Empirical stability limits for a size-based scheduler applied to Network Utility Maximization

Bruno Mendivez Vasquez
Monash University
bruno.mendivezvasquez@monash.edu

Lachlan L. H. Andrew
University of Melbourne
lachlan.andrew@unimelb.edu.au

Julian Garcia
Monash University
julian.garcia@monash.edu

Cao Wang
U. Wollongong
wangc@uow.edu.au

***Abstract*—Network Utility Maximization (NUM) is an optimization problem that models rate allocation in the Internet. NUM is considered to produce fair bandwidth allocations, an advantage for the implementation of TCP queue management protocols. Prioritization of data packets can benefit from knowing transfer sizes *a-priori*. A state-of-the-art NUM formulation considers a size-based approach where transmission flows are scaled by some function of a transfer's residual work (inspired by schedulers like Shortest Remaining Processing Time, SRPT). This new NUM-SRPT hybrid scheduler was proven to be stable for a defined region on any arbitrary topology. We found that for all tested topologies, stability limits are well beyond of what was proven theoretically, showing that performance does benefit from a SRPT-based prioritization at much higher loads. A method to detect empirical stability based on the augmented Dickey-Fuller test was used in order to assess the stability limits.**

I. Introduction

Congestion control is responsible for both fairness and efficiency of capacity allocation in the Internet. Fairness is usually quantified by instantaneous fairness: how fair are the rates given to different flows. This type of *fair* allocation can be posed as an optimization: the network utility maximization (NUM) problem [1].

NUM considers a utility (or benefit) given to any user that transmits at a particular rate. The goal is to maximize the sum of all utilities subject to the rates satisfying the capacity constraints of all the links they use. This optimization models many TCP/active queue management (AQM) algorithms, where the global maximization can be solved in a decentralized manner via the Lagrange dual problem [1], [2].

NUM is a good framework for analyzing TCP/AQM network control performance such as stability, fairness and throughput [3]. However, instantaneous fairness is not what users need. If a user is downloading a file, the only thing that matters is how long it takes to download that file, called the file completion time. Minimizing the average file completion time (AFCT) in TCP was popularized in [4], but for decades it has been the primary objective in queueing theory, under the name average "sojourn" time.

Typical TCP applications consider sources that rely on fair allocations; however there has been an increase of specific scenarios where job prioritization is a desired feature (*e.g.* when streaming media). By taking advantage of *a priori* knowledge of job sizes, the network could potentially modify its allocation strategy in order to benefit some jobs with minimal penalty to others. Prioritizing short jobs is known to reduce mean sojourn time in a single server queueing system, which is equivalent to a single link "network" [5], [6].

The Shortest Remaining Processing Time (SRPT) policy prioritizes the job with the least amount of work to finish and it is the optimal strategy for mean sojourn time in a single server system [7]. Its single-server performance was the motivation for other network applications such as web servers [8], [6]. And even though SRPT needs to know job sizes *a priori*, that information is available at the source of file transfers.

One concern about SRPT is fairness: to what extent this prioritization of small jobs hurts large ones, and if it can even cause starvation. It was shown, also in [8] and [6], that SRPT also tends to outperform Processor Sharing (PS), a fair allocation strategy, in terms of slowdown when job sizes have a heavy tailed distribution, which characterizes file sizes.

Despite the advantages of SRPT and other size-based policies for single hop scheduling, realistic networks have flows that share bandwidth across multiple bottleneck links, equivalent to a multi-server queue. In that case, the performance improvements of SRPT decrease for long routes [9], [10], to the point of reducing the range of workloads for which the system is stable. Indeed, for any per-link load, there is a topology for which that load is unstable under SRPT [10]. As SRPT does not yield optimal mean delay performance, [10] suggests using well tuned NUM policies instead. This paper investigates if it is feasible to *improve* on a NUM formulation for a network resource allocation problem by prioritizing flows with short residual work, but not giving them strict priority.

