Optimization-based approach to congestion control

Resource allocation as optimization problem:
- how to allocate resources (e.g., bandwidth) to optimize some objective function
- may not possible that optimality exactly obtained but...
  - optimization framework as means to explicitly steer network towards desirable operating point
  - practical congestion control as distributed asynchronous implementations of optimization algorithm
  - systematic approach towards protocol design
Model

- network: links \( \{l\} \), capacities \( \{c_l\} \)
- sources \( S \): \((L(s), U_s(x_s)), s \in S\)
  - \(L(s)\) - links used by source \( s \)
  - \(U_s(x_s)\) - utility, strictly concave function of source rate \( x_s \)

![Diagram of network with users and utility functions](image)
Kelly’s System Problem

$$\max_{x_0, x_1, x_2} \sum_{i=1}^{3} U_i(x_i)$$

subject to

$$x_0 + x_1 \leq c_A$$
$$x_0 + x_2 \leq c_B$$
$$x_i \geq 0$$
Optimization Problem

$$\max_{x_s \geq 0} \sum_s U_s(x_s)$$

subject to $$\sum_{s \in S(l)} x_s \leq c_l, \forall l \in L$$

- maximize system utility (note: all sources “equal”)
- constraint: bandwidth used less than capacity
- centralized solution to optimization impractical
  - must know all utility functions
  - impractical for large number of sources
- we’ll see: congestion control as distributed asynchronous algorithms to solve this problem
Issues

- Will users truthfully reveal their utility functions?
- If not, can we design a pricing scheme (mechanism) to induce truth-telling?
- Is there a distributed algorithm to compute the prices?
- What are good choices for utilities?
Max-min Fairness

rates \{x_r\} max-min fair if for any other feasible rates \{y_r\}, if \(y_s > x_s\), then \(\exists p\), such that \(x_p \leq x_s\) and \(y_p < x_p\)
Proportional fairness

- rates \( \{x_r\} \) are proportionally fair if for any feasible \( \{y_r\} \),
  \[
  \sum_{r \in S} \frac{y_r - x_r}{x_r} \leq 0
  \]
- corresponds to \( U_r (x_r) = \log x_r \)
- weighted proportional fairness if \( U_r (x_r) = w_r \log x_r \)
  \[
  \sum_{r \in S} w_r \frac{y_r - x_r}{x_r} \leq 0
  \]
Minimum potential delay fairness

- Rates \( \{x_r\} \) are minimum potential delay fair if \( U_r (x_r) = -w_r/x_r \)

Interpretation: if \( w_r \) is file size, then \( w_r/x_r \) is transfer time; optimization problem is to minimize sum of transfer delays
Max-min Fairness

rates \{x_r\} max-min fair if for any other feasible rates \{y_r\}, if y_s > x_s, then \exists p, such that \ x_p \leq x_s and y_p < x_p

What is corresponding utility function?

\[ U_r(x_r) = \lim_{\alpha \to \infty} \frac{x_r^{1-\alpha}}{1-\alpha} \]
Computing Source Rates

\[ L(x, p) = \sum_r U_r(x_r) - \sum_l p_l(y_l - c_l) \]

\[ y_l = \sum_{r: l \in r} x_r \]

\[ \frac{\partial L}{\partial x_r} = 0 \]
Remove constraints

- consider following problem

\[ V(\mathbf{x}) = \sum_r U_r(x_r) - \sum_{l \in L} \int_0^{\sum x_s} f_l(y) \, dy \]

- \( f_l(y) \) - penalty function
  - \( f_l() \) non decreasing, continuous and
  
  \[ \int_0^y f_l(x) \to \infty \quad \text{as } y \to \infty \]
\[ \max V(x) \]
\[ \frac{\partial V}{\partial x_r} = 0, \quad r \in S \]

\[ U_r'(x_r) - \sum_{l: l \in r} f_l(y_l) = 0, \quad r \in S \]
\[ y_l = \sum_{s: l \in s} x_s, \quad l \in L \]
\( p_l(t) \) price of link \( l \) at time \( t \)

\[
p_l(t) = f_l(y_l(t))
\]

\[
U'_r(x_r) - q_r = 0, \quad r \in S
\]
Source Algorithm

- Source needs only its path price:

\[ \dot{x}_r = k_r(x_r)(U_r'(x_r) - q_r) \]

- \( k_r() \) nonnegative nondecreasing function
- Above algorithm converges for any initial condition to unique solution
- Example: \( q_r \) - loss/marking probability
If utility function is

\[ U_r(x_r) = \omega_r \log x_r, \]

then a controller that implements it is given by
Proportionally-Fair Controller

If utility function is

\[ U_r(x_r) = w_r \log x_r, \]

then a controller that implements it is given by

\[ \dot{x}_r = \kappa_r (w_r - x_r q_r) \]
Computing Source Rates

\[ L(x, p) = \sum_r U_r(x_r) - \sum_l p_l(y_l - c_l) \]
Computing Lagrange Multipliers

