Congestion control and multipath routing in TCP/IP communication networks

Roberto Cominetti
Cristóbal Guzmán

Universidad de Chile
rccc@dii.uchile.cl
cguzman@dim.uchile.cl

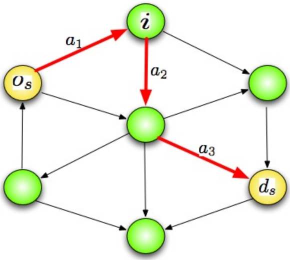
September 2010

Overview

1. Congestion control and network utility maximization
2. Congestion control with Markovian multipath routing
3. Implementation issues

Framework

- Communication network $G = (N, A)$
- Set of sources S
- Source $s \in S$ transmits from origin o_s to destination d_s
- A single route connects o_s to d_s



Notation: $a \in s$ iff link a belongs to route used by source s

TCP/IP Protocols

- **Route selection**
  Routing Information Protocol (RIP)
  slow timescale evolution (15-30 seconds)

- **Rate control**
  Transmission Control Protocol (TCP)
  fast timescale evolution (100-300 milliseconds)

TCP – Congestion control

Sources adjust transmission rates in response to congestion

higher congestion $\Rightarrow$ smaller rates

$(x_s)_{s\in S}$: source transmission rates [packets/sec]
$(\lambda_a)_{a\in A}$: link congestion prices (loss pbb, queuing delay)

Decentralized algorithms

$$x_s^{t+1} = F_s(x_s^t, q_s^t) \quad \text{(source - transmission control protocol)}$$
$$\lambda_a^{t+1} = G_a(\lambda_a^t, y_a^t) \quad \text{(link - active queue management)}$$

where

$$q_s^t = \sum_{a\in s} \lambda_a^t \quad \text{(end-to-end route congestion)}$$
$$y_a^t = \sum_{s\ni a} x_s^t \quad \text{(aggregate rates on links)}$$

TCP – Congestion control

Sources adjust transmission rates in response to congestion

higher congestion $\Rightarrow$ *smaller rates*

$(x_s)_{s \in S}$: source transmission rates [packets/sec]
$(\lambda_a)_{a \in A}$: link congestion prices (loss pbb, queuing delay)

Decentralized algorithms

$$\begin{aligned} x_s^{t+1} &= F_s(x_s^t, q_s^t) \quad \text{(source - transmission control protocol)} \\ \lambda_a^{t+1} &= G_a(\lambda_a^t, y_a^t) \quad \text{(link - active queue management)} \end{aligned}$$

where

$$\begin{aligned} q_s^t &= \textstyle\sum_{a \in s} \lambda_a^t \quad \text{(end-to-end route congestion)} \\ y_a^t &= \textstyle\sum_{s \ni a} x_s^t \quad \text{(aggregate rates on links)} \end{aligned}$$

Example: TCP-Reno/DropTail-RED-REM

TCP-DropTail: control window AIMD (τ_s= round-trip time):

$$W_s^{t+\tau_s} = \begin{cases} W_s^t + 1 & \text{if } W_s^t \text{ packets are successfully transmitted} \\ \lceil W_s^t/2 \rceil & \text{one or more packets are lost (duplicate ack's)} \end{cases}$$

A packet is transmitted successfully with probability

$$\pi_s^t = \prod_{a \in s}(1 - p_a^t)$$

RED-REM: loss probability on links controlled by AQM

$$p_a^t = \varphi_a(r_a^t)$$

as a function of the link's average queue length

$$r_a^{t+1} = (1-\alpha_a)r_a^t + \alpha_a L_a^t$$

Example: TCP-Reno/DropTail-RED-REM

TCP-DropTail: control window AIMD (τ_s= round-trip time):

$$W_s^{t+\tau_s} = \begin{cases} W_s^t + 1 & \text{if } W_s^t \text{ packets are successfully transmitted} \\ \lceil W_s^t/2 \rceil & \text{one or more packets are lost (duplicate ack's)} \end{cases}$$

