Notice of Violation of IEEE Publication Principles

"A edge to edge capacity contracting congestion pricing implementation in IP networks"
by Kai Mao
in the 2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), 2010, pp. 1-4.

After careful and considered review of the content and authorship of this paper by a duly constituted expert committee, this paper has been found to be in violation of IEEE's Publication Principles.

This paper contains significant portions of original text from the paper cited below. The original text was copied with insufficient attribution (including appropriate references to the original author(s) and/or paper title) and without permission.

Due to the nature of this violation, reasonable effort should be made to remove all past references to this paper, and future references should be made to the following articles:

"Congestions Pricing Overlaid on Edge-to-Edge Congestion Control"
by Murat Yuksel, Shivkumar Kalyanaraman and Anuj Goel
in IEEE International Conference on Communications 2003 (ICC 2003), Vol. 2, 2003, 880 – 884.

"Distributed dynamic capacity contracting: an overlay congestion pricing framework"
by Murat Yuksel, Shivkumar Kalyanaraman
in Computer Communications 26, 2003, pp. 1484 – 1503.

A edge to edge capacity contracting congestion pricing implementation in IP networks

Kai Mao
College of Computer Science and Information Engineering,
Chongqing Technology and Business University, Chongqing,
PR China
Maokai_cq@163.com

***Abstract*—Several congestion pricing proposals have been made in the last decade. Usually, however, those proposals studied optimal strategies and did not focus on implementation issues. Our main contribution in this paper is to address implementation issues for congestion-sensitive pricing. We propose a new congestion-sensitive pricing framework edge to edge capacity contracting (EECC), which is able to provide a range of fairness (e.g. max–min, proportional) in rate allocation by using pricing as a tool. We develop a pricing scheme within the EECC framework and investigate several issues such as optimality of prices, fairness and stability. Then, we adapt EECC in current IP networks and evaluate it by extensive simulation.**

Keywords-Congestion pricing; EECC; implementation

I. INTRODUCTION

Implementation of congestion pricing still remains a challenge in computer network, although several proposals have been made, e.g. Refs. [1–3]. Among many others, two major implementation obstacles can be defined: need for timely feedback to users about the price, determination of congestion information in an efficient, low-overhead manner.

The first problem, timely feedback, is relatively very hard to achieve in a wide area network such as the Internet. In Ref. [4], the authors showed that users do want feedback about charging of the network service (such as current price and prediction of service quality in near future). However, in Ref. [5], the authors illustrated that congestion control by pricing cannot be achieved if price changes are performed at a time-scale larger than roughly 40 roundtrip-times (RTTs). This means that in order to achieve congestion control by pricing, service prices must be updated very frequently (i.e. 2–3 s since RTT is expressed in terms of milliseconds for most cases in the Internet).

The second problem, congestion information, is also very hard to solve in a way that does not require a major upgrade at network routers[6].

In order to solve these problems above, we propose edge to edge capacity contracting (EECC) pricing solutions.

EECC pricing architectures is shown in Fig.1. EECC models a short-term contract for a given traffic class as a function of price per unit traffic volume P_v; maximum volume V_{max} (maximum number of bytes that can be sent during the contract) and the term of the contract T (length of the contract):

$$Contract = f(P_v, V_{max}, T). \quad (1)$$

Customers can only access network core by making contracts with the provider stations placed at the edge routers. The stations offer contracts (i.e. Vmax and T) to fellow users. Notice that, in EECC framework, provider stations can implement dynamic pricing schemes. Particularly, they can implement congestion-based pricing schemes, if they have actual information about congestion in network core. This congestion information can come from the interior routers or from the egress edge routers depending on the congestion-detection mechanism being used. EECC assumes that the congestion detection mechanism is able to give congestion information in time scales smaller than contracts.

