Fully Decentralized Emulation of Best-Effort and Processor Sharing Queues

Rade Stanojević
Hamilton Institute, NUIM, Ireland
rade.stanojevic@nuim.ie

Robert Shorten
Hamilton Institute, NUIM, Ireland
robert.shorten@nuim.ie

ABSTRACT

Control of large distributed cloud-based services is a challenging problem. The Distributed Rate Limiting (DRL) paradigm was recently proposed as a mechanism for tackling this problem. The heuristic nature of existing DRL solutions makes their behavior unpredictable and analytically untractable. In this paper we treat the DRL problem in a mathematical framework and propose two novel DRL algorithms that exhibit good and predictable performance. The first algorithm Cloud Control with Constant Probabilities (C3P) solves the DRL problem in best effort environments, emulating the behavior of a single best-effort queue in a fully distributed manner. The second problem we approach is the DRL in processor sharing environments. Our algorithm, Distributed Deficit Round Robin (D2R2), parameterized by parameter α, converges to a state that is, at most, $O(\frac{1}{\alpha})$ away from the exact emulation of centralized processor sharing queue. The convergence and stability properties are fully analyzed for both C3P and D2R2. Analytical results are validated empirically through a number of representative packet level simulations. The closed-form nature of our results allows simple design rules which, together with extremely low communication overhead, makes the presented algorithms practical and easy to deploy.

Categories and Subject Descriptors

C.2.3 [**Computer Communication Networks**]: Network management

General Terms

Algorithms, Management, Performance

Keywords

Rate limiting, CDN, Cloud control, Consensus agreement, Stability and convergence

1. INTRODUCTION

SIGMETRICS'08, June 2–6, 2008, Annapolis, Maryland, USA.

Many internet services are structured around a large number of servers that are distributed worldwide to improve the content availability, fault robustness, end-to-end delays and data transmission rates. Services of this type are sometimes referred to as cloud based services and examples of those include most of Yahoo! and Google services, Amazon's Simple Simple Storage Service (S3), and Akamai's content-distribution network (CDN). Some other applications, such as Google Docs or Microsoft Groove Office, have integrated software-as-a-service paradigm and allow desktop users to utilize cloud-based services in hosted environments.

The ability to control network usage is critical for several important functions of a cloud-based service provider (CBSP):

(1) *Pricing* of service by most of the existing CBSPs is usage-based [1, 26]. Namely, services are charged at a rate that is an increasing (usually concave) function of the total resources used. However, in the history of communications, pricing of various services (eg. ordinary mail, the telegraph, the telephone, and the Internet) followed similar pattern: it started with usage-based pricing and converged at some form of flat-fee pricing. Moreover, enterprises tend to prefer fixed cost of an IT service rather than unlimited/unpredictable usage-based cost, see [12] and [23]

(2) *Provisioning* of high quality services depends on the nature of the service demand pattern. The ability to regulate the usage of individual service allows CBSPs to design networks with predictable performance bounds.

(3) *Fault tolerance* of large-scale distributed services is an important performance objective that is enhanced by resource control by means of fast fault discovery and quick response to these faults.

The paper [26] introduces the notion of Distributed Rate Limiting (DRL) as a mechanism for resource control in cloud-based services. Briefly, DRL stands for any mechanism that controls the aggregate network bandwidth used by a cloud-based service. The idea is to enhance a set of local limiters with the ability to exchange information among them towards the global goal: control of the aggregate network bandwidth that a cloud-based service uses. The main obstacle in the design of a DRL algorithm is the fairness postulate [26]:

Fairness postulate: Flows arriving at different limiters should achieve the same rates as they would if they were traversing a single, shared rate limiter.

Thus, in DRL, a flow traversing one limiter competes for bandwidth with flows that traverse the same limiter and with all other flows traversing other limiters. The worldwide scale[1] of such clouds raises important issues as to how to efficiently control resource usage in such large distributed environments.

Two DRL algorithms are proposed in [26]: Global Random Drop (GRD) and Flow Proportional Share (FPS). Even though both GRD and FPS are decentralized, the main information needed by a local limiter to adjust its behavior is the information on the global demand/weight. This global demand/weight information is obtained by the decentralized epidemic algorithm from [17] and that information is utilized by each of the local limiters in the quest of emulating global token bucket. The ad-hoc nature of the proposed algorithms makes the following important questions analytically untractable: Does system converge to the desired operating point? If it does, how quickly? How does cloud topology affects stability and responsiveness of the system? What are the performance guarantees in terms of aggregate utilization, loss rates and bandwidth allocation?

In this paper we propose a rigorous mathematical framework for the design of fully decentralized DRL algorithms. We use our approach to design two DRL algorithms; one for best effort environments - Cloud Control with Constant Probabilities (C3P) and the other for processor-sharing environments Distributed Deficit Round Robin (D2R2). Both algorithms are fully decentralized (meaning that each node utilizes only state information from its neighbors) which in turn results in significant reduction of the communication overhead and increased fault tolerance compared to GRD and FPS.

1.1 Problem formulation

Let a CBSP controls N hosting centers with each hosting center $i \in \{1, 2, \ldots, N\}$ being able to limit locally the bandwidth of a particular service, serving the flow population $\mathcal{F}_i$, at level C_i. The first constraint of DRL is to keep the aggregate bandwidth at a prescribed level C:

$$\sum_{i=1}^{N} C_i = C. \tag{1}$$

Then the local limiters should collaborate to achieve the fairness postulate. In order to formalize the fairness postulate we need to have a reference point, the service experienced by a user of a single shared rate limiter. In this paper we discuss the following two cases.

1. **Best effort** single shared rate limiter. Denote by $r_1(f)$ the rate a flow $f \in \cup_{i=1}^{N} \mathcal{F}_i$ would obtain by traversing a (virtual) single best effort limiter with capacity C. Since $r_1(f)$ is a random variable (whose distribution is determined by loss rate) it is meaningless to require that rate of flow f be exactly equal to $r_1(f)$. We rather require that the expected value of flow's f rate be equal to its expected value in centralized best effort limiter:

$$E(rate(f)) = E(r_1(f)) \ \ \forall f \in \cup_{i=1}^{N} \mathcal{F}_i. \tag{2}$$

Since, there is a strong connection between expected throughput of a flow f and loss rate[2], condition (2) can be expressed as

$$p_i = \bar{p} \ \ \forall i \in \{1, 2, \ldots, N\}, \tag{3}$$

where $p_i = p(C_i, \mathcal{F}_i)$ is the loss rate at the limiter i, and $\bar{p} = p(C, \cup_{i=1}^{N} \mathcal{F}_i)$ is the loss rate achievable with the centralized best effort limiter. Thus, in the best-effort case, the fairness postulate translates into condition (3) subject to (1).

2. **Processor sharing** single shared rate limiter. In the processor sharing case, the throughput of a flow f is a deterministic function of the "fair share". Denote by v^* the "fair share"[3] of the centralized processor sharing limiter with capacity C serving population of flows $\cup_{i=1}^{N} \mathcal{F}_i$. In order to get some understanding on the nature of v^*, lets denote by $u_s^{(i)}$ the demand of a flow $f_s^{(i)} \in \mathcal{F}_i$: the rate that flow would achieve if limiter i had infinite bandwidth. We will also use the following notation:

$$g_i(v) = \sum_{s=1}^{n_i} \min(u_s^{(i)}, v), \tag{4}$$

and

$$G(v) = \sum_{i=1}^{N} g_i(v). \tag{5}$$

The function $g_i(v)$ represents the throughput of the limiter i when the "fair-share" is equal to v and $G(v)$ represents the aggregate throughput that centralized processor sharing limiter would obtain if the 'fair-share" is equal to v. Now v^* is simply the solution of the following equation:

$$G(v) = C. \tag{6}$$

Note that if the aggregate demand is lower than capacity ($\sum_{i=1}^{N} \sum_{s=1}^{n_i} u_s^{(i)} \leq C$) no such v exists. However, a more interesting problem occurs when the aggregate demand is greater than the available capacity and then the problem of interest translates to distributed computing of v^* such that rate of flow $f_s^{(i)}$ is equal to

$$rate(f_s^{(i)}) = \min(u_s^{(i)}, v^*) \ \ \forall f_s^{(i)} \in \cup_{j=1}^{N} \mathcal{F}_j. \tag{7}$$

Let G be a connected undirected graph with nodes given by N limiters. We allow each limiter i to cooperate with its neighbors in G in adapting its bandwidth limit C_i. The goal of our work is to develop fully distributed algorithms that converge to the bandwidth allocation $(C_1, C_2, \ldots, C_N)$ for which the constraint (1) and either (3) (best effort DRL case) or (7) (in the processor sharing DRL case) are satisfied.

[1]For example, Google's services run on several hundreds of thousands servers distributed worldwide [26, 6]. Akamai's content distribution network utilized more than 12 thousand servers in the year of 2002 [8].

