

# Forgalomszabályozás az Interneten (1)

Sonkoly Balázs  
[sonkoly@tmit.bme.hu](mailto:sonkoly@tmit.bme.hu)

---

2016.10.25.

# Áttekintés

- TCP áttekintése
  - egy működő protokoll
  - torlódásszabályozás
- Értsük meg, mit csináltunk
  - matematikai modellek (utólag)
  - probléma megfogalmazása
- Lehetséges eszközök
  - folytonos idejű visszacsatolt rendszer
  - szabályozástechnika

# Forgalomszabályozás

- Sok területen jön elő a hálózatok világában (is)
- például a transzport rétegben
- egyik legfontosabb protokoll: TCP (Transmission Control Protocol)
- Probléma:
  - a küldő határozza meg az adási sebességet a vevő felé
  - sok küldő használja a közös hálózati erőforrásokat (linkek, routerek, switch-ek, bufferek, ...)
  - ki milyen sebességgel adjon, hogy “optimálisan” használjuk a hálózatot?
  - ne árasszuk el / terheljül túl a vevőket
  - ne árasszuk el / terheljül túl a hálózati csomópontokat

# TCP

- Transzport protokoll (szállítási réteg)
- TCP “szolgáltatás” tulajdonságai
  - hoszt-hoszt (end-to-end) átvitel
  - full-duplex
  - megbízható, sorrendhelyes, duplikáció mentes átvitel
  - megbízhatatlan IP (datagram) hálózat felett
  - multiplexelés az alkalmazási réteg számára (portok)
  - kapcsolat alapú
  - **forgalomszabályozás**
    - flow control
    - congestion control

# TCP átvitel elemei

- TCP retransmit
  - megbízható átvitel kell IP hálózat felett
  - garantálni kell az elveszett csomagok újraküldését
  - nyugtázási mechanizmus kell (ACK)
- TCP flow control
  - vevő oldal védelme
  - ne árasszuk el a vevő oldali buffereket
- TCP congestion control
  - ne terheljük túl a közös hálózati erőforrásokat (switch-ek, routerek, bufferek, linkek ...)
  - használjuk hatékonyan / optimálisan a hálózati erőforrásokat
  - kerüljük el a “*torlódási összeomlást*” (“congestion collapse”)
  - V. Jacobson, 1988

# TCP retransmit: megbízhatóság

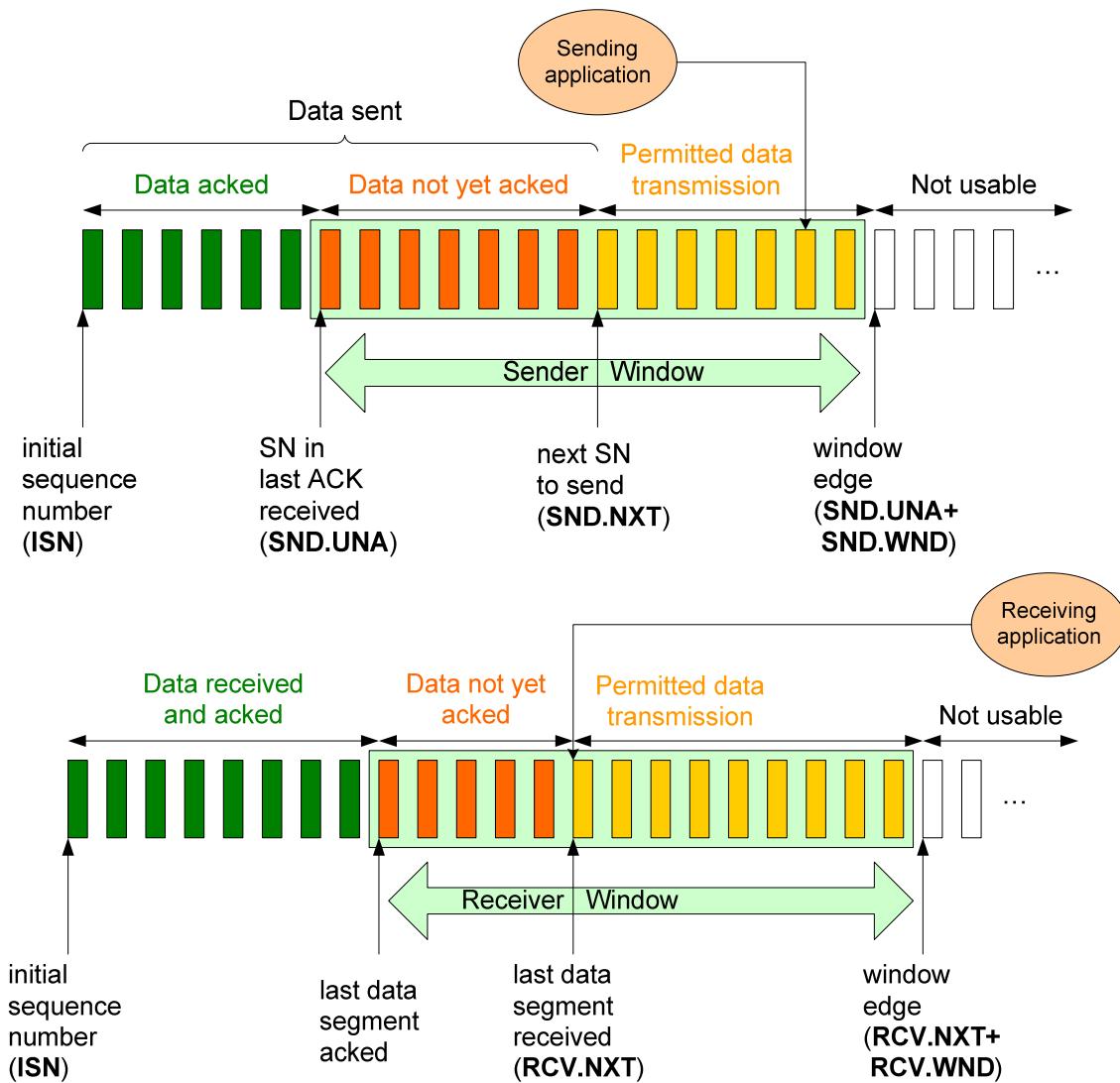
## ■ Probléma

- csomagvesztés } → eldob és újraküld
- csomaghiba } → eldob
- duplikátum } → helyreállítás sorszám alapján
- sorrendcsere }

## ■ Csomagvesztés érzékelése az adó oldalon

- **timeout**
  - ha RTO (Retransmission TimeOut) lejár → újraküldés
  - komoly torlódást jelez
- **dup ACK**
  - 3 duplikált nyugta (dup ACK) vétele → újraküldés (*fast retransmit*)
  - kevésbé komoly torlódás (a vesztés után érkeztek csomagok)

# Csúszóablakos szabályozás



- Mechanizmus: sliding window
- TCP legtöbb szolgáltatása ennek segítségével megvalósítható
- Koncepció:
  - szabályozás az adó oldalon (**smart sender**)
  - egyszerű vevő, csak nyugtákat küld (**dump receiver**)

# TCP congestion control alapok (1)

## ■ Alapelvek

- best-effort hálózatot feltételezünk (IP)
- minden forrás maga határozza meg / számítja a hálóztaik kapacitást
- implicit visszacsatolás alapján
- self-clocking

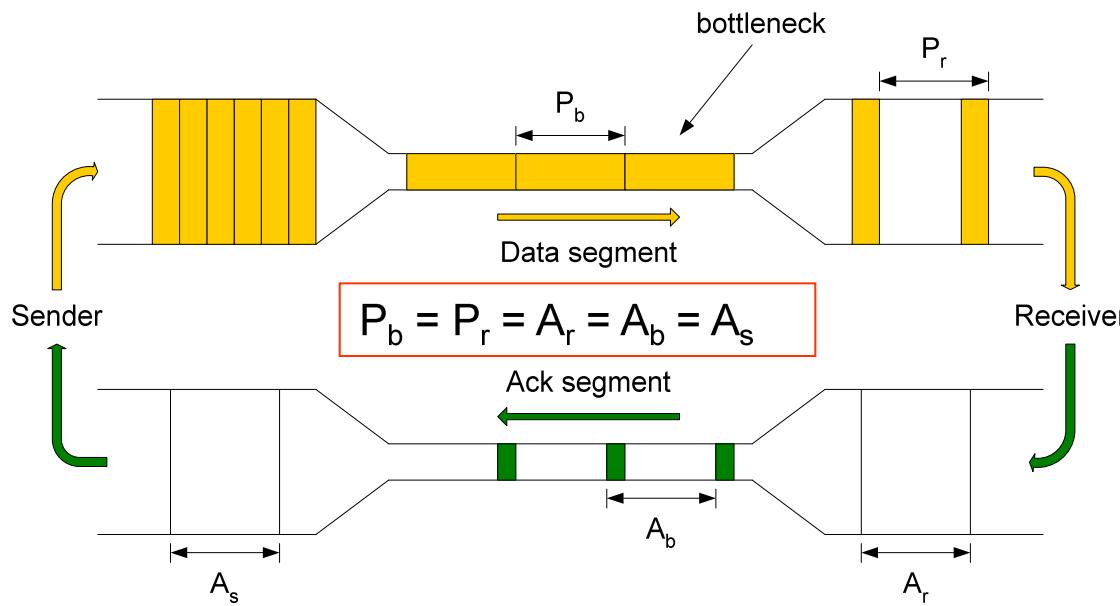
## ■ Cél

- elérhető kapacitás automatikus meghatározása

# TCP congestion control alapok (2)

## ■ Self-clocking

- packet conservation: ne küldjünk új csomagot a hálózatba, amíg egy másik el nem hagyja azt (egyensúlyi állapotban)
- küldő a nyugták érkezési üteme szerint küldi ki az új csomagokat
- vevő nem tud gyorsabban nyugtát generálni, mint ahogy az adatcsomag keresztül jut a hálózaton
- az adás automatikusan igazodik a sávszélesség és késleltetés változásaihoz

