Optimization Framework for Congestion Control Algorithms

Laila Daniel and Krishnan Narayanan

25th March 2013

Outline of the talk

- Complete the proof of the Primal Congestion Control Algorithm (CCA)
- Dual Algorithm for CCA
- Primal-Dual for CCA
- One-bit marking for congestion control
- REM (Random Early Marking) algorithm
- TCP Reno and its Utility function
- Summary and Future directions for study

Slides on primal and dual CCA based on the paper by Kelly et.al. Rate control in communication networks: shadow prices, proportional fairness and stability (P13)
Slides on TCP Reno and utility functions based on the paper by Shakkottai et.al. Network Optimization and Control (P16)

Goal of Congestion Control

- The goal of the congestion control algorithm is to provide the rate allocation of the SYSTEM OPTIMUM RATES (obtained by solving the network optimization problem) as the EQUILIBRIUM RATES when all the flows have enough data to send
- The mechanism should be STABLE for this equilibrium rates and should converge to this equilibrium no matter what the initial state is
- The network equilibrium is characterized by some fairness criterion
- How quickly the algorithm can converge to this equilibrium without large oscillations of the network allocated rates is also an important question
- These are some considerations in the design of a congestion control algorithm

Optimal bandwidth sharing

- Optimal bandwidth sharing is got by solving the following utility maximization problem:

$$\text{Max } \sum_{s \in S} U(x_s)$$

$$\text{Subject to } \sum_{s \in S_l} x_s \leq C_l, \forall l \in L$$

$$x_s \geq 0, \forall s \in S$$

- A unique maximizer called primal optimal solution exists as the objective function is strictly concave and the feasible region is a compact convex set
- A rate vector **x** will be optimal iff KKT holds at this rate vector.
- Now use KKT to get the following relationships between the optimal rates x_s and the dual variables $p_l, l \in L$

System Model: Notation

- Based on the paper: Kelly etal. Rate control for communication networks (P13 in the course page)
- A network with J resources, C_j finite capacity of the resource j, for $j \in J$
- *route* r a non-empty subset of J. R set of possible routes. Associate a route r with a user
- $A_{i,j} = 1$ if $j \in r$, j lies on route r and $A_{i,j} = 1$ otherwise
- A rate x_r is allocated to user r and x_r has a utility $U_r(x_r)$
- Assume $U_r(x_r)$ is increasing, strictly concave and continuously differentiable and utilities are additive,
- Aggregate utility rates $x = (x_r, r \in R)$ is $\sum_{r \in R} U_r(x_r)$
- $U = (U_r(.), r \in R)$ $C = (c_j, j \in J)$

System Model

- $SYSTEM(U, A, C)$:

$$max \sum_{r \in R} U_r(x_r)$$

  subject to

$$Ax \leq C$$

  over

$$x \geq 0$$

- These equations say that the system objective is to maximize the total utility of all the users in the network subject to the link capacity constraints
- This system optimization problem is also called the NUM (Network Utility Maximization) problem
- Here we seek to maximize a concave utility function over a convex feasible region defined by the linear constraints

Decomposing the System Model for distributed solution

- We seek a distributed solution to the NUM problem as a centralized solutions are impractical for large scale networks
- We show that the NUM problem can be decomposed into subproblems for users and a problem for the Network
- w_r be the amount user r chooses to pay per unit time
- User r gets a flow rate x_r in proportion to w_r
- $x_r = \frac{w_r}{\lambda_r}$, λ_r charge per unit flow for user r
- $USER_r(U_r;\lambda_r)$:

$$max \; \left(U_r\left(\frac{w_r}{\lambda_r}\right) - w_r \right)$$

$$\text{over} \quad w_r \geq 0$$

- The above model says that user r tries to maximize its utility 'minus' cost which is to maximize its selfish goal

Network Subproblem

- $NETWORK(A, C; w)$:

$$max \sum_{r \in R} w_r \, log \, x_r$$

subject to

$$Ax \leq C$$

$$\text{over} \quad w_r \geq 0$$

- Network knows the vector the vector $w = (w_r, \, r \in R)$, w_r what user chooses to pay per unit time
- Network does not know the utility of each user and uses $log \, x_r$ to represent users utility with weight w_r
- Network tries to maximize the weighted log utilities of the users in the network subject to link capacity constraints

