Overload Balancing in Single-hop Networks with Bounded Buffers

Xinyu Wu, Dan Wu, Eytan Modiano
Laboratory for Information and Decision Systems, MIT, USA
{xinyuwu1,danwumit,modiano}@mit.edu

***Abstract*—We consider the problem of overload balancing in single-hop networks with bounded buffers. We show that the backpressure policy, which is known to achieve the most balanced overload for networks with unbounded buffers, does not balance the overload for networks with bounded buffers. We formulate the problem of overload balancing in single-hop networks with bounded buffers by leveraging ordinary differential equations (ODE) to model the queue dynamics. We prove that choosing service rates on each transmission link that minimizes the quadratic sum of queue overload rates leads to the most balanced overload. Based on this result, we propose a queue-based policy combining maxweight scheduling with backpressure, which can asymptotically achieve the most balanced overload agnostic of packet arrival rate and capacity information. The proof technique is based on a novel characterization of the policy in a differentiable form, which is of independent interest. We further propose a distributed version of the policy, which reduces overhead by an order of magnitude. We evaluate our proposed policies under single-hop network and their concatenation into Clos structure, under randomly selected packet arrival rates, link capacities, and buffer sizes. Results demonstrate that our proposed policy converges to the most balanced overload in all cases, and the distributed version is nearly optimal.**

I. Introduction

Network overload occurs when user demand surpasses network service capacity. Data packets accumulate in network buffers and cause congestion. Multiple reasons contribute to network overload: demand surge [1], denial-of-service attacks [2], server shutdown and misconfiguration [3], [4], transmission link failure [5], etc. Such overload can result in detrimental consequences such as throughput reduction [6], [7], increased latency [8], or unfairness where some sessions "starve" other sessions [9], [10]. Network overload may occur in datacenters and server farms on a regular basis due to bursty demand [3].

An important measure to impede severe overload is *overload balancing*, which ensures that all sessions are equally affected by the congestion, and is a desired attribute of network control policies [9]. A number of works considered the problem of overload balancing. In [9], Georgiadis and Tassiulas demonstrated that the *backpressure* policy can achieve most balanced overload in a network with unbounded buffers under the criterion of lexicographic minimum, a concept related to min-max optimization [11]. More recent works study specific network structures. For parallel queues, [10] considered overload balancing by introducing explicit constrainits on fairness level, and [12] studied packet dropping policies to control the flow. For server farms, [3] generalized different fairness notions through α-fair penalty functions, which allowed for a convex optimization formulation. For cloud systems, [13] studied detection and balancing the transient overload through distributed optimization.

However, the above works did not study the effect of bounded buffers, whose sizes are limited in any practical setting, from networks on-chip to spacecraft networks [14]. We point out that bounded buffers can significantly affect the resulting policy under network overload. Consider the 3×1 switched network in Fig. 1. According to [9], if the egress node d has unbounded buffer, the backpressure policy[1] guarantees that in the steady state, the queue overload rates in all 4 nodes are the same, which is most balanced. However, if node d has bounded buffer, which means its buffer size is finite, then backpressure will fill up the buffer in the steady state, and lead to overload imbalance as explained in the figure caption. Therefore, figuring out effective transmission policies to balance the queue overload in bounded-buffer systems is nontrivial and practically significant.

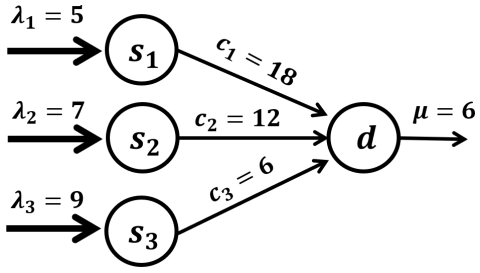

Fig. 1: Suppose all ingress nodes have unbounded buffer. When node d has unbounded buffer, backpressure achieves most balanced overload rates where all 4 nodes grow with rate 3.75; When node d has finite buffer, then $\dot{q}_d = 0$ in the steady state due to buffer saturation, and backpressure achieves overload rates $(\dot{q}_{s_1}, \dot{q}_{s_2}, \dot{q}_{s_3}, \dot{q}_d) = (2, 5, 8, 0)$, deviating significantly from the most balanced one $(5, 5, 5, 0)$.

In this paper, we propose a general analytical framework based on ordinary differential equations (ODE) to characterize queue dynamics, which we demonstrate can capture general

This work was supported by NSF grant CNS-1735463, and by a fellowship from MathWorks. The opinions and views expressed in this publication are of the authors and not necessarily from MathWorks or NSF.

ISBN 978-3-903176-48-5

[1]The definition of backpressure is deferred to Section IV (eqn. (8)).

buffer settings, and facilitate analytical results and policy design. We concentrate on single-hop networks modeled as a bipartite graph connecting ingress and egress nodes, each with a buffer to store incoming packets. Fig. 2 shows an example of bipartite graph, and real world infrastructures matching this structure: switched networks with ingress and egress buffering, and server farms with packet dispatchers as ingress nodes and servers as egress nodes. This work can serve as the foundation for future work on multi-hop structures, such as datacenter network that are made up of multiple single-hop structures [15].

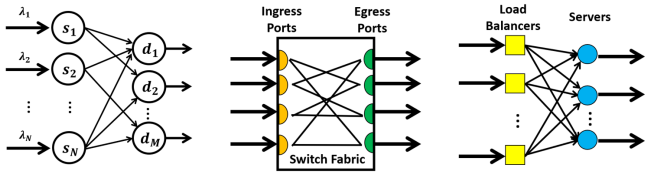

Fig. 2: (a) Single-hop structure as a bipartite graph; (b) Network switch with ingress and egress buffering; (c) Server farm with load balancers and servers.

Specifically, our contributions are as follows. (i) We prove that minimizing the quadratic sum of queue overload rates leads to the minimum of the max overload rate among all nodes, and also lexicographic minimum of queue overload rates. The quadratic sum minimization offers an equivalent but more tractable way to analyze the most balanced overload, compared with lexicographic minimization in [9]. (ii) Agnostic of packet arrival rates and link capacities, we prove that a policy combining *maxweight and backpressure (mw+bp)* achieves the most balanced overload in single-hop networks, which only requires queue information. We show that our ODE formulation can embed queue-based policies under bounded buffers elegantly based on a novel characterization of the policy in differentiable form. (iii) From a practical perspective, we propose a distributed version of the mw+bp policy which significantly reduces communication overhead. (iv) We verify our proposed policies by simulation in single-hop structures and their concatenations (Clos structure [16]), under randomly selected settings of packet arrival and departure rates, link capacities, and buffer settings. We show the mw+bp converges to the most balanced overload in all the test cases, while the distributed version sacrifices little optimality. Both policies work much better than pure backpressure proposed in [9] for unbounded-buffer systems.

II. Model and Problem Formulation

A. Queue Dynamic Model

In this section, we introduce the ODE model for queue dynamics in single-hop network structure. We use a bipartite graph $(\mathcal{V}, \mathcal{E})$ to model the network, where $\mathcal{V} := \{\mathcal{V}_I, \mathcal{V}_E\}$ denotes the node set with $\mathcal{V}_I$ the set of ingress nodes, $\mathcal{V}_E$ the set of egress nodes, $\mathcal{E}$ the set of transmission links between $\mathcal{V}_I$ and $\mathcal{V}_E$, and $|\mathcal{V}_I| = N$ and $|\mathcal{V}_E| = M$. We term it as an $N \times M$ single-hop network. Denote the ith ingress node as s_i and the jth egress node as d_j. Each node k has a buffer that stores the packets, whose size is denoted by b_k. The packets in each node k form a queue, whose length at time t is denoted by $q_k(t)$. Therefore $q_k(t) \in [0, b_k], \forall k, \forall t$, which means that queue length cannot surpass the buffer size. We do not allow packet transmission to a saturated node. This is desirable in practice since it prevents packet dropping and significantly reduces the retransmission delay due to buffer overflow [17], [18], and can be implemented simply with a detection signal of the saturation level of downstream buffers. Packets will be backlogged until there exists spare buffer storage downstream.