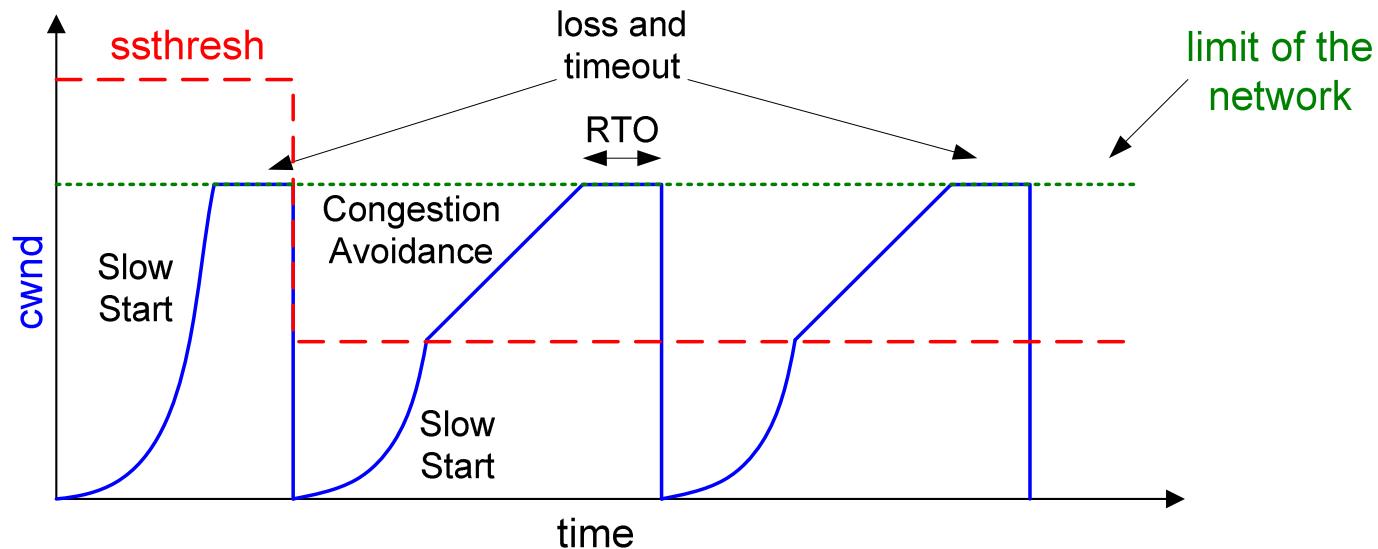


- függőleges dim: sávszélesség (BW)
- vízszintes dim: idő (T)
- $BW \times T = \text{Bits (terület)}$
- “interarrival time” marad!
- TCP sending rate
  - bejövő ACK-ok határozzák meg
  - amit a bottleneck határoz meg!

# TCP congestion control mechanizmusai

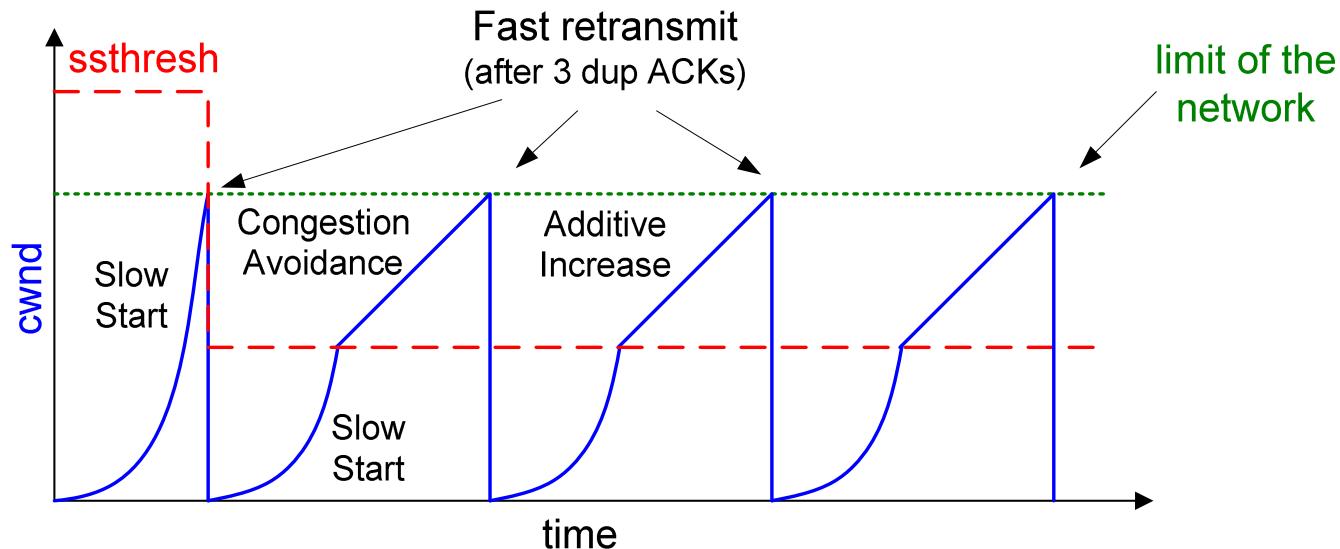
- Csúszóablakos szabályozás
  - flow control és congestion control is ez alapján
  - TCP adó nem tudja, hol a bottleneck:  
vevőnél vagy a hálózatban
  - congestion window (**cwnd**)
    - dinamikusan frissített változó
    - ami meghatározza az adási sebességet
- Négy fő működési fázis
  - (különböző algoritmusok a cwnd szabályozására)
  - Slow start
  - Congestion Avoidance
  - Fast retransmit
  - Fast recovery

# Congestion avoidance – RFC 1122



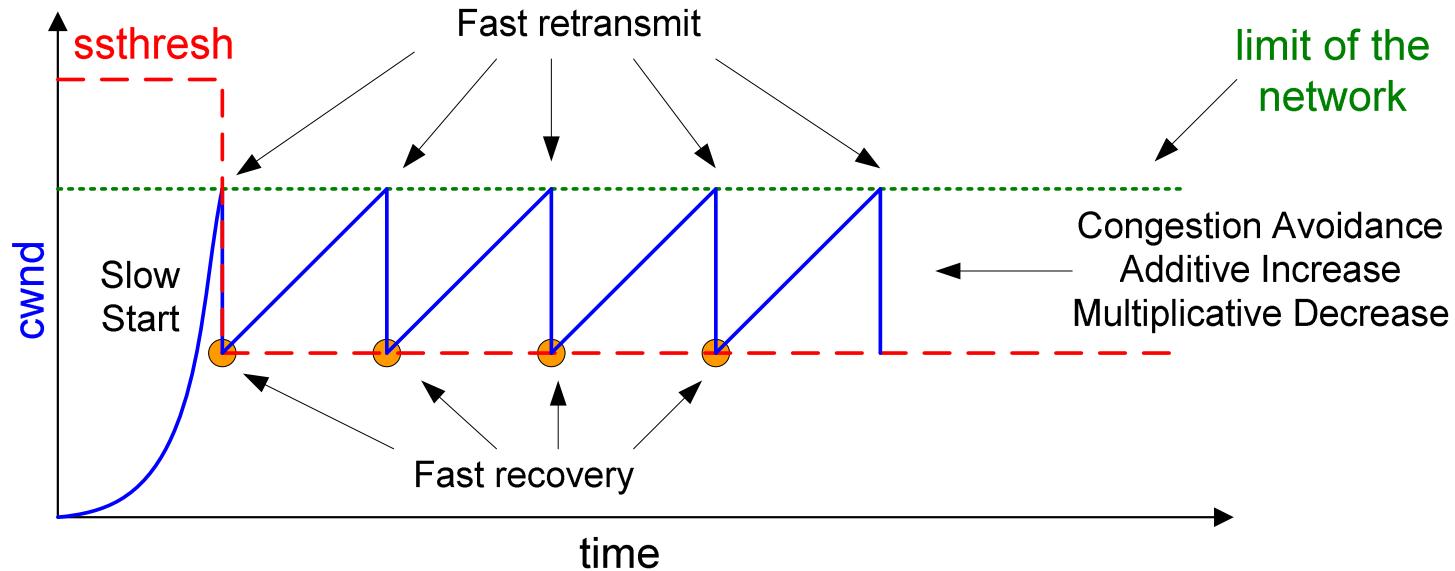
- Célok (RFC 1122)
  - tartsuk a cwnd-t az optimum környékén amennyire csak lehet
  - Slow start
    - indulásnál növeljük gyorsan a cwnd-t, amivel gyorsan elérhetünk egy maximálisan biztonságos adási sebességet
    - max. biztonságos: fele annak az értéknek, ami csomagvesztést okozott (konzervatív!)
  - Congestion avoidance
    - növeljük lassan a cwnd-t, hogy elkerüljük a csomagvesztést, amíg csak lehet

# Fast retransmit – TCP Tahoe



- TCP Tahoe
  - slow start + congestion avoidance
  - + fast retransmit (3 dup ACK után)
- TCP Tahoe problémái
  - fast retransmit után tudjuk, hogy torlódás történt
  - DE slow start-ra visszalépni túl konzervatív
  - tudjuk, hogy későbbi csomagok megérkeztek
  - nagyon érzékeny a csomagvesztésre (1%-os csomgvesztési arány: 50-75% throughput visszaesés!)
- Megoldás: két különböző típusú torlódás
  - RTO lejár → komoly torlódás
  - 3 dup ACK → nincs komoly torlódás (legalább 3 csomag megérkezett)

# Fast recovery – TCP Reno



- TCP Reno
  - első implementáció: 4.3 BSD Reno, Net/2, ~1990
  - Slow start
  - Congestion avoidance: **AIMD** (Additive Increase Multiplicative Decrease)
  - Fast retransmit
  - Fast recovery
- Problémák
  - következő előadás...