[2]In TCP case, this relationship is expressed by square root formula: $E(rate(f)) = \frac{\theta}{\sqrt{p}}$. For any other elastic loss-based protocol such relationship exist as well, although it may not be explicitly known in the literature. For nonelastic flow f, the expected throughput is simply $(1-p)SendingRate(f)$.

[3]The maximal flow throughput or "fair share" is easily measurable at any processor sharing emulator. It is simply the maximum forwarding rate among all flows utilizing the PS queue.

1.2 Our contributions

As we have said, the main concern of this paper is a principled design of algorithms for DRL. Briefly, the main contributions of our work are following:

- We propose two simple, fully decentralized algorithms: Cloud Control with Constant Probabilities (C3P) and Distributed Deficit Round Robin (D2R2). C3P solves best-effort DRL problem while D2R2 gives an asymptotic solution to the processor sharing DRL problem.

- The stability and convergence properties are analyzed for both C3P and D2R2 and the closed-form nature of our results allows simple design rules.

- Empirical evaluation of the proposed schemes is presented, that supports our analytical findings.

We note also that both C3P and D2R2 are easy to implement, computationally light and have extremely low communication requirements. Namely, the only information used by local limiter i are states (loss rate in C3P case, and "fair share" in D2R2 case) from its immediate neighbors in the communication graph G.

The dynamical systems that describe the dynamics of C3P and D2R2 are nonlinear and implicit. This makes the task of analysis quite challenging. Namely, standard theory of consensus algorithms (see [14] and references therein) cannot be employed in our case. The convergence results established in Theorem 1 (for C3P) and Theorem 4 (for D2R2) are nontrivial and represent the main theoretical contributions of this paper.

We also embedded C3P and D2R2 in *ns2* and performed a number of representative packet level simulations. We found that various performance metrics closely match our analytical predictions capturing one of the goals of present paper: principled and performance-predictable design of DRL algorithms.

Finally, we note that the fairness postulate, formalized by (3) for best effort DRL and by (7) for processor sharing DRL, have very interesting interpretation. Namely, the solution $(C_1, C_2, \ldots, C_N)$ to the processor sharing DRL problem, maximizes the minimal "fair-share" (among N limiters). While this feature of processor sharing DRL can be expected the following property of best effort DRL is quite surprising. Among all N-tuples $(C_1, C_2, \ldots, C_N)$ that satisfy the constraint (1) one that satisfies (3) enforces the least number of drops globally in the TCP environments. This feature of C3P is specific only to TCP-like environments in which the square root formula holds, and the proof is given in Section 2.2.

1.3 Related work

The problem of distributed resource allocation has been studied in different application domains in the past. For example, in distributed admission control [4] end users can test for and book bandwidth across a set of network paths. DRL can be seen as a reservation-free version of distributed admission control: each end-user can pick an arbitrary resource to connect to and DRL ensures that service obtained is independent of the choice of the resource. Capacity provisioning in hose model [9] of virtual private networks (VPNs) is another topic that have attracted significant attention in the last decade. In the hose model, a number of nodes have fixed capacities that sums up to a constant C. Two main issues are: (1) provisioning the network infrastructure [20] and (2) routing [19], such that various QoS requirements are met across all traffic matrices that satisfy hose model constrains.

An early DRL-like proposal appeared in [15] which discussed general framework for monitoring and control of distributed systems, with a particular application on Planetlab DRL control. The paper [26] introduces DRL in context of cloud-based service control. Here we briefly overview two algorithms proposed in [26]: Global Random Drop (GRD) and Flow Proportional Share (FPS). GRD works as follows: each local limiter tracks its demand and broadcasts that information using the algorithm from [17]. Then the total demand T is computed as a sum of demands at all limiters and an arriving packet is dropped with probability $(T - C)/T$. As it is noted in [26], GRD exhibits poor performance for large number of limiters. To cite [26]: *"... Beyond 50 limiters, GRD fails to limit aggregate rate, but this is not assuaged with an increasing communication budget. Instead it indicates GRD's dependance on swiftly converging global arrival rate estimates."*.

In FPS, each local limiter i uses a token bucket with service rate C_i and tracks the "weight" w_i defined as $w_i = C_i/MaxRate_i$, where $MaxRate_i$ is the maximal rate among all flows that use limiter i. The "weights" are broadcasted and, in steady state, C_i is updated as:

$$C_i = \frac{w_i}{\sum_{j=1}^{N} w_j} C.$$

The value w_i is used as an estimate of number of unbotlenecked[4] flows. In practice, w_i does not say much about the number of unbotlenecked flows at a token bucket limiter i. To see this lets consider case with two limiters one serving 5 TCP flows $f_1 - f_5$ with RTT equal $100ms$ and another serving 4 TCP flows $f_6 - f_9$ with RTT equal $100ms$ and one TCP flow f_{10} with RTT equal $10ms$; none of the flows $f_1 - f_{10}$ is bottlenecked elsewhere. From the square-root formula in best-effort environments (see eq. (8)) we know that rate of flow f_{10} is 10 times greater than rate of flows $f_6 - f_9$. Therefore in steady state:

$$w_1 = 5$$

and

$$w_2 = 1 + 4 \cdot 0.1 = 1.4.$$

Thus, although both limiters serve 5 unbotlenecked flows the values of w_1 and w_2 are significantly different. Simple calculations show that flows $f_1 - f_5$ and flow f_{10} obtain rates $\frac{1}{6.4}C$ while flows $f_6 - f_9$ receive rates $\frac{0.1}{6.4}C$, which violates the fairness postulate (for both best effort and processor sharing cases).

Both best-effort and processor sharing versions of our problem, formulated in Section 1.1, can be seen as instances of

[4]The term unbotlenecked flow is used in [26] as a flow that is not bottlenecked elsewhere. In the congestion control literature (such as [29]) it has a rather opposite meaning. We use the terminology of [26].

```
1  UpdateCapacities()
2      Once every Δ units of time do
3        for i = 1 : N
4          C_i ← C_i + η Σ_{(i,j)∈E} (p_i − p_j)
5        endfor
6      enddo

7    InitializeCapacities()
8      for i = 1 : N
9        C_i ← C/N
10     endfor
```

Figure 1: Pseudo-code of C3P

the consensus agreement. Consensus algorithms have attracted significant attention over last several years being applied in various topics, such as flocking [27], time synchronization, multi-agent coordination [14], sensor, peer-to-peer and ad hoc networks [5]. In most existing applications consensus algorithms can be modelled as positive linear systems, which then allows the elegant theory of Nonnegative matrices and Markov chains to be employed to capture the convergence properties of the algorithms. However, little is known about implicit nonlinear consensus problems (see [18]moreau) and one of the main contributions of this paper is the proof of global stability for the implicitly given nonlinear systems describing the dynamics of algorithms presented in the next two sections.

2. CLOUD CONTROL WITH CONSTANT PROBABILITIES (C3P)

We now present the C3P algorithm that solves best-effort DRL problem introduced in Section 1.1: allocate the network resources in a manner that equalizes loss rates amongst local-limiters; the rational for doing this is to ensure that all end-users experience a similar quality of service.

Our basic setup is as follows. We use N local limiters to control aggregate network bandwidth at level C. The local limiter i has a capacity C_i that can be adjusted, and this limiter can exchange information with the neighbor limiter j. (i,j) is an edge in the communication graph $G=(N,E)$ and we write $(i,j)\in E$. For a given capacity C_i, and a family $\mathcal{F}_i$ of flows utilizing the limiter i, the loss rate p_i at limiter i can be directly measured, and is a function of $\mathcal{F}_i$ and C_i:

$$p_i = p(C_i, \mathcal{F}_i).$$

The goal here is to obtain a fully decentralized algorithm, for adjusting the C_i such that

$$p_1 = p_2 = \cdots = p_N.$$

At each limiter we use a virtual queue (see [11]) with service rate C_i: on each arrival the packet size is placed in virtual queue. If the packet is discarded from the virtual queue, then the arriving packet is dropped, otherwise it is forwarded to the appropriate output line. Thus, no queueing delay is caused by any limiter. The loss rate is measurable directly, and it depends on the traffic pattern: in general the more flows exist at the limiter (meaning that aggregate aggressiveness is bigger) the higher loss rate is. Note also that p_i is a decreasing function of C_i.

The pseudo-code for control of $(C_1,\ldots,C_N)$ is given in Figure 1. Initially, all the C_i are set by the $1/N$ rule. Then C_i is updated in discrete time steps by the simple rule: $C_i \leftarrow C_i + \eta\sum_{(i,j)\in E}(p_i - p_j)$. The rationale for this update step is the following. The loss rate is the main performance indicator of the quality of service in best-effort token buckets. If the loss rate p_i at limiter i is higher than loss rate p_j at some neighbor j of i (in G), then this indicates that some extra bandwidth should be allocated to limiter i which must be compensated by reducing the capacity of limiter j. Giving more bandwidth to limiters with high loss rates affects reducing their loss rates. The parameter $\eta>0$ determines responsiveness and stability properties of the algorithm and its choice is discussed in the next subsection.

