Stability of the Max-Weight Routing and Scheduling Protocol in Dynamic Networks and at Critical Loads

Matthew Andrews
Bell Laboratories
Murray Hill, NJ
andrews@research.bell-labs.com

Kyomin Jung*
MIT
Cambridge, MA
kmjung@mit.edu

Alexander Stolyar
Bell Laboratories
Murray Hill, NJ
stolyar@research.bell-labs.com

ABSTRACT

We study the stability of the MAX-WEIGHT protocol for combined routing and scheduling in communication networks. Previous work has shown that this protocol is stable for adversarial multicommodity traffic in subcritically loaded static networks and for single-commodity traffic in critically loaded dynamic networks. We show:

- The MAX-WEIGHT protocol is stable for adversarial multicommodity traffic in adversarial dynamic networks whenever the network is subcritically loaded.
- The MAX-WEIGHT protocol is stable for fixed multicommodity traffic in fixed networks even if the network is critically loaded.

The latter result has implications for the running time of the MAX-WEIGHT protocol when it is used to solve multicommodity flow problems. In particular, for a fixed problem instance we show that if the value of the optimum solution is known, the MAX-WEIGHT protocol finds a flow that is within a $(1-\varepsilon)$-factor of optimal in time $O(1/\varepsilon)$ (improving the previous bound of $O(1/\varepsilon^2)$). If the value of the optimum solution is not known, we show how to apply the MAX-WEIGHT algorithm in a binary search procedure that runs in $O(1/\varepsilon)$ time.

Categories and Subject Descriptors: F.2.2 [Nonnumerical Algorithms and Problems]: Sequencing and scheduling.

General Terms: Algorithms.

Keywords: Routing, scheduling, stability.

1. INTRODUCTION

In this paper we consider the problem of combined packet routing and scheduling in communication networks. We study an algorithm introduced by Tassiulas and Ephremides in [22, 23] and Awerbuch and Leighton in [7, 8] that has been referred to by a number of names, including the MAX-WEIGHT algorithm, the Differential Backlog algorithm, the Backpressure algorithm and the Load Balancing algorithm. The exact definition of this algorithm will be given in Section 1.2. The essential idea is that each node maintains a queue for each destination. When an edge needs to schedule a packet it tries to move a packet from a large queue to a small queue. Throughout this paper we shall refer to this as the MAX-WEIGHT algorithm.

Since the MAX-WEIGHT algorithm was first defined, it has been studied in a variety of wireless and wireline contexts. The main reason for its popularity is that it is throughput-optimal in a wide range of circumstances. By throughput-optimal we mean that it can serve all the offered packets and maintain stability whenever this is feasible. The following two results are particularly relevant to our work.

- In [1], Aiello, Kushilevitz, Ostrovsky and Rosen showed that in *static* graphs with *multicommodity* demands, the MAX-WEIGHT algorithm achieves stability whenever the network is subcritically loaded. (The network is subcritically loaded if there is a way to route the traffic so that on any edge the offered load is strictly less than the edge capacity.)
- In [5], Anshelevich, Kempe and Kleinberg showed that in dynamic graphs with *single-commodity* demands, the MAX-WEIGHT algorithm achieves stability whenever the network is critically loaded. (The network is critically loaded if there is a way to route the traffic so that on any edge the offered load is at most the available edge capacity.) A similar result was also proved by Awerbuch et al. [6] although under a slightly different input model. The model of [5] is slightly closer to the model that we consider in our work.

The above two results raised two immediate open questions. First, what is the performance of the MAX-WEIGHT algorithm in dynamic graphs with multicommodity demands? Such a problem is of importance since dynamic graphs represent a simple model of wireless ad-hoc networks and we would like to know how well the MAX-WEIGHT algorithm will operate in such networks. Second, what is the performance of the MAX-WEIGHT algorithm in critically loaded networks with multicommodity demands? This question has relevance to the running time of iterative algorithms for multicommodity flow problems. Before we can describe our results in detail and compare them with previous work we must define our model together with the MAX-WEIGHT algorithm.

*Work partially done while the author was visiting Bell Laboratories.

STOC'07, June 11–13, 2007, San Diego, California, USA.

1.1 The Model

We model a communication network by a graph $G = (V, E)$, where $|V| = n, |E| = m$. Each edge $e \in E$ models a communication channel. We assume that the network has the following properties.

- The network is directed. We model a two-way communication channel by two directed links. Let Δ be the maximum degree of the network.
- Time is slotted and indexed by $t \in \mathbb{N}$.
- The capacity of an edge changes over time. (As mentioned earlier, this allows us to model wireless networks.) We let $c_e(t)$ be the capacity of edge e at time t. If $e = (v, u)$ then data of maximum size $c_e(t)$ can be transferred from node v to node u at time t. The values of this edge capacity process are controlled by an adversary whose properties are described below.

  Let $C = \{c_e(t) : e \in E, t \in \mathbb{N}\}$, i.e. C is the set of all possible edge capacities. Let $c_{\max}$ be the maximum possible edge capacity. If $C = \{0, 1\}$ then we say that G is a *dynamic* graph. In this case we say that edge e is *open* at time t if $c_e(t) = 1$. If $C = \{1\}$ then we say that G is a *static* graph.
- The adversary determines when packets are injected. When packet p is injected, its size ℓ_p, its source node s_p and its destination node d_p are specified. However, the route that the packet must follow is not specified. If there exists a node d such that $d_p = d$ for all d then we say that we have a single-commodity problem, otherwise we have a multicommodity problem. For much of the paper we shall focus on the case in which packets have unit size, i.e. $\ell_p = 1$ for all p.
- The adversary controls packet arrivals and edge capacities with a limitation that the injection sequence does not inherently overload the network. (Definition 1.)
- It is the purpose of a *routing and scheduling protocol*, to determine *which* data (of size at most $c_e(t)$) to transmit along edge e at time t.

As already mentioned, we assume that the adversary does not inherently overload the network, otherwise stability is impossible to achieve. In particular:

DEFINITION 1 ($A(\omega, \epsilon)$-ADVERSARY). *We say that an adversary injecting the packets and controlling the edges is an $A(\omega, \epsilon)$-adversary for some $\varepsilon \geq 0$ and some integer $\omega \geq 1$, if the following holds: For any time $t \in \mathbb{N}$, let $I^{[t,t+\omega-1]}$ be the set of packets injected during the ω time steps from t to $t+\omega-1$. Then the adversary can associate with each packet $p \in I^t$, a simple path Γ_p from s_p to d_p, such that for all $e \in E$,*

$$\sum_{p \in I^{[t,t+\omega-1]}, e \in \Gamma_p} \ell_p \leq (1-\varepsilon) \sum_{t'=t}^{t+\omega-1} c_e(t').$$

1.2 The Protocol

We are now in a position to define the protocol MAX-WEIGHT . We assume that each vertex v has n queues: one queue for each destination. Let $Q_{v,d}$ be the queue at node v for data having destination d. Let $q^t_{v,d}$ be the total size of data in queue $Q_{v,d}$ at time t. We define a general algorithm MAX-WEIGHT(β) that is parameterized by a parameter β. We use MAX-WEIGHT to denote the algorithm with $\beta = 1$. The reason to introduce the parameter β is that the analysis is easier when β is large. Under MAX-WEIGHT(β) , each node v does the following at each time t:

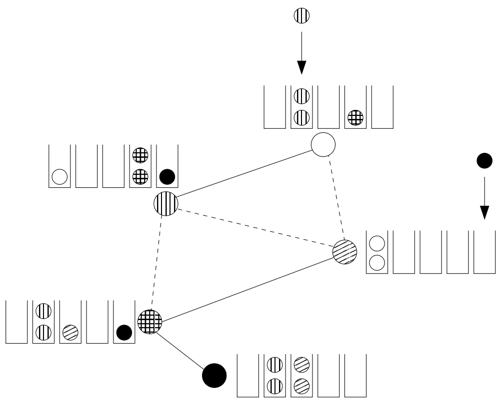

Figure 1: In the dynamic graph adversarial model, at each time slot the adversary determines packet arrivals and edge capacities. The MAX-WEIGHT(β) protocol then determines which packets will be transmitted along each edge. Data is stored at each node according to its eventual destination.