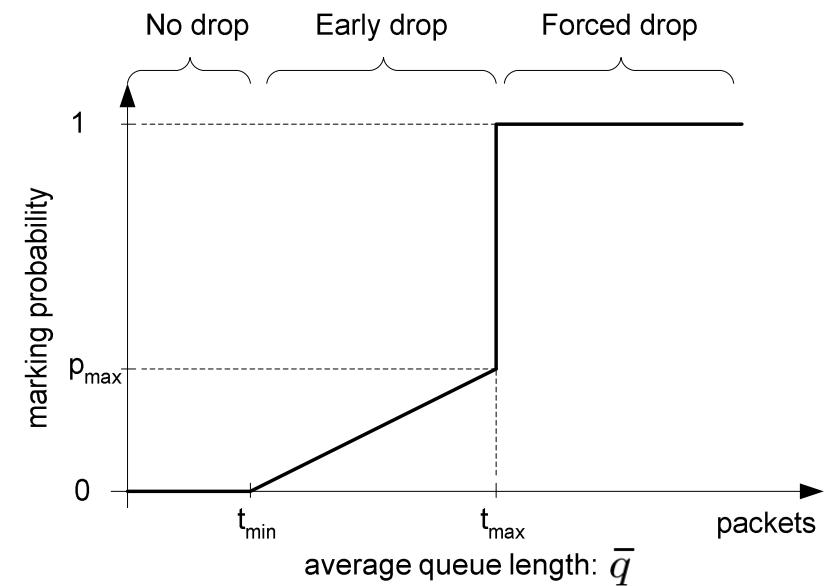
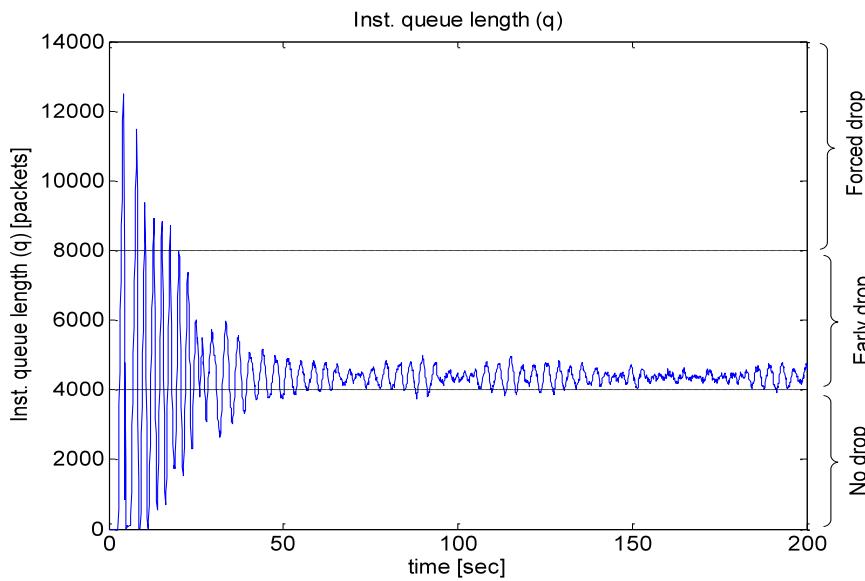
# TCP Reno – summary

- When **cwnd** is below **ssthresh**, sender in slow-start phase, window grows exponentially
- When **cwnd** is above **ssthresh**, sender is in congestion-avoidance phase, window grows linearly
- When a triple duplicate ACK occurs, **ssthresh** set to **cwnd/2** and **cwnd** set to **ssthresh** (fast retransmit, fast recovery then congestion avoidance)
- When timeout occurs, **ssthresh** set to **cwnd/2** and **cwnd** is set to 1 segment (**slow-start**)

# Torlódás elkerülés a routerekben

## ■ RED: Random Early Detection

- active queue management (AQM) mechanizmus
- cél: torlódás megelőzése a routerekben
- random csomagdobás (burst-ös helyett) mielőtt megtelik a várakozási sor
- dropping rate  $\sim$  flow rate
- elkerülhető a globális szinkronizálódás



# Áttekintés

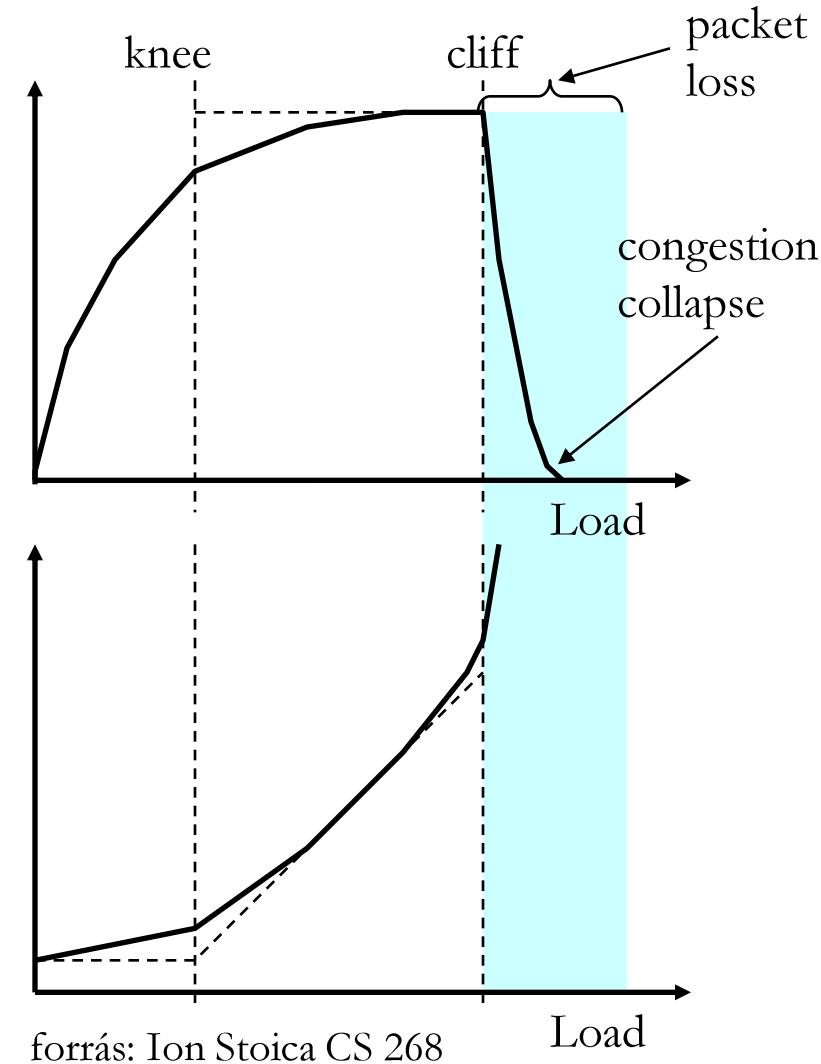
- TCP áttekintése
  - egy működő protokoll
  - torlódásszabályozás
- Értsük meg, mit csináltunk
  - matematikai modellek (utólag)
  - probléma megfogalmazása
- Lehetséges eszközök
  - folytonos idejű visszacsatolt rendszer
  - szabályozástechnika

# Congestion control – általánosan

## ■ Célok

- operátorok oldaláról
  - hálózati elemek túlterhelésének elkerülése
  - DE használjuk hatékonyan az erőforrásokat
  - “**knee-point**” belövése (load-throughput görbén)
  - stabil hálózati működés
- felhasználók oldaláról
  - hatékonyság
  - fairness!

## ■ hatékonyság ↔ fairness



# Congestion control – módszerek

## ■ Ha a hálózat is segít

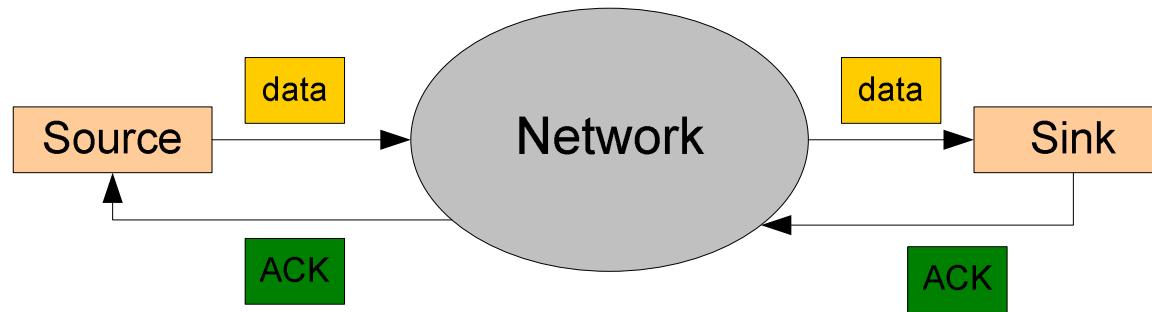
- explicit congestion control
- explicit visszajelzés a routerektől (pl. egy bit beállítása torlódás esetén – ECN, de kifinomultabb módszerek is vannak – XCP, RCP)
- fő probléma: módosítani kell a routereket

## ■ Ha a hálózat nem segít

- NINCS explicit visszajelzés
- következtetni kell implicit torlódási mérőszámokból
  - csomagvesztés → loss-based mechanizmusok (pl. TCP Reno)
  - késleltetés → delay-based mechanizmusok (pl. TCP Vegas, FAST)
    - round-trip time (RTT): csomag kiküldése – ACK vétele
  - kombinált loss- and delay-based mechanizmusok (pl. Compound TCP)
  - bandwidth measurement (pl. TCP Westwood)
    - aktív / passzív mérések

