Flow control and economics
Basic issues

- A large percentage of the flows in the Internet are *elastic*
  - they are subject to flow control
  - TCP is a flow control mechanism
- We need a framework to analyze the properties of such mechanisms
  - stability (delays, randomness)
  - economics (sharing the available bandwidth)
- Generalize existing mechanisms and understand the effect of certain parameters
- Reverse-engineering considerations
Contents

- Elastic traffic
- Fairness
- Flow control algorithms and economics
- TCP
- Reverse engineering concepts
Elastic traffic

Elastic traffic: flexible “contract” with network
- no guarantees on delay, throughput, CLP
- examples: TCP/IP, ABR
- Sources of randomness:
  - number of users + amount of data
  - amount of available resources
Flow control and economics

Notions of fairness in flow allocations
Fairness

- How should the bandwidth be shared among competing flows?
- Max throughput, max-min, proportional fairness,…
Max-min fairness: \( x = (x_r, r \in R) \) is max-min fair if it is feasible and for any other feasible vector \( y \),

\[ \exists r: y_r > x_r \Rightarrow \exists s: y_s < x_s < x_r \]

Proportional fairness:

\[
\sum_r \frac{y_r - x_r}{x_r} \leq 0
\]

Weighted proportional fairness:

\[
\sum_r w_r \frac{y_r - x_r}{x_r} \leq 0
\]

Mechanism-induced fairness: the result of TCP, …
Fairness based on economics

- Define fairness using an economic model of resource sharing
- **Assume** some utility functions for flows $x_r$, solve

$$\max \sum_r U_r(x_r)$$

**NUM:**

$$\{x_r\} \ s.t. \ Ax \leq C, \ x \geq 0$$

$$A_{jr} = 1 \text{ if } j \in r$$

$$A_{jr} = 0 \text{ if } j \not\in r$$

Or, using penalty functions:

$$\max \sum_r U_r(x_r) - \sum_r c_j(y_j)$$

**NUM-P:**

$$\{x_r\} \ s.t. \ Ax = y, \ x \geq 0$$

$$A_{jr} = 1 \text{ if } j \in r$$

$$A_{jr} = 0 \text{ if } j \not\in r$$
Unifying the two

Mo and Walrand: define *weighted α-fair allocations*:

\[ U_r(x_r) = w_r \frac{1}{1 - \alpha} x_r^{1-\alpha} \]

\[ x_r(p) = \left( \frac{w_r}{p_r} \right)^{\frac{1}{\alpha}} \]

- \( \alpha = 0 \) maximize throughput
- \( \alpha = 1 \) proportional fairness
- \( \alpha = \infty \) max-min fairness

Reverse engineering results: all proposed flow control mechanisms solve NUM for some choice of utilities

TCP Reno: \( a = 2, \ldots \)
Example: weighted prop. fairness

- w-Proportional Fairness (WPF): \( U_r(x_r) = w_r \log x_r \)
- Network problem NUM:

\[
\max_{\{x_r\}} \sum_r w_r \log(x_r) \quad s.t. \quad Ax \leq C, \quad x \geq 0
\]

\[
L(x, \mu) = \sum_r w_r \log(x_r) - \sum_j \mu_j \left[ \sum_{s: j \in s} x_s - C_j \right]
\]

\[
\mu_j > 0 \iff \sum_{s: j \in s} x_s - C_j = 0
\]

\[
(x^*, \mu^*) \quad \frac{\partial L}{\partial x_r} = 0 \iff \frac{w_r}{x_r} = \sum_{j \in r} \mu_j \iff x_r = \frac{w_r}{\sum_{j \in r} \mu_j} = \frac{w_r}{\lambda_r}
\]
Flow control and economics

An interesting class of flow control algorithms:
Primal-Dual algorithms
“Economic” flow control

Users: act according
To utility and price

Network: market model
Each link advertises a price

\[ U_r(x_r) \quad x_r \rightarrow \quad \lambda_r = \sum_{j : j \in r} \mu_j \]

If \( u_r(x) = w_r \log x \)
then \( x_r = w_r / \lambda_r \)

Users interact through the prices of the link markets

Network prices: many implementations

Achieves \( \max \sum_{r} U_r(x_r) \) s.t. \( Ax \leq C, x \geq 0 \)
Primal algorithms (theory)

**NUM :** \( \max_{\{x_r\}} \sum_r U_r(x_r) \text{ s.t. } Ax \leq C, x \geq 0 \)

**NUM - P :** \( \max_{\{x_r\}} \sum_{r \in S} U_r(x_r) - \sum_l B_l(\sum_{s : l \in S} x_s) \)

\( B_l(y) = \int_0^y p_l(z) \, dz \)

\( U'_r(x_r) - \sum_{l : l \in r} p_l(\sum_{s : l \in S} x_s) = 0 \quad \forall r \in S \)