Each packet arrives to one of the ingress nodes and departs from one of the egress nodes. The packet arrival rate at ingress node s_i, denoted by λ_i, represents the average number of packets that are injected into s_i in a time unit. We use $\boldsymbol{\lambda} := \{\lambda_i\}_{i=1}^N$ to denote the packet arrival rate vector. Packets in the buffer of s_i are transmitted to an adjacent egress node d_j through link $(s_i, d_j) \in \mathcal{E}$. The transmission rate on link (s_i, d_j) at time t, denoted by $g_{s_i d_j}(t)$, represents the number of packets transmitted over (s_i, d_j) in a time unit. Each link (s_i, d_j) is associated with a capacity value $c_{s_i d_j}$, which represents its maximum transmission rate. Specifically, $0 \leq g_{s_i d_j}(t) \leq c_{s_i d_j},\ \forall (s_i, d_j) \in \mathcal{E}$. We use $\mathbf{c} := \{c_{s_i d_j}\}_{(s_i,d_j)\in\mathcal{E}}$ to denote the capacity vector. Finally, packets in an egress node d_j depart from the networks with departure rate denoted as $g_{d_j}(t)$, and the service rate of node d_j, which is the maximum departure rate for packets in d_j, is denoted by μ_j. Thus $g_{d_j}(t) \in [0, \mu_j],\ \forall j = 1, \ldots, M$. Let $\boldsymbol{\mu} := \{\mu_j\}_{j=1}^M$.

Based on the above setting, we now formulate the queue dynamics according to the flow conservation law, which states that the net increase of queue length equals the difference between the number of new arrivals and departures at a node at any time. Specifically, for any ingress node s_i,

$$\dot{q}_{s_i}(t) = \lambda_i - \sum_{d_j:(s_i,d_j)\in\mathcal{E}} g_{s_i d_j}(t) \tag{1}$$

and for any egress node d_j,

$$\dot{q}_{d_j}(t) = \sum_{s_i:(s_i,d_j)\in\mathcal{E}} g_{s_i d_j}(t) - g_{d_j}(t). \tag{2}$$

$\dot{q}_k(t)$ denotes the *queue overload rate* of node k at time t, and we assume that $\dot{q}_k := \lim_{t\to\infty} \dot{q}_k(t)$ exists where $\dot{q}_k$ denotes node k's queue overload rate in steady state[2]. All the nodes thus have nonnegative queue overload rates in the steady state as the queue length is bounded below by 0. Furthermore, the queue length in nodes with bounded buffers will not grow with a positive rate in the steady state, therefore $\dot{q}_i = 0$ for any $i \in \mathcal{V}$ with bounded buffer.

In this paper, we assume that internal (egress) buffers are bounded while ingress buffers are large enough to avoid saturation. In practice, internal nodes often have limited buffers [14], [20]. For example, on-chip networks have very small internal buffers. Similarly, satellite networks have small buffers on-board the satellite. In contrast, ingress buffers have

[2]The existence holds under most of the policies of interest [3], [19].

sufficient capacity to absorb bursty packet arrivals, e.g. in a satellite network the buffer at the ground terminal can be relatively large.

In reality, even ingress buffers have limited size, and packet loss will be inevitable when the packet arrival rate to an ingress node s_i is larger than the sum of the capacities of its downstream links. We can deal with bounded ingress buffers by introducing a virtual queue for such s_i with unbounded buffer, whose length is q_{s_i} plus the number of dropped packets that are to be retransmitted, and thus the actual overload rate at s_i can be exactly characterized by the overload rate of a virtual queue with unbounded buffer. Therefore, we can assume without loss of generality that ingress buffers are unbounded.

We define $\dot{\mathbf{q}} := \{\dot{\mathbf{q}}_s, \dot{\mathbf{q}}_d\} \in \mathbb{R}^{N+M}$ as the *queue overload rate vector*, where $\dot{\mathbf{q}}_s = \{\dot{q}_{s_i}\}_{i=1}^N \in \mathbb{R}^N$ and $\dot{\mathbf{q}}_d = \{\dot{q}_{d_j}\}_{j=1}^M \in \mathbb{R}^M$ are the ingress and egress queue overload rate vector respectively[3]. Similarly, we define the *transmission rate vector* of the system as $\mathbf{g} := \left\{\{g_{s_i d_j}\}_{(s_i,d_j)\in\mathcal{E}}, \{g_{d_j}\}_{j=1}^M\right\}$. We further define the feasible flow region $\mathcal{G}$ as the set of transmission rate vectors $\mathbf{g}$ that satisfy the flow conservation laws (1) and (2) and capacity constraints. We then define the feasible queue overload rate region $\mathcal{R}$ as the set of queue overload rate vectors $\dot{\mathbf{q}}$ that can be achieved under some element in $\mathcal{G}$.

Remark: In (1) and (2), the queue length can be fractional. This is a fluid approximation to the real case where packets are discrete, which offers a simplified framework for studying flow control [9]. This fluid approximation is different from the fluid model defined in some prior works which captures the scaled limit of the queue backlog [6], [21], [22], an indicator for queue stability but not suited to study finite buffers and queue overload dynamics.

B. Problem Formulation: Overload Balancing

In this section we define the problem of overload balancing. The queue overload rate vector $\dot{\mathbf{q}}$ indicates the severity of queue overload. We need a metric to evaluate how balanced $\dot{\mathbf{q}}$ is. Multiple metrics related to network fairness have been investigated. The very first concept is *min-max fairness* which aims to identify a transmission rate vector such that any other vector that decreases the overload rate at some nodes must be at the expense of increasing the overload rate of some other nodes with a higher overload rate [11]. This concept stems from *max-min fairness* [23] which maximizes the minimum commodity flow to be transmitted, while in overload balancing the direction is reversed as reducing the max queue overload is desired.

Moreover, the min-max fairness solution is closely related to the lexicographic minimum solution [9] defined as follows.

Definition 1. *The queue overload vector $\dot{\mathbf{q}}^*$ is the lexicographic minimum in the feasible overload rate region $\mathcal{R}$ if and only if for $\forall \dot{\mathbf{q}} \in \mathcal{R}$ that $\dot{\mathbf{q}} \neq \dot{\mathbf{q}}^*$, $\sum_{i=1}^k \dot{q}^*_{(i)} \leq \sum_{i=1}^k \dot{q}_{(i)},\ \forall k$, where $\dot{q}^*_{(i)},\ \dot{q}_{(i)}$ denote the ith maximal element of $\dot{\mathbf{q}}^*,\ \dot{\mathbf{q}}$ respectively.*

[3]We neglect time t in the notations for brevity. We will clarify explicitly when notations without t represents steady state value to avoid ambiguity.

The lexicographic minimum $\dot{\mathbf{q}}^*$ represents the most balanced overload vector since it guarantees that the top-k most severely overloaded nodes have been balanced under the metric of the sum of queue overload rates for every k. Therefore the overload balancing problem can be formally stated as: *determine the transmission rate vector* $\mathbf{g}$ *so that the resulting overload rate vector is the lexicographic minimum.*

Nevertheless, the lexicographical minimum is hard to formulate as it involves the ordering of a vector with cumulative sum comparisons. To overcome the challenge, we prove that minimizing the quadratic sum of queue overload rates serves as an equivalent criterion to lexicographic minimum under network flow constraints, which facilitates analysis of overload balancing, as shown in following sections.