# Congestion control → “szabtech”

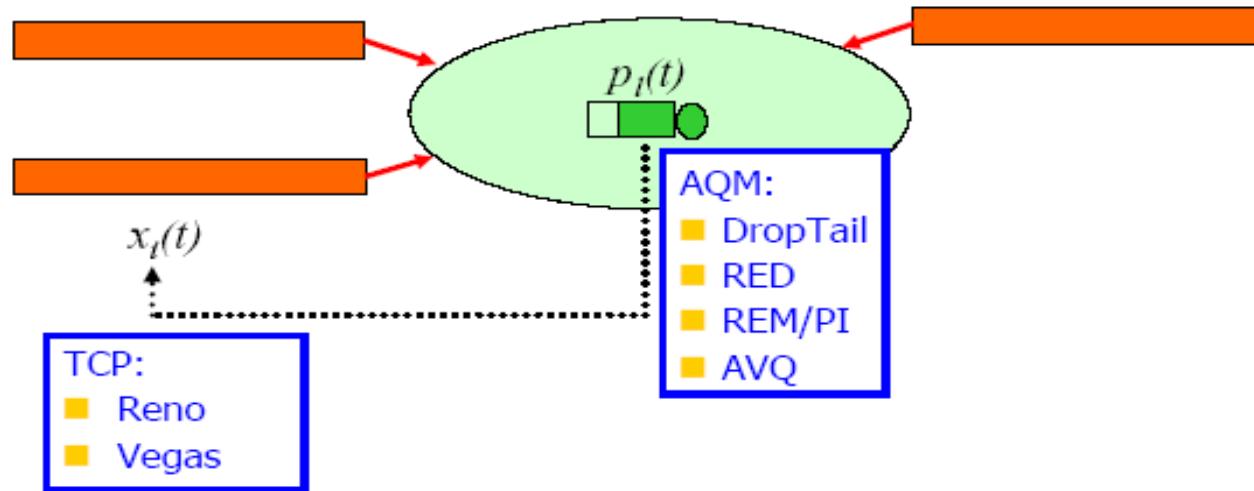
- End-to-end congestion control az Interneten
  - (hatalmas) visszacsatolt rendszer
  - (szabályozástechnika!)
  - elosztott (distributed)
  - nem elhanyagolható késleltetések (delayed)
  - adók:
    - adási sebesség beállítása
    - a mért torlódási jelek alapján



# Modellek és eszözök

- Csomagszint
  - sztochasztikus modellek
  - queueing theory
  - kevés jól használható eredmény visszacsatolás esetén
- Folyamszint
  - adott időskálán érvényes
  - determinisztikus folyadékmodellek
  - eszközök
    - optimization-theory
    - control-theory
    - game-theory
- Vizsgáljuk meg a meglévő protokollokat!

# TCP & AQM



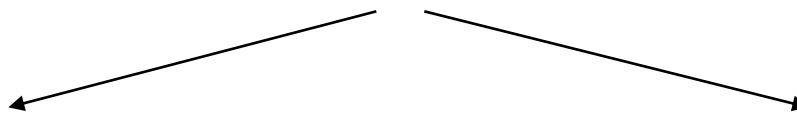
- Transzport protokollok
  - küldési ráta ( $x_i(t)$ )
  - rendszer bemeneti paramétere
- Active Queue Management (AQM)
  - torlódási mérőszám ( $p_l(t)$ ) (pl. loss rate, delay,...)
  - visszacsatolt jel

# TCP & AQM

## TCP – AQM modell:

$$x(t+1) = F(p(t), x(t))$$

$$p(t+1) = G(p(t), x(t))$$



### **Egyensúlyi állapot**

- throughput, loss, delay,...
- fairness
- utility

### **Duality theory (optimization)**

- sending rates → primal variables
- cong. measure → dual variables
- flow / cong. control → optimization

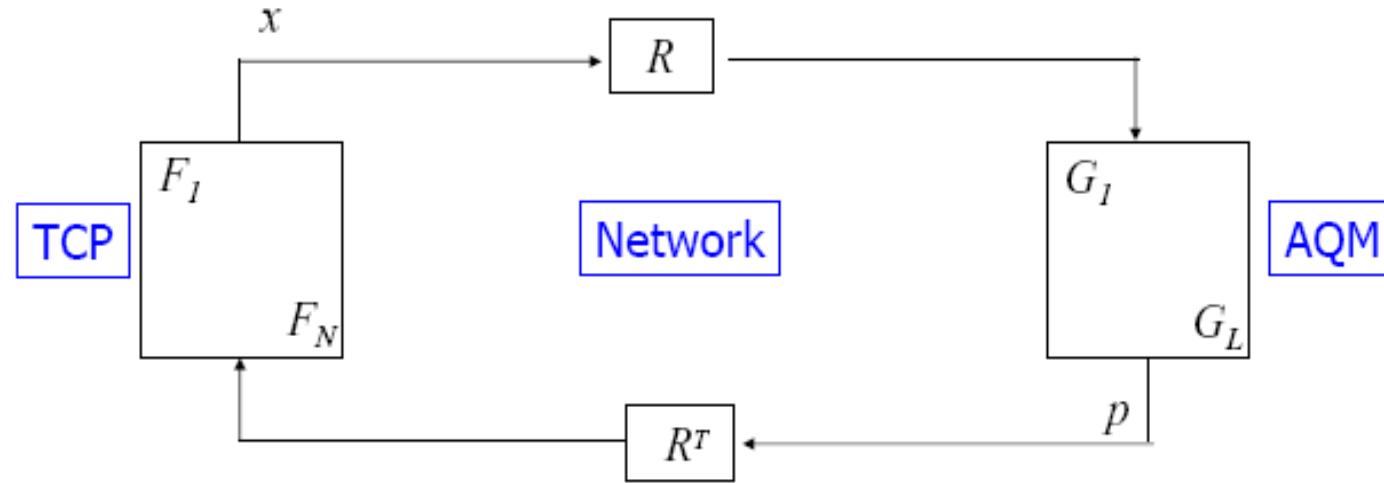
### **Dinamikus viselkedés**

- lokális stabilitás
- globális stabilitás

### **Control theory**

- feedback system
- distributed
- delayed

# TCP & AQM



$R_{li} = 1$  if source  $i$  uses link  $l$

$$x(t+1) = F(R^T p(t), x(t))$$

$$p(t+1) = G(p(t), Rx(t))$$

IP routing

Reno, Vegas

DT, RED, ...

$[R_f(s)]_{li}$  forward delays  
 $[R_b(s)]_{li}$  backward delays

# Áttekintés

- TCP áttekintése
  - egy működő protokoll
  - torlódásszabályozás
- Értsük meg, mit csináltunk
  - matematikai modellek (utólag)
  - probléma megfogalmazása
- Lehetséges eszközök
  - folytonos idejű visszacsatolt rendszer
  - szabályozástechnika

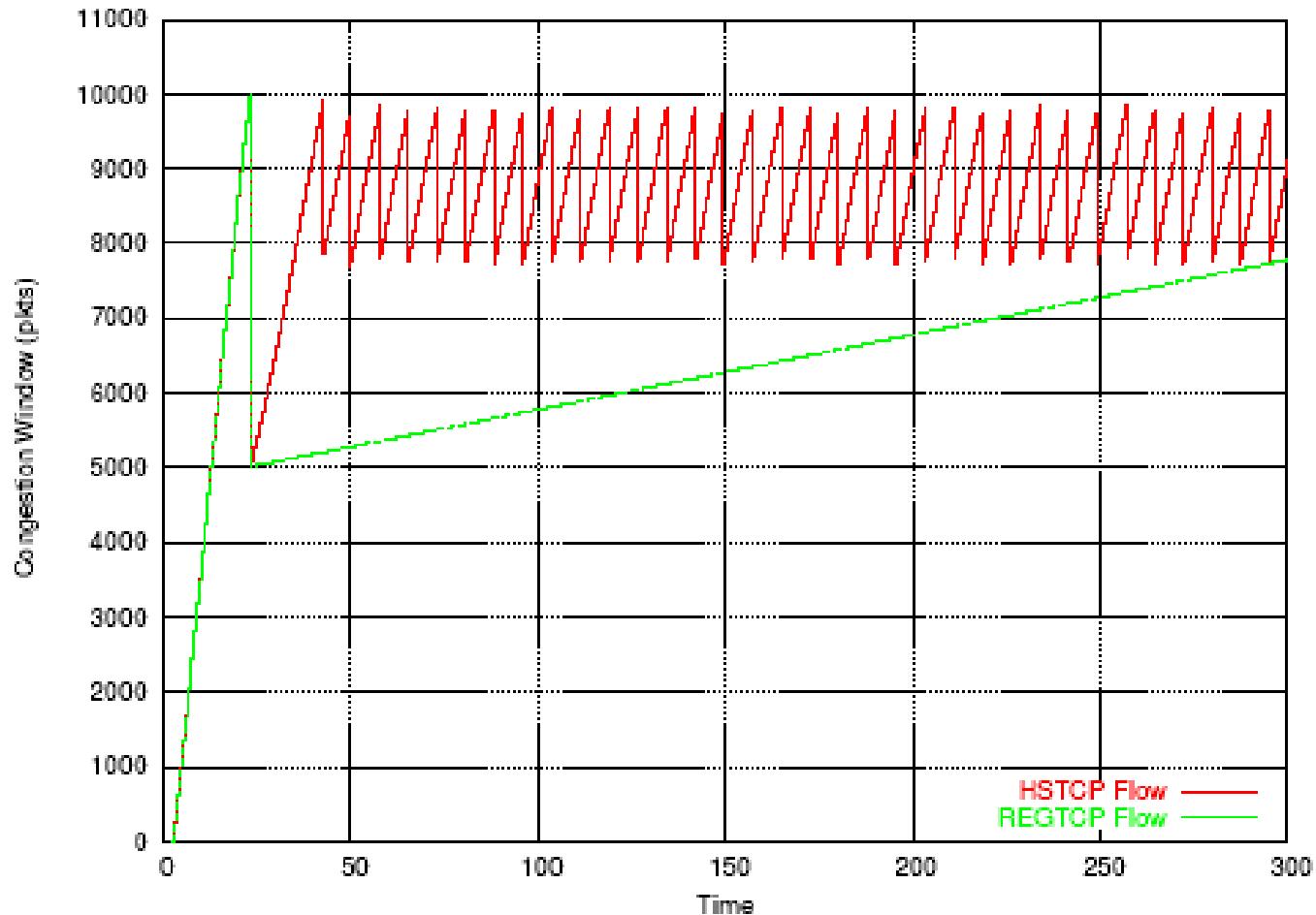
# Egy gyakorlati példa

- HighSpeed TCP / RED hálózatok vizsgálata
- (szabályozástechnika) control-theory módszereivel

