Expert Systems With Applications 91 (2018) 492–512

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A distributed and quiescent max-min fair algorithm for network congestion control☆

Alberto Mozo[a,*], José Luis López-Presa[b,c], Antonio Fernández Anta[d]

[a] Dpto. Sistemas Informáticos., Universidad Politécnica de Madrid, Madrid, Spain
[b] Dpto. Telemática y Electrónica, Universidad Politécnica de Madrid, Madrid, Spain
[c] DCCE, Universidad Técnica Particular de Loja, Loja, Ecuador
[d] IMDEA Networks Institute, Madrid, Spain

ARTICLE INFO

Article history:
Received 30 January 2017
Revised 19 August 2017
Accepted 9 September 2017
Available online 12 September 2017

Keywords:
Max-min fair
Distributed optimization
Proactive algorithm
Quiescent algorithm
Network congestion control

ABSTRACT

Given the higher demands in network bandwidth and speed that the Internet will have to meet in the near future, it is crucial to research and design intelligent and proactive congestion control and avoidance mechanisms able to anticipate decisions before the congestion problems appear. Nowadays, congestion control mechanisms in the Internet are based on TCP, a transport protocol that is totally reactive and cannot adapt to network variability because its convergence speed to the optimal values is slow. In this context, we propose to investigate new congestion control mechanisms that (a) explicitly compute the optimal sessions' sending rates independently of congestion signals (i.e., proactive mechanisms) and (b) take anticipatory decisions (e.g., using forecasting or prediction techniques) in order to avoid the appearance of congestion problems.

In this paper we present B-Neck, a distributed optimization algorithm that can be used as the basic building block for the new generation of proactive and anticipatory congestion control protocols. B-Neck computes proactively the optimal sessions' sending rates independently of congestion signals. B-Neck applies max-min fairness as optimization criterion, since it is very often used in traffic engineering as a way of fairly distributing a network capacity among a set of sessions. B-Neck iterates rapidly until converging to the optimal solution and is also quiescent. The fact that B-Neck is quiescent means that it stops creating traffic when it has converged to the max-min rates, as long as there are no changes in the sessions. B-Neck reacts to variations in the environment, and so, changes in the sessions (e.g., arrivals and departures) reactivate the algorithm, and eventually the new sending rates are found and notified. To the best of our knowledge, B-Neck is the first distributed algorithm that maintains the computed max-min fair rates without the need of continuous traffic injection, which can be advantageous, e.g., in energy efficiency scenarios.

This paper proposes as novelty two theoretical contributions jointly with an in-depth experimental evaluation of the B-Neck optimization algorithm. First, it is formally proven that B-Neck is correct, and second, an upper bound for its convergence time is obtained. In addition, extensive simulations were conducted to validate the theoretical results and compare B-Neck with the most representative competitors. These experiments show that B-Neck behaves nicely in the presence of sessions arriving and departing, and its convergence time is in the same range as that of the fastest (non-quiescent) distributed max-min fair algorithms. These properties encourage to utilize B-Neck as the basic building block of proactive and anticipatory congestion control protocols.

☆ A preliminary version of this work was presented at the 10th IEEE International Symposium on Network Computing and Applications (NCA 2011) (Mozo et al. (2011)). This extended version has several substantial differences from the conference: (1) We provide a new introduction to justify the current interest of researchers in proactive max-min fair algorithms in order to alleviate some of the limitations that TCP exhibits in the high demanding network bandwidth and speed scenarios that the Internet will have to meet in the near future; (2) We formally prove the correctness of the algorithm, i.e., we prove that when no more sessions join or leave the network, B-Neck eventually converges to the max-min fair values and stops injecting packets to the network; (3) We formally prove an upper bound on B-Neck convergence time when no more sessions join or leave the network; and (4) We add a new set of detailed experiments that validate the obtained theoretical results and evaluate how B-Neck performs under real-world settings.

* Corresponding author.
E-mail addresses: a.mozo@upm.es, amozo@etsisi.upm.es (A. Mozo), jllopez@diatel.upm.es (J.L. López-Presa), antonio.fernandez@imdea.org (A. Fernández Anta).

http://dx.doi.org/10.1016/j.eswa.2017.09.015

1. Introduction

Nowadays, congestion control mechanisms in the Internet are based on the TCP protocol that is widely deployed, scales to existing traffic loads, and shares network bandwidth applying a flow-based fairness to the network sessions. TCP entities implement a closed-control-loop algorithm that reactively recompute sessions' rates when congestion signals are received from the network. There is a general consensus regarding the higher demands in network bandwidth and speed that the Internet will have to meet in the near future. In this context of speeds scaling to 100 Gb/s and beyond, TCP and other closed-control-loops approaches converge slowly to the optimal (fair) sessions' rates. Therefore, it is vital to investigate intelligent and proactive congestion control and avoidance mechanisms that can anticipate decisions before the congestion problems appear. Some current research works propose the usage of proactive congestion control protocols leveraging distributed optimization algorithms to explicitly compute and notify sending rates independently of congestion signals (Jose et al. (2015)). Complementary, a growing research trend is to not just react to network changes, but anticipate them as much as possible by predicting the evolution of network conditions (Bui et al. (2017)). We propose to investigate new congestion control mechanisms that (a) explicitly compute the optimal sessions' sending rates independently of congestion signals (i.e., proactive mechanisms) and (b) can leverage the integration of anticipatory components capable of making predictive decisions in order to avoid the appearance of congestion problems.