How to maximize a function \( g(x) \)?
Choose \( x(t) \) s.t \( g(x(t + \delta)) > g(x(t)) \)

\( x(t + \delta) = x(t) + k \frac{dg(x)}{dx} \delta \)

\( \dot{x} = k \frac{dg(x)}{dx} \)

\( \dot{x}_r = k_r \left[ U'_r(x_r) - \sum_{l : l \in r} p_l(\sum_{s : l \in S} x_s) \right] \)

Price function

Economics of elastic flows - 13
Primal-Dual algorithms

**PRIMAL:**
\[
\frac{d}{dt} x_r(t) = k_r x_r(t) \left( U'_r(x_r(t)) - \sum_{j \in r} \mu_j(t) \right)
\]
\[
\mu_j(t) = p_j(y_j(t)), \quad y_j(t) = \sum_{r: j \in r} x_j(t)
\]

**DUAL:**
\[
\frac{d}{dt} \mu_j(t) = k_j \left( \sum_{r: j \in r} x_r(t) - C_j \right)_{\mu_j(t)}^{+}
\]
\[
x_r(t) = \arg\max_x \left\{ U_r(x) - x \sum_{j \in r} \mu_j(t) \right\}
\]

At equilibrium:
\[
U'_r(x_r) = \sum_{j \in r} \mu_j \quad \mu_j \gg 0 \iff \sum_{r: j \in r} x_r = C_j
\]

Solve NUM as the price function becomes steeper!
The case of WPF

**PRIMAL:**

\[
\frac{d}{dt} x_r(t) = k_r \left( w_r - x_r(t) \sum_{j \in r} \mu_j(t) \right)
\]

\[
\mu_j(t) = p_j(y_j(t)), \quad y_j(t) = \sum_{s: j \in s} x_s(t)
\]

**DUAL:**

\[
\frac{d}{dt} \mu_j(t) = k_j \left( \sum_{r: j \in r} x_r(t) - C_j \right)_+ \left/ \mu_j \right.
\]

\[
x_r(t) = \frac{w_r}{\sum_{j \in r} \mu_j(t)}
\]

At equilibrium:

\[
x_r = \frac{w_r}{\sum_{j \in r} \mu_j}
\]

Solves WPF!

Think of price = marking prob

Economics of elastic flows - 15
Remarks

- Primal and Dual algorithms solve approximately NUM, exactly as the price function becomes steeper.

NUM: \[
\max_{\{x_r\}} \sum_r U_r(x_r) \text{ s.t. } Ax \leq C, \ x \geq 0
\]

- The dual price corresponds to queuing delay (see next slide), hence it is “naturally implemented”.

- For any choice of the price function, the primal solves

\[
\max_{\{x_r\}} \sum_r U_r(x_r) - \sum_j c_j(\sum_{s: j \in s} x_s), \quad c_j(y) = \int_0^y p_j(z)dz
\]

Hence the price corresponds to a congestion price for some appropriately defined cost => implemented by sample path congestion marks.
Window flow control (WFC)

Route $r$: WFQ with window $w_r$

\[ x_r = \frac{w_r}{T_r} \]

\[ T_r = \sum_{j: j \in r} T_j \]

RTT on route $r$

\[ \text{Delay at link } j \]

\[ T_j(t) \]

\[ r \]

Network generates link delays

- Flows choose windows $w_r$
- Each link posts a delay as a result (due to queuing)
- Rates get regulated by total delay along the route

Any relation with the economic flow control?

- If $U_r(x_r) = w_r \log x_r$, we get exactly the same equations as in WPF!
- WFC achieves $w$-proportional fairness!!
Delay interpretation for the dual algorithm

Assume all link constraints are active, pick \( k_j = 1/C_j \)

Then

\[
\frac{d}{dt} \mu_j(t) = \frac{1}{C_j} \left( \sum_{r \cdot j \in r} x_r(t) - C_j \right), \quad x_r(t) = \frac{w_r}{\sum_{j \in r} \mu_j(t)}
\]

hence \( \mu_j \) has the interpretation of queuing delay at link \( j \)

If propagation delays = 0, then \( x_r(t) = \frac{w_r}{\sum_{j \in r} \mu_j(t)} \) corresponds to window flow control with window \( w_r \), where \( \sum_{j \in r} \mu_j(t) \) is the round trip time on route \( r \)

Hence WFC with windows \( w_r \) achieves WPF flow allocations!