III. Quadratic Sum Minimization Leads to Lexicographic Minimum

In this section, we prove that identifying $\mathbf{g} \in \mathcal{G}$ to minimize the quadratic sum of queue overload rates leads to lexicographic minimum overload rates. This result is derived under the prior information of $(\boldsymbol{\lambda}, \mathbf{c}, \boldsymbol{\mu})$, and it can capture the most balanced solution at any time shot when $(\boldsymbol{\lambda}, \mathbf{c}, \boldsymbol{\mu})$ is obtained. Analysis of policies without such prior information in following sections is based on it. This result can be directly proved by Cauchy-Schwarz inequality if there are no constraints on $\mathbf{g}$, however there is no general result (and it generally does not hold) when $\mathbf{g}$ is constrained. We, for the first time, prove the result under general single-hop network with bounded buffers. We first introduce an intermediate result that the minimizer of the quadratic sum minimizes the maximum overload rate, and then show the main result.

A. Quadratic Sum Minimization to Maximum Overload Rate Minimization

The quadratic sum minimization framework of overload balancing under an $N \times M$ single-hop network can be formulated as

$$\begin{aligned} \min_{\mathbf{g}} \quad & \frac{1}{2}\sum_{i=1}^N \left(\lambda_i - \sum_{j=1}^M g_{s_i d_j}\right)^2 + \frac{1}{2}\sum_{j=1}^M \left(\sum_{i=1}^N g_{s_i d_j} - g_{d_j}\right)^2 \\ \text{s.t.} \quad & \sum_{i=1}^N g_{s_i d_j} = g_{d_j},\ \forall d_j \in \mathcal{B} \\ & 0 \leq g_{s_i d_j} \leq c_{s_i d_j},\ \forall (s_i, d_j) \in \mathcal{E} \\ & 0 \leq g_{d_j} \leq \mu_j,\ \forall j = 1, \ldots, M \end{aligned} \tag{3}$$

where the objective represents the quadratic sum of queue overload rates at all ingress and egress nodes according to (1) and (2), the variables $\mathbf{g}$ can represent the transmission rate vector at any specific time shot t, with $\mathcal{B}$ denoting the nodes whose buffer has been saturated at this time shot. The constraint $\sum_{i=1}^N g_{s_i d_j} = g_{d_j}$ means that $\dot{q}_{d_j} = 0$ for an egress node d_j saturated at this time shot. It is trivial to verify that the optimum can never be achieved when $\exists i \notin \mathcal{B}, \dot{q}_i < 0$, since the objective function is a quadratic sum of $\dot{\mathbf{q}}$. This property

enables using (3) to study the steady state ($t \to \infty$), since we require $\dot{q}_i \geq 0,\ \forall i$ in steady state mentioned in Section II-A.

The minimization of maximum queue overload rate corresponds to the objective function $\min_{\mathbf{g}\in\mathcal{G}} \max_{i\in\mathcal{V}} \dot{q}_i$. This can be equivalently formulated as a linear programming problem

$$\begin{aligned} \min_{\mathbf{g},v} \quad & v \\ \text{s.t.} \quad & \sum_{i=1}^{N} g_{s_i d_j} = g_{d_j},\ \forall d_j \in \mathcal{B} \\ & 0 \leq g_{s_i d_j} \leq c_{s_i d_j},\ \forall (s_i, d_j) \in \mathcal{E} \\ & 0 \leq g_{d_j} \leq \mu_j,\ \forall j = 1, \ldots, M \\ & \lambda_i - \sum_{j=1}^{M} g_{s_i d_j} \leq v,\ \forall s_i \in \mathcal{V}_I \\ & \sum_{i=1}^{N} g_{s_i d_j} - g_{d_j} \leq v,\ \forall d_j \in \mathcal{V}_E \end{aligned} \tag{4}$$

by introducing an auxiliary variable v and additional constraints $\dot{q}_i \leq v,\ \forall i \in \mathcal{V}$.

Now we state the intermediate result as Lemma 1.

Lemma 1. *Suppose that $\mathbf{g}^* \in \mathcal{G}$ is optimal in* (3)*, then $\mathbf{g}^*$ is optimal in* (4)*.*

The main proof idea is to take advantage of the Karush-Kuhn-Tucker (KKT) conditions of (3) and (4). Details are deferred to the appendix. We present a geometric interpretation of Lemma 1 in Fig. 3 through a 2×1 single-hop networks. Lemma 1 states that the the minimizer of (3) always overlaps with a minimizer of (4) in the feasible flow region $\mathcal{G}$ under the constraints.

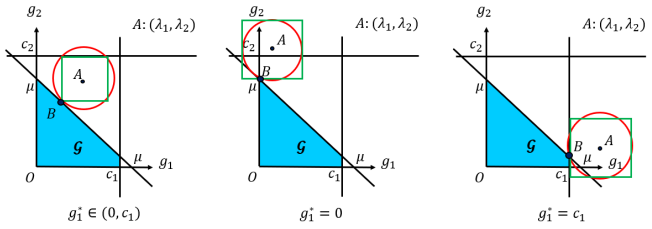

Fig. 3: Geometric interpretation of Lemma 1 through a 2×1 single-hop network. The contour curves of (3) in red and (4) in green coincide at the same optimal point B on the boundary of $\mathcal{G}$ under different arrival rate vectors $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$, denoted as point A where $\lambda_1 + \lambda_2 > \mu$.

B. Quadratic Sum Minimization to Lexicographic Minimum

We now demonstrate the main result in Theorem 1. The idea is that by Lemma 1, $\mathbf{g}^*$ minimizes the maximum queue overload rate, then we can show it must minimize the second maximum queue overload among all $\mathbf{g} \in \mathcal{G}$ that minimizes the maximum queue overload, otherwise it violates Lemma 1. Then iteratively we can obtain lexicographical minimum.

Theorem 1. *Suppose that $\mathbf{g}^* \in \mathcal{G}$ is optimal in* (3)*, then it is lexicographic minimum in $\mathcal{G}$.*

Proof. (sketch) Note that $\mathbf{g}^*$, the minimizer of quadratic sum $\sum_{i=1}^{n} \dot{q}_i^2$ in (3) is the minimizer of max growth rate $\dot{q}_{(1)}$ in (4). Denote the minimum $\dot{q}_{(1)}$ as $\dot{q}_{(1)}^{\triangle}$, then the set $\mathcal{G}_1 = \{\mathbf{g} \in \mathcal{G} \mid \dot{q}_{(1)}^{(\mathbf{g})} = \dot{q}_{(1)}^{\triangle}\}$ contains $\mathbf{g}^*$, where $\dot{q}_{(1)}^{(\mathbf{g})}$ denotes the maximum queue overload rate under the transmission rate vector $\mathbf{g}$. Now we consider (3) and (4) with additional constraint that $\mathbf{g} \in \mathcal{G}_1$. Obviously $\mathcal{G}_1$ is a convex set, thus (4) with $\mathbf{g} \in \mathcal{G}_1$ is still convex. Meanwhile, with additional constraint that $\mathbf{g} \in \mathcal{G}_1$, (3) keeps in the form of a quadratic optimization problem. Therefore we can apply Lemma 1 to (3) and (4) in $\mathcal{G}_1$ similarly to obtain that the $\mathbf{g}^*$ minimizes $\dot{q}_{(2)}$, the second largest queue overload rate. Denote the minimum $\dot{q}_{(2)}$ as $\dot{q}_{(2)}^{\triangle}$, thus $\mathbf{g}^* \in \mathcal{G}_2 := \{\mathbf{g} \in \mathcal{G} \mid \dot{q}_{(1)}^{(\mathbf{g})} = \dot{q}_{(1)}^{\triangle}, \dot{q}_{(2)}^{(\mathbf{g})} = \dot{q}_{(2)}^{\triangle}\}$. Iteratively, $\mathbf{g}^* \in \mathcal{G}_{N+M}$ where any element in $\mathcal{G}_{N+M}$ induces the lexicographic minimum queue overload rate vector. □

IV. Maxweight + Backpressure Leads to Most Balanced Overload

Section III demonstrates that solving (3) can achieve most balanced overload. However, it requires the complete knowledge of network parameters $(\boldsymbol{\lambda}, \mathbf{c}, \boldsymbol{\mu})$, which in real networks may not be available [19]. In practice, the queue backlog $\mathbf{q}(t)$ is often accessible in real-time, thus we consider if there exists any *queue-based* transmission policy, which determines the transmission rate vector $\mathbf{g}(t)$ based on $\mathbf{q}(t)$, that can achieve most balanced overload as (3) does.