Algorithm MAX-WEIGHT(β)

- Phase 1: Accept all packets injected by the Adversary to v.
- Phase 2: For each edge $e = (v, u)$ that appears at time t, let d be such that $(q^t_{v,d})^\beta - (q^t_{u,d})^\beta$ is maximized over all $d \in V$ (with an arbitrary tiebreaking rule). If $q^t_{v,d} - q^t_{u,d}$ is greater than[1] $\Delta \cdot c_{\max}$, send data of size $\max\{c_e(t), q^t_{v,c}\}$ from $Q_{v,d}$ to $Q_{u,d}$ along e.
- Phase 3: Remove any packets that arrive at their destination.

The algorithm can be understood to be designed so that the following *potential function* decreases as much as possible.

$$P(t) \triangleq \sum_{v,d} (q^t_{v,d})^{\beta+1}.$$

[1]The reason we require that the difference between the queue heights is more than $\Delta \cdot c_{\max}$ is that we want to make sure that the potential in the system does not increase when data is transferred from one queue to another. We believe that our results would still hold even if we only require this difference to be positive.

1.3 Contribution

We say that a system is *stable* if the queue sizes are bounded over time. The aim of this paper is to derive conditions under which the MAX-WEIGHT protocol is stable. Our results are:

1. In Section 3 we show that the MAX-WEIGHT(β) protocol is stable in adversarial dynamic graphs with adversarial multicommodity demands for any $\varepsilon > 0$ and for some large β. The reason we study MAX-WEIGHT(β) before the (more natural) MAX-WEIGHT protocol is that MAX-WEIGHT(β) is easier to analyze and allows us to introduce some of the essential ideas gradually.

2. In Section 4 we show that the MAX-WEIGHT protocol is stable in adversarial dynamic graphs with adversarial multicommodity demands for any $\varepsilon > 0$.

3. In Section 5 we briefly sketch how to extend our results to the case of graphs with arbitrary time-varying capacities and arbitrary (upper bounded) packet sizes.

4. In Section 6 we turn our attention to the problem in which $\varepsilon = 0$. Although we are not able to resolve this question in full generality, we show that in the case of static graphs with constant traffic patterns, the MAX-WEIGHT protocol is stable.

5. In Section 7 we show how the previous result can be used to obtain better bounds on the running time of the Awerbuch-Leighton multicommodity flow algorithm. Specifically, for a fixed instance of the problem and assuming that a flow of value λ^* is feasible, they show how to find a flow of value $(1-\varepsilon)\lambda^*$ in time $O(1/\varepsilon^2)$. We show that in fact the running time is only $O(1/\varepsilon)$. Hence if the value of the optimum solution to the multicommodity flow problem is known, the MAX-WEIGHT protocol finds a flow that is within a $(1-\varepsilon)$-factor of optimal in time $O(1/\varepsilon)$. If the value of the optimum solution is not known, we show how to apply the MAX-WEIGHT algorithm in a binary search procedure that runs in $O(1/\varepsilon)$ time.

1.4 Previous work

As already mentioned, the MAX-WEIGHT algorithm was first introduced by Tassiulas and Ephremides [22, 23] and Awerbuch and Leighton [7, 8]. Tassiulas and Ephremides studied the algorithm in the context of time-varying wireless networks and showed stability under the assumption that the traffic injections and the network dynamics are controlled by a stationary stochastic process (rather than an adversary). Awerbuch and Leighton were interested in multicommodity flow problems and showed that MAX-WEIGHT can be used to find a feasible flow in time $O(\frac{LM}{\varepsilon^2}(K + \log(\frac{K}{\varepsilon})))$ assuming that there exists a feasible flow when the demands are increased by a factor of $1+\varepsilon$. Here L is the length of the longest flow path and K is the number of demands. They remark that by using binary search, they can then obtain a $(1-\varepsilon)$-approximation to the optimal value of a flow.

In [1], Aiello et al. extended the study of MAX-WEIGHT to the case of static networks with adversarially generated multicommodity demands. They showed that MAX-WEIGHT is stable as long as $\varepsilon > 0$. They also extended the result to dynamic networks in which $c_e(t) \in \{0,1\}$. However, their restriction on the adversary in this context is that for all e and t, $\sum_{p\in I^{[t,t+\omega-1]}, e\in\Gamma_p} \ell_p + \varepsilon\omega \le \sum_{t'=t}^{t+\omega-1} c_e(t')$. Note that the additive term on the left-hand side means that each edge has to appear frequently, i.e. they preclude a situation in which the adversary can remove an edge from the network for long periods. In such a case it is possible to apply an analysis that is very similar to the case for static graphs. The result that we prove in Section 4 addresses a more general model of dynamic graphs in which the adversary can remove edges from the network for arbitrarily long periods.

In [5], Anshelevich et al. showed that for single-commodity traffic MAX-WEIGHT is stable in adversarial dynamic graphs when $\varepsilon = 0$. Our result in Section 4 can be viewed as extending the result of [5] (for $\varepsilon > 0$) to the case of multicommodity traffic. A similar result for single-commodity traffic was proved by Awerbuch et al. in [6]. However, their adversary was somewhat different (and more general). They showed stability under the assumption that the adversary can construct a schedule under which the queues are stable. Such an adversary may not conform to Definition 1.

The MAX-WEIGHT algorithm has also been studied in wireless networks under the node-constraint model. Some papers that show stability under stochastically generated traffic include [3, 2, 18]. In contrast, it was shown in [4] that for adversarial traffic in which the channel rates can be arbitrarily small, no online algorithm (including MAX-WEIGHT) can be stable. More recently, there is a body of work that aims to combine MAX-WEIGHT with congestion-control algorithms that decide how much data is to be injected into the network. Example papers include [21, 10, 17]. The aim in these papers is to maximize the total utility of traffic injected into the network, for example the sum over all demands of the logarithm of the injected flow rate.

Other situations where the MAX-WEIGHT algorithm has been studied include scheduling input-queued crossbar switches (see e.g. McKeown et al. [15]) and load balancing tasks in networks of processors (see e.g. Muthukrishnan and Rajaraman [16]).