Utility functions determine the rates given to each flow. They can be scaled by a function of remaining file transfer in such a way as to create a NUM-SRPT hybrid policy. A centralized scheduler was recently proposed [11], which allows a continuous tradeoff between NUM and SRPT, parameterized by a prioritization factor θ. Viewed as a queueing system, NUM is stable for all loads in systems that use α-fair, concave increasing utility functions and arbitrary light-tailed file size distributions [12]. On the other hand, SRPT can be unstable for arbitrarily low loads, for topologies with sufficiently long paths [10]. Nonetheless, the scheduler of [11] was proven to be stable for a non-vanishing stability region for all topologies, but not for the maximum possible stability region. This paper will use simulation to study the actual stability region of the new scheduler.

The contributions of this paper are as follows. We propose a technique for determining the stability of a queueing system's simulation. We use this to demonstrate that the stability region of the algorithm of [11] is somewhat less than the maximum stability region, but much larger than proved in [11]. Moreover, the system is able to improve the average performance on all paths, for a sufficiently small prioritization.

The rest of this paper is organized as follows. Section II presents the size-based resource allocation problem first introduced in [11], together with the theoretical condition for stability. We will use this optimization problem to define a job scheduler algorithm implemented in Julia.

The empirical method to determine if a queue has reached equilibrium based on the augmented Dickey-Fuller test is presented in Section III.

To test the empirical stability we present topologies that may resemble practical workloads. Section IV shows numerical results for different route loads, prioritization values and network sizes. Section V concludes the paper.

II. Formulation

We begin by describing the NUM-SRPT hybrid model presented in [11]. It consists of a utility-based resource allocation strategy applied to bandwidth-sharing networks. A network consists of $l \in L$ links and $m \in M$ routes. A link-route allocation matrix is defined as R where $R_{lm} = 1$ if link l is part of route m. For any route, transmitting at rate x_{im} gives job i traversing m a benefit $U_m(x_{im})$.

NUM's goal is to choose rates x_{im} to maximize the sum of all utilities subject to link capacity constraints. Note that NUM is formulated for infinite length flows, whose utility comes from the rate they receive, rather than the length of time they remain in the system.

To introduce the notion of minimizing completion times, we could reformulate the problem so that utility is a function of file completion time. That gives very difficult non-convex optimization to solve, which is also not amenable to a distributed implementation. Instead, we optimize each point in time separately, but scale utilities to prioritize small flows.

Given file sizes are known in advance (although not all flows are file transfers), a job's utility can be time-varying if we prioritize further based on remaining transfer size. If the remaining size is denoted by s_i for job i, a function $h_m(s)$ prioritizes it by scaling the corresponding utility $U_m(x)$. Such an allocation strategy needs to be performed while meeting link capacity constraints. The proposed size-based NUM problem at time t is

$$\text{maximize} \sum_{m} \sum_{i \in I_{m,t}} h_m(s_i(t)) \cdot U_m(x_i(t)) \tag{1a}$$

$$\text{s.t.} \sum_{m} R_{lm} \cdot \left(\sum_{i \in I_{m,t}} x_{im}(t) \right) \leq c_l, \forall l \tag{1b}$$

$$x_{im}(t) \geq 0, \forall i, m, \tag{1c}$$

where c_l is the capacity of link l.

In the rest of this paper, we consider only an α-fair system, in which all flows use the same utility of the form

$$U(x) = (1-\alpha)^{-1} x^{1-\alpha}, \tag{2}$$

where $\alpha \in [0, 1) \cup (1, \infty)$.

Note that for any bandwidth-sharing queueing system, jobs in route m arrive at a mean intensity λ_m, creating a route load of $\rho_m = \lambda_m/\mu_m$, where μ_m is the service rate in jobs per second. Let $G_m(\sigma)$ be the complementary cumulative distribution function (CCDF) of job sizes σ on route m. The main result known about (1) is [11]:

Theorem 1. *Consider a queueing system where the file size distribution has a pth moment for some $p > 1$. If the loads $\rho_m := \lambda_m/\mu_m$ satisfy*

$$\sum_{m} R_{lm} \rho_m \theta_m^{\frac{1}{\alpha+1}}(\alpha, h_m) < c_l, \tag{3}$$

for all l, where

$$\theta_m(\alpha, h_m) = \mu_m \int_0^\infty G_m(\sigma) h_m^{-\alpha-1}(\sigma) d\sigma, \tag{4}$$

then the fluid model of the network is stable when employing the scheduling policy in (1)*, with $h_m(0) = 1$, $h_m(s) > 0$ for all $s \geq 0$ and $h_m(.)$ non-increasing, for all m.*

For each route, we can choose an appropriate prioritization function $h_m(\sigma)$, as long as $G_m(\sigma)h_m^{-\alpha-1}(\sigma)$ is integrable over $[0, \infty), \forall m$. The function h_m is then normalized so $h_m(0) = 1$. If $h_m(\sigma) = 1$ for all sizes σ then (1) reduces to NUM. However, if h_m decreases sufficiently fast with σ, then the problem approaches SRPT.