The ODE dynamical system (1) and (2) under queue-based policy form an autonomous system

$$\begin{cases} \dot{q}_{s_i}(t) = \lambda_i - \sum_{d_j:(s_i,d_j)\in\mathcal{E}} g_{s_i d_j}(\mathbf{q}(t)),\ \forall i = 1, \ldots, N \\ \dot{q}_{d_j}(t) = \sum_{s_i:(s_i,d_j)\in\mathcal{E}} g_{s_i d_j}(\mathbf{q}(t)) - g_{d_j}(\mathbf{q}(t)),\ \forall j \in 1, \ldots, M \end{cases} \tag{5}$$

Due to the absence of prior knowledge of $(\boldsymbol{\lambda}, \mathbf{c}, \boldsymbol{\mu})$, we can no longer achieve most balanced overload in one stroke by optimizing (3). Instead, we aim to propose queue-based policies that can render the queue dynamics (5) to converge to the most balanced overload state. Formally, the problem of overload balancing under queue-based policy can be formulated as: *Is there $\mathbf{g}(\mathbf{q})$ that guarantees $\lim_{t\to\infty} \dot{\mathbf{q}}(t) = \dot{\mathbf{q}}^*$ in* (5)*, where $\dot{\mathbf{q}}^*$ is the overload rate vector induced by $\mathbf{g}^*$, the optimal solution to* (3) *with the oracle $(\boldsymbol{\lambda}, \mathbf{c}, \boldsymbol{\mu})$?* We prove that a *maxweight + backpressure (mw+bp)* queue-based policy yields a solution under $N \times M$ single-hop structure, and propose a distributed version of this policy that reduces communication overhead from $O(N)$ to $O(1)$, with performance close to optimum shown in Section V.

A. Methodology

Our methodology to prove that a queue-based policy $\mathbf{g}(\mathbf{q})$ achieves the most balanced overload is to verify that it satisfies two conditions which together, as illustrated in Fig. 4, is a sufficient condition. For the first condition, we establish the **existence** of a queue vector $\mathbf{q}$ such that the transmission rate vector under the policy at $\mathbf{q}$ is an optimizer of (3) which leads to most balanced queue overload. Specifically, we consider the

following optimization framework in which the only difference with (3) is that the queue vector $\mathbf{q}$ is the decision variable.

$$\begin{aligned}\min_{\mathbf{q}} \quad & \frac{1}{2}\sum_{i=1}^{N}\left(\lambda_i - \sum_{j=1}^{M} g_{s_i d_j}(\mathbf{q})\right)^2 + \frac{1}{2}\sum_{j=1}^{M}\left(\sum_{i=1}^{N} g_{s_i d_j}(\mathbf{q}) - g_{d_j}(\mathbf{q})\right)^2 \\ \text{s.t.} \quad & \sum_{i=1}^{N} g_{s_i d_j}(\mathbf{q}) = g_{d_j}(\mathbf{q}),\ \forall d_j \in \mathcal{B} \\ & 0 \le g_{s_i d_j}(\mathbf{q}) \le c_{s_i d_j},\ \forall (s_i, d_j) \in \mathcal{E} \\ & 0 \le g_{d_j}(\mathbf{q}) \le \mu_j,\ \forall j = 1, \ldots, M \end{aligned} \tag{6}$$

Denote an optimizer of (6) as $\mathbf{q}^*$ and the set of all optimizers of (6) as $\mathcal{Q}^*$. The first condition is to verify that the policy $\mathbf{g}(\mathbf{q})$ satisfies $\forall \mathbf{q}^* \in \mathcal{Q}^*$, $\mathbf{g}(\mathbf{q}^*)$ equals some $\mathbf{g}^* \in \mathcal{G}^*$ where $\mathcal{G}^*$ denotes the set of optimizers of (3). For the second condition, we verify the **convergence** of the ODE dynamics (5) to the most balanced state under the policy $\mathbf{g}(\mathbf{q})$ given any initial queue vector. Namely, as $t \to \infty$, the second condition is to verify that the policy drives the queue vector to $\mathcal{Q}^*$. If the policy $\mathbf{g}(\mathbf{q})$ satisfies both conditions, it can achieve most balanced overload in the steady state.

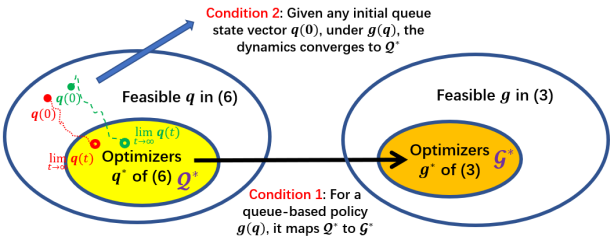

Fig. 4: Condition 1 (existence) and 2 (convergence) to verify that a queue-based policy achieves most balanced overload

B. Maxweight + Backpressure Policy in Differentiable Form

The ODE-based methodology in Section IV-A requires the queue-based policy $\mathbf{g}(\mathbf{q})$ in a differentiable form. We now define the mw+bp policy accordingly. The policy contains two parts: maxweight scheduling and a backpressure mechanism. The idea of maxweight scheduling is to serve input nodes that have longer queue backlogs with higher priority [22]. The backpressure mechanism determines to transmit packets over a link (s_i, d_j) at t with rate $c_{s_i d_j}$ if $q_{s_i}(t) > q_{d_j}(t)$, and does not transmit otherwise [9]. To avoid buffer overflow, backpressure also ensures that packets are not served to a saturated node.

Both maxweight and backpressure were proposed in discrete forms originally. To embed them in the ODE framework, we propose the following differentiable characterization which can well approximate their original version.

Maxweight Scheduling: The maxweight scheduling in differentiable form is defined as

$$g_{s_i d_j}(\mathbf{q}) = c_{s_i d_j} \frac{e^{\gamma q_{s_i}}}{\sum_{k=1}^{N} e^{\gamma q_{s_k}}},\ \forall (s_i, d_j) \in \mathcal{E} \tag{7}$$

where $\gamma > 0$ is a parameter. The larger γ is, the more we favor to serve ingress nodes with longer queue length. An extreme case is $\gamma \to \infty$, which matches to the serve-the-longest-queue policy [24]: only the ingress node with longest queue length will be served, and if there are K ingress nodes that have the same longest queue length, then (7) guarantees that each of these K nodes, say node s_i, will be served with rate $c_{s_i d}/K$. This corresponds to the result under serve-the-longest-queue policy in expectation, in which one of these K nodes is chosen uniformly at random to be served.