Solution Concept

- $SYSTEM(U, A, C) = NETWORK(A, C; w) + USER_r(U_r; \lambda_r)$
- There exists vectors $\lambda = (\lambda_r,\ r \in R)$, $w = (w_r,\ r \in R$ and $x = (x_r,\ r \in R)$ such that $w_r = \lambda_r x_r$
  - w_r solves the $USER_r(U_r; \lambda_r)$ for $r \in R$
  - x solves the $NETWORK(A, C; w)$
  - furthermore, x is the unique solution to the $SYSTEM(U, A, C)$
- This means that the congestion pricing of the network balances the selfish goals of the users with the social goal of the network.
- This **equilibrium** is attained under proportional fairness criterion for resource allocation which **aligns selfish optimum objective** of the individual users with the **social optimum objective** of the network

Primal Algorithm

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

where

$$\mu_j(t) = p_j \left(\sum_{s:j \in S} x_s(t) \right)$$

- $x_r(t)$: flow on route r
- $w_r(t)$: what the user r willing to pay per unit time for its flow
- $p_j(y)$: price charged per unit flow by the resource j through the resource j when the total flow through the resource j is y
- $\mu_j(t)$: price charged per unit flow by the resource j at time t
- The differential equation tries to equalize the total cost of the flow against the price the user w_r the user is prepared to pay

Primal Algorithm (contd)

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

where

$$\mu_j(t) = p_j \left(\sum_{s:j \in S} x_s(t) \right)$$

- This equation corresponds to a rate control algorithm for user r which has two components
- Steady increase at a rate proportional to $w_r(t)$
- Steady decrease proportional to the stream of congestion indication signals received
- Note that this equation corresponds to AIMD model of adjusting the source rates
- Key assumptions: time lags and stochastic perturbations ignored in the model

Lyapunov function and the Global stability of the Primal algorithm

- Lyapunov function is obtained from the system objective of the NUM problem
- More precisely it is a relaxation of the optimization problem $SYSTEM(U, A, C)$
- Lyapunov function, a global view of the local description given by the primal algorithm
- Lyapunov function shows that the primal algorithm converges to a unique equilibrium point given by the system optimum.
- Intuitively, Lyapunov function can be regards as a 'hill' and the primal algorithm advances every step to reach the 'top'

Lyapunov function of the primal algorithm

$$\mathcal{U}(x) = \sum_{r \in R} w_r \, log \, x_r \; - \sum_{j \in J} \int_0^{\sum\limits_{s:j \in s} x_s} p_j(y) dy$$

- We can tidy up the appearance of the above equation by introducing the quantity $C_j(y)$
- Let $C_j(y)$ is the rate at which cost is incurred at resource j when the load through it is y
- $\frac{d}{dy} C_j(y) = p_j(y)$
- $p_j(y)$ strictly increasing function of y bounded above by unity

$$\mathcal{U}(x) = \sum_{r \in R} w_r log x_r \; - \sum_{j \in J} C_j \left(\sum_{s:j \in s} x_s \right)$$

Network interpretation of the Lyapunov function $\mathcal{U}(x)$

$$\mathcal{U}(x) = \sum_{r \in R} w_r log x_r \; - \sum_{j \in J} C_j \left(\sum_{s : j \in s} x_s \right)$$

- $\mathcal{U}(x)$ can be interpreted as the weighted sum of the log utilities of the users minus the total congestion price incurred by the network resources (links) in routing the flows through them.
- So, maximizing $\mathcal{U}(x)$ can be regarded as the network maximizing its (benefits minus cost)

Equilibrium point corresponds to maximizing the Lyapunov function $\mathcal{U}(x)$

$$\mathcal{U}(x) = \sum_{r \in R} w_r \, log \, x_r \, - \sum_{j \in J} \int_0^{\sum\limits_{s:j \in s} x_s} p_j(y) dy$$

- $\mathcal{U}(x)$ is strictly concave, its local maximum coincides with the global maximum (implication of maximizing a concave function over a convex set)
- At the local maximum, $\nabla_x \mathcal{U}(x) = 0$, so we get

$$\frac{\partial}{\partial x_r} \mathcal{U}(x) = \frac{w_r}{x_r} - \sum_{j \in r} p_j \left(\sum_{s:j \in s} x_s \right) = 0 \, \forall r \in R$$