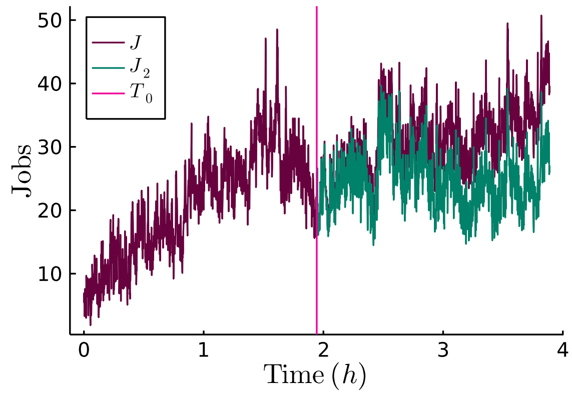
We use the problem in (1) to develop a scheduling policy based on giving priority to jobs that are close to finishing. Given a certain topology that specifies route loads and link capacities, the scheduler calculates the allocation vector $x(t)$ for every active job in the queue. The scheduler can be distributed as a standard NUM scheduler, although evaluating the performance of that is beyond the scope of this paper.

The implementation uses Julia's JuMP and NLopt libraries, which imperfectly optimize problem (1). They may not entirely allocate all of the bottleneck link capacities to active jobs. The capacity constraint in problem (1) is enforced but it does not always find a solution with bottleneck link utilization of 100%. The solution was tweaked to allocate capacity of links that was missed by the optimizer in a max-min fair manner [13] among the flows. This is suboptimal, but increases the objective and does not violate the constraints and so is an improvement over the rates found by the imperfect optimizer.

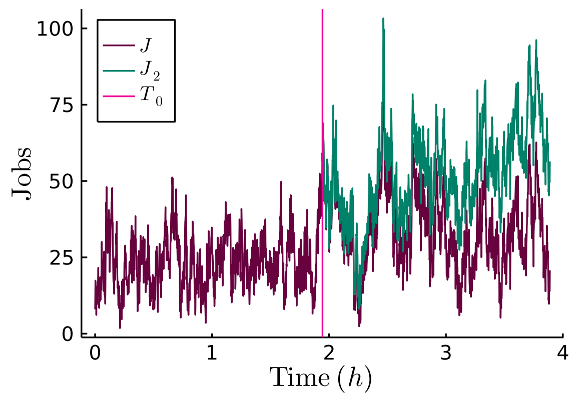
By analyzing the execution of the queue, we can determine congestion and other metrics that will be useful for assessing at what time the system reaches equilibrium.

III. Detecting stability

Relation (3) is a lower bound on the stability limit for any scheduling policy based in (1). However, this bound applies to arbitrary topologies, and it is known [10] that topologies with



(a) Unstable outcome, parking lot, $|L| = 8$, $\eta = 10$, $\rho_m = 0.7, \forall m$. ADF reports J_2 is stationary, confirming J is growing steadily.



(b) Stationary (or stable) outcome. The main outcome for J returned a stationary result, sufficient condition that does not trigger a test on J_2. This experiment considered a parking lot topology with 8 hops, $\eta = 0$ (NUM) and $\rho_m = 0.8, \forall m$.

Fig. 1. Number of jobs in the system versus time. Two series: J and J_2. Warm up period ends at T_0 (vertical line). Test is done over $[T_0, 2T_0]$.

long paths have much smaller stability regions than those with short paths. To determine how loose this bound is in practice, we simulate several networks. This raises the age-old question of how to determine stability from a simulation.