A packet is transmitted successfully with probability

$$\pi_s^t = \prod_{a \in s}(1 - p_a^t)$$

RED-REM: loss probability on links controlled by AQM

$$p_a^t = \varphi_a(r_a^t)$$

as a function of the link's average queue length

$$r_a^{t+1} = (1 - \alpha_a) r_a^t + \alpha_a L_a^t$$

Example: TCP-Reno/RED-REM

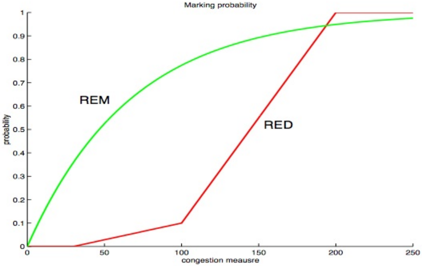


Figura: Loss probability $p_a = \varphi_a(r_a)$ as a function of average queue length

Example: TCP-Reno/RED-REM

Congestion prices

$$\left.\begin{array}{lll} q_s^t & \triangleq & -\ln(\pi_s^t) \\ \lambda_a^t & \triangleq & -\ln(1-p_a^t) \end{array}\right\} \Rightarrow \boxed{q_s^t = \sum_{a\in s} \lambda_a^t}$$

The approximate equality

$$\mathbb{E}(W_s^{t+\tau_s}|W_s^t) \sim e^{-q_s^t W_s^t}(W_s^t+1) + (1-e^{-q_s^t W_s^t})\lceil W_s^t/2\rceil$$

yields the following expected dynamics for rates $x_s^t = W_s^t/\tau_s$

$$\Rightarrow \boxed{x_s^{t+1} = x_s^t + \frac{1}{2\tau_s}\left[e^{-\tau_s q_s^t x_s^t}(x_s^t + \frac{2}{\tau_s}) - x_s^t\right]}$$

Example: TCP-Reno/RED-REM

Congestion prices

$$\left.\begin{array}{rcl} q_s^t & \triangleq & -\ln(\pi_s^t) \\ \lambda_a^t & \triangleq & -\ln(1-p_a^t) \end{array}\right\} \Rightarrow \boxed{q_s^t = \sum_{a\in s} \lambda_a^t}$$

The approximate equality

$$\mathbb{E}(W_s^{t+\tau_s} | W_s^t) \sim e^{-q_s^t W_s^t}(W_s^t + 1) + (1 - e^{-q_s^t W_s^t})\lceil W_s^t/2 \rceil$$

yields the following expected dynamics for rates $x_s^t = W_s^t/\tau_s$

$$\Rightarrow \boxed{x_s^{t+1} = x_s^t + \frac{1}{2\tau_s}\left[e^{-\tau_s q_s^t x_s^t}(x_s^t + \frac{2}{\tau_s}) - x_s^t\right]}$$

Example: TCP-Reno/RED-REM

Congestion prices

$$\left.\begin{array}{lcl} q_s^t & \triangleq & -\ln(\pi_s^t) \\ \lambda_a^t & \triangleq & -\ln(1-p_a^t) \end{array}\right\} \Rightarrow \boxed{q_s^t = \sum_{a\in s} \lambda_a^t}$$

The approximate equality

$$\mathbb{E}(W_s^{t+\tau_s}|W_s^t) \sim e^{-q_s^t W_s^t}(W_s^t + 1) + (1 - e^{-q_s^t W_s^t})\lceil W_s^t/2 \rceil$$

yields the following expected dynamics for rates $x_s^t = W_s^t/\tau_s$

$$\Rightarrow \boxed{x_s^{t+1} = x_s^t + \frac{1}{2\tau_s}\left[e^{-\tau_s q_s^t x_s^t}(x_s^t + \frac{2}{\tau_s}) - x_s^t\right]}$$

Another example: TCP-Vegas

TCP-Vegas uses queueing delay as congestion measure

$$\lambda_a = \frac{L_a}{c_a} = \frac{\text{queue length}}{\text{queue capacity}}$$

A simple model for the dynamics

$$\boxed{\begin{aligned} \lambda_a^{t+1} &= \left[\lambda_a^t + \frac{y_a^t}{c_a} - 1\right]_+ \\ x_s^{t+1} &= x_s^t + \frac{1}{(d_s+q_s^t)^2}\,\mathrm{sign}(\alpha_s d_s - x_s^t q_s^t). \end{aligned}}$$

with

$$\alpha_s = \text{parameter of Vegas}$$
$$d_s = \text{round-trip propagation delay}$$

Network Utility Maximization

- Kelly, Maullo and Tan (1999) proposed an optimization-based model for distributed rate control in networks.
- Low, Srikant, etc. (1999-2002) showed that TCP congestion control algorithms solve implicitly an optimization problem.
- During last decade, the model has been used and extended to study the performance of wired and wireless networks.