# HighSpeed TCP (1)

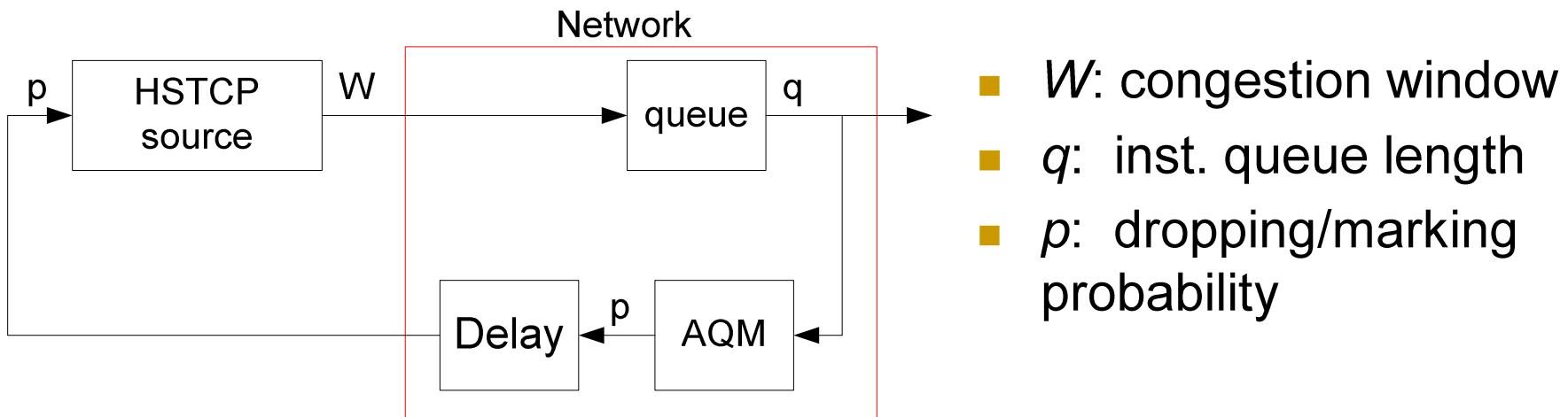
- TCP Reno egyszerű javítása nagysebességű környezethez
- RFC 3649: HighSpeed TCP for Large Congestion Windows (Sally Floyd, 2003)
- csomagvesztés alapú protokoll
- módosított AIMD mechanizmus
  - cwnd növelésének és csökkentésének mértéke az aktuális cwnd értéktől függ
  - “skálázódás”
  - nagy torlódás esetén → TCP Reno szerű működés
  - különben → agresszívabb kontroll

# HighSpeed TCP (2)



# Control-theoretic model

- Modeling network with single bottleneck link
- Plant: HSTCP sources, queue
- Controller: Active Queue Management (e.g., RED)
- Feedback control system



# Fluid-flow model of HSTCP/RED networks

## ■ Notations

- |       |          |   |
|-------|----------|---|
| $W$   | $\doteq$ | expected TCP window size (packets),                 |
| $q$   | $\doteq$ | expected queue length (packets),                    |
| $x$   | $\doteq$ | expected queue length estimation (packets),         |
| $R$   | $\doteq$ | round-trip time (RTT) = $\frac{q}{C} + T_p$ (secs), |
| $C$   | $\doteq$ | link capacity (packets/sec),                        |
| $T_p$ | $\doteq$ | fix round-trip propagation delay (secs),            |
| $N$   | $\doteq$ | load factor (number of TCP sessions),               |
| $p$   | $\doteq$ | probability of packet mark/drop.                    |

# Fluid-flow model of HSTCP/RED networks

- Dynamics can be described by differential equations
- expected (average) transient behavior is captured
- **dynamics of HSTCP source:**

$$\dot{W}(t) = \frac{a(W(t))}{R(t)} - b(W(t)) W(t) \frac{W(t - R(t))}{R(t - R(t))} p(t - R(t))$$

- first term: Additive Increase
  - $W$  is increased by  $a(W(t))$  for one RTT
  - “positive” ACKs arrived at a rate proportional to  $1-p(t) \approx 1 \rightarrow$  not lost packets
- second term: Multiplicative Decrease
  - $W$  is decreased by  $bW(t)$
  - at a rate proportional to
    - $p(t) \rightarrow$  lost packets (loss ratio)
    - $x(t-R) = w(t-R)/R(t-R) \rightarrow$  sending rate at one RTT earlier

# Fluid-flow model of HSTCP/RED networks

- **dynamics of bottleneck queue** → differential equation

$$\dot{q}(t) = N(t) \frac{W(t)}{R(t)} - 1_{q(t)} C$$

- first term
  - sending rate of N identical HSTCP sources
  - aggregate arriving traffic
- second term
  - serving rate
  - according to the link capacity (if the buffer is not empty)
- **queue estimation** (exponentially weighted moving average)

$$\dot{x}(t) = -Kx(t) + Kq(t)$$

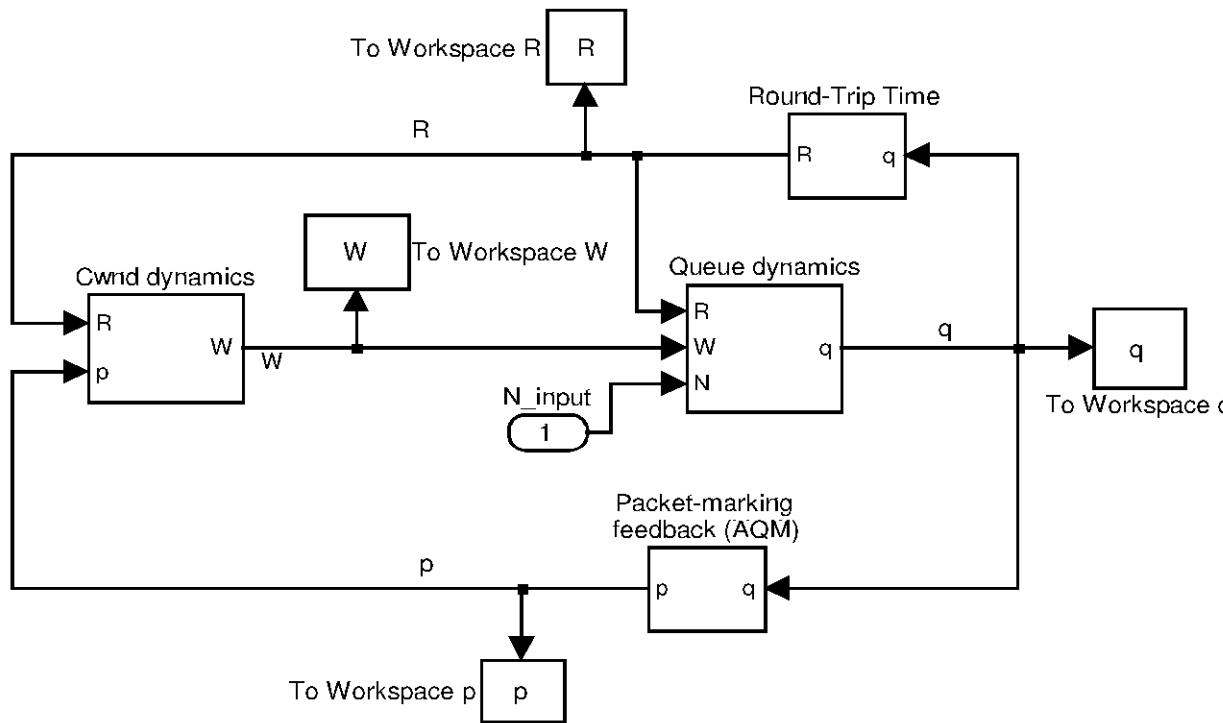
- low-pass filter
- with a cut-off frequency of  $K$ , where  $K = -C \ln(1 - \alpha)$
- $\alpha$  is the forgetting factor

# Fluid-flow model of HSTCP/RED networks

- Dynamics of
  - congestion window (HSTCP source)
  - bottleneck queue
  - queue estimation (at RED)
- coupled differential equations
- complex dependence between the variables
- delay in some arguments
- moreover: variable delay in some arguments
- model is analytically not tractable
- numerical approximations can be applied to solve the equations

# Simulink framework

- MATLAB/Simulink is a suitable framework
- system models can easily be implemented
- numerical approximation based solvers can be applied
- here: Domand-Prince algorithm (“ode45”)

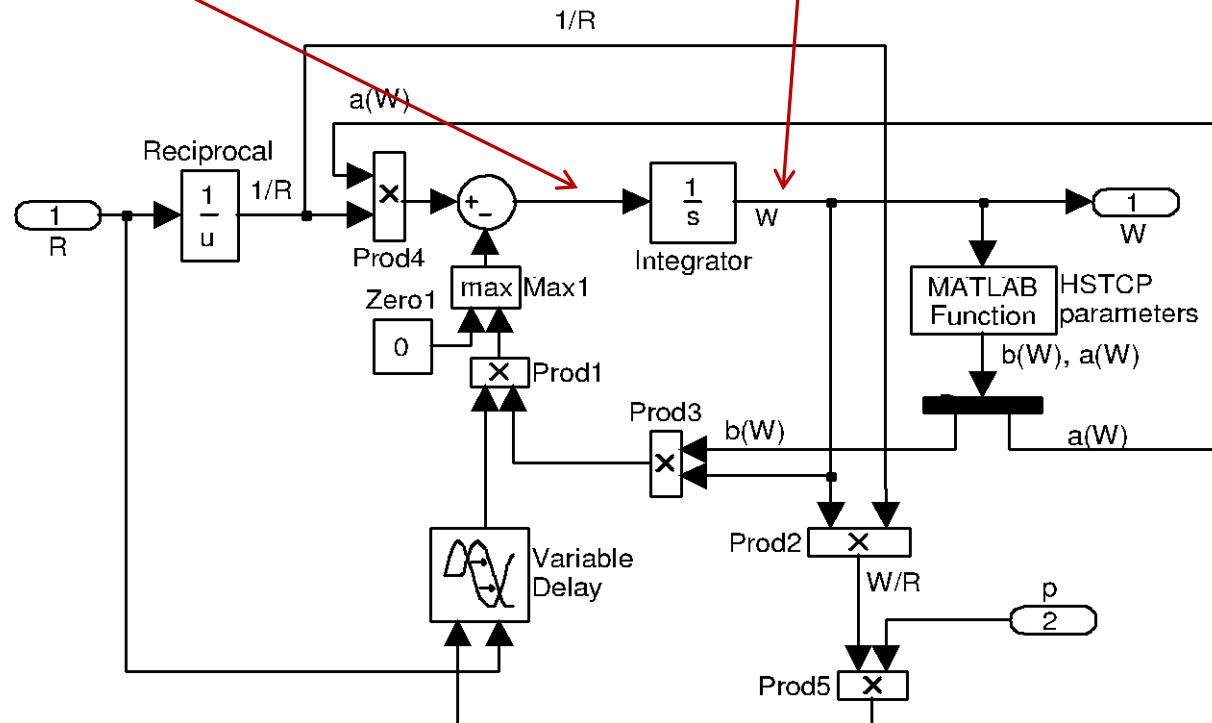


- Plant: HSTCP source + bottleneck queue
- Controller: RED module
- $W$ : cwnd (state variable)
- $q$ : inst. queue length (state variable, output)
- $p$ : dropping/ marking probability (feedback signal, input)
- $R$ : round-trip time
- $N_{input}$ : No. of sources