1.5 Why Previous Analyses Do Not Work Directly for Dynamic Networks

We now briefly explain why we cannot directly apply previous analyses of MAX-WEIGHT that were developed for static networks (or networks that vary either in a stochastic manner or else guarantee that each edge appears frequently). The main idea behind most previous analyses is that at each time step the potential function increases by at most $O((\max_{v,d} q_{v,d}^t)^{\beta-1})$. Moreover, if $\max_{v,d} q_{v,d}^t$ is large then the potential function strictly *decreases* (by an amount $\Theta(\varepsilon(\max_{v,d} q_{v,d}^t)^{\beta}))$. In other words, if the queues are sufficiently large, we always get a decrease in potential. This means that the total potential stays bounded which means in turn that the maximum queue size in the network remains bounded. However, in an adversarial network the adversary is able to disconnect large queues from the network for arbitrarily long periods. This means that we can no longer say that if the maximum queue size is large then the potential strictly decreases.

The method of Anshelevich et al. [5] for single-commodity traffic in dynamic networks involved different techniques and a different potential function. However, their analysis made critical use of the max-flow min-cut theorem and hence it cannot be extended to multicommodity traffic due to the existence of flow-cut gaps in the multicommodity setting.

2. OVERVIEW OF THE STABILITY PROOFS

The proofs of stability of MAX-WEIGHT(β) and MAX-WEIGHT in dynamic networks is quite complex. Hence in this section we give a high-level overview of the proofs. The first part of the analysis concerns the dynamics of the potential function $P(t) = \sum_{v,d}(q^t_{v,d})^{\beta+1}$. By the definition of the MAX-WEIGHT(β) protocol the change in the potential function at any time satisfies the following two key properties.

- Whenever the MAX-WEIGHT(β) protocol moves data from node v to node u, the potential in the system due to this transmission decreases.
- When data is injected into the network, the potential in the system may increase. However, we can obtain the following Lemma (whose proof is contained in the Appendix).

LEMMA 1. *Consider the* MAX-WEIGHT(β) *protocol in adversarial dynamic graphs with multicommodity demands for $\varepsilon > 0$. Then for each injection of packet p at time t having size ℓ_p, we can associate this injection with packet movements along its corresponding simple path so that the sum of potential changes due to these movements are at least $-\frac{\varepsilon}{1-\varepsilon}\ell_p(\beta+1)q^\beta + \ell_p O(q^{\beta-1})$, where q is the height of the queue where the packet is injected. Moreover, all these changes occur in a window of size ω. Hence there is a constant q^* depending on n, ω and ε, so that if $q \geq q^*$ the sum of potential changes due to the injection is less than $-\varepsilon\ell_p q^\beta$.*

The above properties mean that we are able to show a net decrease in the potential in the system, as long as there are "sufficient" injections into queues that are large enough. *A priori*, it is not clear that the adversary has to ever inject packets into large queues. However, we show that for both MAX-WEIGHT(β) and MAX-WEIGHT, if the queue sizes grow without bound, it must be the case that there are enough injections into large queues to cause a decrease in potential. This leads to a contradiction.

2.1 Overview of stability of MAX-WEIGHT(β)

We begin by giving an overview of the stability of MAX-WEIGHT(β) for some parameter β. The analysis is easier than the analysis for MAX-WEIGHT since an injection into a large queue of size q leads to a large decrease in potential, namely a decrease of $O(q^\beta)$. We focus on the case of dynamic graphs (i.e. $c_e(t) \in \{0, 1\}$ for all t), and the case of unit size packets. The key observations are the following:

- Whenever the MAX-WEIGHT(β) protocol serves a packet, the packet moves to a smaller queue. Therefore, the only way to increase the maximum queue size in the network is via a packet injection.
- Suppose that the adversary can make the network unstable. Let S_t be the configuration of the network at time t. Suppose that $S_t = S_{t^*}$ for some $t < t^*$. Then, the period between time t and time t^* was completely redundant in terms of making the queue sizes grow. Hence, we can assume without loss of generality that $S_t \neq S_{t^*}$.
- The number of queues in the network is at most n^2. Therefore, if we let q be the maximum queue size, the number of possible network configurations is $O(q^{n^2})$. Therefore, in time $O(q^{n^2})$, there must be some injection into a queue of size at least q. Let t_q be the first time this happens. By our previous discussion, the decrease in potential due to this injection is $\Omega(q^\beta)$. Moreover, by Lemma 1 we observe that only the injections to queues having size less than some constant q^* can make the potential increase. Hence the maximum increase in potential due to any injection is at most $(q^*)^2$ which implies that the total increase in potential up to time t_q is $O(q^{n^2})$.
- Therefore, for large enough q, the total increase in potential up to time t_q is less than the decrease in potential at time t_q. This is clearly a contradiction since as the queue sizes grow without bound at the time steps t_q, the potential in the network must also grow without bound.

2.2 Overview of stability of MAX-WEIGHT

We now give an overview of the stability of the MAX-WEIGHT protocol in dynamic networks. The proof is more complicated since we do not get such a large decrease in potential when a packet is injected into a large queue. The structure of the proof is as follows. A similar argument can be applied to show that the MAX-WEIGHT(β) protocol for any $\beta \geq 1$ is stable under the same adversarial setup.

- We define a set of queue thresholds $F_1 \gg F_2 \gg F_3 \ldots$. Consider some fixed time t_0 and let the queue sizes at that time be $q_1 \geq q_2 \geq q_3 \ldots$. Then, if the total data in the network is sufficiently large, there must be some k such that $q_k \geq F_k$ and $q_{k+1} \leq F_{k+1}$. The queues having size bigger than F_k are called *tall queues*. The other queues are called *small queues*.
- Let t_1 be the first time after t_0 that there is an injection of a packet to a tall queue or a transmission of a packet from a tall queue to a small queue. By Lemma 1 an event of the first type causes a large decrease in the system potential. The definition of potential directly implies that an event of the second type also causes a large decrease in the system potential. We can therefore show that the total potential remains bounded as long as the increase in potential between times t_0 and t_1 is less than the decrease in potential due to the injection or transmission at time t_1.
- Note that the only increase in potential can come from an injection to a small queue. There are two cases to consider:
  - **Case 1:** The injections into the small queues and the capacities utilized by the transmissions between small queues satisfy the definition of an $A(\omega, \epsilon)$-adversary. In this case we can use an inductive argument to bound the total increase in potential associated with these queues (since the number of these queues is strictly less than n).

– **Case 2:** The capacity associated with some of the injections into the small queues is utilized by transmissions between large queues. We call these injections "bad injections". However, recall that there are no injections into tall queues between time t_0 and t_1. Hence transmissions between tall queues have the effect of "smoothing out" the heights of the tall queues. That is, a transmission between two tall queues reduces the difference of the heights of these queues. If there is a transmission between two tall queues of very different heights, then the resulting loss in potential is enough to compensate for the increase in potential between time t_0 and time t_1. If on the other hand the only transmissions between tall queues is between queues of similar heights then there is a limit to how many such injections there are. This in turn leads to a bound on the number of bad injections. Hence we can use an inductive argument to bound the total increase in potential associated with the small queues, subject to this bound on the number of bad injections. Once again we can make sure that this increase is less than the decrease in potential associated with the injection or transmission at time t_1.