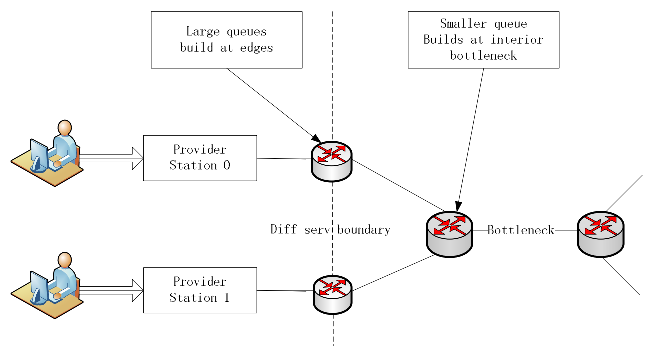


Fig. 1. EECC pricing architectures

In EECC, the prices are calculated on an edge to-edge basis, while traditionally it has been proposed that prices are calculated at each local link and fed back to users. Basically, the links on a flow's route are abstracted out by edge-to-edge capacity estimation and the ingress node communicates with the corresponding egress node to observe congestion on the route. Then, the ingress node uses the estimated capacity and the observed congestion information in price calculation. However, in other congestion pricing framework [10-12], each link calculates its own price and sends it to the user, and the user pays the aggregate price. So, EECC is better in terms of implementation requirements.

II. RELATED WORK

There have been several pricing proposals, which can be classified in many ways: static vs. dynamic, per-packet charging vs. per-contract charging, and charging a priori to service vs. a posteriori to service. Examples of dynamic pricing proposals are Gupta et al.'s Priority Pricing [13], Kelly et al.'s Proportional Fair Pricing (PFP) [8], Semret et al.'s Market Pricing [3,14], and Wang and Schulzrinne's Resource

This research work was supported in part by the National Natural Science Foundation of China (No. 60873201) and the Natural Science Foundaton of Chongqing (No. CSCT 2008BB2189, 2007BB2386), China.

Negotiation and Pricing (RNAP) [2,15]. Odlyzko's Paris Metro Pricing (PMP) [10] is an example of static pricing proposal. Clark's Expected Capacity [16] and Cocchi et al.'s Edge Pricing [9] allow both static and dynamic pricing. In terms of charging granularity, Priority Pricing, PFP employ per-packet charging, whilst RNAP and Expected Capacity employ per-contract charging.

Our work, EECC, is a middle-ground between Priority Pricing and Expected Capacity in terms of granularity. It performs congestion pricing at short-term contracts, which allows more dynamism in prices while keeping pricing overhead small. In the area, another proposal that mainly focused on implementation issues of congestion pricing on diff-serv is RNAP [2,15]. Although RNAP provides a complete picture for incorporation of admission control and congestion pricing, it has excessive implementation overhead since it requires all network routers to participate in determination of congestion prices. This requires upgrades to all routers. We believe that pricing proposals that require upgrades to all routers will eventually fail in implementation phase. This is because of the fact that the Internet routers are owned by different entities who may or may not be willing to cooperate in the process of router upgrades. Our work solves this problem by requiring upgrades only at edge routers rather than at all routers.

III. EECC FRAMEWORK

EECC framework is specifically designed for diff-serv environment. Each edge router is treated as a station of the provider and advertises locally computed prices with information received from other stations. The main framework basically describes how to preserve coordination among the stations such that stability and fairness of the overall network is preserved.

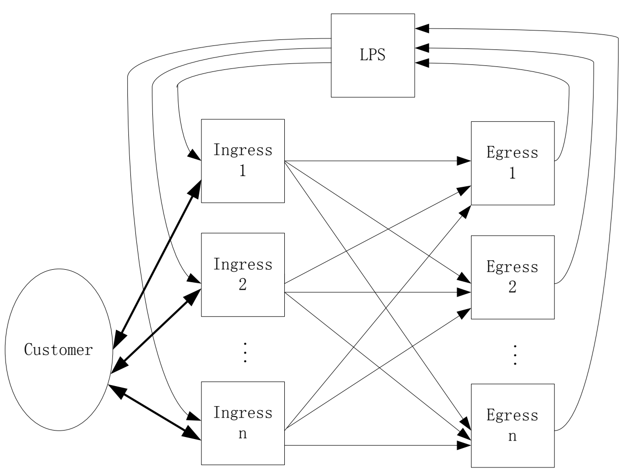


Fig. 2. Components of EECC framework

EECC framework has three major components as shown in Fig. 2: Logical Pricing Server (LPS), Ingress Stations, and Egress Stations. Solid lined arrows in the figure represent control information being transmitted among the components. Basically, Ingress stations negotiate with customers, observe customer's traffic, and make estimations about customer's demand. Ingress stations inform corresponding Egress stations about the observations and estimations about each edge-to-edge flow.

Egress stations detect congestion by monitoring edge-to edge traffic flows. Based on congestion detections, Egress stations estimate available capacity for each edge-to-edge flow, and inform LPS about these estimations.