- We next solve the equation to get x_r corresponding to $\max(\mathcal{U}(x))$

Equilibrium rates obtained from maximizing $\mathcal{U}(x)$

$$\frac{\partial}{\partial x}\mathcal{U}(x) = \frac{w_r}{x_r} - \sum_{j\in R} p_j\left(\sum_{s:j\in s} x_s\right) = 0$$

- So the equilibrium rate x_r is given by

$$x_r = \frac{w_r}{\sum_{j\in R} p_j\left(\sum_{s:j\in s} x_s\right)}$$

- Note that the equilibrium value of x_r is precisely the quantity obtained by setting the primal rate control equation x_r to zero

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

The rate control algorithm climbs the Lyapunov hill to reach the top

$$\frac{d}{dt}\mathcal{U}(x(t)) = \sum_{r \in R} \frac{\partial \mathcal{U}}{\partial x_r} \times \frac{d}{dt} x_r(t)$$

$$= \kappa \sum_{r \in R} \frac{1}{x_r(t)} \left(w_r - x_r(t) \sum_{j \in r} p_j \left(\sum_{s: j \in s} x_s(t) \right) \right)^2$$

- This shows that $\mathcal{U}(x(t)$ is strictly increasing with t unless $x(t)$ corresponds to x which is the unique x maximizing $\mathcal{U}(x(t)$
- So we conclude that the primal algorithm steadily climbs the hill till it reaches the top.

Dual Algorithm - $DUAL(A, c; w)$

- The term PRIMAL Algorithm refers to a congestion control algorithm with dynamics at the sources (as given by a differential equation) and each link computes its congestion price based on the total arrival rates at that link.
- The congestion price at a link is also called 'shadow price' which is nothing but the Lagrangian multiplier corresponding to the link capacity constraint inequality in the optimization problem.
- The optimization problem here has 'zero duality gap', so both the original NUM problem and its dual yield the same optimal value.
- In the dual algorithm the dynamics is implemented at the link and the sources adjusts their rates as functions of the aggregate congestion prices.

Dual Algorithm

$$\frac{d}{dt}\mu_j(t) = \kappa \left(\sum_{r:j \in r} x_r(t) - q_j(\mu_j(t)) \right)$$

$$\text{where } x_r(t) = \frac{w_r}{\sum_{k \in r} \mu_k(t)}$$

- Here $\mu_j(t)$ is the congestion price of link j at time t.
- q_j maps the congestion price of link j to the corresponding arrival rate at the link.
- So $q_j(\eta)$ is the arrival rate at link j corresponding to the congestion price η at the link j
- The RHS of the link dynamics equation gives the excess demand over supply and this determines the price adjustment at the link.

Lyapunov function of Dual Algorithm

$$\psi(\mu) = \sum_{r \in R} w_r \, log \left(\sum_{j \in r} \mu_j \right) - \sum_{j \in J} \int_0^{\mu_j} q_j(\eta) d\eta$$

- $\psi(\mu)$ is the Lyapunov function of the dual algorithm
- Using it we can show that the vector μ maximizing $\psi(\mu)$ is a stable point of the system to which the system dynamics converges.
- $\psi(\mu)$ can be suitably chosen to closely approximate DUAL (A,c;w)

Economics viewpoint of the Dual Algorithm

- The dual algorithm can be described in economics parlance as 'A tatonnement process' that adjusts the price according to supply and demand
- Roughly it means a market in which buyers with limited cash seek to buy goods from sellers with goods to sell according to their own valuations of the goods.
- Buyers buy goods to maximize their own benefits from the goods and the pricing of the goods is such that equilibrium attained when the buyers exhaust all their cash buying the goods and the sellers sell all their goods.
- CCP rate control algorithms can be regarded as 'Walrasian auctioneer' searching for market clearing prices.