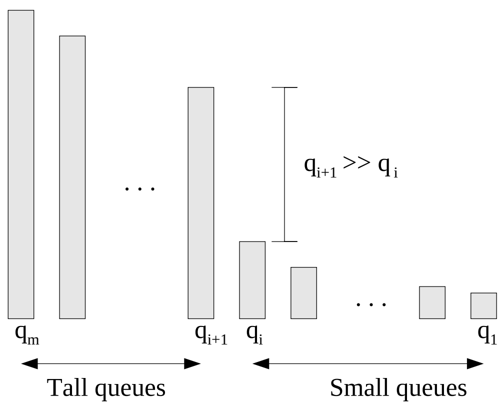

Figure 2: We define a set of queue thresholds $F_1 \gg F_2 \gg \ldots$. Consider some fixed time and let the queue sizes at that time be $q_1 \geq q_2 \geq q_3 \ldots$. Then, if the total data in the network is sufficiently large, there must be some k such that $q_k \geq F_k$ and $q_{k+1} \leq F_{k+1}$. The queues having size at least F_k are called *tall queues*. The other queues are called *small queues*.

3. STABILITY OF MAX-WEIGHT(β) IN DYNAMIC NETWORKS

In this section and the next we present the details of our stability proofs for dynamic networks.

THEOREM 2. *For $\beta > n^2$, the* MAX-WEIGHT(β) *protocol is stable in adversarial dynamic graphs with multicommodity demands for any $\varepsilon > 0$.*

PROOF. First note that whenever the MAX-WEIGHT(β) protocol serves a packet, the packet moves to a smaller queue. Therefore, the only way to increase the maximum queue size in the network is via a packet injection to the tallest queue.

Now, suppose that the adversary can make the network unstable. For $q \in \mathbb{N}$, let t_q be the first time so that the size of the tallest queue becomes at least q. Let S_t be the configuration of the network at time t. This consists of the queue sizes at time t, and for all $t' \in [t - \omega + 1, t]$, the information regarding the set of open edges at time t', and the source and destination information of all the packets injected at time t'. Now suppose that $S_t = S_{t^*}$ for some $t < t^*$. Then, the period between time t and time t^* was completely redundant in terms of making the queue sizes grow, i.e. removing all the processes during time $[t, t^* - 1]$ results in another adversary which make the network unstable. Hence we can assume without loss of generality that $S_t \neq S_{t^*}$ for any $t, t^* \leq t_q$, $t \neq t^*$. Let n be the number of nodes. Then, the number of queues in the network is at most n^2. So, by considering n and ω as constants, the number of possible network configurations in which the maximum queue size is less than q is $O(q^{n^2})$. Therefore, $t_q = O(q^{n^2})$.

Now, by Lemma 1, each injection to a queue of size bigger than some constant q^* makes the potential decrease. Moreover, each injection to a queue of size at most q^* makes the potential increase by at most $(q^* + 1)^{\beta+1}$. Note also that at each time slot $[t, t + \omega - 1]$, the number of injected packets (except packets having same source and destination) during that period of time is at most $\omega|E|$, where $|E|$ is the number of edges. Hence the maximum possible increase in potential up to time $t_q + \omega$ is at most $(q^* + 1)^{\beta+1} \cdot \omega \cdot n^2 \cdot t_q$. If we consider n, ω and β as constants this expression is $O(q^{n^2})$.

Note that by the definition of t_q, at time t_q there must be a packet injection to a queue of size at least $q - 1$. By Lemma 1, the decrease in potential due to this injection is $\Omega(q^{\beta})$. Hence for $\beta > n^2$, if q is large enough, the decrease in potential due to this injection is larger than the maximum possible increase in potential up to time $t_q + \omega$. Hence for large enough q, $P(t_q + \omega) \leq P(0)$. But by definition of t_q, the size of the tallest queue at time t_q is q. Hence $P(t_q) \geq q^2$, which is a contradiction. Therefore, there does not exist an adversary that can make the network unstable. □

4. STABILITY OF MAX-WEIGHT IN DYNAMIC NETWORKS

THEOREM 3. *The* MAX-WEIGHT *protocol is stable in adversarial dynamic graphs with multicommodity demands for any $\varepsilon > 0$.*

PROOF. We prove Theorem 3 by showing a stability result for a more general model, namely for any *general adversarial queuing system* with *bad packets*. We introduce this new model since in our inductive argument, we need to consider the case when there are a finite number of injected packets for which the adversary does not associate a path. These packets are called *bad packets*. A precise definition will be given later.

In the original adversarial dynamic graph model, we have n^2 queues, and there are some constraints on how packets move between queues, e.g. packets cannot be transmitted between two queues located at the same node. A general adversarial queuing system is defined the same as our original model, except that in this model the number of queues can be any finite number, not only n^2, and packet transmissions between any two different queues are allowed. In addition, the number of destination queues can be any positive number. An adversary allowing b many *bad packets* is defined as follows.

DEFINITION 2 ($A(\omega,\epsilon,b)$-ADVERSARY). *We say that an adversary injecting the packets and controlling the edges in a general adversarial queue system is an $A(\omega,\epsilon,b)$-adversary for some $\varepsilon \geq 0$ and some integers $\omega \geq 1$ and $b \geq 0$, if the following holds: Among all the packets injected over all time, the adversary can designate b number of these packets as* bad packets. *We say all the other injected packets are* good packets. *Then for any time $t \in \mathbb{N}$, let $I^{[t,t+\omega-1]}$ be the set of good packets injected during the ω time steps from t to $t+\omega-1$. Then there is a constant $q^* \geq 0$ so that the Adversary can associate with each packet $p \in I^t$ injected into a queue of size $\hat{q}$, a set of edges Γ_p such that*

$$\sum_{e=(i,j)\in\Gamma_p} |q^t(i) - q^t(j)| \geq \hat{q} - q^*,$$

and for all $e \in E$,

$$\sum_{p\in I^{[t,t+\omega-1]}, e\in\Gamma_p} \ell_p \leq (1-\varepsilon)\sum_{t'=t}^{t+\omega-1} c_e(t').$$

From now on in this proof, by Adversary we mean an Adversary of a general adversarial queue system. It can be verified that that Lemma 1 is also true for this model. We now show the following Lemma, which implies Theorem 3.

LEMMA 4. *For any general adversarial queue system with $A(\omega,\epsilon,b)$-adversary, where $\varepsilon > 0$, the* MAX-WEIGHT *protocol is stable.*

PROOF. Let $\varepsilon > 0$ and ω be fixed, and let n be the number of queues. We will show that there exists a constant $B(n, q_0, b)$ such that for a given general adversarial queue system with n queues, for any $A(\omega,\epsilon,b)$-adversary under MAX-WEIGHT , when the size of the tallest queue at time $t = 0$ is at most q_0, the sizes of all queues over all $t \geq 0$ is bounded above by $B(n, q_0, b)$.

We will use induction on n to show that for any $q_0 \geq 0$ and $b \geq 0$, there exists $B(n, q_0, b)$. First, when $n = 1$, since there is at least one destination queue, the queue must be a destination queue. So the claim is true.

Suppose that there exist $B(m, q_0, b)$ for all $1 \leq m \leq n-1$, and for all $q_0 \geq 0$ and $b \geq 0$. Using this induction hypothesis, we will show that for any q_0, $B(n, q_0, 0)$ exists. Then, note that we can set

$$B(n, q_0, 1) = B(n, B(n, q_0, 0) + 1, 0),$$

since at the time when the bad packet arrives, the size of the tallest queue is at most $B(n, q_0, 0)$. Similarly we can set $B(n, q_0, i) = B(n, B(n, q_0, i-1) + 1, 0)$, by considering the time when the ith bad packet arrives.