We propose a procedure to classify a model as stable or unstable, based on the analysis of a time series that describes the congestion of the system: number of active jobs over time,

$$J(t) = \sum_{m \in M} |\mathcal{I}_{m,t}|. \tag{5}$$

For some warm-up time T_0, we perform a pair of tests on the interval $[T_0, 2T_0]$. We declare the model "stable" if the time series passes a statistical test for stationarity. Otherwise, we declare "unstable" if the detrended version

$$J_2(t, T_0) = J(t) - t\frac{J(2T_0) - J(T_0)}{T_0}, \tag{6}$$

passes the test for stationarity. If it is not declared stable, then T_0 is doubled and the test is repeated until reaching an arbitrary maximum time. At that stage, we perform one final test by analyzing the second half of the entire simulation.

If the initial interval is too short, starting far from equilibrium, then the procedure may falsely conclude "unstable". This never causes the test to overstate the stability region. Still, to reduce it, we use as the initial simulation duration the length of the next smaller load (scaled by $(1-\rho_2)/(1-\rho_1)$), or next smaller prioritization (scaled by η_2/η_1).

The stability tests are based on the Augmented Dickey-Fuller (ADF) test, with unit lag parameter [14]. The test examines the null hypothesis that considers the time series as non-stationary (first order autoregressive with unit root), against the alternative hypothesis that the sequence is stationary. The ADF statistic does not follow a well-known distribution. Instead, threshold values are tabulated for various confidence levels (p-values). For our experiments, we used $p = 0.05$.

To improve accuracy, the time series we evaluate is actually the average of $J(t)$ for three time series with different random seeds.

Fig 1 shows an example for the ADF test, where the final outcome is determined by the functions J and J_2. The stationary condition is mainly decided by the series J, as shown in Fig. 1b. Only when the outcome for J is not definitive, J_2 is evaluated. If the trend is stationary, then the system is flagged as unstable (Fig. 1a).

IV. Simulations

A. Workload

To test the policy (1), we use arrivals that follow a homogeneous Poisson process and job sizes that are *i.i.d.* Pareto with complementary cumulative distribution function (CCDF)

$$G(\sigma) = \begin{cases} \sigma^{-a} & \text{if } \sigma \geq 1 \\ 1 & \text{if } \sigma < 1, \end{cases} \tag{7}$$

where a is a distribution shape parameter, related to the mean job size by $E[S] = a/(a-1) = 1/\mu$. A shape parameter of $a = 1.9$ was used in order to get job sizes with finite mean but infinite variance.

Experiments use a prioritization function

$$h(\sigma, \eta) = [\log(e + \sigma)]^{-\eta},$$

where e is the base of the natural logarithm. As a result, the stability condition in (3) becomes

$$\begin{aligned} \theta(\eta) = \mu & \int_1^\infty \sigma^{-a} [\log(e+\sigma)]^{\eta(\alpha+1)/\alpha} \, d\sigma \\ & + \mu \int_0^1 [\log(e+\sigma)]^{\eta(\alpha+1)/\alpha} \, d\sigma, \end{aligned} \tag{8}$$

where $\alpha = 0.95$, which is close to proportional fairness. The above integral is bounded since for any $\epsilon > 0$ there is a K such that the integrand is bounded above by $K\sigma^{-a+\epsilon}$.

The prioritization function h is chosen so the scheduler approaches SRPT as $\theta \to \infty$. Prioritization of small jobs is beneficial up to some point but for higher levels it increases average delay, eventually leading to instability (*i.e.*, the backlog grows unboundedly). Therefore, we want to know how large θ can be made while maintaining stability. This, along with network size and load, determine congestion and stability.