In this context, we present B-Neck, a distributed optimization algorithm that can be used as the basic building block for the new generation of proactive and anticipatory congestion control protocols. B-Neck computes proactively the optimal sessions' rates independently of congestion signals iterating rapidly until converging to the optimal solution.

B-Neck applies max-min fairness as optimization criterion, since it is often used in traffic engineering as a way of fairly distributing a network capacity among a set of sessions. The max-min fairness criterion has gained wide acceptance in the networking community and is actively used in traffic engineering and in the modeling of network performance (Bertsekas and Gallager (1992), Nace and Pióro (2008)) as a benchmarking measure in different applications such as routing, congestion control, and performance evaluation. A paradigmatic example of this is the objective function of Google traffic engineering systems in their globally-deployed software defined WAN, which delivers max-min fair bandwidth allocation to applications (Jain et al. (2013)). Max min fairness is closely related to max-min and min-max optimization problems that are extensively studied in the literature. Intuitively, to achieve max-min fairness first the total bandwidth is distributed equally among all the sessions on each link. Then, if a session can not use its allocated bandwidth due to restrictions arising elsewhere on its path, then the residual bandwidth is distributed between the other sessions. Thus, no session is penalized, and a certain minimum quality of service is guaranteed to all sessions. More precisely, max-min fairness takes into account the path of each session and the capacity of each link. Thus each session s is assigned a transmission rate λ_s so that no link is overloaded, and a session could only increase its rate at the expense of a session with the same or smaller rate. In other words, max-min fairness guarantees that no session s can increase its rate λ_s without causing another session s' to end up with a rate $\lambda_{s'} < \lambda_s$.

Up to the date, all proposed proactive max-min fair algorithms require packets being continuously transmitted to compute and maintain the max-min fair rates, even when the set of sessions does not change (e.g., no session is arriving nor leaving). One of the key features of B-Neck is that, in absence of changes (i.e., session arrivals or departures), it becomes quiescent. The quiescence property guarantees that, once the optimal solution is achieved and the max-min fair rates have been assigned, the algorithm does not need to generate (nor assume the existence of) any more control traffic in the network. Moreover, B-Neck reacts to variations in the environment, and so, changes in the sessions (e.g., arrivals and departures) reactivate the algorithm, and eventually the new sending rates are found and notified. As far as we know, this is the first quiescent distributed algorithm that solves the max-min fairness optimization problem. In an exponentially growing IoT scenario of connected nodes (Cisco and Ericsson predict around 30 billions of connected devices by 2020) where different strategies and algorithms are required for energy-efficiency, B-Neck offers, due to its quiescence, an advantageous alternative to the rest of distributed max-min fair algorithms that need to periodically inject traffic into the network to recompute the max-min fair rates.

This paper proposes as novelty two theoretical contributions jointly with an in-depth experimental analysis of the B-Neck algorithm. We formally prove that B-Neck is correct, and second, an upper bound for its convergence time is obtained. Additionally, extensive simulations were conducted to validate the theoretical results and compare B-Neck with the most representative competitors. In these experiments we show that B-Neck behaves nicely in the presence of sessions arriving and departing, and its convergence time is in the same range as that of the fastest (non-quiescent) distributed max-min fair algorithms. These properties encourage to utilize B-Neck as the basic building block of proactive and anticipatory congestion control protocols.

1.1. Related work

We are interested in solving the max-min fair optimization problem in a packet network with given link capacities to compute the max-min fair rate allocation for single path sessions. These rates can be efficiently computed in a centralized way using the Water-Filling algorithm (Bertsekas and Gallager (1992), Nace and Pióro (2008)). From a taxonomic point of view, centralized and distributed algorithms have been proposed. In addition, max-min fair algorithms may also be classified in those which need the routers to store per-session state information, and those which only need a constant amount of information per router. To our knowledge, the proposals of Hahne and Gallager (1986) and Katevenis (1987) were the first to apply max-min fairness to share out the bandwidth, among sessions, in a packet switched network. No max-min fair rate is explicitly calculated, and a window-based flow control is needed in order to control congestion. Per-session state information, which is updated by every data packet processed, is needed at each router link. Additionally, a continuous injection of control traffic is needed to maintain the system in a stable state.

Then, with the surge of ATM networks, there were a collection of distributed algorithms proposed to compute the rates to be used by the virtual circuits in the Available Bit Rate (ABR) traffic mode (Afek, Mansour, and Ostfeld (2000), Bartal, Farach-Colton, Yooseph, and Zhang (2002), Charny, Clark, and Jain (1995), Kalyanaraman, Jain, Fahmy, Goyal, and Vandalore (2000), Cao and Zegura (1999), Hou, Tzeng, and Panwar (1998), Tsai and Kim (1999), Tzeng and Sin (1997)). The computed rates were in fact max-min fair. These algorithms assign the exact max-min fair rates using the ATM explicit End-to-End Rate-based flow-Control protocol (EERC). In this protocol, each source periodically sends special Resource Management (RM) cells. These cells include a field called the Explicit Rate field (ER), which is used by these algorithms to carry per-session state information (e.g., the potential max-min fair rate of a session). Then, router links are in charge of executing the max-min fair algorithm. The EERC protocol, jointly with the former distributed max-min fair algorithms, can be con-