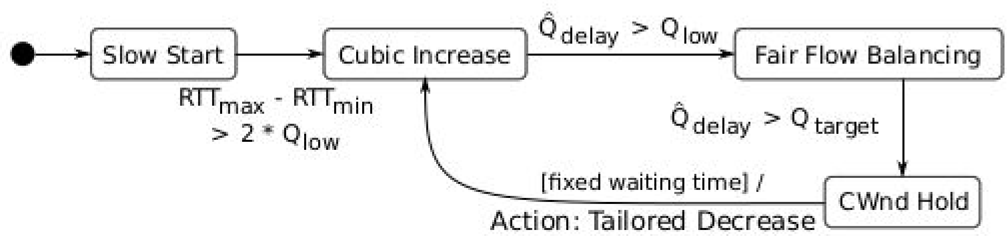
TCP LoLa

Congestion Control for
Low Latencies and High Throughput

What is TCP LoLa?

A **delay-based** congestion control algorithm that supports both, **low queuing delay** and **high network utilization** in high speed wide-area networks.

How TCP LoLa works?



RTT_{max} and RTT_{min}: maximal and minimal measured RTT respectively

Q_{low} and Q_{target}: threshold values

$\hat{Q}_{delay}$: queuing delay caused by the standing queue

Standing queue: a small queue deliberately created by TCP LoLa; Standing queue exists => overall amount of in-flight data is sufficient to fully utilize the bottleneck link

(a) Slow Start

TCP LoLa enters the slow start state after its initial start or after a retransmission timeout.

(b) Cubic Increase

Increase function used by TCP LoLa:

$$CWnd(t) = C \cdot (t - K)^3 + CWnd_{max}$$

t: time since last window reduction

C: unit-less factor (C = 0.4)

K: recalculated whenever CWnd has to be reduced

$CWnd_{max}$: size of CWnd before last reduction

TCP LoLa uses this function only if the potential bottleneck link is most likely not fully utilized, i.e., no standing queue is detected.

(c) Fair Flow Balancing

Basic idea: Each flow should keep a low but similar amount of data (X) in the bottleneck queue.

To keep the overall queuing delay between the thresholds Q_{low} and Q_{target}, X has to be dynamically scaled:

$$X(t)[Byte] = (\frac{t[ms]}{\phi})^3$$

t = 0 when fair flow balancing is entered; ϕ is a constant.

CWnd is adapted as follows:

$$\begin{array}{l} \textbf{if } \widehat{Q}_{data} < X(t) \textbf{ then} \\ \quad CWnd \leftarrow CWnd + (X(t) - \widehat{Q}_{data}) \\ \textbf{else} \\ \quad CWnd \leftarrow CWnd \\ \textbf{end if} \end{array}$$

$\hat{Q}_{data}$: amount of data the flow itself has queued at the bottleneck

Fair flow balancing requires that all competing flows enter and leave it at similar points in time => design puts a strong emphasis on synchronized state changes.

(d) CWnd Hold

In this state, the CWnd is unchanged for a fixed amount of time t_{sync} (default value = 250 ms). The hold time is necessary to ensure that all flows quit the current round of fair flow balancing.

(e) Tailored Decrease

Tailored decrease adjusts the CWnd reduction to the amount of congestion:

$$CWnd \leftarrow (CWnd - \widehat{Q}_{data}) \cdot \gamma$$

Each flow reduces its CWnd by $\hat{Q}_{data}$ – this should already empty the queue. CWnd is further reduced by the factor $\gamma < 1$ to ensure that the queue will actually be drained completely. To achieve this, K is calculated as follows:

$$K = \sqrt[3]{\left(CWnd_{max} - \widehat{RTT}_{min} \cdot \frac{CWnd_{max}}{\widehat{RTT}_{now}} \cdot \gamma\right) / C}$$

$R\hat{T}T_{min}$: RTT without any queuing delays

$R\hat{T}T_{now}$: RTT including the standing queue

How are queuing delay measurements done?

$$\widehat{RTT}_{now} = \min\{RTT(t_k) | t_k \in [t - t_{measure}, t]\}$$

RTT (t_k): an individual RTT measurement at time t_k

$t_{measure}$: a certain time interval independent of a flow's RTT

$$\widehat{RTT}_{min} \leftarrow \min(\widehat{RTT}_{min}, \widehat{RTT}_{now})$$
$$\widehat{Q}_{delay} = \widehat{RTT}_{now} - \widehat{RTT}_{min}$$

$$\widehat{Q}_{data} = \widehat{Q}_{delay} \cdot \frac{CWnd}{\widehat{RTT}_{now}}$$

The validity of $R\hat{T}T_{\text{min}}$ is checked after tailored decrease. $R\hat{T}T_{\text{min}}$ is reset, if no $R\hat{T}T_{\text{now}}$ value close to $R\hat{T}T_{\text{min}}$ has been measured for a certain number (e.g., 100) of tailored decreases.