Backpressure Mechanism:

$$g_{s_i d_j}(\mathbf{q}) = c_{s_i d_j} \alpha_{s_i d_j} \beta_{d_j},\ \forall (s_i, d_j) \in \mathcal{E} \tag{8}$$

where

$$\alpha_{s_i d_j} = \frac{1}{1 + e^{-a(q_{s_i} - q_{d_j})}}, \quad \beta_{d_j} := \frac{1}{1 + e^{-a(b_{d_j} - q_{d_j} - \epsilon)}}$$

and $a > 0$ and $\epsilon > 0$ are preset values. Note that if $a \to \infty$ and ϵ is close enough to 0, then the term $\alpha_{s_i d_j} = 1$ if $q_{s_i} > q_{d_j}$ and $\alpha_{s_i d_j} = 0$ if $q_{s_i} < q_{d_j}$; the term $\beta_{d_j} \to 1$ if $q_{d_j} < b_{d_j}$ and $\beta_{d_j} \to 0$ if $q_{d_j} \to b_{d_j}$. Therefore the policy (8) transmits the packets from an ingress node s_i to an egress node d_j with maximum service rate $c_{s_i d_j}$ if and only if the queue length in s_i is greater than in d_j, and meanwhile the buffer of node d is not saturated, which shows that (8) is an approximation to backpressure under sufficiently large a and small ϵ.

Maxweight + Backpressure (mw+bp): Combining (7) and (8), we can formulate the mw+bp policy as

$$g_{s_i d_j}(\mathbf{q}) = c_{s_i d_j} \alpha_{s_i d_j} \beta_{d_j} \frac{e^{\gamma q_{s_i}}}{\sum_{k=1}^{N} e^{\gamma q_{s_k}}},\ \forall (s_i, d_j) \in \mathcal{E} \tag{9}$$

In (9), a transmission link is activated if and only if its corresponding ingress queue length satisfies the link activation requirements under both maxweight and backpressure.

In addition, egress nodes operate in a work-conserving manner, where each egress node d_j serves packets with rate μ_j whenever the buffer is nonempty. This guarantees that the egress nodes reduce the queue overload at the maximum rates. This work-conserving policy can also be formulated into a differentiable form as

$$g_{d_j}(\mathbf{q}) = \mu_j \frac{1}{1 + e^{-a(q_{d_j} - \epsilon)}}, \forall j \in 1, \ldots, M \tag{10}$$

under sufficiently large a and $\epsilon \to 0$.

C. MW+BP in Single-hop Networks with Sufficient Capacity

In this part, we prove that the mw+bp policy (9) can achieve most balanced overload as by optimizing (3) if every transmission link has sufficient capacity, and all egress nodes run (10), stated in Theorem 2.

Theorem 2. *The queue dynamics under* (9) *and* (10) *converges to the most balanced overloading if* $c_{s_i d_j} > \mu_j,\ \forall (s_i, d_j) \in \mathcal{E},\ j = 1, \ldots, M.$

The proof idea is to verify the existence and convergence of most balanced state. More details are deferred to appendix. Implementing (10) clearly makes for overload mitigation as all egress nodes do their best to send packets out. The intuition why (9) achieves most balanced overload is three-fold: (i)

The maxweight (7) balances the ingress nodes as it favors serving queues with longer length; (ii) The backpressure (8) balances any connected ingress s_i and egress d_j as it sets the threshold for transmission decision at $q_{s_i} = q_{d_j}$; (iii) The condition $c_{s_i d_j} > \mu_j,\ \forall (s_i, d_j) \in \mathcal{E}$ guarantees that mw+bp can achieve maximum throughput since it makes any egress node d_j never be idle. The sufficient link capacity generally holds in data center networks and server farms which have sufficient transmission resources to guarantee high quality-of-service requirements [3].

D. MW+BP in Single-hop Networks with Limited Capacity

Next we consider limited capacity where $c_{s_i d_j} > \mu_j,\ \forall (s_i, d_j) \in \mathcal{E}$ does not hold. In this case, mw+bp policy (9) may not achieve most balanced overload since the maximum throughput may not be achieved, compared with sufficient capacity case in Section IV-C. We show an example in Fig. 5. Our solution is to consider a generalized version of mw+bp, where we run (9) and in the meantime additionally serve the ingress nodes with longest queues in order until maximum throughput is achieved. A special case is to consider $\gamma \to \infty$ in (9) so that the whole mechanism is exactly an extended version of serving-the-longest-queue policy.

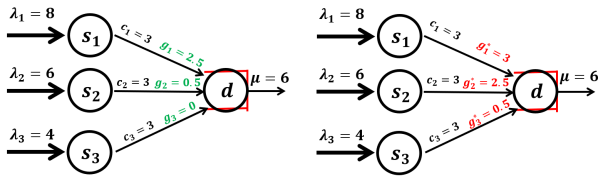

Fig. 5: Example that (9) does not achieve most balanced overload with limited capacity with $\gamma \to \infty$, where node d has bounded buffer. On the left, $(g_1, g_2, g_3) = (2.5, 0.5, 0)$ denotes the transmission rates under (9) in steady state, under which $\dot{\mathbf{q}} = [5.5, 5.5, 4]$. It is not the most balanced overload as presented on the right, where $\dot{\mathbf{q}}^* = [5, 3.5, 3.5]$ under $(g_1^*, g_2^*, g_3^*) = (3, 2.5, 0.5)$, the optimal solution to (3). Other γ values also suffer from the suboptimality.

We summarize our solution in Algorithm 1. Algorithm 1 is a real-time algorithm that determines the service rates on every link given the current queue information $\mathbf{q}(t)$. We run (9) and calculate the remaining capacity on each link. Then we sort the ingress nodes in non-increasing order, and further follow the order to inject packets to egress nodes whose packet injection rate is lower than its service rate. Algorithm 1 presents our solution in a quantified way, which uses the information of $\mathbf{c}$ and $\boldsymbol{\mu}$ that may not be available. However, in practical implementation we do not require them: (i) We can sense if a link has been served with full capacity. If so, we do not inject more packets through this link. This replaces the need of calculation in line 3 and 8; (ii) Each egress node d_j can send the information of whether the queue length is increasing to ingress nodes through broadcasting or a controller, serving as an alternative indicator of $r_{d_j} < \mu_j$ in line 7 and replaces the need of calculation in line 4 and 9.

We can show that the generalized maxweight scheduling leads to most balanced overloading.

Algorithm 1: Generalized maxweight + backpressure with limited capacity

1 **Input:** current queue vector $\mathbf{q} := \mathbf{q}(t)$;
2 Run (9) and (10), and obtain $\mathbf{g}(\mathbf{q})$;
3 For all $(s_i, d_j) \in \mathcal{E}$, calculate the remaining capacity $\tilde{c}_{s_i d_j} := c_{s_i d_j} - g_{s_i d_j}(\mathbf{q})$;
4 Calculate the packet injection rate to all $d_j \in \mathcal{V}_E$ as $r_{d_j} := \sum_{s_i:(s_i,d_j)\in\mathcal{E}} g_{s_i d_j}(\mathbf{q})$;
5 Sort the queue length of ingress nodes in non-increasing order $q_{s_{(1)}} \geq q_{s_{(2)}} \geq \cdots \geq q_{s_{(N)}}$, where $\{s_{(i)}\}_{i=1}^N$ is a permutation of $\{s_i\}_{i=1}^N$;
6 **for** $i = 1, \ldots, N$ **do**
7 **for** *all* d_j *that* $r_{d_j} < \mu_j$ **do**
8 $g_{s_{(i)} d_j}(\mathbf{q}) \leftarrow g_{s_{(i)} d_j}(\mathbf{q}) + \min\{\tilde{c}_{s_{(i)} d_j}, \mu_j - r_{d_j}\}$;
9 $r_{d_j} \leftarrow r_{d_j} + \min\{\tilde{c}_{s_{(i)} d_j}, \mu_j - r_{d_j}\}$;
10 **Return** $\mathbf{g}(\mathbf{q})$ as the transmission policy;

Theorem 3. *The generalized mw+bp policy in Algorithm 1 achieves most balanced overloading.*

The proof idea is similar to the proof of Theorem 2 by expressing the generalized mw+bp policy into a differentiable form, and then verify the conditions 1 and 2 in Section IV-A. The intuition is that Algorithm 1 achieves maximum throughput, and is a combination of mw+bp (9) and the serving-the-longest-queue, both of which achieve most balanced overload once maximum throughput can be achieved. Due to space limitation, we omit the proof.