LPS receives capacity estimations from Egress stations, calculate the congestion price and allocates the network available capacity to edge-to-edge flows according to different criteria (such as fairness, price optimality).

Below, we describe functions and sub-components of these three components in detail.

A. *Ingress station*

Ingress station includes two sub-components: Pricing Scheme and Budget Estimator. It keeps a 'current" price p_{ij}; where p_{ij} is the price for the flow from ingress i to egress j. Pricing Scheme is the sub-component that calculates price p_{ij} for each edge-to-edge flow starting at Ingress i: It uses allowed flow capacities c_{ij} and other local information (such as Estimation for budget of flow from i to j), in order to calculate price p_{ij}: The station, then, uses p_{ij} in negotiations with customers.

Also, the ingress station uses the total estimated network capacity C in calculating the V_{max} contract parameter defined in Eq. (1). We use a simple method which does not put any restriction on V_{max}; i.e. $V_{max} = C*T$, where T is the contract length.

Budget Estimator is the sub-component that observes demand for each edge-to-edge flow. We implicitly assume that user's 'budget' represents user's demand. Budget Estimator performs a very trivial operation to estimate budgets b_{ij} of each flow. The Ingress station basically knows its current price for each flow, p_{ij}. By monitoring the packets transmitted for each flow, Budget Estimator can estimate the budget of each flow. Let x_{ij} be the total number of packets transmitted for flow i to j in unit time, then the budget estimate for the flow i to j is $b_{ij} = x_{ij}p_{ij}$.

B. *Egress station*

Sub-components of Egress station include: Congestion Detector and Congestion-Based Capacity Estimator.

Congestion Detector implements an algorithm to detect congestion in network core by observing traffic arriving. Congestion detection can be done in several ways. We assume that interior routers mark (i.e. sets the ECN bit) the data packets if their local queue exceeds a threshold. Congestion Detector generates a 'congestion indication' if it observes a marked packet in the arriving traffic.

Congestion-Based Capacity Estimator estimates available capacity c_{ij} for each edge-to-edge flow exiting at Egress station: In order to calculate c_{ij}, it uses congestion indications from Congestion Detector and actual output rates μ_{ij} of the flows. The crucial property of Congestion-Based Capacity Estimator is that, it estimates capacity in a congestion-based manner, i.e. it decreases the capacity estimation when there is congestion indication and increases when there is no congestion indication. Basically, an observation interval is congested if a congestion indication was received from Congestion Detector during that observation interval. At the end of each observation interval t; Congestion-Based Capacity Estimator updates the estimated capacity c_{ij} as follows:

$$c_{ij}(t)=\begin{cases}\beta\cdot\mu_{ij}(t), & congested\\ c_{ij}(t-1)+\Delta c, & non-congested\end{cases}, \quad (2)$$

where β is in (0,1), $\mu_{ij}(t)$ is the measured output rate of flow i to j during observation interval t; and Δc is a pre-defined increase parameter.

C. Logical Pricing Server(LPS)

LPS receives information from egresses and calculates congestion price for each edge-to-edge flow. LPS allocates capacity to edge-to-edge flows based on their budgets. The flows with higher budgets are given more capacity than the others.

At first for ingress i to egress j flow f_{ij}, EECC framework provides an allowed capacity c_{ij} and an estimation of total user budget b_{ij} at ingress i. LPS can use these information to calculate price. We propose a simple price formula to balance supply and demand:

$$p_{ij}=b_{ij}/c_{ij}\,. \quad (3)$$

Here, b_{ij} represents user demand and c_{ij} is the available supply.

Further, to achieve the fairness objectives, we adopt new parameters for tuning rate allocation to flows. In order to adjust flow i to j; LPS can increase or reduce flow rate. It uses the following formula to do so:

$$r_{ij}(t)=f(r_{ij}(t-1),\alpha,p_{\min})=\frac{r_{ij}(t-1)}{p_{\min}+(p_{ij}(t)-p_{\min})\alpha}, \quad (4)$$

where $r_{ij}(t)$ is the flow rate from i to j; p_{min} is the minimum possible congestion price for the flow, and α is fairness coefficient. When α is 0, LPS is employing max–min fairness. As it gets larger, rate allocation gets closer to proportional fairness.

IV. SIMULATION EXPERIMENTS AND RESULTS

We now present simulation experiments for EECC by ns2, on IP networks topology. Our goals are to illustrate fairness and stability properties.

