Rate Allocation in the Internet

Yang Richard Yang
10/29/2001

Review:

- End-to-end congestion control
  - the focus is on an individual flow
  - end hosts change their behaviors by observing the status of the network
  - assumes the routers provide FIFO service
  - the behaviors of end hosts determine rate allocation
  - end hosts adapt to network
- Starting from this class, we will look at the interactions among different flows
  - we don't know much yet
**Objective of Rate Allocation**

**Heavy tail ➔ Mice-elephants**

- Efficient & fair sharing
- Small delay
- Queueing + propagation

**End Hosts and the Network**

- Example adaptation algorithms:
  - TCP/Reno
  - TCP/SACK
  - TCP/Vegas
  - TCP-Friendly congestion

- Example congestion measure $p_j(t)$:
  - Loss/marking
  - Queue length
  - Queueing delay
  - Delay jitter
  - Others
Three Perspectives

- Efficiency and fairness
- Stability and robustness
- Reality

Rate Allocation: What is the State at Equilibrium and How to Achieve it?

Optimization theory $\rightarrow$ efficiency and fairness
- Source rates $x_i(t)$ are primal variables
- Congestion measures $p_j(t)$ are dual variables
- Congestion control is an optimization process over the Internet
Rate Allocation: Will We Always Achieve the Optimal Equilibrium?

Control theory → stability & robustness
- Internet is a gigantic feedback system
- Distributed & delayed

Rate Allocation: Reality

- Dynamic system (may never achieve equilibrium)
- Many small flows (how to consider them?)
- What exactly do applications need?
- What if applications do not cooperate?
Schedule

- Today:
  - efficiency and fairness: optimization
- Wednesday:
  - stability and robustness: control theory
- Reality: ??

Rate Allocation: Objective
Resource Allocation:
Formalizing the Objective [Kelly97]

\[ \text{SYSTEM}(U, A, C) \]
\[
\begin{align*}
\text{maximize} & \quad \sum_{i \in I} U_i(x_i) \\
\text{subject to} & \quad A^T x \leq C \\
& \quad x \geq 0
\end{align*}
\]

- \( U \): The column vector of \( U_i \), the utility function of user \( i \); \( U_i \) is strictly concave
- \( A \): The connectivity matrix; \( a_{ij} = 1 \) if user \( i \) uses link \( j \); otherwise 0
- \( x \): The column vector of \( x_i \)
- \( C \): The column vector of link capacity \( c_j \)

- What assumptions have already been made?
- Why is \( \text{SYSTEM}(U, A, C) \) hard to solve?

Solving the Optimization Problem

- Using Lagrange multiplier (shadow price, congestion measure)
  \[ L(x, z, \mu) = \sum_i U_i(x_i) + \mu^T (C - A^T x - z) \]

- For derivative over \( x_i \), we have
  \[ \frac{\partial}{\partial x_i} L(x, z, \mu) = U_i'(x_i) - \sum_{j \in J_i} \mu_j \]

- Let \( p_i = \sum_{j \in I_i} \mu_j \) denote the price for user \( i \), we have
  \[ U_i'(x_i) = p_i \]
TCP Reno:
Relationship Between p and x

for every ack (\(ca\))
{ \(W \leftarrow \frac{1}{W}\) }
for every loss
{ \(W \leftarrow W/2\) }

\[
\Delta w_i(t) = \frac{x_i(t)(1 - p(t))}{w_i} - \frac{w_i(t)}{2} x_i(t) p(t)
\]

\[
\Rightarrow (1 - p_i) = \frac{x_i^2}{D_i^2} p_i
\]

\[
\Rightarrow p_i = \frac{1}{1 + \frac{D_i^2 x_i^2}{2}} = U_i^{'\text{reno}}(x_i)
\]

\[
\Rightarrow U_i^{'\text{reno}}(x_i) = \frac{\sqrt{2}}{D_i} \tan^{-1}\left(\frac{x_i D_i}{\sqrt{2}}\right)
\]

TCP/Vegas

for every RTT
{ if \(W_{\text{RTT}_m} - W_{\text{RTT}} < \alpha\) then \(W++\)
    if \(W_{\text{RTT}_m} - W_{\text{RTT}} > \alpha\) then \(W-\) }

for every loss
\(W := W/2\)

\[
x_i(t+1) = \begin{cases} 
    x_i(t) + \frac{1}{D_i^2} & \text{if } w_i(t) - d_i x_i(t) < \alpha_i d_i \\
    x_i(t) - \frac{1}{D_i^2} & \text{if } w_i(t) - d_i x_i(t) > \alpha_i d_i \\
    x_i(t) & \text{else}
\end{cases}
\]

\[
U_i^{\text{vegas}}(x_i) = \alpha_i d_i \log x_i
\]
Rate Allocation: Fairness

- For each utility function, we have an implicit fairness definition.
- When we achieve the maximal of \( \sum_{i \in I} U_i(x_i) \) we should have
  \[ d \sum_{i \in I} U_i(x_i) = \sum_{i \in I} U'_i(x_i) dx_i \leq 0 \]
- Consider \( U_i(x_i) = p_i \log(x_i) \), we have
  \[ d \sum_{i \in I} U_i(x_i) = \sum_{i \in I} U'_i(x_i) dx_i = \sum_{i \in I} p_i \frac{dx_i}{x_i} \leq 0 \]