E. Distributed MW + BP

In mw+bp policy (9), collecting real-time queue information of all ingress nodes is required at each ingress node or through a centralized controller. This induces large communication overhead in large-scale networks. We consider a distributed version of (9) to reduce overhead. The idea is that each ingress node gets access to another r ingress nodes, and run (9) where the maxweight part only depends on the queue length of itself and these r ingress nodes. One extreme case is that $r = 1$, where each ingress node s_i has the information of s_{i+1} (s_N has the information of s_1). The ingress node s_i serves packets to egress nodes only if $q_{s_i} > q_{s_{i+1}}$ (for s_N the condition is $q_{s_N} > q_{s_1}$), thus no need of sharing queue information of ingress nodes other than s_{i+1} to s_i, which reduces the communication overhead at the ingress side from $N-1$ to 1 for each ingress node. The intuition this distributed mechanism works for overload balancing is that balancing the pairs (s_{i-1}, s_i) and (s_i, s_{i+1}) together indirectly balances the pair (s_{i-1}, s_{i+1}).

The above distributed mw + bp policy at any ingress s_i can be formalized into a differentiable form as

$$g_{s_i d_j}(\mathbf{q}) = c_{s_i d_j} \alpha_{s_i d_j} \beta_{d_j} \frac{e^{\gamma q_{s_i}}}{e^{\gamma q_{s_i}} + e^{\gamma q_{s_{i+1}}}} \tag{11}$$

where $\alpha_{s_i d_j}$ and β_{d_j} represents the backpressure terms as in (8). We show in Section V that this distributed variant, even

under $r = 1$, is close to the optimum achieved by mw+bp (9) in a large portion of test cases, which serves as a promising alternative of (9) to reduce communication overhead with low performance sacrifice in practical implementation.

V. Performance Evaluation

In this section, we verify our proposed policies and theories through experiments over (i) single-hop network: server farm, packet switch, etc.; (ii) tree-structured datacenter network, for example Clos structure [16]. Clos concatenates multiple stages of single-hop structures. Although not proved analytically, verification results over Clos structure below demonstrate the extendability of our proposed policies to multi-hop networks.

We evaluate three policies: (i) Pure backpressure (8) [9]; (ii) Centralized Maxweight + Backpressure ((9) and Algorithm 1); (iii) Distributed Maxweight + Backpressure under $r = 1$ (11). We evaluate their performance in overload balancing through measuring the gap between $\dot{\mathbf{q}}^*$, the optimal solution to (3) achieved with prior knowledge of $(\boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{c})$ in steady state, and $\dot{\mathbf{q}}^\pi$, the steady state queue overload rate vector under a particular queue-based policy π. Closer gap represents superior overload balancing performance. Specifically, we consider two gap ratio metrics: (i) Quadratic sum gap ratio: $||\dot{\mathbf{q}}^\pi||^2/||\dot{\mathbf{q}}^*||^2$ which is exactly the metric we postulate; (ii) Max overload rate gap ratio: $\max_{i\in\mathcal{V}} \dot{q}_i^\pi \,/\, \max_{i\in\mathcal{V}} \dot{q}_i^*$ which reflects particularly the balancing of the most severe overload. For both metrics, the closer to 1, the better π is. The first metric reflects more on overall balancing while the second fits into cases where maximum overload is more important.

To demonstrate the universality of our proposed policy, we evaluate it using (i) different $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, which represents different overload levels; (ii) different $\mathbf{c}$, which represents different service capacity; (iii) different buffer values $\mathbf{b}$, which represents different buffer settings, including the spatial distribution of sufficient and limited buffers. We consider multiple networks instances with randomly sampled values of the above parameters, and measure the empirical cumulative distribution function (CDF) of the two gap metrics.

As introduced at length below, we see that maxweight + backpressure achieves most balanced overload rates in steady state, far better than pure backpressure in both metrics, while the performance of distributed maxweight + backpressure approaches that of maxweight + backpressure.

A. Single-hop Networks

We evaluate on a 64×32 single-hop network with full connection between ingress and egress nodes, modeling real switched networks [16], [20]. We consider 200 randomly selected parameter settings: (i) The arrival rate to each ingress port is uniformly distributed in $[0, 4]$; (ii) The service rates of each egress port is uniformly distributed within $[0, 6]$; (iii) The capacity $c_{s_i d_j}$ for each ingress-egress pair (s_i, d_j) is uniformly distributed in $[0, 10]$; (iv) The buffer size for any ingress node is 10,000 so that it is never saturated during the simulation, and the buffer size for any egress node is 10,000 with probability 0.2 and uniformly distributed within $[30, 80]$ with probability 0.8. The initial queue length in each node is set to be uniformly distributed within $[0, \min(30, \text{buffer size})]$, so that no queues are overflow initially. The rationales behind the settings are: (i) and (ii) guarantee that the system is overloaded with high probability, as the expected sum of arrival rates is $2 \times 64 = 128$, 133% of the expected sum of egress service rates $3 \times 32 = 96$; (iii) considers both sufficient and limited capacity values; (iv) realizes different buffer settings, which follows the real case that buffers at ingress are large [17] while egress ports may have limited buffers [14], [20] generally.

We plot the CDFs of the quadratic sum gap ratio and max overload rate gap ratio of all 200 sampled settings in steady state in Fig. 6 and Fig. 7, where the x-axis is in logarithmic scale to make details clearer. Our proposed mw+bp (9) achieves quadratic sum gap close to 1 for nearly all test instances, and achieves max overload rate gap close to 1 for more than 70% instances. The optimality shown in the results is not 100% as in theory due to limited time span. The value difference between these two gaps is due to the balancing effect of quadratic sum[4]. The distributed version of mw + bp (11) loses some accuracy while still performs generally well, as in more than 85% instances, the quadratic sum overlaod gap is less than 1.15 and the max overload rate gap is less than 1.4, as pointed out in the figures. Comparatively, the backpressure policy incurs large gaps in the steady state and achieves the optimum in none of the cases. Typically, with more than 75% of instances the quadratic sum gap exceeds 1.4 and the max overload rate gap exceeds 1.5.

