Congestion control methods: A, B, C and D

Method A:

- if \( Q(t) = Q^* \) then \( \lambda(t + 1) \leftarrow \lambda(t) \)
- if \( Q(t) < Q^* \) then \( \lambda(t + 1) \leftarrow \lambda(t) + a \)
- if \( Q(t) > Q^* \) then \( \lambda(t + 1) \leftarrow \lambda(t) - a \)

where \( a > 0 \) is a fixed parameter

\[ \rightarrow \text{ linear increase and linear decrease} \]

Question: does it work?

Example:

- \( Q^* = 100 \)
- \( \gamma = 10 \)
- \( Q(0) = 0 \)
- \( \lambda(0) = 0 \)
- \( a = 1 \)
With $a = 0.5$:
With $a = 3$: 

![Load Evolution](image)

![Lambda Evolution](image)

![Gamma](image)
With $a = 6$: 

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{load_evolution.png}
\caption{Load Evolution}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{lambda_evolution.png}
\caption{Lambda Evolution}
\end{figure}
Remarks:

- Method A isn’t that great no matter what $a$ value is used
  -> keeps oscillating

- Actually: would lead to unbounded oscillation if not for physical restriction $\lambda(t) \geq 0$ and $Q(t) \geq 0$
  -> easily seen: start from non-zero buffer
  -> e.g., $Q(0) = 110$
With $a = 1$, $Q(0) = 110$, $\lambda(0) = 11$: 
Method B:

- if \( Q(t) = Q^* \) then \( \lambda(t + 1) \leftarrow \lambda(t) \)
- if \( Q(t) < Q^* \) then \( \lambda(t + 1) \leftarrow \lambda(t) + a \)
- if \( Q(t) > Q^* \) then \( \lambda(t + 1) \leftarrow \delta \cdot \lambda(t) \)

where \( a > 0 \) and \( 0 < \delta < 1 \) are fixed parameters

Note: only decrease part differs from Method A.

\[ \text{\( \rightarrow \)} \quad \text{linear increase with slope } a \]
\[ \text{\( \rightarrow \)} \quad \text{exponential decrease with backoff factor } \delta \]
\[ \text{\( \rightarrow \)} \quad \text{e.g., binary backoff in case } \delta = 1/2 \]

Similar to Ethernet and WLAN backoff

\[ \text{\( \rightarrow \)} \quad \text{question: does it work?} \]
With $a = 1$, $\delta = 1/2$: 

---

**Load Evolution**

**Target**

---

**Lambda Evolution**

**Gamma**

---
With $a = 3$, $\delta = 1/2$:
With $a = 1, \delta = 1/4$:
With $a = 1$, $\delta = 3/4$: 
Note:

• Method B isn’t that great either

• One advantage over Method A: doesn’t lead to unbounded oscillation

→ note: doesn’t hit “rock bottom”

→ due to asymmetry in increase vs. decrease policy

→ typical “sawtooth” pattern

• Method B is used by TCP

→ linear increase/exponential decrease

→ additive increase/multiplicative decrease (AIMD)

Question: can we do better?

→ what “freebie” have we not utilized yet?
Method C:

\[ \lambda(t + 1) \leftarrow \lambda(t) + \varepsilon(Q^* - Q(t)) \]

where \( \varepsilon > 0 \) is a fixed parameter

Tries to adjust magnitude of change as a function of the gap \( Q^* - Q(t) \)

\[ \rightarrow \text{ incorporate distance from target } Q^* \]

\[ \rightarrow \text{ before: just the sign (above/below)} \]

Thus:

- if \( Q^* - Q(t) > 0 \), increase \( \lambda(t) \) proportional to gap
- if \( Q^* - Q(t) < 0 \), decrease \( \lambda(t) \) proportional to gap

Trying to be more clever...

\[ \rightarrow \text{ bottom line: is it any good?} \]
With $\varepsilon = 0.1$: 

![Graph 1: Load Evolution](image1)

![Graph 2: Lambda Evolution](image2)
With $\varepsilon = 0.5$: 

![Graph 1: Load Evolution and Target](image1.png)

![Graph 2: Lambda Evolution](image2.png)
Answer: no

→ looks good

→ but looks can be deceiving

Time to try something strange

→ any (crazy) ideas?

→ good for course project (assuming it works)
Method D:

\[ \lambda(t + 1) \leftarrow \lambda(t) + \varepsilon(Q^* - Q(t)) - \beta(\lambda(t) - \gamma) \]

where \( \varepsilon > 0 \) and \( \beta > 0 \) are fixed parameters

\[ \rightarrow \quad \text{odd looking modification to Method C} \]

\[ \rightarrow \quad \text{additional term } -\beta(\lambda(t) - \gamma) \]

\[ \rightarrow \quad \text{what’s going on?} \]

Sanity check: at desired operating point \( Q(t) = Q^* \) and \( \lambda(t) = \gamma \), nothing should move

\[ \rightarrow \quad \text{check with methods A, B and C} \]

\[ \rightarrow \quad \text{fixed-point property} \]

\[ \rightarrow \quad \text{what about Method D?} \]

Now: does Method D get to the target fixed point?
With $\varepsilon = 0.2$ and $\beta = 0.5$: 

[Diagrams showing Load Evolution and Lambda Evolution]
With $\varepsilon = 0.5$ and $\beta = 1.1$:
With $\varepsilon = 0.1$ and $\beta = 1.0$: 

\begin{figure}
\centering
\includegraphics[width=\textwidth]{load_plot.png}
\caption{Load Evolution}
\end{figure}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{lambda_plot.png}
\caption{Lambda Evolution}
\end{figure}
Remarks:

- Method D has desired behavior
- Is superior to Methods A, B, and C
- No unbounded oscillation
- In fact, dampening and convergence to desired operating point
  \[ \rightarrow \text{converges to target operating point } (Q^*, \gamma) \]
  \[ \rightarrow \text{asymptotic stability} \]
Why does it work?

What is the role of the $-\beta(\lambda(t) - \gamma)$ term in the control law:

$$\lambda(t + 1) \leftarrow \lambda(t) + \varepsilon(Q^* - Q(t)) - \beta(\lambda(t) - \gamma)$$

Need to look beneath the hood ...
Visualize action in 2-D \((Q(t), \lambda(t))\)-space:

\[
\begin{align*}
\lambda(t) > \gamma & \quad Q(t) < Q^* \\
\lambda(t) > \gamma & \quad Q(t) > Q^*
\end{align*}
\]

\[
\begin{align*}
\lambda(t) < \gamma & \quad Q(t) < Q^* \\
\lambda(t) < \gamma & \quad Q(t) > Q^*
\end{align*}
\]
Convergent trajectory:

\[ \rightarrow \text{ asymptotically stable \& optimal} \]
Divergent trajectory:

\[ \rightarrow \text{ unstable} \]
Stable (but not asymptotically so) trajectory:

\[ \rightarrow \text{ limit cycle} \]
Which case arises depends on the specifics of protocol actions.

For example:

- Methods A and C: divergent
- Method B: stable (but not asymptotically)
  \(\rightarrow\) TCP
- Method D: asymptotically stable & optimal
  \(\rightarrow\) “optimal control”

Why does Method D work:

\(\rightarrow\) overview of underlying mathematics