# Implementation of HSTCP source

- Module “Cwnd dynamics” → describe the corresponding differential equation

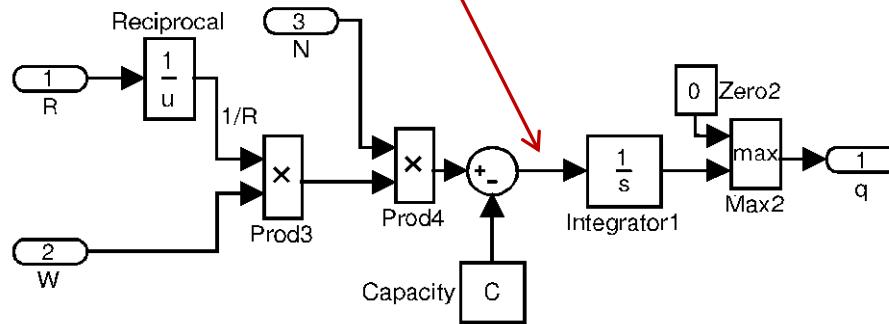
$$\dot{W}(t) = \frac{a(W(t))}{R(t)} - b(W(t)) W(t) \frac{W(t - R(t))}{R(t - R(t))} p(t - R(t))$$



# Implementation of network model

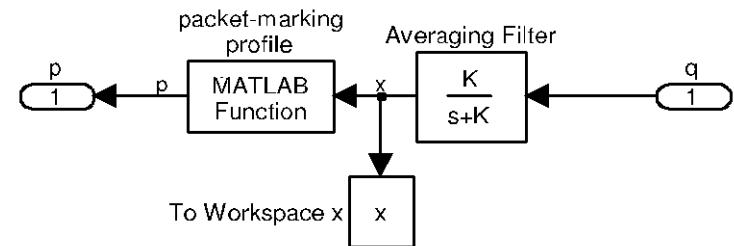
- Module “queue dynamics” → describe the corresponding differential equation

$$\dot{q}(t) = N(t) \frac{W(t)}{R(t)} - 1_{q(t)} C$$



- AQM module → implementing RED packet marking profile
  - averaging filter (low-pass filter)
  - packet-marking profile
- RTT module: simple relation

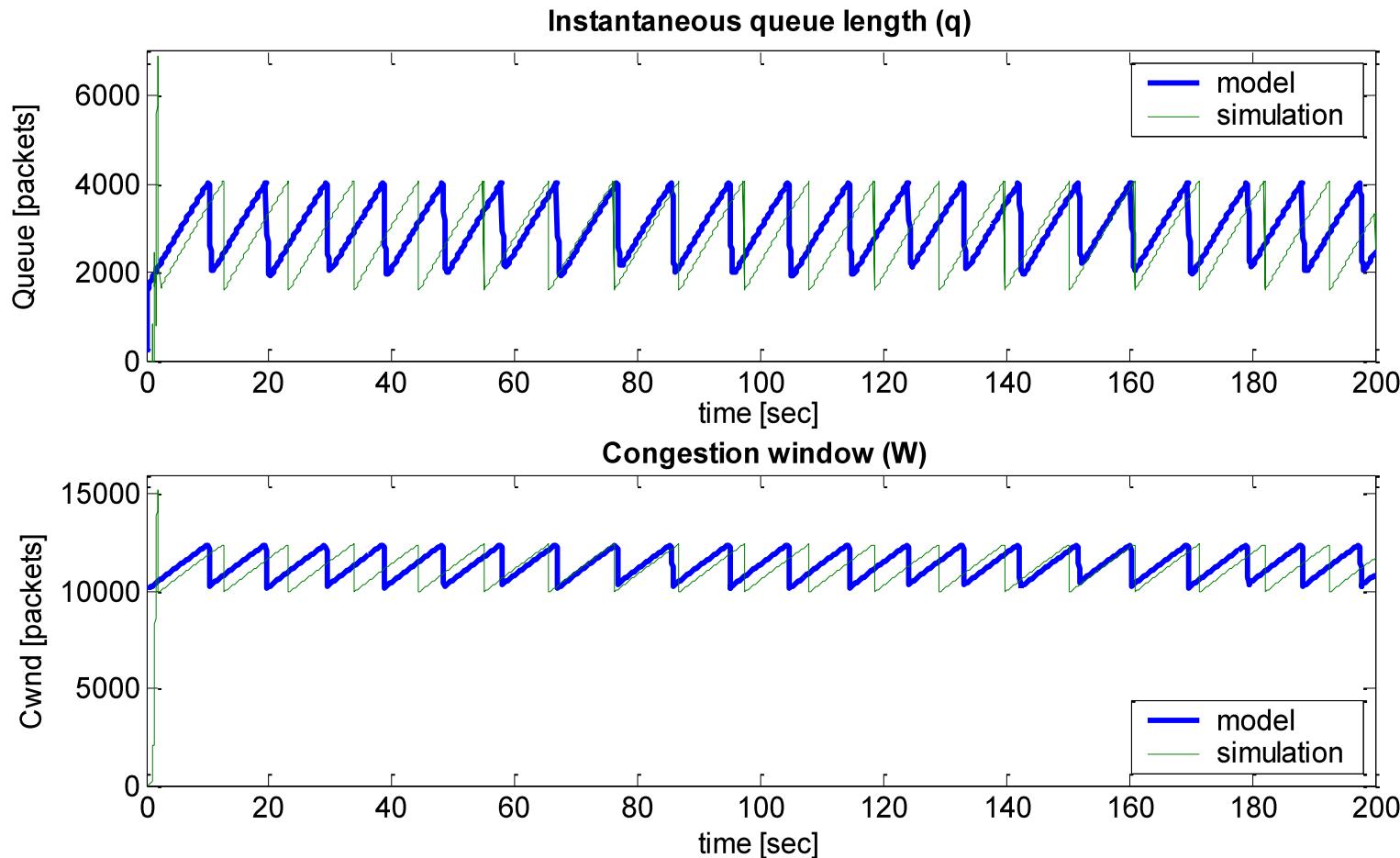
$$R = T_p + \frac{q(t)}{C}$$



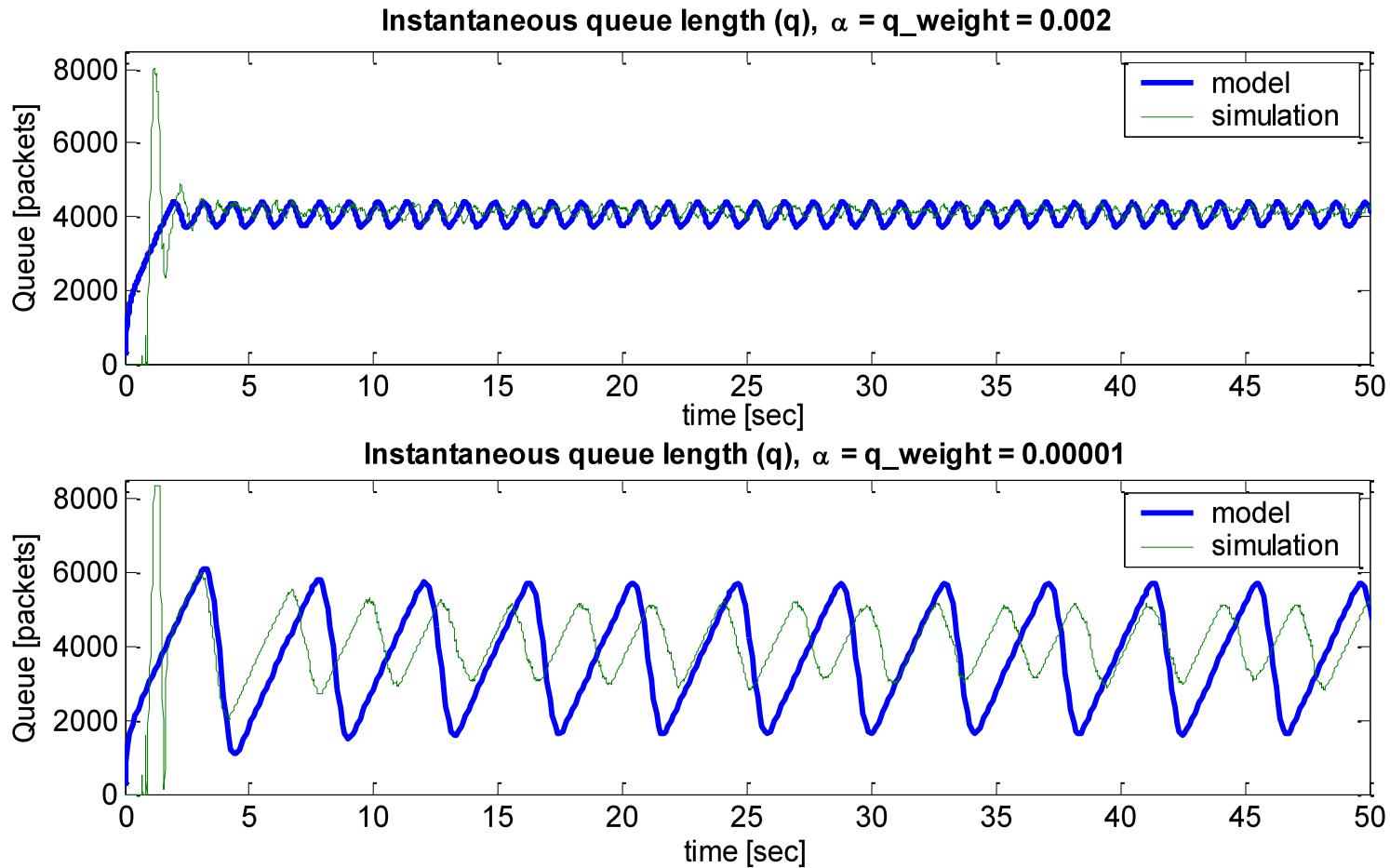
# Validation of the model

- Flow-level model  $\leftrightarrow$  packet-level simulations
- MATLAB/Simulink  $\leftrightarrow$  Ns-2
  - with similar parameter sets
  - impact of Slow-Start is modeled by initial condition of congestion window at the flow-level
- single HSTCP flow
- more HSTCP flows, impact of different parameters
- oscillation or stability at the flow-level
- (expected values!)