Fairness: An Example

Max-min fairness vs. Proportional fairness

What is the rate allocation for max-min fairness?
What is the rate for proportional fairness?
Rate Allocation: Implementation

Decomposition of the Objective

USER(U_i; \lambda_i)
\begin{align*}
\text{maximize} & \quad U_i \left( \frac{p_i}{\lambda_i} \right) - p_i \\
\text{over} & \quad p_i \geq 0
\end{align*}

NETWORK(A, C; p)
\begin{align*}
\text{maximize} & \quad \sum_{i=1}^{k} p_i \log(x_i) \\
\text{subject to} & \quad A^T x \leq C \\
\text{over} & \quad x \geq 0
\end{align*}
Relationship Between Problems

- Theorem 2: There exist vectors
  \( \lambda^* = (\lambda^*_i, i \in I) \),
  \( p^* = (p^*_i, i \in I) \), and
  \( x^* = (x^*_i, i \in I) \), such that
  i) \( p^* \) solves \( \text{USER}_i(U_i, \lambda^*_i) \) for \( i \in I \)
  ii) \( x^* \) solves \( \text{NETWORK}(A, C; p^*) \)
  iii) \( p^*_i = \lambda^*_i x^*_i \) for \( i \in I \)

Furthermore, the vector \( x^* \) then also solves \( \text{SYSTEM}(U, A, C) \).

Structure of Algorithms

- Rate allocation problem
  \[
  \max_{x \geq 0} \sum_i U_i(x_i)
  \]
  subject to \( x_i' \leq c_i \), \( \forall i \in L \)

- Primal-dual algorithm
  \[
  x(t+1) = F(p(t), x(t)) \quad \text{Reno, Vegas}
  \]
  \[
  p(t+1) = G(p(t), x(t)) \quad \text{DropTail, RED, VQ}
  \]

- CC/AQM protocols \((F, G)\)
  - maximize aggregate source utility
  - with different utility functions \( U_i(x_i) \)
Previous Result by Kelly et al. 1998

User i updates its rate $x(t)$ by

$$\frac{d}{dt} x_i(t) = \kappa (w_i - x_i(t) \sum_{i \in I_j} \mu_j(t)), \text{ where}$$

$$\mu_j(t) = p_j \left( \sum_{i \in I_j} x_i \right)$$

If user i updates $w_i$ as follows:

$$w_i(t) = x_i(t) U'_j(x_i(t))$$

Then $x(t)$ converges to a unique stable point that maximizes

$$U(x) = \sum_i U_i(x_i) - \sum_j \int_0^{\sum_{i \in I_j} x_i} p_j(y) dy$$

Main Idea of This Paper

- Two phases
  - given $p$, solve NETWORK($A, C; p$)
  - drive $p$ to $p^*$
Solving NETWORK($A, C; p$)

- Use the (p,1)-proportional algorithm [MW98]
- Each user has a fixed target queue size $p_i$
- Window-based protocol
- Each user updates its window size $w_i(t)$ as follows:
  \[ \frac{d}{dt} w_i(t) = -\kappa \frac{d_i}{d_i(t)} \frac{w_i(t) - x_i(t) \overline{d_i} - p_i}{w_i(t)} \]
- At stable point, the rate allocation is weighted proportionally fair

Driving $p$ to $p^*$

- At given $p_i(t^-)$, after $x_i(t^-)$ is stable, user $i$ updates $p_i(t)$ as the following:
  \[ p_i(t) = \arg\max_{p_i} \left( U_i\left( \frac{p}{\lambda_i(t^-)} \right) - p_i \right) \]
  \[ = \begin{cases} 
   0 & \text{if } (\frac{\gamma(t^-)}{\lambda_i(t^-)} \geq U'_i(0)) \\
   p_i : U'_i\left( \frac{p}{\lambda_i(t^-)} \right) = \lambda_i(t^-) & \text{else}
  \end{cases} \]
Backup Slides

Decomposition 1 of the Objective

\[
\text{USER}_i(U; \lambda_i) \\
\left\{ \begin{array}{l} 
\text{maximize} & U_i(x_i) - \lambda_i x_i \\
\text{over} & x_i \geq 0 
\end{array} \right.
\]

\[
\text{NETWORK}(A, C; \lambda) \\
\left\{ \begin{array}{l} 
\text{maximize} & \sum_{i=1}^{\lambda} \lambda_i x_i \\
\text{subject to} & A^T x \leq C \\
\text{over} & x \geq 0 
\end{array} \right.
\]
Theorem 1: There exists a price vector \( \lambda = (\lambda_i, i \in I) \), such that the vector \( x = (x_i, i \in I) \), formed from the unique solution \( x_i \) to \( \text{USER}_i(U; \lambda_i) \) for each \( i \in I \), solves \( \text{NETWORK}(A, C; \lambda) \). The vector \( x \) then also solves \( \text{SYSTEM}(U, A, C) \).