\( w_r = 1 \) corresponds to TCP Vegas!!
Discussion

- Both algorithms reach proportionally fair equilibria

  - Primal algorithm
    - response to congestion signals, TCP-like
    - congestion prices
    - averaging at sources

  - Dual algorithm
    - delay-based, sources estimate delays in queues
    - averaging at resources

- Need to consider stability properties
  - delay instability (high gains – delays)
  - stochastic instability (randomness in marking)
Flow control and economics

An analysis of TCP-like flow control algorithms
Goal

- Characterize flow allocations achieved by TCP-like algorithms (self-clocking)
- Generalize the primal algorithm
- Understand the effects of the various parameters
- Propose way to improve TCP
TCP

**Simple Algorithm:** Increase $cwnd$ with constant rate $a$, when a loss is detected then $cwnd = \frac{1}{2} cwnd$

![Diagram showing the relationship between cwnd and time](image)

$x = \frac{cwnd}{RTT}$

**Implementation:**
For each positive ack: $cwnd := cwnd + \frac{1}{cwnd}$

Hence $a = \frac{1}{RTT}$
TCP capacity sharing

How is fairness achieved by TCP?

For flows with the same rate of increase the capacity is shared fairly.

What happens when the rate of increase is different?

Flows with a large RTT have a smaller rate of increase, hence get smaller share.

Observation: using TCP, the network remain simple internally, but the sharing of the bandwidth depends on the “honesty” of the sources!
Generating congestion signals

- TCP packets are acknowledged
- Loss is interpreted as an indication of congestion
- Better ways to signal congestion: ECN
  - Set a bit in the header of a packet that encounters congestion
  - Endpoints detect ECN marks, react by reducing rates
  - IETF proposed standard
Modelling TCP

- TCP: self-clocking (sender uses acks to step forward)
- Window flow control: \( \text{cwnd} = xT \)
- General model of wfc:
  - positive ack: \( \text{cwnd} := \text{cwnd} + a \text{cwnd}^n \)
  - congestion signal: \( \text{cwnd} := \text{cwnd} - b \text{cwnd}^m \) \( n < m \)
- Expected change in the congestion window per update step
  \[
  \Delta \text{cwnd} = a(xT)^n (1 - p) - b(xT)^m p \quad \Delta t = 1/x = T/cwnd
  \]
- Expected change in the rate \( x \) per unit time
  \[
  \Delta x = \Delta \text{cwnd} / T \cdot (1/x)^{-1} = \frac{x}{T} (a(xT)^n (1 - p) - b(xT)^m p)
  \]
Modelling TCP (2)

Hence use this system of differential equations:

\[
\frac{d}{dt} x_r(t) = \frac{x_r(t)}{T_r} \left( a_r(x_r(t)T_r)^n (1 - p_r) - b_r(x_r(t)T_r)^m p_r \right)
\]

where

\[
p_r = 1 - \prod_{j \in r} (1 - \mu_j) \approx \sum_{j \in r} \mu_j \quad \text{for small } \mu_j \text{s}
\]

Unique equilibrium

\[
x_r = \frac{1}{T_r} \left( \frac{a_r}{b_r} \frac{1 - p_r}{p_r} \right)^{\frac{1}{m-n}} \quad \text{weighted a-fair}
\]

TCP: \[a = 1, b = 1/2, m = 1, n = -1\]

\[
x_r = \frac{1}{T_r} \left( \frac{2}{p_r} \right)^{\frac{1}{2}}
\]
Remarks

- TCP-like mechanisms generalize the primal alg discussed earlier.
- They achieve weighted α-fair allocations:
  - but sharing depends on rtt $T_r$
  - like if flows with large rtts are willing to pay less
  - to remedy this, we must make $a_r, b_r$ depend on $T_r$

- Can we improve TCP?
An improvement: scalable TCP

- Which is the optimal choice of the parameters \( a, b, m, n \) ?
- Stability considerations:
  - delay stability: \( a_r (x_r T_r)^n < \pi / 2 \beta \)
  - delay invariance: \( a_r = \bar{a} r T_r^{1-n} \quad b_r = \bar{b} r T_r^{1-m} \)
  - stochastic stability: \( \text{Var}(x_r(t)) = \frac{b_r x_r^{m+1} T_r^{m-1}}{2\alpha (1 - \lambda_r)} \)  
    \[
    \Rightarrow m = 1
    \]

\[
\begin{align*}
x_r &= \frac{1}{T_r} \left( \frac{a_r}{b_r} \frac{1 - \lambda_r}{\lambda_r} \right)^{1/m-n} \\
&= \frac{w_r \bar{b}}{\lambda_r} \\
&= \frac{\bar{a}}{w_r \bar{b}} \left( \frac{1}{1 - \lambda_r} \right)^{1/m-n}
\end{align*}
\]

A possible choice:

\[
\begin{align*}
0 &= \frac{\bar{a}}{w_r \bar{b}} \\
0 &= \frac{a_r}{w_r T_r} \\
0 &= \frac{\bar{b}}{w_r \bar{b}} \\
0 &= \frac{\bar{a}}{w_r \bar{b}} \left( \frac{1}{1 - \lambda_r} \right)^{1/m-n}
\end{align*}
\]
Conclusions