While the basic algorithm makes sense intuitively, many questions need to be answered before it can be deployed. Paramount among these concerns under which conditions does the algorithm C3P converge to the desired (unique) equilibrium, and if so, how fast. These questions provide the focus for the investigation presented in the next section.

2.1 Model and analysis of C3P

In this section we analyze the model of C3P utilized by standard TCP end-users. The result for general traffic mix is given by Theorem 2.

The starting point of our model is well known square-root formula, that relates the loss rate and the expected sending rate of TCP flow with round-trip time given by RTT [10, 24]:

$$x(p, RTT) = \frac{\theta}{RTT\sqrt{p}}. \tag{8}$$

This formula is widely accepted as explaining many observations in networks characterised by TCP traffic[5].

Now, suppose that limiter i serves population $\mathcal{F}_i$ of n_i TCP flows with round-trip times $RTT_1^{(i)},\ldots,RTT_{n_i}^{(i)}$. Then the limiter i is utilized with capacity C_i if the sum of sending rates from all n_i flows is equal to C_i:

$$C_i = \sum_{j=1}^{n_i} \frac{\theta}{RTT_j^{(i)}\sqrt{p_i}}. \tag{9}$$

Equation (9) represents the key relationship between C_i and p_i. Given this, the dynamical system describing the evolution of $C_i(t)$, in time t, is given by:

$$C_1(0) = C_2(0) = \cdots = C_N(0) = C/N, \tag{10}$$

$$C_i(t+1) = C_i(t) + \eta \sum_{(i,j)\in E} \left(\frac{\beta_i^2}{C_i^2(t)} - \frac{\beta_j^2}{C_j^2(t)} \right) \tag{11}$$

where, we used the notation:

$$\beta_i = \sum_{j=1}^{n_i} \frac{\theta}{RTT_j^{(i)}}.$$

The following lemma is a straightforward consequence of the fact that G is an undirected graph.

LEMMA 1. *For all t, the capacity constraint is satisfied:*

$$C_1(t) + C_2(t) + \cdots + C_N(t) = C. \tag{12}$$

[5]In [10], the value of θ is explicitly computed for TCP flows without delayed acking: $\theta = 1.3098$. If TCP delayed acking is turned on, then the value $\theta = 0.87$ is used.

PROOF. For $t = 0$ the statement is true from the definition. Suppose that it is valid for $t = k$, then for $t = k + 1$:

$$\sum_{i=1}^{N} C_i(t+1) = \sum_{i=1}^{N} \left[C_i(t) + \eta \sum_{(i,j)\in E} \left(\frac{\beta_i^2}{C_i^2(t)} - \frac{\beta_j^2}{C_j^2(t)} \right) \right] =$$

$$\sum_{i=1}^{N} C_i(t) + \eta \sum_{(i,j)\in E} \left(\frac{\beta_i^2}{C_i^2(t)} - \frac{\beta_j^2}{C_j^2(t)} \right) + \left(\frac{\beta_j^2}{C_j^2(t)} - \frac{\beta_i^2}{C_i^2(t)} \right) =$$

$$= \sum_{i=1}^{N} C_i(t) = C.$$

□

The following theorem gives a sufficient condition under which system (10)-(11) converge.

THEOREM 1. *Let d_i be the degree of node i in the communication graph G and let η satisfies:*

$$0 < \eta < \frac{1}{3} \left(\frac{C}{N} \right)^3 \frac{1}{\max_{1\le i\le N} \beta_i^3} \min_{1\le i\le N} \frac{\beta_i}{d_i}. \tag{13}$$

Then

$$\lim_{t\to\infty} C_i(t) = \frac{\beta_i}{\sum_{j=1}^{N} \beta_j} C \tag{14}$$

and

$$\lim_{t\to\infty} p_i(t) = p^* = \left(\frac{\sum_{j=1}^{N} \beta_j}{C} \right)^2. \tag{15}$$

PROOF. We find it more convenient to write the dynamics of (11) in terms of $p_i(t)$:

$$\frac{\beta_i}{\sqrt{p_i(t+1)}} = \frac{\beta_i}{\sqrt{p_i(t)}} + \eta \sum_{(i,j)\in E} (p_i(t) - p_j(t)), \tag{16}$$

or

$$p_i(t+1) = \frac{p_i(t)}{\left(1 + \sqrt{p_i(t)} \frac{\eta}{\beta_i} \sum_{(i,j)\in E} (p_i(t) - p_j(t)) \right)^2}. \tag{17}$$

We denote

$$m(t) = \min_{1\le i\le N} p_i(t),$$

and

$$M(t) = \max_{1\le i\le N} p_i(t).$$

Step 1. First we prove that under condition (13) the sequence $m(t)$ is nondecreasing and the sequence $M(t)$ is nonincreasing.

Let $p_i(t) = M(t) - \lambda$. Then from the equation (17) we have:

$$p_i(t+1) = \frac{M(t) - \lambda}{\left(1 + \sqrt{p_i(t)} \frac{\eta}{\beta_i} \sum_{(i,j)\in E} (M(t) - \lambda - p_j(t)) \right)^2} \le$$

$$\frac{M(t) - \lambda}{\left(1 - \sqrt{p_i(t)} \frac{d_i\eta}{\beta_i} \lambda \right)^2} \le \frac{M(t) - \lambda}{1 - 2\sqrt{p_i(t)} \frac{d_i\eta}{\beta_i} \lambda} =$$

$$M(t) - \frac{\lambda \left(1 - 2\sqrt{p_i(t)} \frac{d_i\eta}{\beta_i} M(t) \right)}{1 - 2\sqrt{p_i(t)} \frac{d_i\eta}{\beta_i} \lambda} \le M(t) - \frac{\lambda \left(1 - 2\sqrt{M(t)^3} \frac{d_i\eta}{\beta_i} \right)}{1 - 2\sqrt{p_i(t)} \frac{d_i\eta}{\beta_i} \lambda}$$

For $t = 0$, $M(0) = \max_{1\le i\le N} (\frac{N\beta_i}{C})^2$, and from (13) we have that $1 > 2\sqrt{M(0)^3} \frac{d_i\eta}{\beta_i}$, concluding that $M(1) \le M(0)$. Now, we use the mathematical induction principle. If $M(t) \le M(t-1)$ for all $t < k$ for $t = k$, $M(k) \le M(0)$ and thus $1 > 2\sqrt{M(k)^3} \frac{d_i\eta}{\beta_i}$ and therefore

$$M(k+1) \le M(k).$$

Thus, we showed that sequence $M(t)$ is nonincreasing.

Now we show that $m(t)$ is nondecreasing. Let, $p_i(t) = m(t) + \lambda$, for $\lambda \ge 0$. Then:

$$p_i(t+1) = \frac{m(t) + \lambda}{\left(1 + \sqrt{p_i(t)} \frac{\eta}{\beta_i} \sum_{(i,j)\in E} (m(t) + \lambda - p_j(t)) \right)^2} \ge$$

$$\frac{m(t) + \lambda}{\left(1 + \sqrt{p_i(t)} \frac{d_i\eta}{\beta_i} \lambda \right)^2}$$

Now we prove that the last expression is not smaller than $m(t)$.

$$m(t) \left(1 + \sqrt{p_i(t)} \frac{d_i\eta}{\beta_i} \lambda \right)^2 =$$

$$m(t) \left(1 + 2\sqrt{p_i(t)} \frac{d_i\eta}{\beta_i} \lambda + \left(\sqrt{p_i(t)} \frac{d_i\eta}{\beta_i} \lambda \right)^2 \right) \le$$

$$m(t) + 2\lambda M(t)^{\frac{3}{2}} \frac{d_i\eta}{\beta_i} + \lambda \left(M(t)^{\frac{3}{2}} \frac{d_i\eta}{\beta_i} \right)^2 \le$$

$$m(t) + 2\lambda \frac{1}{3} + \lambda (\frac{1}{3})^2 = m(t) + \frac{7}{9}\lambda \le m(t) + \lambda.$$

Above we used $\lambda = p_i(t) - m(t) \le M(t) - m(t) \le M(t)$ and $M(t)^{\frac{3}{2}} \frac{d_i\eta}{\beta_i} \le M(0)^{\frac{3}{2}} \frac{d_i\eta}{\beta_i} \le \frac{1}{3}$.