Steady state equations

$$
\begin{aligned}
x_s^{t+1} &= F_s(x_s^t, q_s^t) \quad \text{(TCP - source dynamics)} \\
\lambda_a^{t+1} &= G_a(\lambda_a^t, y_a^t) \quad \text{(AQM - link dynamics)}
\end{aligned}
$$

$$
\begin{array}{lcll}
x_s = F_s(x_s, q_s) & \Leftrightarrow & q_s = f_s(x_s) & \text{(decreasing)} \\
\lambda_a = G_a(y_a, \lambda_a) & \Leftrightarrow & \lambda_a = \psi_a(y_a) & \text{(increasing)} \\
 & & q_s = \sum_{a \in s} \lambda_a & \\
 & & y_a = \sum_{s \ni a} x_s &
\end{array}
$$

Example: TCP/REM steady state

$$\boxed{q_s = f_s(x_s) \triangleq \frac{1}{\tau_s x_s} \ln(1 + \frac{2}{\tau_s x_s})}$$

$$\left.\begin{array}{l} p_a = \varphi_a(r_a) \triangleq 1 - \exp(-\delta r_a) \\ r_a = \mathbb{E}(L_a) = \frac{y_a}{c_a - y_a} \end{array}\right\}$$

$$\boxed{\lambda_a = \psi_a(y_a) \triangleq -\ln(1 - \varphi_a(\frac{y_a}{c_a - y_a})) = \delta \frac{y_a}{c_a - y_a}}$$

Steady state - primal optimality

$$\begin{cases} q_s = f_s(x_s) \\ \lambda_a = \psi_a(y_a) \\ q_s = \sum_{a \in s} \lambda_a \\ y_a = \sum_{s \ni a} x_s \end{cases}$$

$$f_s(x_s) = \sum_{a \in s} \lambda_a = \sum_{a \in s} \psi_a(y_a) = \sum_{a \in s} \psi_a\left(\sum_{s \ni a} x_s\right)$$

$\equiv$ optimal solution of strictly convex program

$$(P) \quad \boxed{\min_x \ \sum_{s \in S} F_s(x_s) + \sum_{a \in A} \Psi_a\left(\sum_{s \ni a} x_s\right)}$$

$$F_s'(\cdot) = -f_s(\cdot)$$

$$\Psi_a'(\cdot) = \psi_a(\cdot)$$

Steady state - primal optimality

$$\begin{cases} q_s = f_s(x_s) \\ \lambda_a = \psi_a(y_a) \\ q_s = \sum_{a \in s} \lambda_a \\ y_a = \sum_{s \ni a} x_s \end{cases}$$

$$f_s(x_s) = \sum_{a \in s} \lambda_a = \sum_{a \in s} \psi_a(y_a) = \sum_{a \in s} \psi_a\left(\sum_{s \ni a} x_s\right)$$

$\equiv$ optimal solution of strictly convex program

$$(P) \quad \boxed{\min_x \ \sum_{s \in S} F_s(x_s) + \sum_{a \in A} \Psi_a\left(\sum_{s \ni a} x_s\right)}$$

$$F_s'(\cdot) = -f_s(\cdot)$$

$$\Psi_a'(\cdot) = \psi_a(\cdot)$$

Steady state - dual optimality

Alternatively

$$\psi_a^{-1}(\lambda_a) = y_a = \sum_{s \ni a} x_s = \sum_{s \ni a} f_s^{-1}(q_s) = \sum_{s \ni a} f_s^{-1}\left(\sum_{a \in s} \lambda_a\right)$$

$\equiv$ optimal solution of strictly convex program

$$(D) \quad \boxed{\min_{\lambda} \sum_{a \in A} \Psi_a^*(\lambda_a) + \sum_{s \in S} F_s^*\left(\sum_{a \in s} \lambda_a\right)}$$

Steady state - dual optimality

Alternatively

$$\psi_a^{-1}(\lambda_a) = y_a = \sum_{s \ni a} x_s = \sum_{s \ni a} f_s^{-1}(q_s) = \sum_{s \ni a} f_s^{-1}\left(\sum_{a \in s} \lambda_a\right)$$