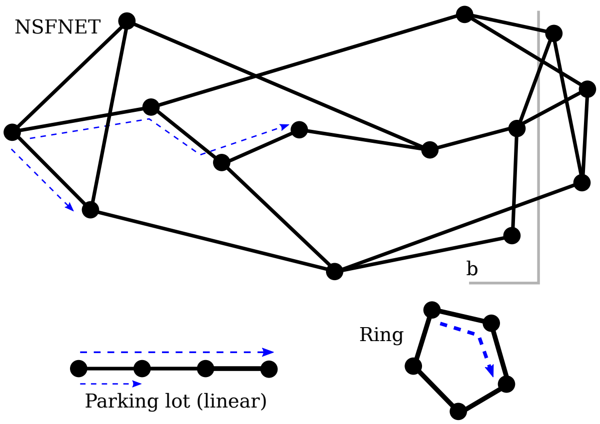
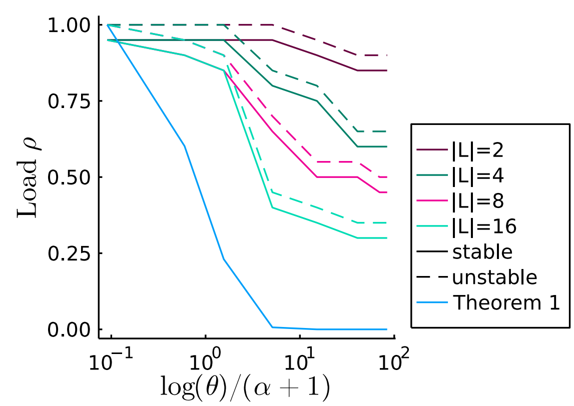
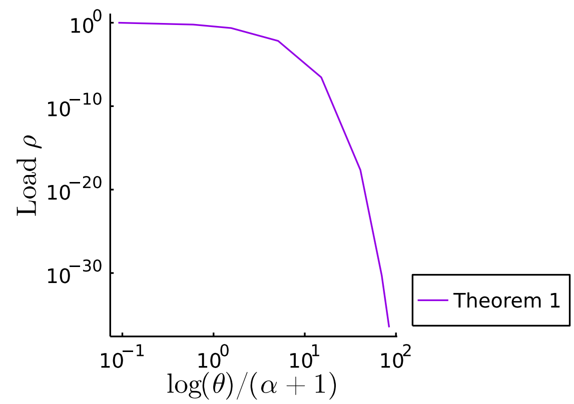


Fig. 2. Topologies considered in our simulations. Blue lines represent routes at variable path lengths. For NSFNET we consider the additional sub-graph *b*. A linear topology consists of one multi hop route while the rest has dedicated links. All routes in the ring topology consist of half begin single hop and the other 2-hop.



(a) Load thresholds for linear networks of various lengths.



(b) Load threshold for Theorem 1.

Fig. 3. Plot of load ρ stability threshold versus prioritization $\theta(\eta)$. Stable below solid line, unstable above dashed line, indeterminate in between.

B. Scenarios

Figure 2 shows the three topologies used for our simulations. We are interested in characterizing networks that offer good variability in terms of path length, as well as depicting practical communication scenarios.

The first network depicted in Fig. 2 is NSFNET, a T1 backbone network of supercomputers established in the United States, circa 1988 [15]. It has the advantage to simulate real world traffic conditions when using node population figures in order to determine "accurate" link loads [16]. To compare against possible stability improvements we consider a sub-graph of NSFNET, left of line *b*.

The second topology is a linear (parking lot) topology, consisting of $|L|$ links and $|L|+1$ routes. The only multi path route is the one that traverses all links in the network, while the remaining routes have dedicated links. This determines that jobs assigned to the multi hop route will become more starved (due to shared links) as the SRPT prioritization and path length increases. This topology maximizes the network diameter given the number of hops, allowing long paths with reasonable simulation times. That is useful, since the disadvantage due to prioritization increases with the path length.

The third case is a ring topology of five nodes. The distribution of path lengths considers half being single hop while the other is multi hop with two links. It was chosen so we could assess a potential case of overall stability, considering the homogeneous allocation of routes and links.

Parameters that apply to all networks are route loads ρ that vary from 0.1 to 0.95 and SRPT prioritization factor $\eta \in [0, 20]$. Network characterization will depend on path length, with the linear topology considering hop counts of 2, 4, 8 and 16. NSFNET will vary its average path length depending on which graph is simulated. As for the ring topology, path length can be 1 or 2 depending on the route, and the overall node count is 5.

C. Results

1) Load stability threshold analysis: We first study the stability threshold load ρ versus $\theta(\eta)$ of equation (8) for the parking lot topology based on the method of Section III. We calculated the maximum load for which all runs were deemed stable, and the minimum load for which all were deemed unstable. The stability of the region in between these series could not be determined from simulations of up to three flows. The results are shown in Fig. 3(a).

