ABC: A Simple Explicit Congestion Controller for Wireless Networks

- Accelerate Brake Control (ABC) is an end-to-end congestion control algorithm that uses explicit congestion notifications to signal the end point to decrease or increase the sending rate. Instead of focusing on buffer capacity or drop rates, **ABC tries to match the sending rate to the dequeue rate of the bottleneck link**. Existing explicit schemes like XCP calculate their feedback based on enqueue rate and the link capacity. As a result, ABC can respond and adjust faster to changes in link capacity and hence the authors claim that it is better suited for wireless networks which tend to have frequent changes in link capacity.
- It is important to note the ABC router can be present at any point in the entire data path, and is responsible for regulating its own local link, i.e. it signals accelerate/brake based on the comparison its dequeue rate and the capacity of the link to the next hop. It doesn't do an estimation of the bottleneck capacity over the entire path like Sprout.
- PBECC, which also performs estimation of the capacity and number of users in LTE cell, to regulate the sending rate, however for PBECC to work the cellular layer needs to be last hop, as it needs access to the link layer of the client.
- Operational Principles:
  - The ABC Router performs estimation of the link capacity and the target rate (Usually the routers try to match the target rate to the capacity however they also take into account the queuing delay as well, if queueing delay is more than a threshold, they reduce the target rate in order so the queues can get empty.)
  - Based on the differences between current sending rate and target rate, routers will mark packets as accelerate or brake.
  - When senders receive an accelerate, it increases the window size by 1 and reduces it by 1 when receives a brake.
  - An ABC router is allowed to change a packet marked as accelerate to brake, however not vice-versa. As a result, even if one router marks the packet as brake, in the entire path, the packet will be reported as brake to the sender.
  - While the based operation of ABC is MIMD (Multiplicative increase Multiplicative decrease), in order to achieve fairness between different users the authors modify it slightly to be MAIMD (Multiplicative-Additive increase multiplicative decrease)
  - Since a non-ABC router can be bottleneck link, the ABC client maintains two congestion windows, one based on ABC and other based conventional TCP, and selects the minimum of the two in order to decide the overall window size.
- ABC uses the ECN bits to inform the sender whether to accelerate or brake. Authors says that using other fields in TCP header is not reliable due to middle boxes.
- An important question is, how will the sender know, which algorithm set the ECN bit --- ABC or XCP (say). This is not very clearly addressed in the paper.
- For co-existence between ABC and non-ABC flows, the authors propose that there is a separate queue of the two and weighted sharing of capacity between them. However it is not very clear, about how an intermediate router can differentiate/identify between the two.

- A key evaluation insight is comparison of ABC and XCP. While ABC outperforms traditional XCP for wireless networks, the authors identify that this is because traditional XCP is slow to adjust to link changes as it computes aggregate over the entire RTT to inform to the sender. A modified version of XCP, that calculates aggregate over smaller time, performs closer to ABC.
- ABC doesn’t take into account the RTT and hence suffers from RTT unfairness. This is similar to other algorithms that ignore RTT.