Step 2. In this step we rewrite the dynamics of $p_i(t)$ in a more practical form. Consider the representation of dynamics of $p_i(t)$ given by (16). From the Lagrange's mean value theorem, applied to the function $h(s) = \frac{1}{\sqrt{s}}$, differentiable on the interval $(p_i(t), p_i(t+1))$ (from Step 1, we know that $p_i(t) \ge m(0) > 0$ which implies the differentiability of function $h(s)$ on the interval $(p_i(t), p_i(t+1))$), we have that

$$\frac{1}{\sqrt{p_i(t+1)}} - \frac{1}{\sqrt{p_i(t)}} = (p_i(t+1) - p_i(t)) \left(-\frac{1}{2q_i(t)\sqrt{q_i(t)}} \right)$$

for some $q_i(t) \in (p_i(t), p_i(t+1))$. From the last relationship we conclude that the dynamics of $p_i(t)$ satisfies:

$$p_i(t+1) = p_i(t) - \frac{2q_i(t)^{\frac{3}{2}}\eta}{\beta_i} \sum_{(i,j)\in E} (p_i(t) - p_j(t)).$$

Therefore, the evolution of the vector $P(t) = (p_1(t), \ldots, p_N(t))$ can be written as:

$$P(t+1) = B(t)P(t),$$

where matrix $B(t)$ is given by $B(t) =$

$$\begin{bmatrix} 1-\frac{2q_1(t)^{\frac{3}{2}}d_1\eta}{\beta_1} & \frac{2q_1(t)^{\frac{3}{2}}\eta}{\beta_1}e_{1,2} & \cdots & \frac{2q_1(t)^{\frac{3}{2}}\eta}{\beta_1}e_{1,N} \\ \frac{2q_2(t)^{\frac{3}{2}}\eta}{\beta_2}e_{2,1} & 1-\frac{2q_2(t)^{\frac{3}{2}}d_2\eta}{\beta_2} & \cdots & \frac{2q_2(t)^{\frac{3}{2}}\eta}{\beta_2}e_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2q_N(t)^{\frac{3}{2}}\eta}{\beta_N}e_{N,1} & \cdots & \cdots & 1-\frac{2q_N(t)^{\frac{3}{2}}d_n\eta}{\beta_N} \end{bmatrix}$$

with $e_{i,j}$ being the elements of the adjacency matrix of G, ie. if $(i,j) \in E$, then $e_{i,j} = 1$ otherwise $e_{i,j} = 0$.

Step 3. We now use the monotonicity of sequences $M(t)$ and $m(t)$ proved in Step 1, to prove that nonzero elements of $B(t)$ are positive and uniformly bounded away from zero. Indeed, recall that $q_i(t) \in (p_i(t), p_i(t+1))$, and therefore $q_i(t) \leq \max(M(t), M(t+1)) \leq M(0)$, and therefore for diagonal entries we have that

$$1 - \frac{2q_i(t)^{\frac{3}{2}}d_i\eta}{\beta_i} \geq 1 - \frac{2M(0)^{\frac{3}{2}}d_i\eta}{\beta_i} \geq 1 - \frac{2}{3} = \frac{1}{3}.$$

For nonzero off-diagonal entries note that $q_i(t) \geq \min(m(t), m(t+1)) \geq m(0) > 0$ and thus

$$\frac{2q_i(t)^{\frac{3}{2}}\eta}{\beta_i} \geq \frac{2m(0)^{\frac{3}{2}}\eta}{\beta_i} = \delta_i > 0.$$

Take $\delta = \min\{\frac{1}{3}, \delta_1, \ldots, \delta_N\} > 0$. Then for all t, all nonzero elements of $B(t)$ are not smaller than δ.

Step 4. Finally, we use the fact that G is connected to show that $M(t) - m(t)$ converges to zero. This implies that $\lim_{t\to\infty} m(t) = \lim_{t\to\infty} M(t) = \lim_{t\to\infty} p_i(t) = p^*$. Let k be the diameter of graph G, i.e. the smallest integer such that there exist a path in G between each two nodes of length not greater than k. Then for all t:

$$D(t) = B(t+k-1)B(t+k-2)\cdots B(t)$$

is a stochastic matrix[6] with strictly positive entries and each entry of $D(t)$ is greater or equal than δ^k.

$$p_i(t+k) = \sum_{i=j}^{N} D_{ij}(t)p_i(t) \leq M(t)(1-\delta^k) + m(t)\delta^k$$

and

$$p_i(t+k) = \sum_{i=j}^{N} D_{ij}(t)p_i(t) \geq m(t)(1-\delta^k) + M(t)\delta^k.$$

Thus

$$M(t+k) - m(t+k) \leq (1-2\delta^k)(M(t) - m(t)). \qquad (18)$$

Since $M(t) - m(t)$ is a nonincreasing sequence and $\delta > 0$ is independent of t, we conclude that $M(t) - m(t) \to 0$, as $t \to \infty$. Now, convergence of $C_i(t)$ and (14) and (15) follow directly from the constraint (10). □

COMMENT 1. *From the bound (18), we can observe that the system converges to the equilibrium exponentially, with a rate bounded above by* $(1-\delta^k)^{\frac{1}{k}}$.

[6]Stochastic matrix is square matrix with nonnegative entries and sum of each row is 1. Since each of $B(t)$ is stochastic, their product is stochastic as well [2].

COMMENT 2. *In the networking community, a widely used approach (see [21, 30]) for analysis of nonlinear dynamical systems is linearization around the equilibrium, and presentation of some kind of local stability result: if system is close to equilibrium then it will stay there. We stress that our result posses an extra feature, it says that the system will actually reach the equilibrium.*

COMMENT 3. *While in the presented model we assume synchronous updates of C_i, this property of the model is not critical. See Section 5 for more details.*

The model analyzed, relies on the assumption that traffic mix is consisted of only TCP flows, which gives an exact relationship (9) between loss rate and the forwarding rate. However, this assumption is not necessary for the stability of C3P. Namely, suppose that C_i and p_i are related by a general relationship

$$C_i = g_i(p_i),$$

where $g_i : (0,1) \to (0,\infty)$. Then the following theorem gives a sufficient condition for η under which the C3P converge to equilibria. The proof follows the same lines as the proof of Theorem 1 and is omitted here.

THEOREM 2. *Let d_i be the degree of limiter i in the communication graph, and suppose that $g_i(\cdot)$ is a differentiable, convex function on (0,1), for all $1 \leq i \leq N$. Then if η satisfies:*

$$0 < \eta < \frac{1}{2} \min_{1\leq i\leq N}(-g_i'(p_i(0))) \min_{1\leq i\leq N} \frac{1}{d_i}, \qquad (19)$$

the following limits exist

$$\lim_{t\to\infty} C_i(t) = C_i^*$$

and

$$\lim_{t\to\infty} p_i(t) = p^*.$$

COMMENT 4. *C3P as well as the algorithm presented in the next section, D2R2, can be seen as instances of "distributed equation solving". Suppose that N agents want to solve the following equation in a distributed manner*

$$G(x) = \sum_{i=1}^{N} g_i(x) = C.$$

If each agent i is able to solve the equation $g_i(x) = y$ for every y (in C3P this translates to: for a given capacity y, x is measured loss rate for which the forwarding rate $g_i(x)$ is equal to y) then C3P-like algorithm with appropriate η converges to the solution of the above equation.

2.2 C3P minimizes the total number of drops in the TCP environments

C3P resource allocation exhibits a surprising feature in the TCP AIMD environments.

THEOREM 3. *Suppose that end-users employ AIMD congestion control algorithms. Then the allocation of bandwidth among N best-effort local limiters $(C_1, \ldots, C_N)$ which minimize the total number of drops across all N limiters is the one that C3P converges to.*

PROOF. From the assumption that all users use AIMD congestion control, we have that capacity C_i and loss rate p_i at the limiter i are related through a square root formula:

$$C_i = \frac{\beta_i}{\sqrt{p_i}}.$$

The number of drops at limiter i is $ND_i(C_i) = C_i p_i$, and the total number of drops across all limiters is:

$$ND(C_1, \ldots, C_N) = \sum_{i=1}^{N} ND_i(C_i) = \sum_{i=1}^{N} C_i p_i =$$

$$= \sum_{i=1}^{N} \frac{\beta_i^2}{C_i}.$$

From the Cauchy-Schwartz inequality:

$$\sum_{i=1}^{N} x_i^2 \sum_{i=1}^{N} y_i^2 \geq \left(\sum_{i=1}^{N} x_i y_i \right)^2$$

used for $x_i = \frac{\beta_i}{\sqrt{C_i}}$ and $y_i = \sqrt{C_i}$ we get:

$$ND(C_1, \ldots, C_N) \cdot C = \sum_{i=1}^{N} \left(\frac{\beta_i}{\sqrt{C_i}} \right)^2 \sum_{i=1}^{N} \sqrt{C_i}^2 \geq \left(\sum_{i=1}^{N} \beta_i \right)^2.$$

Thus

$$ND(C_1, \ldots, C_N) \geq \frac{\left(\sum_{i=1}^{N} \beta_i \right)^2}{C}$$

with equality if and only if there exist some γ such that for all i: $x_i = \gamma y_i$ which is equivalent to

$$p_i = \left(\frac{\beta_i}{C_i} \right)^2 = \left(\frac{x_i}{y_i} \right)^2 = \gamma^2.$$

Therefore, we showed that the vector $(C_1, \ldots, C_N)$ that minimizes total number of drops is the one for which the loss rates among all local limiters are equal. □

3. DISTRIBUTED DEFICIT ROUND ROBIN (D2R2)

The design of D2R2 is in some sense similar to the design of C3P. The terminology is the same: N local limiters need to control aggregate bandwidth at level C. The local limiter i has a controllable capacity C_i, and can exchange information with limiters j, such that (i, j) is an edge in the communication graph $G = (N, E)$ which is undirected and connected. However, the reference point in the formulation of the fairness postulate require that flows located at different limiters emulate global processor sharing. Formally it is given by (7).