- define

\[ D(p) = \max_{\{x_r\}} L(x, p) \]

- dual problem:

\[ \min_{\{p_l \geq 0\}} D(p) \]
Dual Algorithm

\[ U_i'(x_i) = \sum_{l \in i} p_l \]

\[ \frac{dp_l}{dt} = \frac{1}{c_l} (y_l - c_l) \]

- \( p_l \) delay at link \( l \)
- TCP-Vegas: modify source rates in response to measured delay
Dual algorithm

- converges to optimum rates
Primal-Dual Algorithm

\[ \dot{x}_i = \kappa_i(x_i) \left( U_i(x_i) - q_i \right) \]

\[ \dot{p}_l = f_l(p_l)(y_l - c_l) \]

- source can be TCP-Reno
- feedback generated by active queue management algorithms
Active Queue Management

- feedback function of queue length $b_l$

\[ p_l = f_l(b_l) \]

\[ \dot{b}_l = y_l - c_l \]
Random Early Detection (RED)

- Simplified view:

\[ p_l = K_l q_l \]

\[ \dot{p}_l = K_l (y_l - c_l) \]
Random Early Marking (REM)

\[ p_l(b_l) = 1 - e^{-\gamma b_l} \]

\[ \dot{p}_l = \gamma (1 - p_l)(y_l - c_l) \]
Exponential-RED

\[ p_l(b_l) = e^{K_l b_l} \]

\[ \dot{b}_l = K_l p_l(y_l - c_l) \]
Pricing

- can network choose pricing scheme to achieve fair resource allocation?
- suppose network charges price \( q_r \text{ ($/bit)} \)
  where \( q_r = \sum p_l \)
- user’s strategy: spend \( w_r \text{ ($/sec.)} \) to maximize

\[
U_r \left( \frac{w_r}{q_r} \right) - w_r
\]
Optimal User Strategy

\[
\frac{1}{q_r} U_r'(\frac{w_r}{q_r}) = 1
\]

- equivalently,

\[
\omega_r = x_r U_r'(x_r)
\]
Distributed Computation

- with optimal choice of $w_r$, controller becomes

$$\dot{x}_r = x_r \left( U'_r(x_r) - q_r \right)$$

- We have already seen that this solves

$$\max_{\{x_r\}} \sum_r U_r(x_r)$$
Price Takers vs. Strategic Users

\[
\max_{w_r} U_r \left( \frac{w_r}{q_r} \right) - w_r
\]

- Kelly Mechanism: users are price takers, i.e., user does not know the impact of its action on the price

- strategic users:

\[ q_r = g_r(w_r) \]
Efficiency and Competition

- price takers: selfish users can maximize social welfare

- strategic users: competition leads to loss of efficiency, i.e., social welfare is not maximized
TCP-Reno

The condition for optimality is

\[ U'_i(x_i) - p = 0 \]

or

\[ x_i = U_i^{-1}(p) \]

if we have an expression for \( x_i \), we can use to obtain \( U_i \).
TCP-Reno

- TCP-Reno in equilibrium:
  \[ \frac{2}{2 + T_i^2 x_i^2} = p \]

- utility function:
  \[ U_i(x_i) = \frac{\sqrt{2}}{T_i} \arctan \frac{x_i T_i}{\sqrt{2}} \]
Simplified TCP-Reno

- suppose

\[ \hat{x} = \sqrt{\frac{2(1 - \hat{p})}{\hat{p}} \frac{1}{T}} \approx \sqrt{\frac{2}{\hat{p}T}} \]

- then,

\[ U_r(x_r) = -\frac{1}{T_r x_r} \]

- interpretation: minimize (weighted) delay
Price versus Probabilistic Feedback

- price: \( q_i = \sum_{l=2} \pi_i \)

- loss probability: \( q_i = 1 - \prod_{l=2} (1 - \pi_i) \). If the \( \pi_i \)'s are small, they are approximately equal.

- else, TCP solves a modified version of the resource allocation problem.
Explicit Congestion Notification (ECN)

- Instead of dropping packets, **mark** packets to indicate incipient congestion.
- **Marking:** Router flips a bit in the packet header from 0 to 1 to indicate congestion.
- Destination echoes the ECN bit back to the source in the ack.
Multicast

- **single rate**
  - sender determines rate
  - $U(x) = |R|V(x)$ preferred

- **multirate**
  - each rcvr determines rate
  - each rcvr has own utility function
Other issues

- joint congestion control and routing
- high performance environments
Optimization-based congestion control: summary

- bandwidth allocation as optimization problem:
- practical congestion control (TCP) as distributed asynchronous implementations of optimization algorithm
- optimization framework as means to explicitly steer network towards desirable operating point
- systematic approach towards protocol design