- We have provided a framework to analyze flow control algorithms
- The primal algorithms can be implemented using existing mechanisms
- The dual algorithms need delay estimations, may have stability issues
Flow control and economics

Reverse engineering flow control algorithms
Congestion control algs

- Engineering components:

\[
\begin{align*}
x_s(t + 1) &= F_s(x_s(t), q_s(t)) \\
\lambda_l(t + 1) &= G_l(y_l(t), \lambda_l(t), v_l(t)) \\
v_l(t + 1) &= H_l(y_l(t), \lambda_l(t), v_l(t))
\end{align*}
\]

- Two types of price calculation: congestion price type (A), tatonnement type (B)
- Examples: ENG(A)

TCP Reno

\[
F_s(t + 1) = \left[ x_s(t) + \frac{1}{T_s^2} - \frac{2}{3} q_s(t)x_s^2(t) \right]^+
\]

RED

\[
\begin{align*}
b_l(t + 1) &= [b_l(t) + y_l(t) - c_l]^+ \\
r_l(t + 1) &= (1 - \omega_l)r_l(t) + \omega_l b_l(t) \\
\lambda_l(t) &= \begin{cases} 
0, & r_l(t) \leq b_l \\
\rho_1(r_l(t) - \overline{b}_l), & \overline{b}_l \leq r_l(t) \leq \overline{r}_l \\
\rho_2(r_l(t) - \overline{b}_l) + M, & \overline{b}_l \leq r_l(t) \leq \overline{r}_l \\
1, & r_l(t) \geq 2\overline{b}_l.
\end{cases}
\end{align*}
\]
More examples

- TCP Vegas

\[
\lambda_l(t + 1) = \left[ \lambda_l(t) + \frac{y_l(t)}{c_l} - 1 \right]^+
\]

\[
x_s(t + 1) = x_s(t) + \frac{1}{T^2_s(t)} \mathbf{1}(\alpha_s d_s - x_s(t)q_s(t))
\]
The reverse engineering theorem

Assume some (plausible) conditions on $F_s, G_l, H_l, R$

Then:

- ENG have unique equilibrium
- ENG(A) solve NUM-P and ENG(B) solves NUM
  for $U_s(x_s) = \int f_s(x_s)dx_s$, where $f_s$

is the implicit function $q_s = f_s(x_s)$ defined by solving $x_s = F_s(x_s, q_s)$

- ENG correspond to different distributed solutions of BNUM (ENG(A)<-primal, ENG(B)<-dual)
Conclusions

- Engineering specifications (ENG) of flow control protocols can be thought to be motivated by (primal or dual) ways to solve a basic network utility maximization problem (NUM) for appropriate choice of utility functions.
- Utility functions defined by the source reaction dynamics.
- They characterize the sharing of resources wrt fairness.
- Generation of pricing signals affect utilization.
- Nice tools to study stability of ENG and design new protocols.
Flow control and economics

Background: Congestion pricing
Congestion pricing

\[ \max_x u(x) - c(x) \]

**Congestion price:** = marginal congestion cost

**optimal point:** marginal utility = marginal congestion cost
Computing congestion prices

1. Congestion charge rate $p_x$ is computed on an average basis

\[ \frac{\partial c}{\partial x} \]

where $x = \text{average flow}$ and $c = \text{average cost}$.

2. Each packet is charged the cost increment that it causes

Assume: cost unit = extra cost caused by a single packet when loss occurs

Packet $a$ is charged the **extra cost** it causes (sender of $a$ receives a congestion mark= 1c)

The rate of charge $p_x$ is averaged on the particular sample path.

In most systems marking prob $p = \frac{\partial c}{\partial x} = p$
Sample path congestion prices

Example: Server that serves up to 10 packets in each time slot
Congestion prices and flow control

- Key idea: use congestion price signals for flow control
- Resources (links) send congestion marks to sources
  - ECN marks, packet losses, other mechanisms
- Sources react to congestion signals (like in TCP)

\[
    y_j(t) = \sum_{s:j \in s} x_s(t)
\]

\[
    p_j(y_j) = \sum_{j \in r} x_j 
\]

\[
    p_k(y_k) = \frac{d}{dy} c_j(y)
\]

Resource \( j \) marks packets with proportion \( p_j \)

Economics of elastic flows - 39
A note on window flow control

Route $r$:

- Assume acks do not queue in buffers

**Little’s Law:** average number of packets in the system = average rate of packets x average time a packet spends in the system

$$W_r = x_rRTT = x_rT_0 + \sum x_rT_i = N_0 + \sum N_i$$

$N_0$ = packets in fiber, $N_i$ = packets in buffer $i$