$\equiv$ optimal solution of strictly convex program

$$(D) \quad \boxed{\min_{\lambda} \ \sum_{a \in A} \Psi_a^*(\lambda_a) + \sum_{s \in S} F_s^*\left(\sum_{a \in s} \lambda_a\right)}$$

Theorem (Low'2003)

Let x^*, λ^* *and set* $y_a^* = \sum_{s \ni a} x_s^*$ *and* $q_s^* = \sum_{a \in s} \lambda_a^*$. *Then:*

$$\left.\begin{array}{r} x_s^* = f_s(q_s^*) \\ \lambda_a^* = \psi_a(y_a^*) \end{array}\right\} \Leftrightarrow \left\{ \begin{array}{l} x^* \text{ and } \lambda^* \text{ are optimal solutions} \\ \text{for } (P) \text{ and } (D) \text{ respectively} \end{array}\right.$$

Usefulness?

- Reverse engineering of existing protocols
- Forward engineering of new protocols: f_s and ψ_a
- Conceive distributed algorithms to optimize prescribed utilities
- Flexible choice of congestion measure q_s
  - loss probability (TCP Reno/DropTail-RED-REM)
  - propagation delay (TCP Vegas of FAST)

Limitations?

- Delays in transmission of congestion prices
- Improper account of stochastic phenomena
- Single-path routing

Usefulness?

- Reverse engineering of existing protocols
- Forward engineering of new protocols: f_s and ψ_a
- Conceive distributed algorithms to optimize prescribed utilities
- Flexible choice of congestion measure q_s
  - loss probability (TCP Reno/DropTail-RED-REM)
  - propagation delay (TCP Vegas of FAST)

Limitations?

- Delays in transmission of congestion prices
- Improper account of stochastic phenomena
- Single-path routing

Markovian network utility maximization

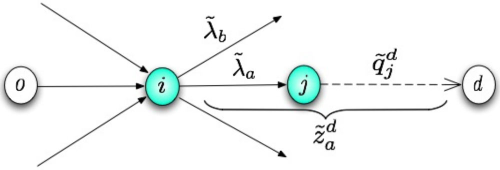
- Want to design a distributed protocol that supports source congestion control and multi-path routing
- Goal: packet-level communication protocol that satisfies some prescribed optimality criteria at equilibrium
- Model based on the notion of Markovian traffic equilibrium

The Model

- Communication network $G = (N, A)$
- Source $s \in S$ sends packets from o_s to d_s at rate x_s
- Links have random prices $\tilde{\lambda}_a = \lambda_a + \epsilon_a$ with $\mathbb{E}(\epsilon_a) = 0$.

At switch i, packets headed to destination d are routed through the link $a \in A_i^+$ that minimizes the cost to destination

$$\tilde{q}_i^d = \text{mín}_{a \in A_i^+} \underbrace{\tilde{\lambda}_a + \tilde{q}_{j_a}^d}_{\tilde{z}_a^d}$$

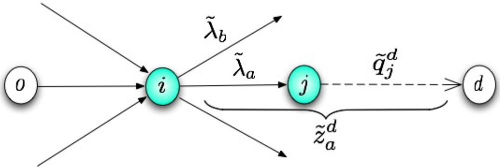


Markov chain with transition matrix

$$P_{ij}^d = \begin{cases} \mathbb{P}(\tilde{z}_a^d \leq \tilde{z}_b^d, \forall b \in A_i^+) & \text{if } i = i_a, j = j_a \\ 0 & \text{otherwise} \end{cases}$$

At switch i, packets headed to destination d are routed through the link $a \in A_i^+$ that minimizes the cost to destination

$$\tilde{q}_i^d = \text{mín}_{a \in A_i^+} \underbrace{\tilde{\lambda}_a + \tilde{q}_{j_a}^d}_{\tilde{z}_a^d}$$



Markov chain with transition matrix

$$P_{ij}^d = \begin{cases} \mathbb{P}(\tilde{z}_a^d \leq \tilde{z}_b^d, \forall b \in A_i^+) & \text{if } i = i_a,\, j = j_a \\ 0 & \text{otherwise} \end{cases}$$