# Single HSTCP flow



# 100 HSTCP flows



Kitekintés

# MIRE IS HASZNÁLHATÓ A MODELL?

# Stability analysis

- Non-linear model (diff. equations)
  - global asymptotic stability could be analyzed
  - starting from any (feasible) initial condition, the algorithms converge to the unique equilibrium state
  - BUT this model is analytically not tractable!
- Linearization
  - our non-linear model can be linearized about an operating point (now at the stable point)
  - dynamics → approximated by the first order derivatives
  - local behavior
  - local asymptotic stability can be analyzed
  - with reasonable region of attraction
- Stability condition
  - designing RED (parameters) for stable behavior
- Numerical evaluation

# Linearized model

- Operating point  $(W_0, q_0, p_0) \rightarrow dW/dt = 0$  and  $dq/dt = 0$  (equilibrium)
- some variables can be approximated by constant values  $(R_0, a_0, b_0)$
- first-order dynamics based on first-order partial derivatives

$$\begin{aligned}\delta \dot{W}(t) &= -\frac{a_0 N}{R_0^2 C} (\delta W(t) + \delta W(t - R_0)) \\ &\quad - \frac{b_0 R_0 C^2}{N^2} \delta p(t - R_0) \\ \dot{\delta q}(t) &= \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t)\end{aligned}$$

- where the variables denote perturbations

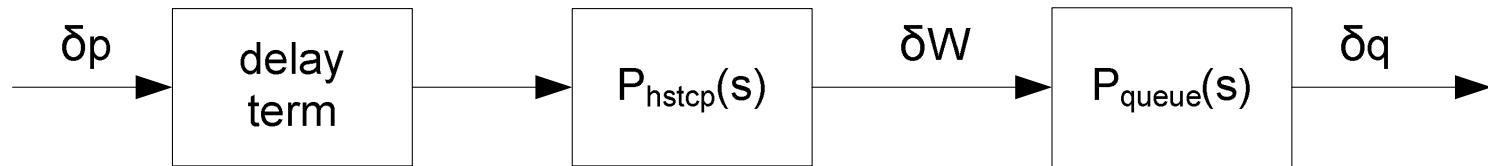
$$\delta W \doteq W - W_0, \delta q \doteq q - q_0 \text{ and } \delta p \doteq p - p_0$$

# Linearized model

- this linear system (plant) can be transformed into the Laplace transform domain, transfer functions can be derived

$$\begin{aligned} P(s) &= -e^{-sR_0} P_{hstcp}(s) P_{queue}(s) = \\ &= -e^{-sR_0} \frac{\frac{b_0 R_0 C^2}{N^2}}{s + \frac{a_0 N}{R_0^2 C} (1 + e^{-sR_0})} \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}} \end{aligned}$$

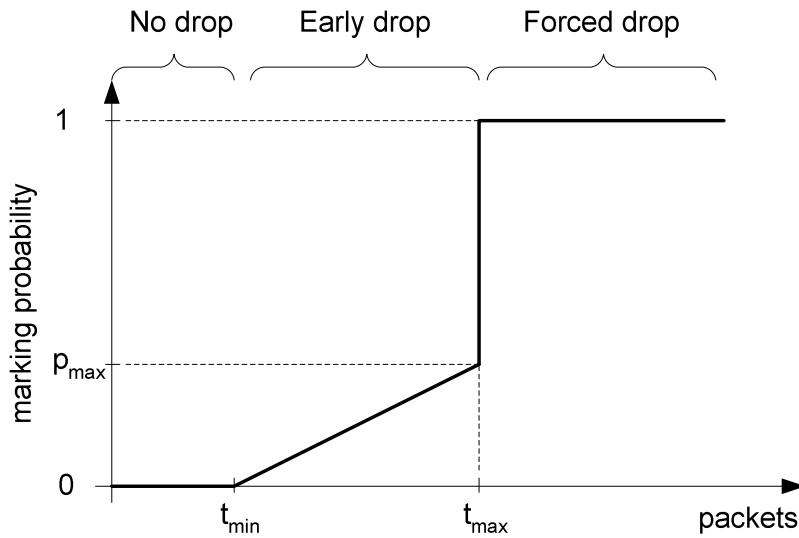
- and can be further simplified



$$P_{hstcp}(s) = \frac{\frac{b_0 R_0 C^2}{N^2}}{s + \frac{2a_0 N}{R_0^2 C}} \quad ; \quad P_{queue}(s) = \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}}$$

# RED controller for the linear plant

- Linear phase of the packet-marking profile is considered
- (early drop)



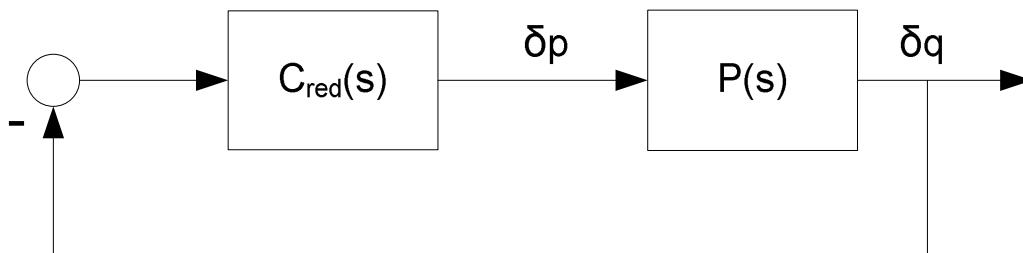
- $C(s)$ : transfer function of RED controller
- $\alpha$ : forgetting factor
- $\delta$ : sample time ( $\approx 1/C$  in steady-state)
- $L$ : slope of the curve of RED profile
- $K$ : cut-off frequency of RED controller

$$C(s) = C_{red}(s) = \frac{L}{\frac{s}{K} + 1}$$

$$L = \frac{p_{max}}{t_{max} - t_{min}} \quad \text{and} \quad K = -\frac{\ln(1 - \alpha)}{\delta} \approx -C \ln(1 - \alpha),$$

# RED controller for the linear plant

- Goal of the RED controller design
- to select RED parameters ( $L$ ,  $K$ )
- to stabilize the feedback control system
- for a given range of  $N$  and  $R_0$   $\rightarrow N \geq N^-$  and  $R_0 \leq R^+$



$$P(s) = \frac{\frac{b_0 C^2}{N} e^{-sR_0}}{\left(s + \frac{2a_0 N}{R_0^2 C}\right) \left(s + \frac{1}{R_0}\right)} \quad ; \quad C_{red}(s) = \frac{L_{hstcp}}{\frac{s}{K_{hstcp}} + 1}$$

# Stability condition

- Stability condition based on Nyquist-criterion can be derived analytically

If  $L_{hstcp}$  and  $K_{hstcp}$  satisfy

$$\frac{L_{hstcp} b_0 (R^+ C)^3}{2a_0 (N^-)^2} \leq \sqrt{\frac{\omega_g^2}{K_{hstcp}^2} + 1}$$