Now we only need to prove that $B(n, q_0, 0)$ exists. Let $P(t)$ be the potential of the queues at time t. First note that by Lemma 1, and the fact that each injection to a queue of size at most q^* makes the potential increase by at most $(2q^*+1)$, the maximum possible increase of potential induced by all the injections during any time window of size ω is bounded by some constant P_0. Now, for fixed n, we will define some constants F_k, $k = 1, 2, \ldots, n$, which are decreasing over k, and show that for any $A(\omega,\epsilon)$-adversary for a general adversarial queue system with n queues, for all time $t \geq 0$, $P(t)$ is bounded by some value that is independent of t. More precisely we will show that

$$P(t) \leq (n-1)F_1^2 + \max\{nq_0^2, nF_1^2 + 2\sqrt{n}F_1\}. \qquad (1)$$

We will define F_k so that if the size of the kth tallest queue is smaller than F_k, and the size of the $(k-1)$th tallest queue is bigger then F_{k-1}, than for all the time afterward the size of the kth tallest queue stays much smaller than F_{k-1}. For completeness, first we give definition of F_k. Let $F_n = 0$. Given F_{k+1}, for $j = 1, 2, \ldots, k$, define

$$S_j \triangleq B\left(n-k, F_{k+1}, (j-1)(H_1 + H_2 \ldots + H_{j-1})^2\right),$$

$$H_j \triangleq \frac{(n-k)S_j^2}{\varepsilon},$$

$$F_k \triangleq H_k + S_k + \frac{P_0}{\varepsilon}.$$

The motivation for these definitions will appear in the course of the proof.

Now suppose that we are given a general adversarial queue system with n queues controlled by an $A(\omega,\epsilon,0)$-adversary and MAX-WEIGHT, such that all the initial queue sizes are at most q_0.

Suppose that for all time t, such that $P(t) < nF_1^2$. Then it directly shows that the MAX-WEIGHT protocol is stable. Now suppose that there is t_0 such that $P(t_0) \geq nF_1^2$. By choosing the smallest such t_0, we may assume that $P(t_0) \leq \max\{nq_0^2, nF_1^2 + 2\sqrt{n}F_1\}$ since if $P(t_0-1) < nF_1^2$, the change of potential between time t_0-1 and t_0 is at most $2\sqrt{n}F_1$. Note that for such t_0, the size of the tallest queue at that time is at least F_1.

Let $q_1 \geq g_2 \geq \ldots q_n = 0$ be the ordered sizes of the queues at time t_0. For $1 \leq j \leq n$, let Q_j be the corresponding jth tallest queue at time t_0. Then since $q_1 \geq F_1$ and $q_n = 0$, there exists some $1 \leq k \leq (n-1)$ such that $q_k \geq F_k$ and $q_{k+1} \leq F_{k+1}$.

Now fix one such k. We will call all the queues having size at least F_k at time t_0 *"tall queues"*, and all the other queues *"small queues"*. Then note that a packet in a small queue will never move to a tall queue by MAX-WEIGHT . Hence we can consider the set of all the small queues as a separate general adversarial queue system. We will call this queue system the *system of small queues*. We will use an inductive argument on this system of small queues to guarantee that their sizes are bounded by a constant over all time. Now, let t_1 be the first time after t_0 such that there is an injection of a packet to a tall queue or a transmission of a packet from a tall queue to a small queue. When there is such a t_1, our main argument is that during time $t_0 \leq t \leq t_1$, the sizes of the small queues stay much smaller than q_k, hence the sizes of tall queues are much bigger than those of the small queues. Hence, by Lemma 1, one injection to a tall queue or one transmission of a packet from a tall queue to a small queue creates a sufficient decrease in potential. Formally, we will prove the following Lemma.

LEMMA 5. *There is t^*, $t_0 < t^* \leq t_1 + \omega - 1$, such that $P(t^*)$ is smaller than $P(t_0)$, and during $t_0 \leq t \leq t^*$ the sizes of small queues are bounded by F_k.*

Suppose that Lemma 5 is true. Note that until the time t^*-1, the potential of all the tall queues, $P_T(t) \triangleq \sum_{i=1}^{k}(q_i^t)^2$, is non-increasing over time, and the potential of all the small queues, $P_S(t) \triangleq \sum_{i=k+1}^{n}(q_i^t)^2$, is bounded above by $(n-1)F_1^2$ since the sizes of all the small queues cannot be bigger than F_k for any time $t_0 \le t \le t^*-1$. Moreover, at time t^* we know that the total potential $P(t^*)$ becomes smaller than $P(t_0)$. So for $t_0 \le t \le t^*$, the potential $P(t)$ is bounded by

$$(n-1)F_1^2 + P(t_0) \le (n-1)F_1^2 + \max\{nq_0^2, nF_1^2 + \sqrt{n}F_1^2\}.$$

Now, again choose the first time $t \ge t^*$, if there exists such t, so that $P(t) \ge nF_1^2$, and set this time as a new t_0. Then by applying the same argument, we obtain that for all time $t \ge 0$, (1) holds. Now, in the case when there is no $t_1 > t_0$ such that at time t_1 an injection of a packet to a tall queue or a transmission of a packet from a tall queue to a small queue occurs, then for this $A(\omega, \epsilon, 0)$-adversary, the same argument that will be presented in the proof of Lemma 5 can be applied to show that the sizes of all the small queues cannot be bigger than F_k for all time $t \ge t_0$. Also the potential of tall queues are non-increasing over all time. Hence also in this case, we obtain that $P(t)$ is bounded by $(n-1)F_k^2 + P(0) \le (n-1)F_1^2 + \max\{nq_0^2, nF_1^2 + 2\sqrt{n}F_1\}$ for all $t \ge t_0$ as required in (1). So for all $t \ge 0$, (1) holds. Hence $B(n, q_0, 0)$ exists. □

PROOF OF LEMMA 5. Note that, between time t_0 and t_1, there may be some injection of packets to a small queue so that its corresponding set of edges includes some edges between tall queues. We will regard these kinds of injected packets as "bad packets" for the system of small queues, and we will call these injections "bad injections". Note that by considering these packets as bad packets, the dynamics of small queues can be thought as an independent general adversarial queue system having $n-k$ queues. Then essentially, we will show that the total number of these bad injections over all time $t_0 \le t \le t_1$ is bounded by some number which is independent of t. We now consider four cases to obtain the required t^*.

• **Case 1** If there are no bad injections to small queues for all time $t_0 \le t \le t_1$, then for all $t_0 \le t \le t_1$, the sizes of small queues are bounded by $S_1 = B(n-k, F_{k+1}, 0)$ by the induction hypothesis. Hence the potential of all the small queues at time t_1 is at most $\varepsilon H_1 = (n-k)S_1^2$. By Lemma 1 and the definition of potential, the decrease of potential due to a injection to a tall queue or a transmission from a tall queue to a small queue at time t_1 is at least $\varepsilon(F_k - S_1)$. Note also that from the definition of F_k,

$$F_k \ge H_1 + S_1 + \frac{P_0}{\varepsilon}.$$

Therefore $\varepsilon(F_k - S_1) \ge \varepsilon H_1 + P_0$, which means that the decrease of potential due to an injection to a tall queue or a transmission from a tall queue to a small queue at time t_1 is bigger than the potential of all the small queues at time t_1 plus P_0. Note that maximum possible increase of potential induced by injections during the time $[t_1, t_1+\omega-1]$ is bounded by P_0, and that all the packet movement associated with the injection to a tall queue at time t_1 occurs no later than $t_1+\omega-1$. In addition, since there was no injection to any of the tall queues during $t_0 \le t \le (t_1-1)$, the potential of the tall queues is non-increasing for $t_0 \le t < t_1$. Hence, by letting $t^* = t_1+\omega-1$, we obtain that $P(t^*) \le P(t_0)$.