Load thresholds decrease with higher prioritization values, θ, which makes the scheme closer to SRPT, as well as hop count, following what happens under SRPT [10]. The "stable" threshold matches the maximum simulated load of $\rho = 0.95$, since $\rho = 1$ was not included in the experiments. This is consistent with the fact that pure NUM is stable for any load below the maximum simulated load, (*i.e.* the stability threshold tends to 1 as $\theta \to 0$).

When path length increases, there is a significant drop in the stability threshold, especially for small θ. Nonetheless, our results indicate that a 2-hop network is stable under all prioritizations. Beyond that, the threshold decreases to a non-zero asymptote. This is because SRPT always has a non-empty

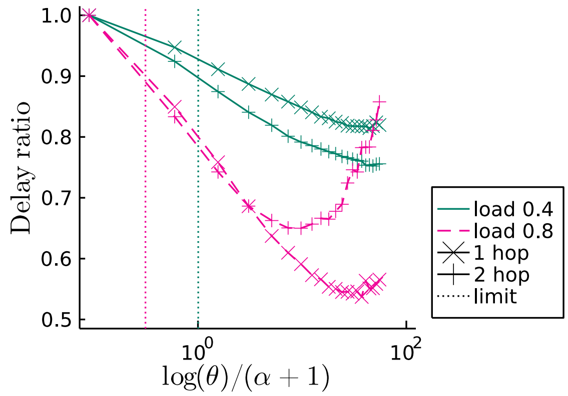


Fig. 4. Plot of mean delay divided by mean delay for pure NUM ($\theta = 0$) versus prioritization factor, load and path length. Network is ring topology.

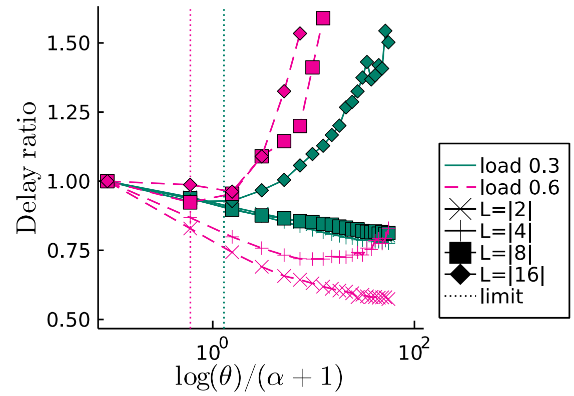


Fig. 5. Plot of mean delay divided by mean delay for pure NUM ($\theta = 0$) versus prioritization factor and load. Network is parking lot and routes are multi hop.

stable region, which is approached as prioritization $\theta \to \infty$.

The lower bound decreases to under 10^{-35} in the range of this graph, showing that the algorithm is more stable in practical networks than the theorem guarantees. Moreover, Fig. 3(b) shows the load threshold limit for Theorem 1, indicating that stability drops sharply for priorities where linear networks are still stable.

The stability of the ring network was also tested, indicating stability for a 5 node network, at any congestion level. This topology is interesting, because it does not result in instability, even though it consists of multi hop flows. The reason is that the network is homogeneous in terms of single and multi hop routes ratio. To some extent, it resembles the stability of a 2-hop linear network.

Another practical case is NSFNET. It was shown that the stability threshold for the two graphs reaches a minimum of $\rho = 0.7$. Naturally, we expect the lower node count network (b) to be more stable at higher prioritization values. Nonetheless, they both remain stable for a large subset of the prioritization domain, considering the the node count difference between the two is only 3. It is worth mentioning for future delay performance metrics, that NSFNET-b generates a small amount of 5-hop paths, while the maximum for full NSFNET is only 3.

Despite these practical topologies compared against Theorem 1, the theorem may not be loose, as it applies to all topologies, including the limit of paths with an unbounded number of hops. Further study is required to determine whether or not the theorem is tight.

2) Performance analysis: The motivation for modifying NUM was to reduce the average file completion time. To see if this has been achieved, we studied the mean delay ratio versus prioritization factor for link loads (0.3, 0.6) and 2, 4, 8 and 16 link networks. This is shown in Fig. 5 for multi hop routes. Vertical doted lines indicate theoretical stability limits (Theorem 1) for the corresponding route loads.