A. Experimental configuration

The network topology has a bottleneck link, which is connected to n edge nodes at each side where n is the number of users. There are n ingress and n egress edge nodes. The bottleneck links have a capacity of 10 *Mb*/*s* and all other links have 15 *Mb*/*s*. Propagation delay on each link is 5 *ms*, and users send UDP traffic with an average packet size of 1000*B*. To ease understanding the experiments, each user sends its traffic to a separate egress. The queues at the interior nodes (i.e. nodes that stand at the tips of bottleneck links) mark the packets when their local queue size exceeds 30 packets. Fig. 3 shows network topology with n=3. The user flow tries to maximize its total utility by contracting for b/p amount of capacity, where b is its budget and p is congestion price.

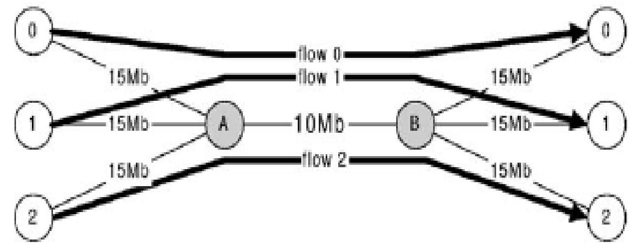


Fig. 3. the network topology for simulation

B. Experiments on single-bottleneck topology

We run simulation experiments for EECC and in this experiment; there are three users with budgets of 30 *Mb*/*s*, 20 *Mb*/*s*, 10 *Mb*/*s*, respectively, for users 1, 2, 3. Total simulation time is 15,000 s, and at the beginning only the user 1 is active in the system. After 5000 s, the user 2 gets active. Again after 5000 s at simulation time 10,000, the user 3 gets active. And the edge queues mark the packets when queue size exceeds 200 packets.

In terms of results, the volume given to each flow is very important. Figs. 4a show the volumes given to each flow in EECC. We see the flows are sharing the bottleneck capacity in proportion to their budgets.

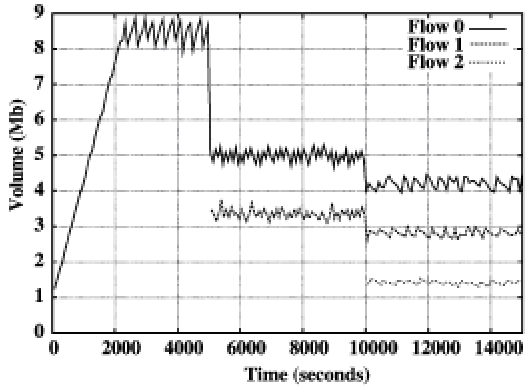
Figs. 4b show the price being advertised to flows. As the new users join in, the pricing scheme increases the price in order to balance supply and demand.

Figs. 4c show the bottleneck queue size. Notice that queue sizes make peaks transiently at the times when new users get active. Otherwise, the queue size is controlled reasonably and the system is stable. We can conclude that EECC manages the bottleneck queue much better because of the tight control enforced by the underlying edge-to-edge congestion control algorithm.

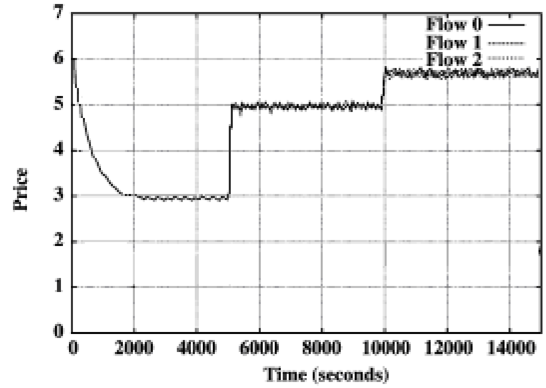
Figs. 4d show estimation of available capacity for flow i to j. Compared with Figs. 4a, similarly, with new users coming, the c_{ij} decreased smoothly.