To achieve this, we require that each limiter utilizes a processor sharing emulator; in our implementation it is Deficit Round Robin (DRR) [28]. While in the best-effort case the global objective was distributed emulation of a single best-effort queue in the PS case the global objective is distributed emulation of a single processor sharing queue.

At each limiter we use a number of virtual queues emulating token buckets of DRR scheduler. If the packet is discarded from the virtual DRR scheduler, then the arriving packet is dropped, otherwise it is forwarded to the appropriate output line. Thus, no queueing delay is caused by any limiter. The "fair-share" is measurable directly, and it depends on the traffic pattern. Roughly speaking, for a given bandwidth, the more flows exist at the limiter (meaning that aggregate aggressiveness is bigger) the lower "fair-share" is.

```
1   UpdateCapacities()
2     Once every Δ units of time do
3       for i = 1 : N
4         ṽ_i ← v_0(C_i, F_i)
5         R_i ← C_i − g_i(ṽ_i)
6         v_i ← ṽ_i + αR_i
7         C_i ← C_i + η Σ_{(i,j)∈E} (v_j − v_i)
8       endfor
9     enddo

10  InitializeCapacities()
11    for i = 1 : N
12      C_i ← C/N
13    endfor
```

Figure 2: Pseudo-code of D2R2

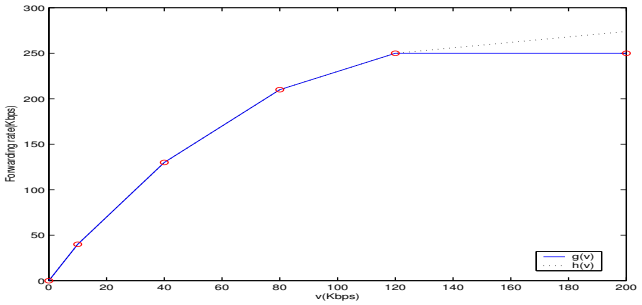

Figure 3: The piecewise-linear dependance between "fair share" and the forwarding rate of a PS scheduler. $n = 4$ flows with demands $u_1 = 10Kbps$, $u_2 = 40Kbps$, $u_3 = 80Kbps$, $u_4 = 120Kbps$.

Before we proceed, we need to explore the nature of the relationship between the demand, "fair-share" and the capacity of a DRR scheduler. Let $\mathcal{F}_i = \{f_1, f_2, \ldots, f_{n_i}\}$ a family of n_i flows, with demands $u_s^{(i)}$, and $g_i : R^+ \to R^+$ be a function given by

$$g_i(v) = \sum_{s=1}^{n_i} \min(u_s^{(i)}, v).$$

Without loss of generality we can assume that $u_1^{(i)} \leq u_2^{(i)} \leq \cdots \leq u_{n_i}^{(i)}$. Then, $g_i(\cdot)$ is a piecewise linear concave function on $(0, \infty)$ and its graph has a form depicted in Figure 3. The "fair-share", $v_0(C_i, \mathcal{F}_i)$, defined as the maximal rate across the all flows from $\mathcal{F}_i$ utilizing DRR scheduler with bandwidth C_i is either:

(1) (the unique) solution of the equation $g_i(v) = C_i$ if $\sum_{s=1}^{n_i} u_s^{(i)} > C_i$, or

(2) $v_0(C_i, \mathcal{F}_i) = u_{n_i}^{(i)}$, if $\sum_{s=1}^{n_i} u_s^{(i)} \leq C_i$.

The residual bandwidth $R_i(C_i) = C_i - g_i(v_0(C_i, \mathcal{F}_i))$ is strictly positive if the demand $\sum_{s=1}^{n_i} u_s^{(i)}$ is strictly smaller than capacity (otherwise $R_i(C_i) = 0$).

Now we are ready to present the Distributed DRR (D2R2) algorithm. Its pseudo-code is given in Figure 2. The algorithm has two parameters $\alpha \geq 1$ and $\eta > 0$. The parameter α determines the accuracy of the algorithm, while η

determines responsiveness and stability properties of the algorithm. The algorithm updates the local limiter's rates $(C_1, \ldots, C_N)$ in discrete time steps. In each time step, "fair-share" $\tilde{v}_i$ and the residuum R_i is computed at each limiter. The augmenting "fair-share" is the sum $v_i = \tilde{v}_i + \alpha R_i$. The algorithm uses a consensus approach similar to those of C3P to equalize all v_i at some value v^*_α. The key observation is that, v^*_α is at most $O(\frac{1}{\alpha})$ away from the solution v^* of the equation (6) (Theorem 5).

3.1 Model and analysis of D2R2

The goal of this section is to analyze the dynamics and the convergence properties of the D2R2 algorithm. For a given α we will establish sufficient conditions on η under which set of local limiter rates converge (to some $C(\alpha) = (C_1(\alpha), \ldots, C_N(\alpha))$) in Theorem 4. Then, in Theorem 5, we will show that vector $C(\alpha)$ is at most $O(\frac{1}{\alpha})$ away to the vector C^* for which (7) is satisfied.

Denote by $C_i(t)$ and $v_i(t)$ the capacity of the limiter i and the augmenting "fair-share" (v_i) at the time step t. From the initialization step we know that: $C_i(0) = C/N$. The update rule (line 7 in Figure 2) allows us to write the dynamics of C_i in the following form:

$$C_i(t+1) = C_i(t) + \eta \sum_{(i,j)\in E} (v_j(t) - v_i(t)). \qquad (20)$$

Note that since G is an undirected graph the capacity constraint (12) is satisfied for all t. From the definition of the augmenting "fair-share" (lines 4-6 in Figure 2), we have that $C_i(t)$ and $v_i(t)$ are related in the following manner:

$$C_i(t) = h_i(v_i(t)), \qquad (21)$$

where $h_i : R^+ \to R^+$ is the following, *strictly increasing* function:

$$h_i(v) = g_i(v) + \frac{1}{\alpha} max(0, v - \sum_{s=1}^{n_i} u_s)$$

The following lemma is an obvious consequence of the definition of $h_i(\cdot)$.

LEMMA 2. *Let $\alpha \geq 1$. Then:*
(1) $h_i(\cdot)$ is a piecewise linear, strictly increasing, concave function on the whole domain $(0, \infty)$.
(2) For any $x, y \in R^+$:

$$|h_i(x) - h_i(y)| \geq \frac{1}{\alpha}|x - y|. \qquad (22)$$

The following result gives simple sufficient condition under which the D2R2 algorithm is stable, and that augmenting "fair-shares" at all local limiters are equal in steady-state.

THEOREM 4. *Let $\alpha \geq 1$. If*

$$0 < \eta \leq \min_{1\leq i\leq N} \frac{1}{2\alpha d_i},$$

then the system (20)-(21) converges to a stable point $(C^{(\alpha)}, v^{(\alpha)})$ and all components of vector $v^{(\alpha)}$ are equal.

PROOF. We will write the system the system (20)-(21) in terms of v_i only:

$$h_i(v_i(t+1)) = h_i(v_i(t)) + \eta \sum_{(i,j)\in E} (v_j(t) - v_i(t)). \qquad (23)$$

From the Lemma 2, for the increasing function h_i, we have that there exists $q_i(t) \geq \frac{1}{\alpha}$ such that:

$$h_i(v_i(t+1)) - h_i(v_i(t)) = q_i(t)(v_i(t+1) - v_i(t)).$$

Thus

$$v_i(t+1) = v_i(t) + \frac{\eta}{q_i(t)} \sum_{(i,j)\in E} (v_j(t) - v_i(t)),$$

which can be rewritten in vector form as

$$v(t+1) = B(t)v(t)$$

where:

$$B(t) = \begin{bmatrix} 1 - \frac{d_1\eta}{q_1(t)} & \frac{\eta}{q_1(t)}e_{1,2} & \cdots & \frac{\eta}{q_1(t)}e_{1,N} \\ \frac{\eta}{q_2(t)}e_{2,1} & 1 - \frac{d_2\eta}{q_2(t)} & \cdots & \frac{\eta}{q_2(t)}e_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\eta}{q_N(t)}e_{N,1} & \cdots & \cdots & 1 - \frac{d_N\eta}{q_N(t)} \end{bmatrix}.$$

For the diagonal elements of $B(t)$ we have:

$$1 - \frac{d_i\eta}{q_i(t)} \geq 1 - \alpha d_i \eta \geq 1 - \frac{1}{2} = \frac{1}{2}.$$

