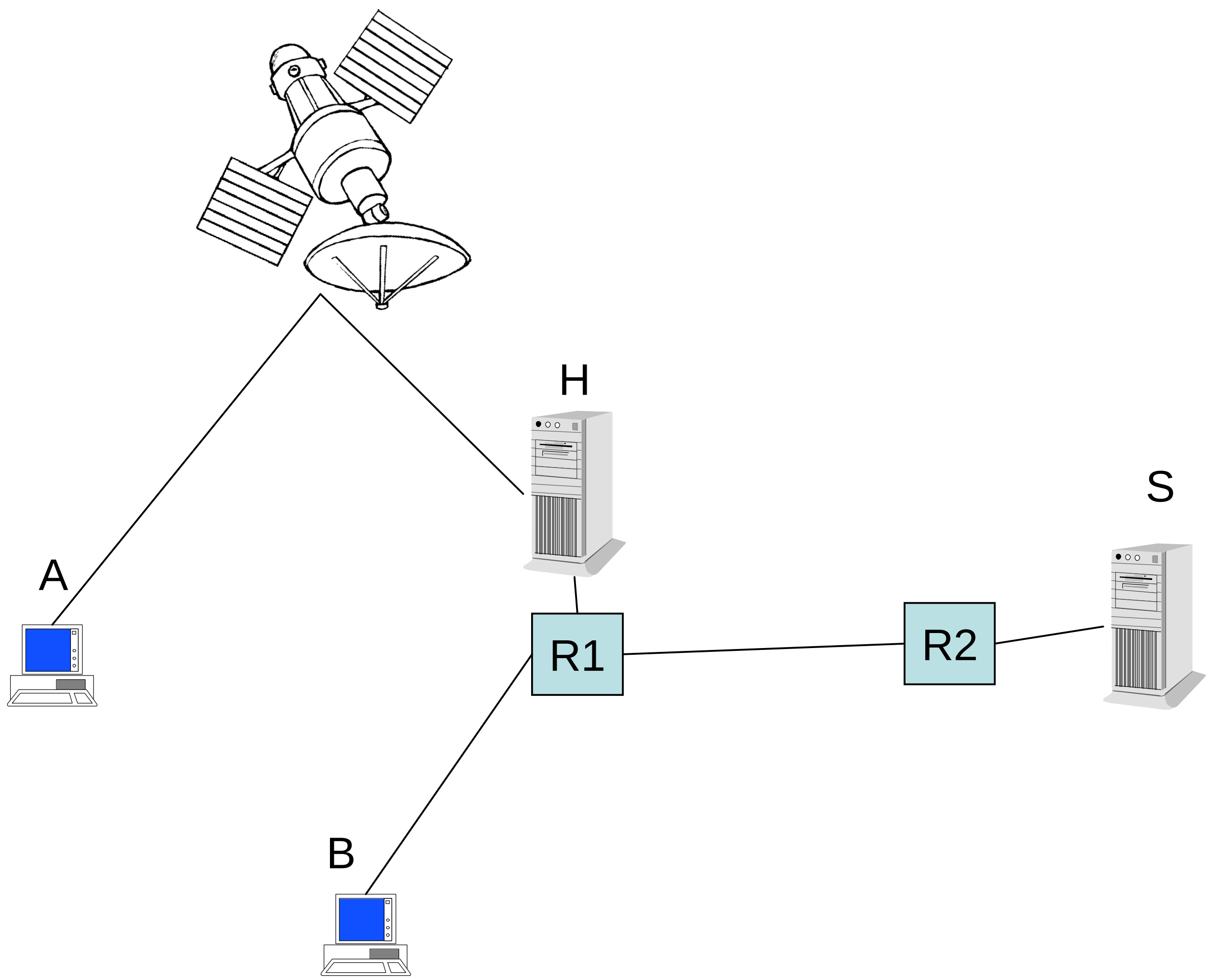

Figure 2: Network for exercise 3.4, step 2.

Exercise 3.5 *Assume that we change the code of TCP by modifying the congestion avoidance phase as follows. For every acknowledgement received, we increase the window size as in slow start. Give the pros and cons of this modification.*

Solution:

This is roughly equivalent to performing multiplicative increase by a factor of 2, instead of additive increase. But we know that this is bad because it will not provide fairness.

Pros: if I am the only one doing this, I am more aggressive and will gain higher throughput, at the expense of others.

Cons: if everyone does this, the network will not distribute rates according to any form of fairness. Connections that have a high share (because the network was lightly loaded when they started) will continue to have a higher share than others during periods of congestion.

Exercise 3.6 *Assume two hosts have a very large file to transfer, and decide to open n parallel TCP connections for the transfer.*

1. *Find one advantage and one drawback, for the user of the two hosts, of having $n > 1$. Justify your answer.*
   Solution:
   Advantage: *We know that TCP approximatively shares the network in order to maximize a utility function close to proportional fairness. By using n TCP connections, the user gets n shares instead of 1.*
   Now the result depends on where the bottleneck is, and with whom it is shared. We can distinguish two extreme cases.
   On one hand, if the bottleneck is a link of large capacity, where the user uses a small fraction of the link. by increasing the number of connections from 1 to n, we have a negligible impact on this link. The fair share for every user remains the same, thus in this case we obtain approximately n times more throughput than before.
   On the other hand, if the bottleneck is not shared, for example it is a modem line used only by this user. Every fair share will be $\frac{1}{n}$ of the capacity, so there is no benefit. As we discuss later, we in fact even obtain less in this case because we have more overhead.
   In summary, one advantage to the user is to obtain more of the network in the case where the limiting resource is shared with others.
   Drawback: *the overhead of managing the n connections (CPU, memory) can become quickly expensive with a growing n.*
2. *Find one advantage or one drawback, for the rest of the network, of having $n > 1$. Justify your answer.*
   Solution:
   Advantage: *there is no clear advantage for the network.*
   Drawback: *the user behaves unfairly and "steals" network resources.*