Primal-Dual Algorithm

- Here dynamics (differential equations) is implemented in both the sources and the links

$$\frac{dx_r}{dt} = \kappa_r(x_r)(\mathcal{U}'_r(x_r) - q_r)$$

- Link prices are adjusted according to the differential equation

$$\frac{dp_l}{dt} = h_l(p_l)(y_l - c_l)^+_{p_l}$$

- $\mathcal{U}'_r$ is the derivative of the utility function of user r
- q_r: the route price/unit flow on route r. i.e., $q_r = \sum_{l:l\in r} p_l$
-

$$\text{Definition :} \quad (g(x))^+_y = \begin{cases} g(x) & if \ y > 0 \\ max(g(x), 0), & y = 0 \end{cases}$$

- $\kappa_r(x_r)$ and h_l are the scaling functions that determine the sensitivity of response and their values influence the stability of the system in the presence of delay
- Via Lyapunov function the primal-dual algorithm can be shown to be asymptotically stable.

One-bit marking approach to congestion price

- The primal algorithm for a source needs the *total* congestion price on its route to adjusts its rate.
- Internet provides only one-bit of information per packet to convey link prices back to the source (e.g., ECN marks, a technology dependent issue).
- How to convey the pricing information with such limited feedback?
- Solution: Link l marks a packet with a probability p_l that reflects its congest ion price.
- Marking is implemented by setting a special bit in the packet header to indicate congestion

Marking Scheme

- q_r denotes the probability that a packet is marked on route r.

$$q_r = \prod_{l:l \in r} (1 - p_l)$$

- The source algorithm is to increase the sending rate only when an unmarked packet is received. The source algorithm is as follows.

$$\frac{dx_r}{dt} = k_r(x_r)((1 - q_r)\mathcal{U}_r'(x_r) - q_r)$$

- By suitably choosing the Lyapunov function, the primal algorithm is shown to be stable.
- An assumption here is that link prices are scalars lying between 0 and 1 and so can be interpreted as probabilities.

REM (Random Early Marking): a one-bit marking Scheme

- We relax the earlier assumption in the 1-bit marking scheme that the link prices lie between 0 and 1. We allow link prices to be arbitrary non-negative real numbers.
- Goal: To convey the path price q_r given by $q_r = \sum_{l:l \in r} p_l$ using just one bit per packet.
- Suppose each link l marks each packet passing through it with a probability $(1 - e^{-\gamma p_l})$ for a fixed $\gamma > 0$
- As link price p_l increases, the marking probability of a packet increases.
- When a packet traverses all the links in route r it is not marked only if it is not marked by any of the links in its path.
- The probability that a packet is not marked on its route r is given by
$$\prod_{l:l \in r} e^{-\gamma p_l} = e^{-\gamma \sum_{l \in r} p_l} = e^{-\gamma q_r}$$
- By measuring the fraction of the packets that not marked, each source can infer its congestion price on its route.

TCP Algorithm and its Utility function

- Assumption: Bulk of the data transfer in TCP done in the congestion avoidance phase (CA).
- We model the CA phase of TCP for representing the TCP dynamics for resource allocation and view it as a gradient-like algorithm.
- Let $W_r(t)$: window size, T_r: RTT and q_r: fraction of the packets lost at time t.
- CA phase of TCP Reno can be modelled as

$$\frac{dW_r}{dt} = \frac{x_r(t - T_r)(1 - q_r(t))}{W_r} - \beta x_r(t - T_r) q_r(t) W_r(t)$$

- $x_r(t) = W_r(t)/T_r$ is the transmission rate, β decrease factor is taken as $1/2$. Here $W_r(t)$ is the window size of the source r at time t.
- The logic underlying this equation is described next

TCP Algorithm and its Utility function - contd

$$\frac{dW_r}{dt} = \frac{x_r(t - T_r)(1 - q_r(t))}{W_r} - \beta x_r(t - T_r)q_r(t)W_r(t)$$

- Each ACK leads to an increase by $1/W(t)$. This explains the first term of the equation. As this causes the window to increase by 1 per RTT, it results in an additive increase of window.
- Similarly the rate at which the packets are lost is $x_r(t - T_r)q_r(t)$. Each loss event would cause the rate of transmission to be decreased proportional to the current window size $W(t)$(multiplicative decrease). The constant of proportionality β has a typical value $1/2$
- To obtain the corresponding rate, we substitute $W_r = x_r T_r$ in the above equation

$$\frac{dx_r}{dt} = \frac{x_r(t - T_r)(1 - q_r(t))}{T_r^2 x_r} - \beta x_r(t - T_r)q_r(t)x_r(t)$$