V. CONCLUSION

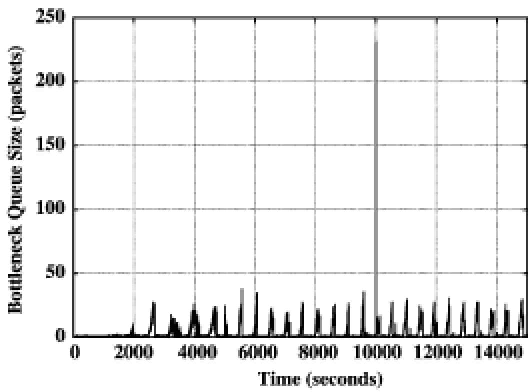
In this paper, we presented a new framework, EECC, for congestion pricing. EECC can provide a contracting framework based on short-term contracts between user application and the service provider. Since contracts are short-term, it becomes possible to update prices frequently and hence to advertise dynamic prices. Particularly, on a totally edge-to-edge basis, we described ways of calculating congestion-based prices, which enables congestion pricing in the proposed EECC framework. By extensive simulations, we demonstrated that EECC's edge-to-edge capacity allocation algorithm has dominant effects especially when ratio of flows' budgets is large.



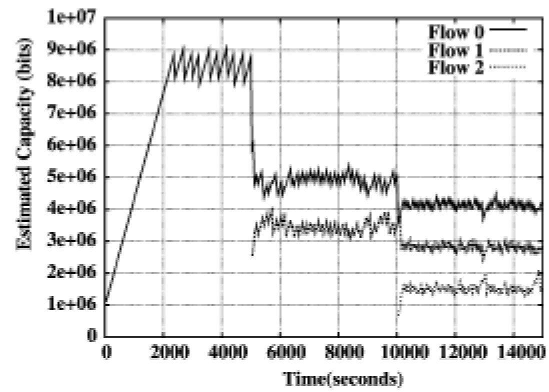
(a) Flow rates



(b) Prices



(c) Bottleneck queue



(d) c_{ij} of flows

Fig. 4. Results of IP networks experiment for EECC

REFERENCES

[1] J.K. MacKie-Mason, H.R. Varian, Pricing the Internet, Kahin, Brian and Keller, James, 2003.

[2] X. Wang, H. Schulzrinne, An integrated resource negotiation, pricing, and QoS adaptation framework for multimedia applications, IEEE Journal of Selected Areas in Communications 18, 2000.

[3] N. Semret, R.R.-F. Liao, A.T. Campbell, A.A. Lazar, Pricing, provisioning and peering: dynamic markets for differentiated Internet services and implications for network interconnections, IEEE Journal of Selected Areas in Communications 18, 2000.

[4] A. Bouch, M.A. Sasse, Why value is everything? A user-centered approach to Internet quality of service and pricing, Proceedings of theIEEE/IFIP International Workshop on Quality of Service (IWQoS) (2001).

[5] M. Yuksel, S. Kalyanaraman, B. Sikdar, Effect of pricing intervals on congestion-sensitivity of network service prices, Technical Report ECSE-NET-2002-1, Rensselaer Polytechnic Institute, ECSE Networks Lab, 2002.

[6] S. Blake, O. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An architecture for Differentiated Services, IETF RFC 2475.

[7] S. Kunniyur, R. Srikant, End-to-end congestion contro: utility functions, random losses and ecn marks, Proceedings of the Conference on Computer Communications (INFOCOM) (2000).

[8] R. Cocchi, S. Shenker, D. Estrin, L. Zhang, Pricing in computer networks: Motivation, formulation and example, IEEE/ACM Transactions on Networking 1.

[9] A.M. Odlyzko, The economics of the Internet: utility, utilization, pricing, and quality of service, Technical Report, AT&T Labs, 2005.

[10] A.M. Odlyzko, Internet pricing and history of communications, Technical Report, AT&T Labs, 2000.

[11] I.C. Paschalidis, J.N. Tsitsiklis, Congestion-dependent pricing of network services, IEEE/ACM Transactions on Networking 8 (2) (2006) 171–184.

[12] A. Gupta, D.O. Stahl, A.B. Whinston, in: McKnight, Bailey (Eds.), Priority Pricing of Integrated Services Networks, MIT Press, Cambridge, 2007.

[13] N. Semret, R.R.-F. Liao, A.T. Campbell, A.A. Lazar, Market pricing of differentiated Internet services, Proceedings of the IEEE/IFIP International Workshop on Quality of Service (IWQoS) (1999) 184–193.

[14] X. Wang, H. Schulzrinne, Pricing network resources for adaptive applications in a Differentiated Services network, Proceedings of the Conference on Computer Communications (INFOCOM) (2001).

[15] A.M. Odlyzko, A modest proposal for preventing Internet congestion, Technical Report, AT&T Labs, 2007.

[16] D. Clark, in: McKnight, Bailey (Eds.), Internet Cost Allocation and Pricing, MIT Press, Cambridge, 2007.