For nonzero off-diagonal entries of $B(t)$ we have

$$\frac{\eta}{q_i(t)} \geq \alpha\eta.$$

Thus, for every $t \geq 0$, all nonzero entries of $B(t)$ are greater than $\delta = \min(\frac{1}{2}, \alpha\eta)$. Thus $B(t)$ is a stochastic matrix, and therefore $M(t) = \max_{1\leq i\leq N}(v_i(t))$ is nonincreasing sequence and $m(t) = \min_{1\leq i\leq N}(v_i(t))$ is a nondecreasing sequence. Now, after establishing that all nonzero elements of $B(t)$ are uniformly bounded away from zero, using the same argument from the proof of Theorem 1 (step 4) we have that $M(t) - m(t) \to 0$ as $t \to \infty$. Thus, $\lim_{t\to\infty} M(t) = \lim_{t\to\infty} m(t) = v^{(\alpha)}$ for some $v^{(\alpha)}$ and for all i:

$$\lim_{t\to\infty} v_i(t) = v^{(\alpha)}. \qquad (24)$$

And also:

$$\lim_{t\to\infty} C_i(t) = h_i(v^{(\alpha)}) = C_i^{(\alpha)}. \qquad (25)$$

□

From the definition of the augmenting "fair-share" we know that the rate of the flow f_s^i at the limiter i, with steady-state capacity $C_i^{(\alpha)}$ is

$$rate(f_s^i) = \min(u_s^i, v^{(\alpha)}).$$

From the capacity constraint we know that

$$\sum_{i=1}^{N} h_i(v^{(\alpha)}) = \sum_{i=1}^{N} C_i^{(\alpha)} = C \qquad (26)$$

Thus $v^{(\alpha)}$ is the unique solution of the equation

$$\sum_{i=1}^{N} h_i(v) = C. \qquad (27)$$

Since v^* is the solution of the equation (6) and $g_i(v) \leq h_i(v)$ for all $v > 0$, we conclude that:

$$v^{(\alpha)} \leq v^*.$$

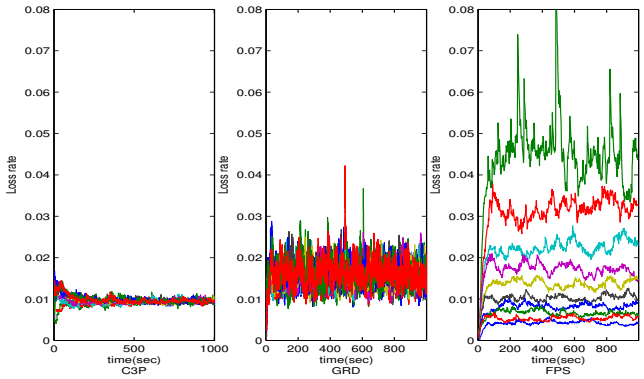

Figure 4: Loss rates of three oblivious C3P, GRD and FPS. Different lines correspond to loss rates at 10 different local limiters.

The next theorem gives a conservative bound, on how far is $v^{(\alpha)}$ from v^*.

THEOREM 5. *Let $\alpha \geq 1$ and M be the number of flows in $\mathcal{F}$ with demand not less than v^*. Then:*

$$0 \leq \frac{v^* - v^{(\alpha)}}{v^*} \leq \frac{N}{M\alpha}.$$

PROOF. To begin the proof, note that for all $v > 0$:

$$h_i(v) = g_i(v) + \frac{1}{\alpha} max(0, v - \sum_{s=1}^{n_i} u_s) \leq g_i(v) + \frac{1}{\alpha} v.$$

Then, from the fact that $v^{(\alpha)}$ is the solution of the equation (27) we have:

$$C = \sum_{i=1}^{N} h_i(v^{(\alpha)}) \leq \sum_{i=1}^{N} (g_i(v^{(\alpha)}) + v^{(\alpha)} \frac{1}{\alpha}).$$

Thus

$$G(v^{(\alpha)}) = \sum_{i=1}^{N} g_i(v^{(\alpha)}) \geq C - \frac{N v^{(\alpha)}}{\alpha}. \qquad (28)$$

The right-hand derivative of $G(v)$ at $v = v^*$ is equal to

$$G'_+(v^*) = \sum_{i=1}^{N} \sum_{s=1}^{n_i} (\min(u_s^{(i)}, v^*))'_+ = M.$$

From the concavity of G we have that

$$G(v^*) - G(v^{(\alpha)}) \geq G'_+(v^*)(v^* - v^{(\alpha)}) = M(v^* - v^{(\alpha)}).$$

Combining the last inequality with (28) and using fact that $G(v^*) = C$ and we obtain:

$$M(v^* - v^{(\alpha)}) \leq C - G(v^{(\alpha)}) \leq \frac{N v^{(\alpha)}}{\alpha} \leq \frac{N v^*}{\alpha},$$

which implies the statement of the theorem. □

COMMENT 5. *The parameter α, does not affect the algorithm as long as the demands at the limiters are greater than their capacities, since in that case the residuum bandwidth, R_i, is actually zero. In that case the augmenting fair-share is equal to the fair-share and $v^{(\alpha)} = v^*$ for any $\alpha \geq 1$ meaning that D2R2 converges exactly to the solution of processor sharing DRL problem. It is only in low demand regimes when the need of introducing α and augmenting fair-share arises.*

4. EVALUATION

In this section we present empirical results obtained by *ns2* packet level simulation. We first compare C3P and D2R2 with existing DRL algorithms (GRD and FPS) using several performance metrics. We then validate our theoretical results experimentally and present a number of results that shows the behavior of our DRL algorithms under more dynamic traffic-mix settings. Finally, we discuss the effects of the cloud size and topology on the speed of convergence of C3P and D2R2.

The DRL algorithms are implemented and evaluated using *ns2*. The results presented assume loss-free communication between the limiters. Some small loss (say less than 1%) of information between limiters have negligible effects on all four algorithms evaluated here and will not be discussed in the rest of this section.

4.1 Comparison with GRD and FPS

The first set of experiments compare the proposed algorithms with schemes from [26], Global Random Drop (GRD) and Flow Proportional Share (FPS), over a number of performance metrics. The simulation setup is the following. There are $N = 10$ local limiters indexed with numbers 1,2, ...,10. Limiter i communicates with limiters $(i-1)$ *mod* 10 and $(i+1)$ *mod* 10 (which means that the communication graph is the ring). Limiters collaborate in order to achieve aggregate rate limiting at the level of $C = 40Mbps$. The limiter i serves i TCP flows (therefore there are $1+2+\cdots+10 = 55$ flows in total) with packet sizes 1000 *bytes* and round trip times

$$RTT_j^{(i)} = (8 \cdot i + 30 \cdot j)ms \quad \text{for } j \in \{1, 2, \ldots, i\}.$$

The choice of GRD and FPS parameters is based on the suggestions from [26]. The estimate intervals for GRD and FPS are $100ms$ and $500ms$ respectively and exponential weighted mean averaging (EWMA) is performed once every 1 second with EWMA parameter 0.1. The parameter $\eta_{C3P} = 200000bps$ is just under the upper bound (13) for stability. For D2R2 we use $\alpha = 1$ and $\eta_{D2R2} = 0.1$. The exchange of information between limiters is $\Delta = 2sec$ in both C3P and D2R2.

We evaluate the following performance parameters.

(a) Loss rates. Per-limiter loss rates provide information on the aggregate bandwidth allocation. From the discussion in Section 1.1 we know that a DRL algorithm emulates a centralized best effort FIFO queue if and only if loss rates at all limiters are equal. Thus, by looking at per-limiter loss rates it is easy to test whether a DRL algorithm emulates centralized best effort queue or not. We look at three DRL schemes with token bucket local limiters: C3P, GRD and FPS. In case of D2R2, local limiters use processor sharing queues and therefore each flow has its own loss rate, so per-limiter loss rate is not a meaningful performance metric. The results are presented in Figure 4. The C3P and GRD algorithms have relatively uniform loss rates, indicating that they emulate the behavior of the centralized limiter relatively well. We also note that loss rates in GRD case are

.	C3P	GRD	FPS	D2R2
mean($Mbps$)	39.98	35.68	38.66	40.00
std($Mbps$)	2.98	2.72	2.57	1.82

Table 1: Mean and standard deviation (std) of the $500ms$-aggregate forwarding rates in C3P, GRD, FPS and D2R2.

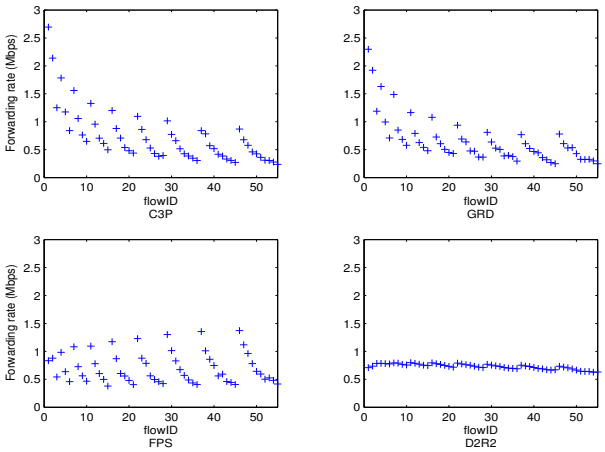

Figure 5: Achieved flow rates of 55 concurrent flows.

slightly bigger, and have a larger variance than in C3P. This is a consequence of the "synchronization" effect observed in [26]: when aggregate forwarding rate reaches aggregate limit C many flows experience (unnecessary) multiple losses. On the other hand, FPS per-limiter loss rates vary greatly indicating that FPS does not match the behavior of the centralized best effort limiter.