We now consider the cases when there are some bad injections to small queues. Let $0 \le r_1 \le r_2 \le \ldots r_{k-1}$ be the sorted list of (q_1-q_2), $(q_2-q_3), \ldots, (q_{k-1}-q_k)$.

• **Case 2** Suppose that $r_1 > H_1$. Then by Lemma 1, any transmission between two tall queues at some time $t_0 < t \le t_1$ will make the potential decrease more than εH_1. Let t^* be the smallest time $t^* > t_0$ so that there is a transmission between two tall queues at time t^*. Note that for all time $t_0 \le t \le t^*$, the sizes of the small queues are bounded by $S_1 = B(n-k, F_{k+1}, 0)$ by the induction hypothesis and the potential of the small queues is bounded by $\varepsilon H_1 = (n-k)S_1^2$. Then from the same argument as the first case, $P(t^*) \le P(t_0)$.

• **Case 3** Suppose that there is $1 \le \ell \le k-1$ such that for all $1 \le j \le \ell$, $r_j \le H_j$, and $r_{\ell+1} > H_{\ell+1}$. We will show that in this case the potential of all the small queues is bounded by $\varepsilon H_{\ell+1}$. We may assume that bad injections to small queues induce transmissions just between neighboring tall queues. Note that the number of bad injections to small queues during some period of time is bounded by the total number of transmissions between tall queues during that period of time.

We say an edge $e_j = (Q_j, Q_{j+1})$ between two neighboring tall queues is a *tall edge* if $q_j - q_{j+1} > H_{\ell+1}$ and a *small edge* otherwise. Then, we can obtain the following Lemma, whose proof is presented after the proof of Lemma 5.

LEMMA 6. *Let $r_1, r_2 \ldots, r_\ell$ be the sizes of the small edges at time t_0 and assume that for $1 \le j \le \ell$, $r_j \le H_j$. If there is no transmission via* tall edges *for $t_0 \le t < t'$ and all the transmissions occur via small edges, then the total number of packet transmissions via* small edges *during that period of time is bounded by*

$$\ell(r_1 + r_2 \ldots + r_\ell)^2 \le \ell(H_1 + H_2 \ldots + H_\ell)^2.$$

• **Case 3-1** Suppose there are no transmissions via tall edges for all time $t_0 \le t \le t_1$. Then by Lemma 6, the total number of bad injections to the small queues during $t_0 \le t \le t_1$ is bounded by $\ell(H_1 + H_2 \ldots + H_\ell)^2$. Hence by So, for all time $t_0 \le t \le t_1$, the sizes of small queues are bounded by $S_{\ell+1} = B(n-k, F_{k+1}, \ell(H_1 + H_2 \ldots + H_\ell)^2)$ by the induction hypothesis. Hence the potential of all the small queues at time t_1 is at most

$$\varepsilon H_{\ell+1} = (n-k)S_{\ell+1}^2.$$

Therefore the potential for the tall queues is non-increasing for $t_0 \le t \le t_1$.

By Lemma 1, the decrease in potential due to the injection to a tall queue or the transmission from a tall queue to a small queue at time t_1 is at least $\varepsilon(F_k - S_{\ell+1})$. Note that from the definition of F_k,

$$\varepsilon(F_k - S_{\ell+1}) \ge \varepsilon H_{\ell+1} + P_0.$$

This implies that the potential decrease at time t_1 is more than the potential of all the small queues at t_1 plus P_0. Hence, by letting $t^* = t_1+\omega-1$, we obtain that $P(t^*) \le P(t_0)$.

• **Case 3-2** If there is a transmission via some tall edge for some time $t_0 < t \le t_1$, let t^* be the smallest such t. Then similarly, by Lemma 6, the total number of bad injections to the small queues during $t_0 \le t \le t^*$ is bounded by $\ell(H_1 + H_2 \ldots + H_\ell)^2$. Hence the sizes of the small queues during this time interval are bounded by $S_{\ell+1}$ by the induction hypothesis and the potential of all the small queues at time t^* is at most $\varepsilon H_{\ell+1}$. In addition, note that during $t_0 \le t \le t^*$, for any tall edge $e_j = (Q_j, Q_{j+1})$, q_j is non-decreasing and g_{j+1} is non-increasing. So for $t = t^*$, $q_j - q_{j+1} \ge H_{\ell+1}$. Hence, a transmission via a tall edge at time t^* will make the potential decrease by at least $\varepsilon H_{\ell+1}$, which is more than the potential of all the small queues at time t^*. The potential for the tall queues is non-increasing for $t_0 \le t < t^*$. Hence, we obtain that $P(t) \le P(t^*)$.

• **Case 4** The only remaining case is when $r_\ell \le H_\ell$ for all $1 \le \ell \le k-1$. Then by Lemma 6 and the induction hypothesis, for all time $t_0 \le t \le t_1$, the sizes of the small queues are bounded by $S_k = B\left(n-k, F_{k+1}, (k-1)(H_1 + H_2 \ldots + H_{k-1})^2\right)$. Hence the potential of all the small queues at time t_1 is at most $\varepsilon H_k = (n-k)S_k^2$. Then by Lemma 1, the decrease in potential due to the injection to a tall queue or the transmission from a tall queue to a small queue at time t_1 is at least $\varepsilon(F_k - S_k) = \varepsilon H_k + P_0$, which is more than the potential of all the small queues at time t_1 plus P_0. The potential of the tall queues is non-increasing for $t_0 \le t < t_1$. Hence, by letting $t^* = t_1 + \omega - 1$, we obtain that $P(t^*) \le P(t_0)$.

Note that in all the above four cases, for $t_0 \le t \le t^*$, the sizes of the small queues are bounded by $S_j + \omega n$ for some $1 \le j \le k$. Therefore they are bounded by F_k. □

Proof of Lemma 6. Let $e_{j_1}, e_{j_2}, \ldots e_{j_\ell}$ be the set of small edges, where $j_1 < j_2 < \ldots < j_\ell$. For $1 \le i \le \ell$, let s_i be $(q_{j_i} - q_{j_i+1})$. Hence $\{s_i\}_{1\le i\le\ell}$ is a permutation of $\{r_i\}_{1\le i\le\ell}$.