The proposed algorithm affects delay in two ways. A. Prioritizing short jobs within a route acts directly to reduce delay. B. As found in [10], prioritizing short jobs indirectly prioritizes short routes, to a greater extent as the length of the competing routes increases. Since single hop routes benefit from both of these, it is expected that their performance will improve as θ and $|L|$ increase, which is shown to happen in Fig. 4 (ring, 1 hop) and Fig. 6, 7 (NSFNET/b, 1 hop).

Notice that only points for which the system is stable are plotted, even if the short routes have an equilibrium delay in systems for which long routes are unstable. Performance may continue to increase beyond that point, at the expense of stability.

For multi hop routes, A above still reduces delay, but B increases delay, as shown more noticeably in Figs. 5 and 6, 7. There is still a visible benefit of slightly prioritizing short jobs for multi hop routes, as A dominates, but for larger prioritization, B dominates as the stability limit is approached. Fig. 6 shows an expected multi hop behavior for NSFNET, with higher hop counts decreasing performance at higher prioritizations. The same applies to the ring topology in Fig. 4, where the 2-hop routes perform worse than single hop at low loads. However, to the extent of path lengths, we can see that both single-hop and multi-hop routes have better performance at higher loads when prioritizations are more than 0 (NUM). The delay ratio drops because mean delay for SRPT increases with load much slower than PS [17], and even FCFS, when considering the M/G/1 queue. Nonetheless, there seems to be a limit imposed by path length, where in Fig. 5 such advantage no longer applies for 8 and 16-hop linear topologies.

In terms of routes with different path lengths competing for service, NSFNET provides an interesting view on multi-hop route performance. Fig. 6 and 7 show a breakdown of route performance by path length. We can see that multi-hop routes perform batter than single hope ones due to having more shared resources that can be allocated to jobs. Nonetheless, the main trade-off is stability at higher prioritizations, causing the advantage to quickly disappear as θ reaches its maximum. For example, NSFNET-b having some routes with 3 and 5-hop

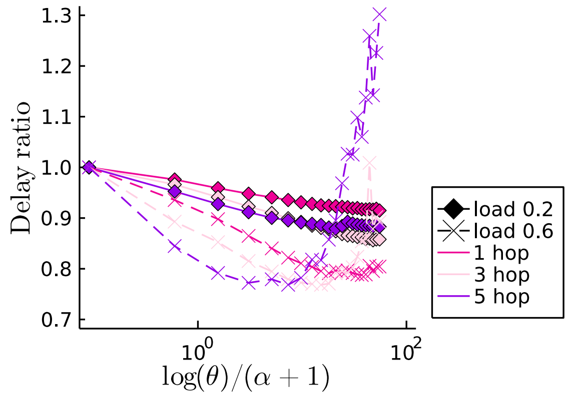


Fig. 6. Plot of mean ratio of delay to delay with $\theta = 0$ versus prioritization factor for routes with different hop counts. Network is NSFNET-b.

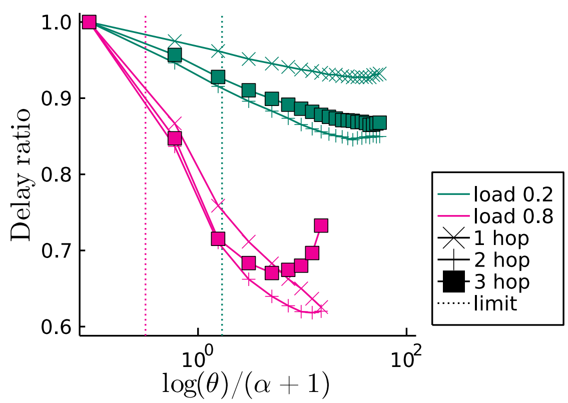


Fig. 7. Plot of mean ratio of delay to delay with $\theta = 0$ versus prioritization factor for routes with different hop counts. Network is NSFNET.

length that exceed in performance to every other one, until it get worse than pure NUM ($\theta = 0$) when prioritization is close to the maximum considered.

