Convergence to optimal solution of (P) can be shown by using Lyapunov function approach (Wu & Arak 04)

**Relationship to TCP-reno**

Window evolution for congestion avoidance in TCP-reno can be approximated as

\[
W(t+\delta) \approx \frac{A(t,t+\delta)}{W(t)} + \left(\frac{1}{e}\right)^{\frac{N(t,t+\delta)}{W(t)}} = e^{-\ln 2 N(t,t+\delta)} \\
\approx 1 - \ln 2 N(t,t+\delta)
\]

Where \( A(t,t+\delta) = \# \text{ACKs in } (t,t+\delta) \)
\( N(t,t+\delta) = \# \text{losses in } (t,t+\delta) \)

→ Continuous time approximation for loss probability \( p(t) \), small \( d \)
\[
\frac{dW}{dt} = \frac{r(t)(1-q(t))}{W(t)} - \beta r(t) q(t) W(t)
\]
where \( \beta \approx \ln 2 \)

⇒ \[
\frac{dr}{dt} = \frac{r(t)(1-q(t))}{d^2 r(t)} - \beta r(t)^2 q(t)
\]

- for small \( q(t) \),
\[
\frac{dr}{dt} \approx \frac{1}{d^2} - \beta r^2 q(t)
\]
\[
= \beta r^2 \left( \frac{1}{\beta d^2 r^2} - q(t) \right)
\]

- from (1),
\[
\frac{\partial U}{\partial r} = \frac{1}{\beta d^2 r^2}
\]
\[
U(r) = -\frac{1}{\beta d^2 r} + \text{const}
\]

\( q_\sigma \rightarrow \text{path loss probability for } \sigma \rightarrow \)
\[ q_\sigma = 1 - \prod_{k \in L_\sigma} (1 - p_k) \]
\[ = \sum_{k \in L_\sigma} p_k \text{ for small } p_k, \quad q_\sigma \]
Queueing disciplines

- **Queueing discipline**: governs how packets are buffered & selected for transmission at a router/switch
  
  a) **Drop policy**: determines which packets are dropped
    
    - **Tail drop**: any packet that arrives when the buffer is full is dropped.
      - Simplest & most common drop policy
    
    - **Random early detection (RED)**: packets are dropped (or marked) with a probability that is a function of the average queue length

    RED probability profile

    ![RED probability profile graph]

    - Breaks synchronization among TCP flows, maintains small queues
b) **scheduling policy**: determines the order in which queued packets are selected for transmission

- *first-in-first-out (FIFO)* or *first-come-first-served (FCFS)*
  - packets are transmitted in the same order as they arrived

- **priority queueing**
  - packets arriving at the output link are classified into priority classes, each typically with its own FIFO queue
  - packets are transmitted from the highest priority class with a nonempty queue
  - used in the Internet to prioritize routing updates

- **fair queueing / round robin**
  - maintain a separate queue for each flow / class
  - scheduler rotates service among the flows in a fixed sequence, equally
- when a queue reaches a particular length, additional packets from the corresponding flow are discarded
- segregates traffic so that an ill-behaved flow not following the congestion control algorithm cannot arbitrarily increase its bandwidth at the expense of other flows

- **Weighted Fair queuing (WFQ)**
  - similar to fair queuing in that classes/flows are served in a circular sequence, but they receive different amounts of service according to their weight
  - useful for QoS

- Detailed description of FQ & WFQ in terms of fluid version called **Generalized Processor Sharing (GPS)**

- FQ, WFQ & GPS are work-conserving, i.e. link is not left idle as long as there is at least 1 nonempty queue