TCP Algorithm and its Utility function - contd

- The equilibrium value of x_r can be obtained by setting $\frac{dx_r}{dt} = 0$, we get

$$\hat{x}_r = \sqrt{\frac{1-\hat{q}}{\beta\hat{q}}}\frac{1}{T_r}$$

- $\hat{q}$ is the loss probability. For small values of $\hat{q}$ we get

$$x_r \propto \frac{1}{T_r\sqrt{\hat{q}_r}}$$

- This is the well-known result on the steady-state behaviour of TCP Reno.
- If the RTT is negligible, We can approximate the equation as

$$\frac{dx_r}{dt} = \frac{1-q_r}{T_r^2} - \beta x_r^2 q_r$$

TCP Algorithm and its Utility function - contd

$$\frac{dx_r}{dt} = \frac{1 - q_r}{T_r^2} - \beta x_r^2 q_r$$

- Comparing to the primal algorithm formulation in NUM

$$\frac{dx_r}{dt} = \kappa_r(x_r)(\mathcal{U}_r'(x_r) - q_r)$$

$$\frac{dx_r}{dt} = (\beta x_r^2 + \frac{1}{T_r^2}\left(\frac{1}{\beta T_r^2 x_r^2 + 1} - q_r\right)$$

- We get

$$\mathcal{U}_r'(x_r) = \frac{1}{\beta T_r^2 x_r^2 + 1}$$

- We get TCP Reno utility function as

$$\mathcal{U}_r(x_r) = \frac{arctan(x_r T_r \sqrt{\beta}}{\sqrt{\beta T_r}})$$

TCP Algorithm and its Utility function - contd

- For small q_r, a similar approximation of x_r with small RTT, T_r leads to the utility

$$\mathcal{U}_r(x_r) = -\frac{1}{x_r T_r}$$

- The above utility function shows that TCP solves weighted minimum potential delay (MPD) fair resource allocation problem.

Many Questions and Some Answers

- The negligible delay assumption in the NUM model and its implications to the stability
- Study the effect of stochastic perturbations, inaccurate measurements, the impact of uncontrolled flows etc on the stability of the model
- Can the model be refined to support QoS based on end-to-end source adaptations only

Directions for further study/research

- The NUM philosophy has been widely extended to study network architectures for both wireless and wireline networks.
- See Chiang et.al paper ' Layering as optimization decomposition ...'
- Game theory, Economics and Algorithms have combined to yield many interesting results in a field Algorithmic Game Theory (also called Mechanism Design)
- See the reference Nisan etal Algorithmic Game Theory book which is available online

Summary of the CCP lectures

- Chiu-Jain Model: CCA as resource sharing principle in networks
  - Choose a Network Operating Point (NOP) that is efficient and fair (in sharing network resources)
  - CCA should converge to NOP, i.e, equilibrium structure of the protocol should coincides to the NOP
- Local-Global Principle
  - In analogy to the least action principle in Mechanics, design Microscopic rules (local) which correspond to Macroscopic laws (global). Often this correspondence is governed by some optimization principles
- Introduction to optimization framework using a simple linear network
  - The notion of utility function that captures the valuation of a source for resource
  - Utility functions can capture fairness
  - Continuum of fairness corresponding to the parameter $\alpha \in (0, \infty)$
  - Proportional fairness (PF) and log utility function
  - In PF, source rate inversely proportional to the congestion price (Lagrangian multipliers) on its route

Summary of the CCP lectures - contd

- Kelly framework
  - The optimization problem of NUM for congestion control can be decomposed into subproblems solved by each SOURCE and each LINK in the network
  - Congestion control algorithm can be interpreted as a gradient algorithm for the NUM problem, to get primal or dual forms of the algorithm
  - Pricing interpretation of the Lagrangian multipliers (dual variable)
  - The operating point of the network is given by the equilibrium between the users willingness to pay (in price per unit time) and the system allotted rates computed (packets/unit price) are in equilibrium
  - The equilibrium point is described using the some fairness criterion
  - The equilibrium point is socially optimal and source algorithm corresponds to selfish optimum
  - Prices align social optimum with selfish optimum