V. Conclusion and Future Work

We proposed a method to detect empirical stability based on the Dickey-Fuller statistical test. It was used to compare against theoretical stability limits introduced in [11]. It is shown that maximum stability limits are well beyond the ones demonstrated theoretically, allowing for residual-size-based prioritization in NUM to show benefits well above the route loads predicted theoretically. Other schedulers that are size-based can always be stable, for example, when based on the Fair Sojourn Protocol. Nonetheless, that is less implementable than the present scheduler, but finding an implementable approximation may be useful in the future.

It will be useful to define a formal hypothesis test for stability based on the method proposed here. This will need to take into account the sequential decisions, as in [18].

This study also provides guidance for a deeper study into the relationship between the different types of cross-traffic on a path, the prioritization, the load on the path and stability. For example, it will be useful to quantify the effective bandwidth consumed by cross traffic on a long path, which is greater than the actual bandwidth because of the prioritization effect. Once these effects are understood, the full potential of partial prioritization of short flows will be revealed.

Acknowledgment

This work was funded in part by Australian Research Council grant DP190102134.

References

[1] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, vol. 49, no. 3, pp. 237–252, 1998.

[2] S. H. Low and D. E. Lapsley, "Optimization flow control I: Basic algorithm and convergence," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[3] D. P. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: Framework and applications," *IEEE Trans. Automatic Control*, vol. 52, no. 12, pp. 2254–2269, 2007.

[4] N. Dukkipati and N. McKeown, "Why flow-completion time is the right metric for congestion control," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 59–62, 2006.

[5] M. Harchol-Balter, B. Schroeder, N. Bansal, and M. Agrawal, "Size-based scheduling to improve web performance," *ACM Trans. Comp. Sys.*, vol. 21, no. 2, pp. 207–233, 2003.

[6] N. Bansal and M. Harchol-Balter, "Analysis of SRPT scheduling: Investigating unfairness," in *Proc. ACM SIGMETRICS*, 2001, pp. 279–290.

[7] L. E. Schrage and L. W. Miller, "The queue M/G/1 with the shortest remaining processing time discipline," *Operations Research*, vol. 14, no. 4, pp. 670–684, 1966.

[8] M. Harchol-Balter, M. Crovella, and S. Park, "The case for SRPT scheduling in web servers," Technical Report MIT-LCS-TR-767, MIT Lab for Computer Science, Tech. Rep., 1998.

[9] S. Yang and G. De Veciana, "Size-based adaptive bandwidth allocation: Optimizing the average qos for elastic flows," in *Proc. IEEE INFOCOM*, 2002, pp. 657–666.

[10] M. Verloop, S. Borst, and R. Núñez-Queija, "Stability of size-based scheduling disciplines in resource-sharing networks," *Performance Evaluation*, vol. 62, no. 1-4, pp. 247–262, 2005.

[11] C. Wang, L. L. H. Andrew, B. Mendivez, and H. L. Vu, "Size-based congestion control using network utility maximization," *Submitted*. [Online]. Available: https://people.eng.unimelb.edu.au/llandrew/NUM-SRPT-proof.pdf

[12] A. Ferragut and F. Paganini, "Network resource allocation for users with multiple connections: fairness and stability," *IEEE/ACM Transactions on networking*, vol. 22, no. 2, pp. 349–362, 2013.

[13] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data networks*. Prentice-Hall International New Jersey, 1992, vol. 2.

[14] Y.-W. Cheung and K. S. Lai, "Lag order and critical values of the augmented Dickey–Fuller test," *Journal of Business & Economic Statistics*, vol. 13, no. 3, pp. 277–280, 1995.

[15] k. c. Claffy, G. C. Polyzos, and H.-W. Braun, "Traffic characteristics of the T1 NSFNET backbone," in *Proc. IEEE INFOCOM*, 1993, pp. 885–892.

[16] Y. Qin, D. Xue, C. K. Siew, I. Saniee, and I. Widjaja, "A topology based dynamic traffic scheduling in time-domain wavelength interleaved networks," in *Proc IEEE Int. Conf. Networks (ICON)*, 2004, pp. 142–146.

[17] M. Lin, A. Wierman, and B. Zwart, "Heavy-traffic analysis of mean response time under shortest remaining processing time," *Performance Evaluation*, vol. 68, no. 10, pp. 955–966, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166531611000721

[18] G. I. Webb and F. Petitjean, "A multiple test correction for streams and cascades of statistical hypothesis tests," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1255–1264.