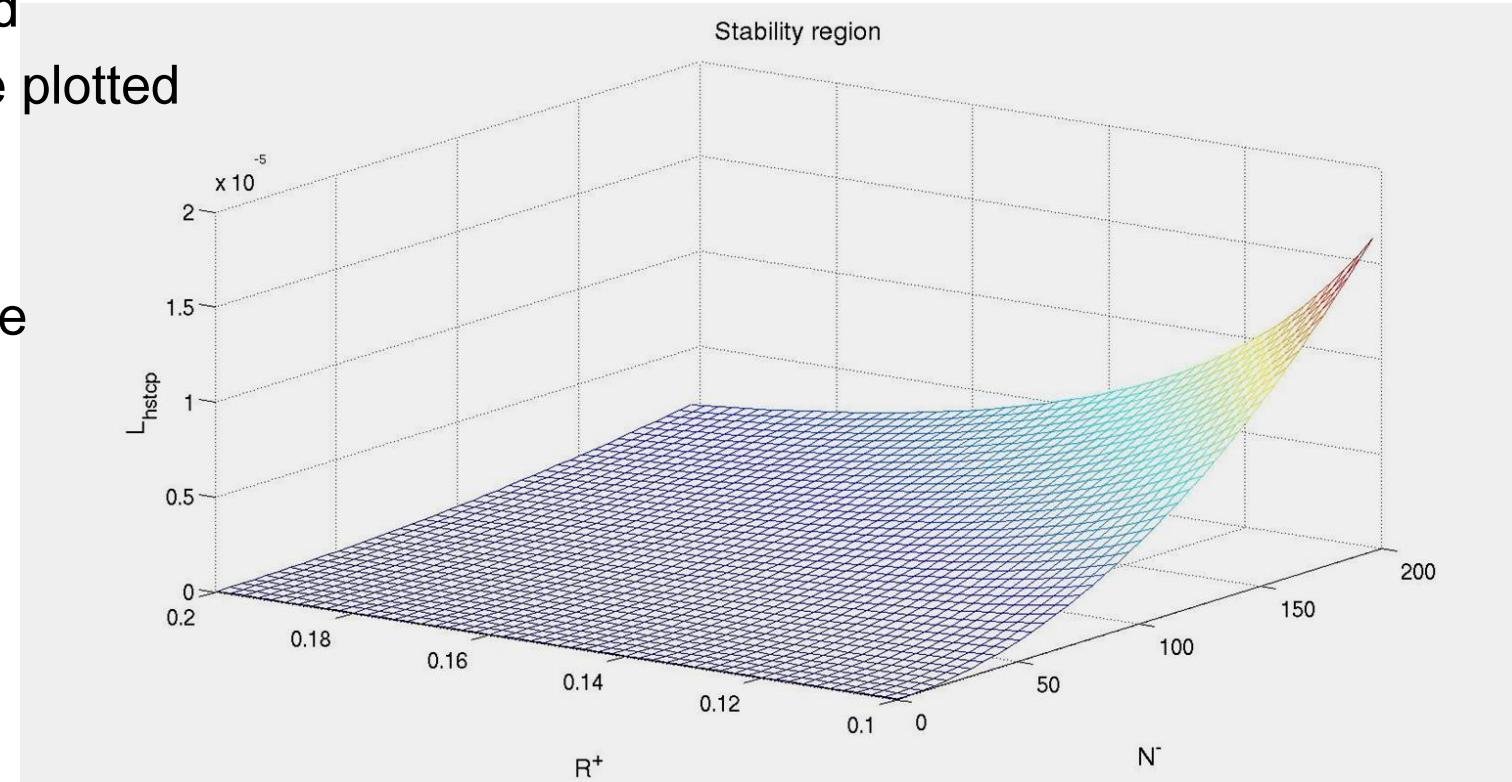
where

$$\omega_g = 0.1 \min \left\{ \frac{2a_0 N^-}{(R^+)^2 C}, \frac{1}{R^+} \right\}$$

then, the linear feedback control system is stable for all  $N \geq N^-$  and  $R_0 \leq R^+$ .

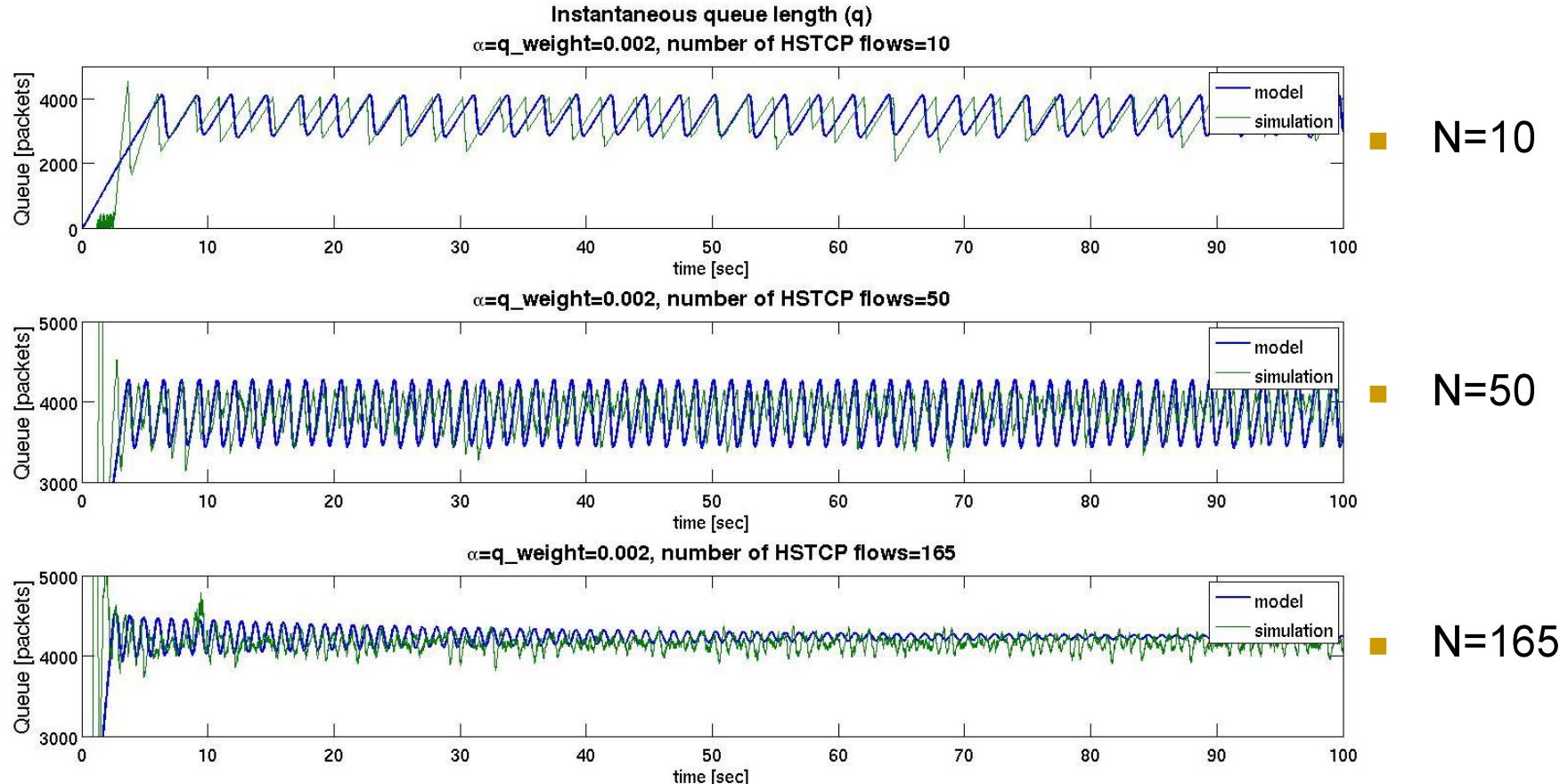
# Stability region

- Based on the condition, a stability region can be derived for given network parameters
- simple example:
  - $K$  is fixed
  - $L$  can be plotted in 3D
  - stability: under the surface



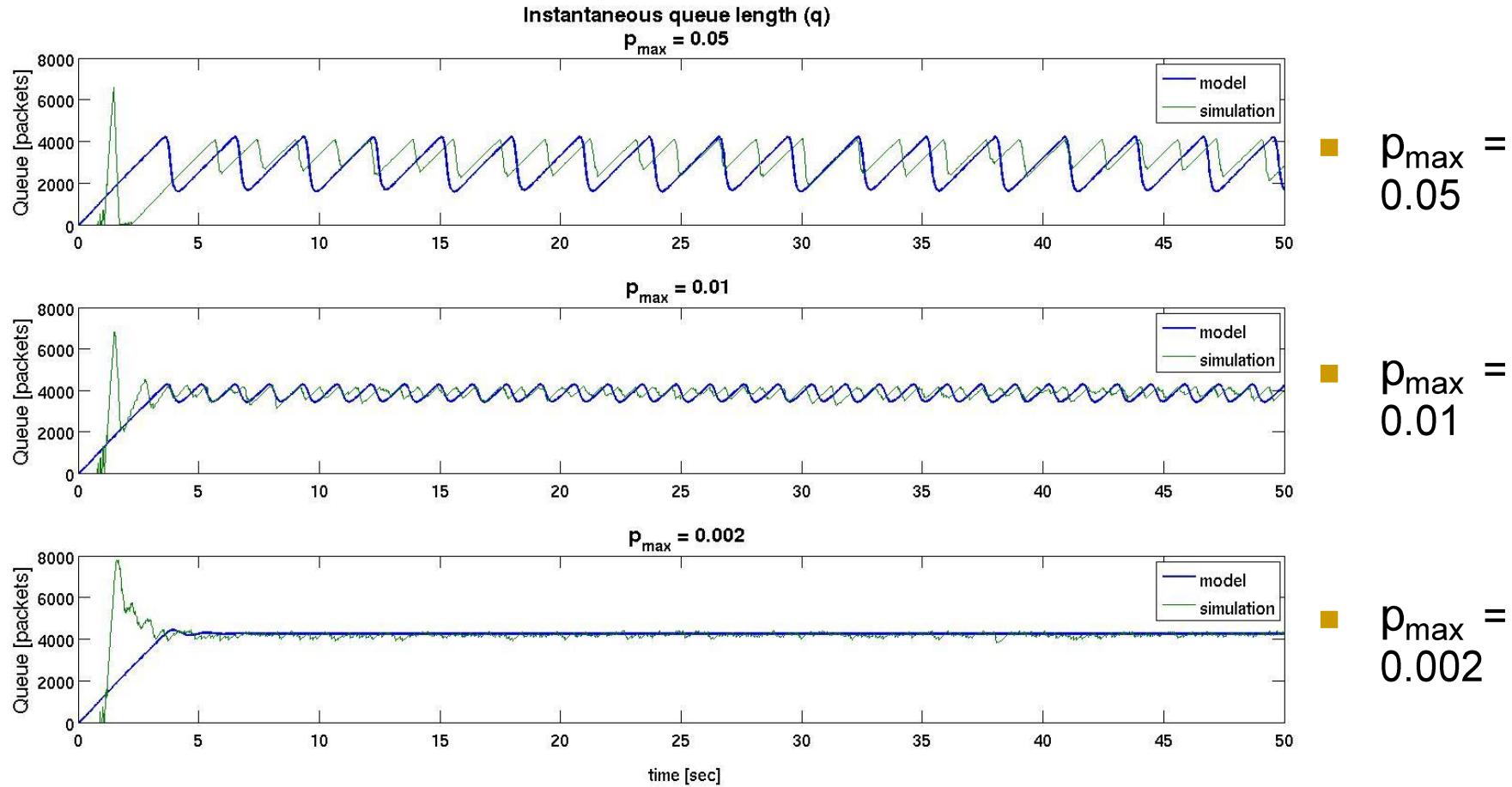
# Numerical examples (1)

- Fixed parameters, only the flow number ( $N$ ) is changed
- according to the stability condition → stability at around 160 flows



# Numerical examples (2)

- Stabilizing the behavior → tuning RED parameter  $p_{\max}$



# Összefoglalás

- TCP áttekintése
  - egy működő protokoll
  - torlódásszabályozás
- Értsük meg, mit csináltunk
  - matematikai modellek (utólag)
  - probléma megfogalmazása
- Lehetséges eszközök
  - folytonos idejű visszacsatolt rendszer
  - szabályozástechnika