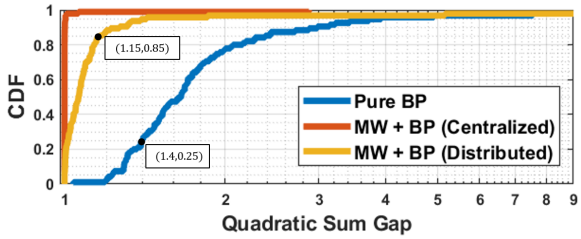

Fig. 6: Quadratic Sum Gap Comparison in Steady State

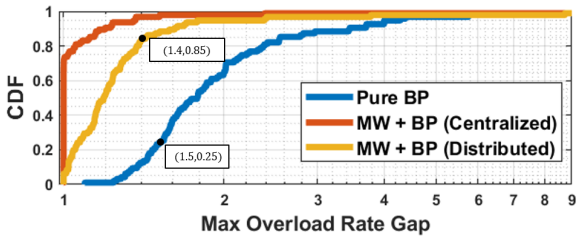

Fig. 7: Max Overload Rate Gap Comparison in Steady State

We further present the transient process of the quadratic sum gap ratio in Fig. 8 under the ODE. Note that negative queue

[4]Consider, for example, $\dot{\mathbf{q}}^* = [0.5, 0.5]$ and $\dot{\mathbf{q}}^\pi = [0.6, 0.4]$, then a gap of $0.6/0.5 = 1.2$ in the max overload rate only leads to a gap of $(0.6^2 + 0.4^2)/(0.5^2 + 0.5^2) = 1.04$ in quadratic sum, where the gap is smaller.

overload rate may exist in transient states, which is desirable for overload mitigation at a node. Therefore the quadratic sum gap ratio at any time in Fig. 8 only considers the sum of positive queue overload rates. At the beginning, the gap ratio is high for all three schemes, because the queue is far from the equilibrium point. For example, in pure backpressure, when all connected nodes satisfy backpressure constraints, then all intermediate links will be activated, thus all ingress nodes have negative overload rates while egress nodes have large positive rates due to the arriving packets from all upstream links. Approaching the steady state, shown in the zoomed-in subfigure, the gap of mw + bp, both centralized and distributed, converges close to 1, while the gap of pure backpressure converges a value greater than 1, not optimal solution for overload balancing.

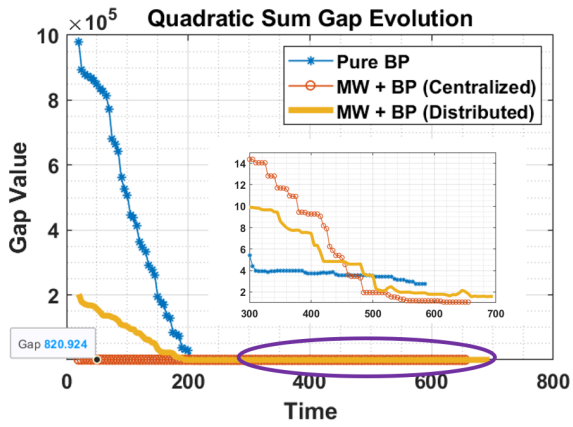

Fig. 8: Transient Process of Quadratic Sum Gap

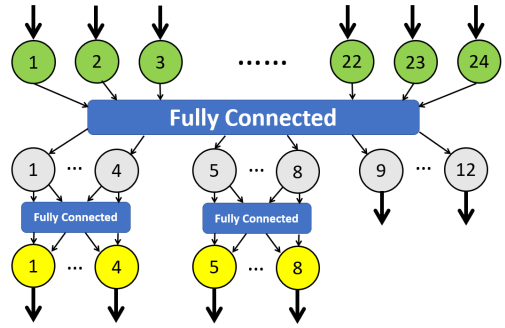

Fig. 9: Example of a 3-stage Clos structure from [16]

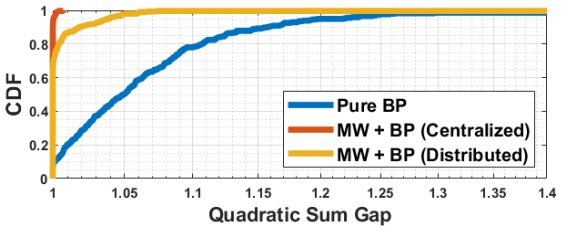

Fig. 10: Quadratic Sum Gap Comparison (Clos)

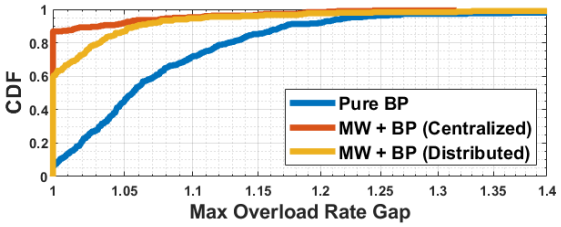

Fig. 11: Max Overload Rate Gap Comparison (Clos)

B. Clos Network Structure

We further test on a 3-stage Clos structure abstracted from Google's work on their Jupiter datacenter [16], shown in Fig. 9. It contains 24 ingress blocks at the top, fully connected to 12 aggregation blocks at the middle, and the aggregation blocks 1 to 4 are connected to 4 egress blocks, as are aggregation blocks 5 to 8. Packets depart from the network from the 8 egress blocks and aggregation blocks 9 to 12, which may have buffers with limited size. Similarly, we randomly take 200 different settings of parameters: Arrival rate is uniformly distributed within $[0, 12]$; Service rate of blocks where packet depart from the network is uniformly distributed within $[0, 12]$; Capacity values of different links, and buffer size setting at blocks from which packets depart are identical to the single-hop case in Section V-A. The performance of the network policies are presented in Fig. 10 and Fig. 11. The results share a similar trend with the single-hop case.

VI. Conclusion

In this paper, we study overload balancing in single-hop networks with bounded buffers. We show that bounded buffer affects the resulting policy to achieve most balanced overload. We leverage ordinary differential equations to model the queue dynamics in bounded buffer systems. We first prove that setting link service rates to minimize the quadratic sum of the queue overload rates leads to the lexicographic minimum queue overload. Based on this result, we prove that a maxweight scheduling and backpressure policy asymptotically achieves most balanced overload, through a novel formulation of the policy in a differentiable form which may be of independent interest. We further propose a distributed maxweight + backpressure policy that can reduce communication overhead by one order of magnitude. We validate the performance of our proposed policies by simulation over single-hop structure and Clos networks under different packet arrival rates, link capacities, and buffer settings. Extension of the results in this work to multi-hop networks, and exploitation of the differential equation formulation for other network performance metrics, are promising future directions.

References

[1] J. David and C. Thomas, "Discriminating flash crowds from ddos attacks using efficient thresholding algorithm," *Journal of Parallel and Distributed Computing*, vol. 152, pp. 79–87, 2021.