Example: Multinomial Logit routing

$\tilde{z}_a^d = z_a^d - \epsilon_a^d$ with ϵ_a^d i.i.d. Gumbel $\Rightarrow$

$$\mathbb{P}\left(\tilde{z}_a^d \leq \tilde{z}_b^d, \forall b \in A_i^+\right) = \frac{e^{-\beta z_a^d}}{\sum_{b \in A_i^+} e^{-\beta z_b^d}}.$$

Remark:

$$\begin{array}{lcl} \beta \to 0 & \Leftrightarrow & \text{random walk} \\ \beta \to \infty & \Leftrightarrow & \text{shortest path} \end{array}$$

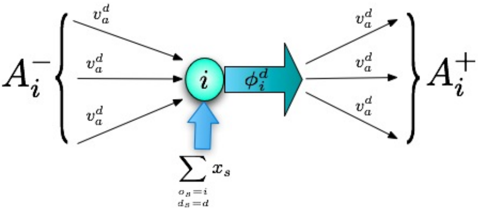
Expected flows

The flow ϕ_i^d entering node i and directed towards d

$$\phi_i^d = \sum_{\substack{o_s=i \\ d_s=d}} x_s + \sum_{a\in A_i^-} v_a^d$$

splits among the outgoing links $a = (i,j)$ according to

$$v_a^d = \phi_i^d P_{ij}^d$$



Expected costs

Letting $z_a^d = \mathbb{E}(\tilde{z}_a^d)$ and $q_i^d = \mathbb{E}(\tilde{q}_i^d)$, we have

$$\begin{aligned} z_a^d &= \lambda_a + q_{j_a}^d \\ q_i^d &= \varphi_i^d(z^d) \end{aligned}$$

with

$$\boxed{\varphi_i^d(z^d) \triangleq \mathbb{E}(\text{mín}_{a \in A_i^+}[z_a^d + \epsilon_a^d])}$$

Moreover

$$\mathbb{P}\left(\tilde{z}_a^d \leq \tilde{z}_b^d, \forall b \in A_i^+\right) = \frac{\partial \varphi_i^d}{\partial z_a^d}(z^d)$$

Given rate control functions $f_s(\cdot)$ and congestion functions $\psi_a(\cdot)$.

Definition (Markovian NUM)

(x, y, v, z, q, λ) is MNUM if $\lambda_a = \psi_a(y_a)$ and $q_s = f_s(x_s)$ with

$$\begin{array}{ll} y_a = \sum_d v_a^d & \text{(total flow on link } a\text{)} \\ q_s = q_{o_s}^{d_s} & \text{(congestion price for source } s\text{)} \end{array}$$

where the expected costs (q, z) are such that

$$(ZQ) \quad \left\{ \begin{array}{l} z_a^d = \lambda_a + q_{j_a}^d \\ q_i^d = \varphi_i^d(z^d) \end{array} \right.$$

and the expected flows v^d satisfy

$$(FC) \quad \left\{ \begin{array}{ll} \phi_i^d = \sum_{\substack{o_s=i \\ d_s=d}} x_s + \sum_{a \in A_i^-} v_a^d & \forall i \neq d \\ v_a^d = \phi_i^d \frac{\partial \varphi_i^d}{\partial z_a^d}(z^d) & \forall a \in A_i^+ \end{array} \right.$$

The dual problem

- (ZQ) defines z_a^d and q_i^d as implicit functions of λ
- then $x_s = f_s^{-1}(q_s(\lambda))$ yields x_s as function of λ
- (FC) then defines v_a^d as functions of λ

Thus: MNUM conditions $\Leftrightarrow$ $\psi_a^{-1}(\lambda_a) = y_a = \sum_d v_a^d(\lambda)$

Theorem

λ supports a MNUM iff it is an optimal solution of

$$(D) \quad \min_{\lambda} \sum_{a \in A} \Psi_a^*(\lambda_a) + \sum_{s \in S} F_s^*(q_s(\lambda))$$

The primal problem

Theorem

MNUM is the optimal solution of

$$\min_{(x,y,v)\in P} \sum_{s\in S} F_s(x_s) + \sum_{a\in A} \Psi_a(y_a) + \sum_{d\in D} \chi^d(v^d)$$

where

$$\chi^d(v^d) = \sup_{z^d} \sum_{a\in A} (\varphi^d_{i_a}(z^d) - z^d_a) v^d_a$$

and P is the polyhedron defined by flow conservation constraints.