(b) Aggregate rate control. A major performance feature of a DRL algorithm is its ability to control the aggregate forwarding rate at the prescribed level C. Table 4.1 contains the mean and standard deviation of time series representing the $500ms$-aggregate-forwarding-rate. We can notice high accuracy of C3P and D2R2 in terms of achieving the mean aggregate forwarding rate very close to C. GRD exhibits almost 10% smaller aggregate rate mainly because of the "synchronization" effect (see previous paragraph).

(c) Flow rate allocation. Figure 5 depicts the average flow rates of 55 flows[7] utilizing 10 limiters employing four architectures C3P, GRD, FPS and D2R2. The Jain's fairness indices (JFI) [16] of r flows achieving sending rates $\mathbf{x} = (x_1, \ldots, x_r)$ is given by

$$JFI(\mathbf{x}) = \frac{\left(\sum_{i=1}^{r} x_i\right)^2}{r \sum_{i=1}^{r} x_i^2}. \qquad (29)$$

JFI's under each of four schemes are given in Table 4.1. We note the following facts. (1) In C3P and GRD case,

[7]Flows with FlowID from $\frac{i(i-1)}{2}+1$ to $\frac{i(i+1)}{2}$ are served by limiter i.

.	C3P	GRD	FPS	D2R2
JFI	0.702	0.722	0.871	0.996

Table 2: Jain's fairness indices for four schemes.

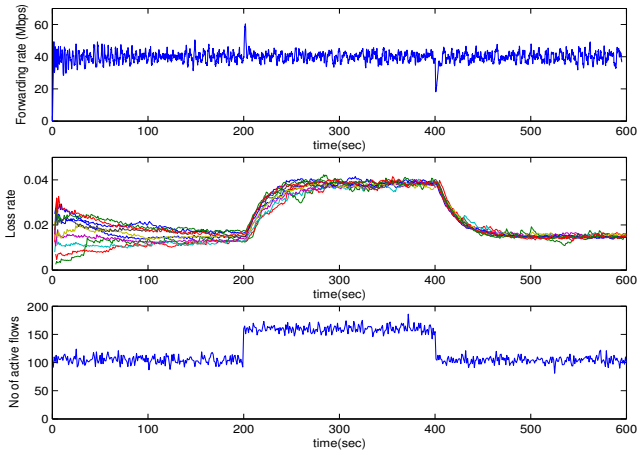

Figure 6: C3P under dynamic traffic. Aggregate forwarding rate (top), per-limiter loss rates (middle) and the number of active flows (bottom).

per-limiter loss rates are almost identical which implies that bandwidth allocated to each flow is inversely proportional to its round-trip time[8]. (2) Jain's fairness index under D2R2 is close to 1 indicating that problem of distributed processor sharing discussed in Section 1.1 is successfully solved by D2R2 in the given example. (3) The heuristic nature of FPS that uses per-limiter weight (defined as the ratio between the capacity of the limiter and the maximum flow-rate among all flows utilizing the limiter) as an estimator of the number of flows exhibits unpredictable bandwidth allocation among flows that use different limiters.

(d) Communication overhead. Since all four algorithms use UDP packets (with size of approximately 50 bytes) the communication overhead of any of those DRL algorithms is inversely proportional to the update interval Δ. Suggested values from [26] are $\Delta_{GRD} = 0.1sec$ and $\Delta_{FPS} = 0.5sec$. Those sub-second update intervals in GRD and FPS are necessary because their actions require instantaneous information of the global aggregate demand/weight. In contrast, C3P and D2R2, emulate centralized queue on a longer time scale requiring less frequent updates. In our implementations we use $\Delta_{C3P} = \Delta_{D2R2} = 2sec$.

4.2 Dynamic traffic distributions

The previous subsection dealt with static traffic conditions and compares the main performance metrics over four DRL schemes. Internet traffic changes its traffic mix patterns and here we evaluate the ability of C3P and D2R2 to adapt their behavior under those changes. The basic setup is the same as in the previous subsection: there are $N = 10$ limiters, with limiter i serving i long-lived TCP flows. In addition to those long-lived TCP flows each limiter serves 50 on-off sources with demand of $100Kbps$ during on-times. On and off times are drown with Pareto distributions with shape 1.5 and means $1sec$ and $9sec$ respectively. Those on-off sources introduce low-intensity changes on short time scales. To evaluate how algorithms behave under more abrupt traffic changes, we duplicate the number of long-lived TCP flows at time $t_1 = 200sec$ and reduce this to 55 long-lived flows at

[8]This follows directly from the square root formula (8).

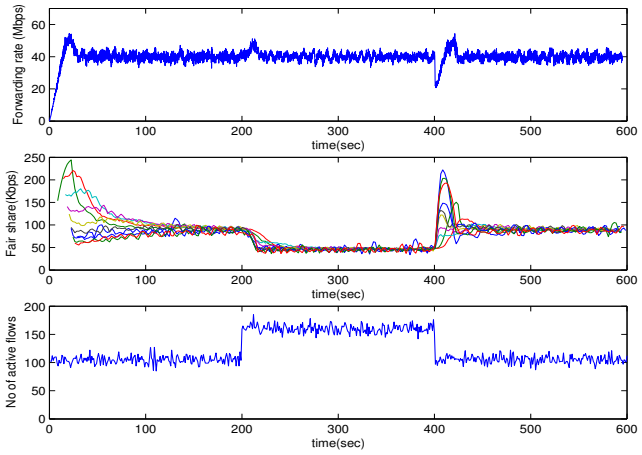

Figure 7: D2R2 under dynamic traffic. Aggregate forwarding rate (top), per-limiter fair share (middle) and the number of active flows (bottom).

time $t_2 = 400sec$.

The resulting per-limiter loss rates (in C3P) and fair-shares (in D2R2) are depicted in Figure 6 and 7, along with aggregate forwarding rate and number of active flows (summed over all limiters). From those figures we can see that: (1) short-time scale changes of traffic mix, induced by on-off sources have negligible effect in the stability of C3P and D2R2, as both of them adapt to those changes in an instantaneous manner. (2) more abrupt traffic changes require some time for the algorithms to converge to the desired regime. However, Theorems 1 and 4 guarantee convergence and this is empirically confirmed by the presented simulations.

4.3 Speed of convergence

To evaluate how fast the algorithms converge to the equilibrium one requires a metric that measures how equalized are the loss rates (in C3P) and "fair shares" (in D2R2). We find it convenient to use Jain's fairness index given by (29) to measure the discrepancies between per-limiter loss rates ($p(t)$) and per-limiter fair shares ($v(t)$).

The simulation setup is the following. There are N local limiters, and limiter i serves n_i TCP flows, where n_i is randomly drown from the interval [1,10] with uniform distribution. The TCP flows have a packet size of 1000 bytes and RTT randomly drown from the interval [100ms,500ms] with uniform distribution. The communication graph is a random d-regular graph[9]. The update interval is $\Delta = 2sec$, and the total capacity is $C_N = (N)Mbps$. The η is chosen at the upper bounds from the Theorems 1 (for C3P) and 4 (for D2R2). Two choices of N and two choices of d are evaluated, and the evolution of $JFI(p(t))$ and $JFI(v(t))$ are depicted in Figures 8 and 9. From those Figures we can observe: (1) It takes a couple of minutes for $JFI(p(t))$ to converge to a value close to 1 in C3P while the convergence of D2R2 is faster (less than a minute in the presented simulations). (2) More surprising and a rather intriguing observation is that for the choice of η given by the upper bounds from the Theorems 1 and 4, the convergence speed (in terms of JFI dynamics) does not appear to depend on the size of cloud or density of the communication graph, but rather on the traffic mix distributions.

[9]An undirected graph is d-regular if its every node has degree d.

5. IMPLEMENTATION ISSUES

Asynchronous updates. The algorithms C3P and D2R2 assume that local-limiter capacities are updated in a synchronized manner using a connected undirected communication graph G. However, this is not necessary to ensure convergence. To see this, suppose that in each time instant t only some subset of nodes exchange information, and that that information exchange is characterized by an undirected graph G_t. If there is some $T > 0$ such that for every τ, $\cup_{t=\tau}^{\tau+T} G_t$ is connected, then the method developed in [14] can be used to establish the convergence of the C3P and D2R2.

Message passing. Communication between two local limiters is performed via small UDP packets. Each packet should contain a field for loss rate in C3P case (augmenting fair share in D2R2 case), as well as some control overhead to ensure that if a loss of a communication packet occurs no local limiter gains or loses extra capacity, and that the capacity constraint (1) is not violated.