[2] Y. Kim, W. C. Lau, M. C. Chuah, and H. J. Chao, "Packetscore: a statistics-based packet filtering scheme against distributed denial-of-service attacks," *IEEE transactions on dependable and secure computing*, vol. 3, no. 2, pp. 141–155, 2006.
[3] C.-p. Li, G. S. Paschos, L. Tassiulas, and E. Modiano, "Dynamic overload balancing in server farms," in *2014 IFIP Networking Conference*. IEEE, 2014, pp. 1–9.
[4] "Facebook is back online after a massive outage that also took down instagram, whatsapp, messenger, and oculus," https://www.theverge.com/2021/10/4/22708989/instagram-facebook-outage-messenger-whatsapp-error.
[5] G. Como, K. Savla, D. Acemoglu, M. A. Dahleh, and E. Frazzoli, "Robust distributed routing in dynamical networks—part i: Locally responsive policies and weak resilience," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 317–332, 2012.
[6] D. Shah and D. Wischik, "Fluid models of congestion collapse in overloaded switched networks," *Queueing Systems*, vol. 69, no. 2, p. 121, 2011.
[7] O. Perry and W. Whitt, "Chattering and congestion collapse in an overload switching control," *Stochastic Systems*, vol. 6, no. 1, pp. 132–210, 2016.
[8] V. Venkataramanan and X. Lin, "On the queue-overflow probability of wireless systems: A new approach combining large deviations with lyapunov functions," *IEEE transactions on information theory*, vol. 59, no. 10, pp. 6367–6392, 2013.
[9] L. Georgiadis and L. Tassiulas, "Optimal overload response in sensor networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2684–2696, 2006.
[10] C. W. Chan, M. Armony, and N. Bambos, "Fairness in overloaded parallel queues," *arXiv preprint arXiv:1011.1237*, 2010.
[11] B. Radunovic and J.-Y. Le Boudec, "A unified framework for max-min and min-max fairness with applications," *IEEE/ACM Transactions on networking*, vol. 15, no. 5, pp. 1073–1083, 2007.
[12] C.-p. Li and E. Modiano, "Receiver-based flow control for networks in overload," *IEEE/ACM Transactions on Networking*, vol. 23, no. 2, pp. 616–630, 2014.
[13] C. Qu, R. N. Calheiros, and R. Buyya, "Mitigating impact of short-term overload on multi-cloud web applications through geographical load balancing," *concurrency and computation: practice and experience*, vol. 29, no. 12, p. e4126, 2017.
[14] A. Baron, R. Ginosar, and I. Keslassy, "The capacity allocation paradox," in *IEEE INFOCOM 2009*. IEEE, 2009, pp. 1359–1367.
[15] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM computer communication review*, vol. 38, no. 4, pp. 63–74, 2008.
[16] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano *et al.*, "Jupiter rising: A decade of clos topologies and centralized control in google's datacenter network," *ACM SIGCOMM computer communication review*, vol. 45, no. 4, pp. 183–197, 2015.
[17] L. B. Le, E. Modiano, and N. B. Shroff, "Optimal control of wireless networks with finite buffers," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–9.
[18] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu *et al.*, "B4: Experience with a globally-deployed software defined wan," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013.
[19] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.
[20] "Juniper qfx5210 switch," https://www.juniper.net/us/en/products/switches/qfx-series/qfx5210-switch-datasheet.html.
[21] J. G. Dai and W. Lin, "Maximum pressure policies in stochastic processing networks," *Operations Research*, vol. 53, no. 2, pp. 197–218, 2005.
[22] M. G. Markakis, E. Modiano, and J. N. Tsitsiklis, "Delay analysis of the max-weight policy under heavy-tailed traffic via fluid approximations," *Mathematics of Operations Research*, vol. 43, no. 2, pp. 460–493, 2018.
[23] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data networks*. Prentice-Hall International New Jersey, 1992, vol. 2.
[24] P. Dupuis, K. Leder, and H. Wang, "Importance sampling for weighted-serve-the-longest-queue," *Mathematics of Operations Research*, vol. 34, no. 3, pp. 642–660, 2009.

APPENDIX

A. Proof Sketch of Lemma 1

We term (3) as l_2 problem and (4) as l_∞ problem, since they respectively minimize the l_2 and l_∞ norm of $\dot{\mathbf{q}}$. The Lagrangian functions of (3) and (4) are[5]

- $\mathcal{L}^{(2)}(\mathbf{g},\mathbf{a},\mathbf{b},h) = \frac{1}{2}\sum_{i\in\mathcal{V}}(\dot{q}_i)^2 + \sum_{(i,j)\in\mathcal{E}} a_{ij}(g_{ij} - c_{ij}) - \sum_{(i,j)\in\mathcal{E}} b_{ij}g_{ij} + \sum_{i\in\mathcal{B}} h_i\dot{q}_i$.
- $\mathcal{L}^{(\infty)}(\mathbf{g},v,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\bar{h}) = v + \sum_{i\in\mathcal{V}}\gamma_i(\dot{q}_i - v) + \sum_{(i,j)\in\mathcal{E}}\alpha_{ij}(g_{ij}-c_{ij}) - \sum_{(i,j)\in\mathcal{E}}\beta_{ij}g_{ij} + \sum_{i\in\mathcal{B}} h_i\dot{q}_i$.

where $\dot{q}_i$ follows (1) and (2). Their KKT conditions are

l_2 **problem:**

$$\begin{cases} -\dot{q}_i + \dot{q}_j + a_{ij} - b_{ij} - h_i + h_j = 0,\ \forall (i,j)\in\mathcal{E} \\ h_i\dot{q}_i = 0,\ \forall i\in\mathcal{B} \\ a_{ij}(g_{ij}-c_{ij}) = 0,\ a_{ij}\geq 0;\ b_{ij}g_{ij}=0,\ b_{ij}\geq 0,\ \forall (i,j)\in\mathcal{E} \end{cases} \tag{12}$$

l_∞ **problem:**

$$\begin{cases} 1 - \sum_{i\in\mathcal{V}}\gamma_i = 0 \\ -\gamma_i + \gamma_j + \alpha_{ij} - \beta_{ij} - \bar{h}_i + \bar{h}_j = 0,\ \forall (i,j)\in\mathcal{E} \\ \bar{h}_i\dot{q}_i = 0,\ \forall i\in\mathcal{B} \\ a_{ij}(g_{ij}-c_{ij}) = 0,\ a_{ij}\geq 0;\ b_{ij}g_{ij}=0,\ b_{ij}\geq 0,\ \forall (i,j)\in\mathcal{E} \\ \gamma_i(\dot{q}_i - v) = 0,\ \gamma_i\geq 0,\ \forall i\in\mathcal{V} \end{cases} \tag{13}$$

Denote an optimizer of the l_2 problem as $\mathbf{g}^{(2)}$, and it suffices to show that given $(\mathbf{g}^{(2)},\mathbf{a},\mathbf{b},\mathbf{h})$ that satisfies (12), there exists $(\bar{\mathbf{g}},v,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\bar{h})$ that satisfies (13), and $\mathbf{g}^{(2)} = \bar{\mathbf{g}}$. If this holds, then $\mathbf{g}^{(2)} = \bar{\mathbf{g}}$ minimizes (4) as the l_∞ problem is convex.

B. Proof Sketch of Theorem 2

Our proof idea is to verify the *existence* and *convergence* conditions for mw+bp (9) mentioned in Section IV-A. The *existence* condition can be proved in a similar way as Lemma 1, where we prove that under (9), for any given solution $\mathbf{q}^*$ and feasible values of Lagrangian multipliers to the KKT conditions of (6), then under $\mathbf{g}^* = \mathbf{g}(\mathbf{q}^*)$ there exists feasible Lagrangian multipliers to the KKT conditions of (3). The proof requires the property that the policy should satisfy

$$\frac{\partial g_{s_i d_j}}{\partial q_{s_i}} > 0,\ \frac{\partial g_{s_k d_j}}{\partial q_{s_i}} < 0,\ \forall k\neq i, \forall d_j\in\mathcal{B}, \tag{14}$$

which the mw+bp policy (9) satisfies. The *convergence* condition is proved by verifying that under mw+bp (9), the $\dot{\mathbf{q}}$ at the steady state of (5) is the most balanced state as induced by the optimum $\mathbf{g}^*$ of (3). We present the proof idea of balancing the ingress nodes for brevity, while taking into consideration of balancing the egress nodes is the same. Due to overload, there exists at least one s_{i_0} that will have $\dot{q}_{s_{i_0}} > 0$ in the steady state. Suppose the most balanced state at the N ingress nodes satisfies $\dot{q}_{s_i} = \xi_{i,i_0}\dot{q}_{s_{i_0}}$ for some $\xi_{i,i_0}\geq 0,\ \forall i\neq i_0$, then it suffices to verify if the ODE dynamics under new variables $\mathbf{x} := \{x_i\}_{i\neq i_0}$ where $x_i := q_{s_i} - \xi_{i,i_0}q_{s_{i_0}}$ will converge to the equilibrium point $\mathbf{x}^*$, indicating $\dot{\mathbf{x}} = 0$ in the steady state, which means the most balanced overload is achieved. By transforming the ODE (5) with respect to $\mathbf{q}$ to $\mathbf{x}$ under mw+bp (9), we can prove the convergence to $\mathbf{x}^*$ by Lyapunov function method, conditioned on (14) which (9) satisfies.

[5] $\mathbf{b}$ in the proof are Lagrangian multipliers rather than buffer sizes.