Implementation

- Implement a distributed algorithm for the previous model
- Protocol with 2 time-scales: a slow one for choosing prices, and a faster one for rate control and price estimation for users
- Notification between routers using a RIP protocol, changing the measure of distance (number of hops) for link prices. This enables dynamic routing.
- Communicate prices to users. Routers can communicate the prices to the links, but at a slower time-scale than rate control.

Implementation

- Implement a distributed algorithm for the previous model
- Protocol with 2 time-scales: a slow one for choosing prices, and a faster one for rate control and price estimation for users
- Notification between routers using a RIP protocol, changing the measure of distance (number of hops) for link prices. This enables dynamic routing.
- Communicate prices to users. Routers can communicate the prices to the links, but at a slower time-scale than rate control.

Price estimation for sources

- Use Explicit Congestion Notification (ECN)
- Assume that prices take values between 0 and 1
- Adapt Adler *et al.* to estimate end-to-end prices $q_s = q_{o_s}^{d_s}$

ECN: Markov chain for marking with $X_0 = 0$ and transitions

$$X_i = \begin{cases} X_{i-1} & \frac{i-1}{i} \\ 1 & \frac{\lambda_{a_i}}{i} \\ 0 & \frac{1-\lambda_{a_i}}{i}. \end{cases}$$

- For fixed route $\mathbb{E}(X_k) = \frac{1}{k}\sum_{i=1}^{k} \lambda_{a_i}$
- Knowing the route length k we can estimate end-to-end price
- Routers need their position i in the route for the pbb
- Markovian case: unknown length of routes and positions

ECN with Markovian route choice

$$\mathbb{E}(X^s) = \sum_{r \in R^s} \left(\prod_{a \in r} p_{i_a j_a} \right) \left(\frac{1}{|r|} \sum_{a \in r} \lambda_a \right),$$

and

$$q^s = \sum_{r \in R^s} \left(\prod_{a \in r} p_{i_a j_a} \right) \left(\sum_{a \in r} \lambda_a \right).$$

If we correct the factor $1/|r|$ we get an unbiased estimator of q^s using sample averages.

Modify the Markov chain using the TTL (time-to-live) value on the IP header. As the TTL value is decreasing, we define a Markov chain X^s with $X_0^s = 0$ and

$$X_i^s = \begin{cases} X_{i-1}^s & \dfrac{T_i}{T_{i-1}} \\[2ex] 1 & \dfrac{T_i}{M}\lambda_{a_i} \\[2ex] 0 & 1 - T_i\left[\dfrac{1}{T_{i-1}} + \dfrac{\lambda_{a_i}}{M}\right]. \end{cases}$$

Where M is an upper bound for the square of all TTL values (this makes that the three transitions have positive probability).

If the final TTL value is T then

$$\mathbb{E}(X^s) = \sum_{r \in R^s} \left(\prod_{a \in r} p_{i_a j_a} \right) \left(\frac{T}{M} \sum_{a \in r} \lambda_a \right)$$

and we get an unbiased estimator for q^s

$$Y^s = \tfrac{M}{T} X^s$$

Conclusions

- We proposed a model that considers multipath routing for NUM, inspired in a packet-level dynamics
- Communication of variables required by protocol is possible under current TCP/IP
- ECN yields unbiased estimation of prices for users, that work on the same time-scale as rate control

Future work

- Design a distributed algorithm for the model
- Study stochastic stability of the markovian equilibrium
- Find a ECN mechanism that works for unbounded prices
- Analyze MNUM when randomness tends to 0

References

- H. Yaïche, R. Mazumdar and C. Rosenberg: *A game theoretic framework for bandwidth allocation and pricing in broadband networks*, IEEE/ACM Transactions on Networking, (2000).
- M. Chiang, S. H. Low, A. R. Calderbank, J. C. Doyle: *Layering as optimization decomposition*, Proceedings of IEEE, (2006).
- F. Kelly, A. Maulloo, D. Tan: *Rate control for communication networks: Shadow prices, proportional fairness and stability*, Journal of Operation Research, (1998).
- J. B. Baillon, R. Cominetti: *Markovian traffic equilibrium*, Mathematical Programming, 2007.
- M. Adler, J. Y. Cai, J. K. Shapiro, D.Towsley: *Estimation of congestion price using probabilistic packet marking*. Proceedings of IEEE INFOCOM, 2002.