Communication delays. The message passing between two local limiters causes some communication delay on a time scale from few milliseconds up to a couple of hundreds of milliseconds. These communication delays could cause some issues related to the stability of the distributed algorithms if the update interval is on some small time scale. However, the time between updates, given by Δ, is on the order of magnitude of several seconds. This is necessary to obtain a good estimate of loss rates ("fair share" in D2R2). This resulting separation of time-scales ensures that effects of the communication delays on the stability of our algorithms may be neglected. Notwithstanding this fact, the issue of delays is a topic for future research.

Node Failures. In cases of a node (local-limiter) failure, it is possible for a loss of aggregate bandwidth to occur (since the capacity constraint (1) would be violated). A simple method for resolving this issue is the following. Let each local-limiter i chose a *best-friend* local-limiter[10] b_i among neighbor nodes in the communication graph G, and let each node inform node b_i of its local rate limit C_i. In the case of failure, local limiter b_i inherits bandwidth of node i, by simply setting $C_{b_i} = C_{b_i} + C_i$. Then the algorithms themselves will eventually adapt capacities of the non-failed limiters to the desired regime.

6. SUMMARY

Issues related to service reliability, service availability, and fault tolerance, have encouraged many service providers in the Internet to shift from traditional centric services to cloud based services. This trend appears to be a dominant mechanism for ensuring robustness of internet services with many "big players", such as Google, Yahoo!, Akamai, Amazon, already offering a suit of cloud-based services.

Pricing, usage control, and resource allocation of cloud based services represent important technical challenges for the networking community. The Distributed Rate Limiting paradigm is a step forward in resolving those issues. The

[10]Note that if j is best-friend of node i, that it does not necessarily mean that i is the best-friend of node j.

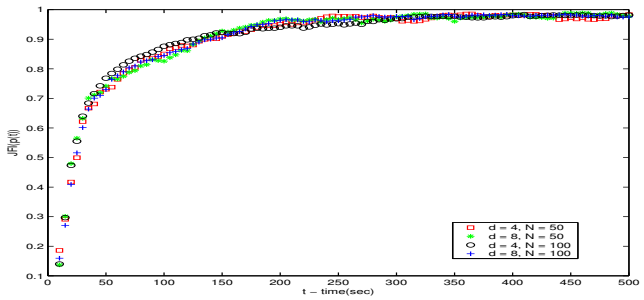

Figure 8: C3P convergence times: $JFI(p(t))$ dynamics.

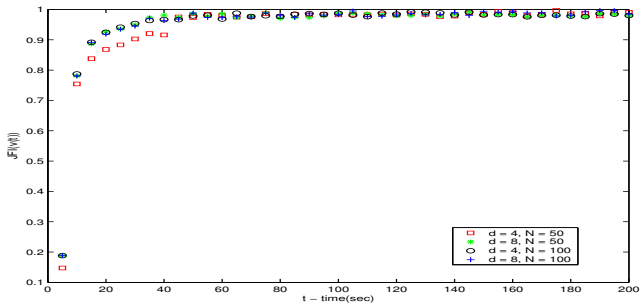

Figure 9: D2R2 convergence times: $JFI(v(t))$ dynamics.

two algorithms, GRD and FPS, presented in [26] are designed in an ad hoc manner using some global information (obtainable via some distributed algorithm). The heuristic nature of those algorithms gives rise to unpredictable performance (in terms of aggregate utilization, loss rates, bandwidth allocation etc.) and, equally importantly, makes those algorithms analytically untractable. In this paper we make a step forward in the design of DRL algorithms. Our first algorithm, Cloud Control with Constant Probabilities (C3P) gives an elegant, fully decentralized, solution to the DRL problem in best effort environments. The dynamics of C3P is analyzed and closed-form sufficient conditions are obtained that ensure *global* stability of the nonlinear system describing the dynamics of C3P.

The DRL problem in processor sharing environments simply states that we need a distributed algorithm for emulating centralized processor sharing queue on each local limiter. It turns out that this is slightly harder problem to solve compared to the best effort environments. Our algorithm Distributed Deficit Round Robin (D2R2), parameterized with parameter α, converges to a solution that is $O(\frac{1}{\alpha})$ away from the global optimum. We note here, that in case of large demands D2R2 indeed converge to the global optimum and it is only in low demand case when we cannot ensure the exact convergence. However, it would be interesting to obtain a distributed algorithm that converge to the global optimum (rather than $O(\frac{1}{\alpha})$ neighborhood of it) in all cases.

7. ACKNOWLEDGEMENTS

This work is supported by the Science Foundation Ireland grant 04/IN3/I460. The authors would like to thank Martin Corless for valuable discussions related to the stability of the algorithms.

8. REFERENCES

[1] Amazon Simple Storage Service(S3):http://aws.amazon.com/s3.
[2] A. Berman, R. Plemmons. "Nonnegative matrices in the mathematical sciences". SIAM, 1979.
[3] D. Bertsekas, R. Gallager. "Data Networks". 1987.
[4] S. Bhatnagar, B. Nath. "Distributed admission control to support guaranteed services in core-stateless networks". In Proceedings of IEEE INFOCOM, 2003.
[5] S. Boyd, A. Ghosh, B. Prabhakar, D. Shah. "Gossip algorithms: Design, analysis and applications". In Proceedings of IEEE INFOCOM, 2005
[6] D. F. Carr. "How Google works". Baseline Magazine, July 2006.
[7] G. Carraro, F. Chong. "Software as a service (SaaS): An enterprise perspective". MSDN Solution Architecture Center, Oct. 2006.
[8] J. Dilley et al., "Globally Distributed Content Delivery". IEEE Internet Computing, vol. 6(5), 2002.
[9] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, J. van der Merive "A flexible model for resource management in virtual private networks". In Proceedings of ACM SIGCOMM 1999.
[10] V. Dumas, F. Guillemin, P. Robert. "A Markovian analysis of additive-increase multiplicative-decrease algorithms". Adv. in Appl. Probab. 34 (2002), no. 1, 85Ű-111.
[11] R. Gibbens, F. Kelly. "Distributed Connection Acceptance Control for a Connectionless Network", 16th International Teletraffic Conference, Edimburgh, June 1999.
[12] D. Hinchcliffe. "2007: The year enterprises open thier SOAs to the Internet". Enterprise Web 2.0, Jan. 2007.
[13] M. Huang. "Planetlab bandwidth limits". Available online: http://www.planet-lab.org/doc/BandwidthLimits.
[14] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules". IEEE Transactions on Automatic Control, vol. 48(6), 2003
[15] A. Jain, J. M. Hellerstein, S. Ratnasamy, D. Wetherall. "A wakeup call for internet monitoring systems: The case for distributed triggers". In Proceedings of HotNets-III, 2004.
[16] R. Jain. "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling". John Wiley and Sons, INC., 1991.
[17] D. Kempe, A. Dobra, J. Gehrke. "Gossip-based computation of aggregate information". In Proceedings of IEEE FOCS, 2003.
[18] C. King, R. Shorten, F. Wirth, M. Akar. "Growth Conditions for the Global stability of Highspeed Communication Networks". To appear in IEEE Transactions on Automatic Control, 2008.
[19] M. Kodialam, T. Lakshman, S. Sengupta. "Maximum Throughput Routing of Traffic in the Hose Model". In Proceedings of IEEE INFOCOM 2006.
[20] A. Kumar, R. Rastogi, A. Siberschatz, B. Yener. "Algorithms for provisioning virtual private networks in the hose model". IEEE/ACM Trans. on Networking, vol 10(4), 2002.
[21] S. Kunniyur, R. Srikant. "Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management". *IEEE/ACM Trans. on Networking*, vol. 12(2).
[22] L. Moreau. "Stability of multiagent systems with time-dependent communication links". IEEE Transactions on Automatic Control, 2005.
[23] A. Odlyzko. "Internet pricing and the history of communications". Computer Networks, vol. 36, 2001.
[24] J. Padhye, V. Firoiu, D. F. Towsley, J. F. Kurose. "Modeling TCP Reno performance: a simple model and its empirical validation". *IEEE/ACM Trans. on Networking*, vol 8(2), 2000.
[25] A. Parekh, R. Gallager. "A generalized processor sharing approach to flow control in integrated services networks: the single-node case". IEEE/ACM Trans. on Networking, vol. 1(3).
[26] B. Raghavan, K. Vishwanath, S. Rambhadran, K. Yocum, A. Snoeren. "Cloud Control with Distributed Rate Limiting".In Proceedings of ACM SIGCOMM 2007.
[27] R. Olfati-Saber. "Flocking for multi-agent dynamic systems: algorithms and theory". IEEE Trans. on Auto. Control, 2006.
[28] M. Shreedhar, G. Varghese. "Efficient fair queueing using deficit round-robin". *IEEE/ACM Trans. on Networking*, 1996.
[29] R. Srikant. "Internet congestion control". Control theory, 14, Birkhäuser Boston Inc., Boston, MA, 2004.
[30] D. Wei, C. Jin, S. Low, S. Hegde. "FAST TCP: motivation, architecture, algorithms, performance". IEEE/ACM Trans. on Networking, 2007.