Recall that the sizes of the queues at time t_0 are non-increasing with respect to their indices. Moreover, note that if $j_{i+1} - j_i \ge 2$ for some i, then any packet p that was originally located at Q_m, with $m \le j_i + 1$ cannot move to Q_{j_i+2} for all time $t_0 \le t \le t'$. Hence we can consider each subset of consecutive small edges separately. For example if $j_1 \ldots j_\ell$ are 2,3,4,7,8, then we will consider 2,3,4 and 7,8 separately. Suppose that $j_1, j_2 \ldots, j_m$ are consecutive integers. Then, consider the following potential function grounded at level q_{j_m+1}.

$$R(t) = \sum_{i=1}^{m} (q_{j_i}^t - q_{j_m+1}^t)^2.$$

Then, $R(t)$ strictly decreases at least by 1 for each packet transmission via one of the edges $e_{j_1}, \ldots e_{j_m}$. Hence the total number of packet transmission via $e_{j_1}, \ldots e_{j_m}$ for time $t_0 \le t \le t'$ is bounded by $R(t_0)$. Note that $R(t_0) \le m(s_1 + s_2 + \ldots + s_m)^2$. A similar argument holds for other consecutive values of j_i's, and the sum of the $R(t_0)$ values for each of these consecutive small edges is at most $\ell(s_1 + s_2 \ldots + s_\ell)^2 \le \ell(H_1 + H_2 \ldots + H_\ell)^2$. Hence the total number of transmissions via small edges during time $t_0 \le t \le t'$ is bounded by $\ell(H_1 + H_2 \ldots + H_\ell)^2$. □

5. EXTENSIONS TO ARBITRARY LINK CAPACITIES AND PACKET SIZES

A similar argument to the proof of Theorem 3 can be applied to show that Max-Weight is stable in the arbitrary link capacity and packet size case. In this case, let t_1 be the first time after t_0 such that the total data injected to tall queues during $[t_0, t_1]$ plus the total data transmitted from tall queues to small queues during $[t_0, t_1]$ is at least 1. We consider those injections and transmissions as bad injections to small queues. Then using a similar inductive argument, we can show that for all $t \ge 0$, $P(t)$ is bounded above by some value depending on $P(0)$ and independent of t. This shows the stability of Max-Weight for this case.

6. Max-Weight UNDER CRITICAL LOAD IN STATIC NETWORKS

All of the results in the previous sections have been for dynamic networks under subcritical load. In this section we study the stability of Max-Weight when the load is critical. We are unfortunately unable to resolve this issue in the most general case of dynamic graphs with adversarially generated traffic. However, we show that Max-Weight *is* stable for the special case in which the network is static and the traffic generated is identical in each time slot. We believe that this result is of interest for two reasons. First, as far as we are aware it is the first result to show stability of Max-Weight in a critically loaded context. Second, it allows us to tighten the analysis of the running time bound of the Awerbuch-Leighton algorithm for multicommodity flow. In particular, for a fixed instance of the problem it allows us to approximate the value of the optimal solution to within a factor $(1-\varepsilon)$ in time $O(\frac{1}{\varepsilon})$. We discuss this in more detail in Section 7.

The situation that we study in this section is the following. We have a fixed set of commodities indexed by i. Each commodity has a source node s_i, a destination node d_i and an injection rate ρ_i. At each time step data of size ρ_i is injected at node s_i with destination d_i. We assume in addition that each edge (v, u) has a fixed capacity c_{uv}. The injection rates and edge capacities are such that the network is critically loaded. In particular we assume that the injections conform to the definition of an $A(1, 0)$-adversary. For convenience, we also assume that at time 0, all queues are empty.

We remark that there is a close connection between this scheduling problem and the following multicommodity flow (MCF) problem,

$$\max \quad \lambda$$

$$\text{subject to} \quad \sum_i x_{vu}^i \le c_{vu} \quad \forall v, u$$

$$\sum_u (x_{vu}^i - x_{uv}^i) = \begin{cases} \lambda f_i & v = s_i \\ -\lambda f_i & v = d_i \\ 0 & \text{o.w.} \end{cases} \quad \forall i.$$

Let λ^* be the optimal value of this problem. Without loss of generality, all of the c_{uv} and f_i are integers (by scaling). Moreover, it is well known (see e.g. [12]) that if L is the number of bits required to represent the problem then we can scale all capacities by at most $poly(L)$ so that λ^* is an integer multiple of four lying between 0 and 2^L.

It is immediate from the definition of criticality that if $\rho_i = \lambda f_i$, the network is subcritically loaded if and only if $\lambda < \lambda^*$, critically loaded if and only if $\lambda = \lambda^*$ and supercritically loaded if and only if $\lambda > \lambda^*$. Moreover, the following lemma states that if the Max-Weight algorithm keeps the system stable then we can derive a solution to the MCF problem.

LEMMA 7. *Suppose that we inject data into each commodity at rate λf_i. If we run the* MAX-WEIGHT *algorithm for time n^2B/ε and the total amount of data in any queue is never more than B, then from the evolution of the system we can derive a solution to the MCF problem of value $(1-\varepsilon)\lambda$.*

PROOF. For each edge (u,v), let X^i_{uv} be the number of commodity i packets that traversed edge (u,v) during the algorithm *and that reached their destination.* (We can record this number by having each packet keep track of how many times it crosses each edge and then recording these numbers whenever a packet reaches its destination.) Let $x^i_{uv} = X^i_{uv}\varepsilon/n^2B$. Since data of size at most c_{uv} crosses edge (u,v) during each time step, we must have $\sum_i x^i_{uv} \le c_{vu}$. Since at the time n^2B/ε the total number of packets that can remain in the network is n^2B and since we are only counting the number of packets that reach their destination, the x^i_{uv} variables must represent a flow of size $(\varepsilon/n^2B)(n^2B((\lambda f_i/\varepsilon)-1)) \ge \lambda f_i(1-\varepsilon)$. In other words, we have derived a solution to the multicommodity flow problem of value at least $(1-\varepsilon)\lambda$. $\Box$

We now present the main result of this section.

THEOREM 8. *For static networks and static injection patterns, the* MAX-WEIGHT *protocol is stable even at critical loads (i.e. when $\varepsilon = 0$.)*

We remark that for ease of analysis we actually consider a slightly different version of the MAX-WEIGHT algorithm than the one described in the Introduction. Specifically, we no longer require that $q^t_{v,d} - q^t_{u,d}$ is greater than $\Delta \cdot c_{\max}$ when deciding whether to send data from $Q_{v,d}$ to $Q_{u,d}$.

PROOF. We utilize techniques that were introduced in [19, 20] to analyze the performance of MAX-WEIGHT under "heavy traffic". Let $r_{vu,d}(t)$ represent the number of packets transmitted from $Q_{v,d}$ to $Q_{u,d}$ during time step t. Let Υ be the set of all vectors that represent the feasible transmissions of packets between queues, i.e.,

$$\Upsilon = \{(\ldots, r_{vu,d}, \ldots) : \sum_d r_{vu,d} \le c_{vu}, r_{vu,d} \in \mathbb{Z}\}.$$

For any vector $(\ldots, r_{vu,d}, \ldots)$ in Υ, let $\omega_{v,d} = \sum_u (r_{vu,d} - r_{uv,d})$, i.e. $\omega_{v,d}$ is equivalent to the change $q^t_{v,d} - q^{t+1}_{v,d}$ when the packets move according to the $r_{vu,d}$ and $q^t_{v,d}$ is sufficiently large. Let Ω be the set of all such "service" vectors. Note that if the queues are continually served according to $\omega = \{\ldots\omega_{v,d}\ldots\} \in \Omega$ then the long-term drift of queue $Q_{v,d}$ is $a_{v,d} - \omega_{v,d}$, where $a_{v,d}$ is the exogenous rate at which data is injected into queue $Q_{v,d}$ (i.e. $a_{v,d} = \sum_{i \in D(v)} \rho_i$, where $D(v)$ is the set of demands whose source node is v). We define $\bar{\Omega} = conv(\Omega)$ and $V = conv\{\omega - a : \omega \in \Omega\}$ (where $conv\{\cdot\}$ denotes convex hull). From now on we shall work with vectors that are indexed by each queue. Note that such vectors have length n^2.

The geometry of V

In order to prove our stability result we shall need to consider the geometry of the rate region V and some other related regions. First note that V is convex polyhedral set. The fact that the system is critically loaded implies that the origin is on the boundary of V. We use C to denote the "normal cone" to V at the origin, defined by

$$C = \{\gamma : (\omega - a) \cdot \gamma \le 0 \quad \forall \omega \in \Omega\}.$$

Without loss of generality, we can assume that for each queue, there exists a demand i such that the queue is reachable from the source node of demand i. This means that for a sufficiently small δ, for any vector x that lies in the negative orthant such that $||x|| \le \delta$, we can make the queue-size vector grow with positive drift $-x$ and so $x \in V$. This in turn implies,

LEMMA 9. *The normal cone C lies in the positive orthant.*

Let ν be any unit vector that lies in the relative interior of C, i.e. the interior of C with respect to the smallest subspace of Euclidean space that contains C. The vector ν has the property that if $\omega \in \bar{\Omega}$ and $\nu \cdot (\omega - a) = 0$ then $\gamma \cdot (\omega - a) = 0$ for all $\gamma \in C$. Let $\alpha = \min\{\nu \cdot (a - \omega) : \omega \in \Omega, \nu \cdot (a - \omega) > 0\}$. Since Ω is finite this minimum exists.

We now let W equal the set of vectors in V that are orthogonal to a vector in C and cannot be extended while remaining in V, i.e. $W = \{\omega - a : \omega \in \bar{\Omega}, \exists \gamma \in C\ \gamma \cdot (\omega - a) = 0, \forall \delta > 0\ (1+\delta)(\omega - a) \notin V\}$. Let $\mu = \min_{\omega - a \in W} ||\omega - a||$. The polyhedral nature of V means that $\mu > 0$. Let $q(t)$ represent the vector of queue sizes, let $||\cdot||$ denote Euclidean distance and let $q^*(t)$ be the closest point to $q(t)$ that lies in the normal cone C. The distance from $q(t)$ to the cone is defined to be $||q(t) - q^*(t)||$. By projecting the vector $q(t) - q^*(t)$ onto V it is not hard to see that,

LEMMA 10. $||q(t) - q^*(t)|| = \max_{\omega - a \in W}(q(t) \cdot (\omega - a))/||\omega - a||$.

Stability proof

We now return to the proof of stability. Let $\omega(t)$ be the vector in Ω that represents how the queues change at time t under the MAX-WEIGHT algorithm. Let $Z_1 = n^4(c_{\max})^2$. (Recall that $c_{\max} = \max c_{uv}$ equals the maximum link capacity in the network.

LEMMA 11. $q(t) \cdot \omega(t) \ge \max_{\omega \in \bar{\Omega}} q(t) \cdot \omega - Z_1$.

PROOF. First note that since the maximum occurs at a vertex of $\bar{\Omega}$, we have that $\max_{\omega \in \bar{\Omega}} q(t) \cdot \omega = \max_{\omega \in \Omega} q(t) \cdot \omega$, i.e. we can focus on integral service vectors that could actually be realized by a scheduling algorithm. Recall that if $r_{uv,d}$ represents the amount of data that is scheduled from $Q_{v,d}$ to $Q_{u,d}$ at time t then $\omega_{v,d} = \sum_u (r_{vu,d} - r_{uv,d})$ and so

$$q(t) \cdot \omega(t) = \sum_{uv,d} r_{uv,d}(q^t_{u,d} - q^t_{v,d}).$$

Therefore we can view the MAX-WEIGHT algorithm as trying to maximize the dot-product $q(t) \cdot \omega(t)$. However, the reason that we do not always utilize the service vector $\omega^* = \arg\max_{\omega \in \Omega} q(t) \cdot \omega$ is that we also have a constraint that the queues never go below zero, i.e. $\sum_v r_{uv,d}(t) \le q^t_{u,d}$. (Note that we are considering a model in which data is removed from queues at the beginning of the time step and then added to queues at the end of the time step.) However, one feasible solution involves taking the service vector ω^* and then creating a new service vector ω' by ignoring any queue for which $\sum_v r_{uv,d} > q^t_{u,d}$. Since $\sum_v r_{uv,d} \le nc_{\max}$

we have that $q(t)\cdot\omega' \geq q(t)\cdot\omega^* - n^2(nc_{\max})^2$. Therefore, since $Z_1 = n^4(c_{\max})^2$ and $\omega(t)$ is the feasible service vector that achieves the maximum dot-product with $q(t)$ we have,

$$q(t)\cdot\omega(t) \geq q(t)\cdot\omega' \geq q(t)\cdot\omega^* - Z_1.$$

□

Let $Z_2 = 4n^4(c_{\max})^2$. The next lemma is immediate from the definitions,

LEMMA 12. $||\omega(t) - a||^2 \leq Z_2$.

We now show that if $q(t)$ is sufficiently far from the cone C, then $q(t+1)$ has to be closer to C. (Cf. [20].)

LEMMA 13. $||q(t+1) - q^*(t+1)||^2 \leq ||q(t) - q^*(t)||^2 - \mu||q(t) - q^*(t)|| + Z_1 + Z_2$.

PROOF. An intuitive explanation of this result is that $q(t+1)\cdot q^*(t) \geq q(t)\cdot q^*$ and if $q(t)$ is far from C then $||q(t+1)|| < ||q(t)||$. Hence $q(t+1)$ is shorter than $q(t)$ but its dot-product with $q^*(t)$ is larger and so it must be closer to the cone.

More formally, we know by the definition of $q^*(t+1)$ that $||q(t+1) - q^*(t+1)|| \leq ||q(t+1) - q^*(t)||$. From the manner in which the queue vector evolves we have,

$$\begin{aligned} & ||q(t+1) - q^*(t)||^2 \\ = \; & ||q(t) - q^*(t)||^2 - q(t)\cdot(\omega(t) - a) + \\ & q^*(t)\cdot(\omega(t) - a) + ||\omega(t) - a||^2. \end{aligned}$$

By the definition of C we know $q^*(t)\cdot(\omega(t) - a) \leq 0$. By Lemma 11, $q(t)\cdot(\omega(t) - a) \geq \max_{\omega\in\bar\Omega} q(t)\cdot(\omega - a) \geq \max_{\omega-a\in W} q(t)\cdot(\omega - a)$. By Lemma 10 there exists a vector $\omega - a \in W$ such that $q(t)\cdot(\omega - a) = ||\omega - a||||q(t) - q^*(t)|| \geq \mu||q(t) - q^*(t)||$. Therefore $q(t)\cdot(\omega(t) - a) \geq \mu||q(t) - q^*(t)|| - Z_1$. By Lemma 12, $||\omega(t) - a||^2 \leq Z_2$. Putting all these inequalities together,

$$\begin{aligned} & ||q(t+1) - q^*(t+1)||^2 \\ \leq \; & ||q(t+1) - q^*(t)||^2 \\ \leq \; & ||q(t) - q^*(t)||^2 - \mu||q(t) - q^*(t)|| + Z_1 + Z_2. \end{aligned}$$

□

LEMMA 14. *For all* t, $||q(t) - q^*(t)||^2 \leq Z_1 + Z_2 + (Z_1 + Z_2)^2/\mu^2$.

PROOF. By the previous lemma, if $||q(t) - q^*(t)|| \geq (Z_1 + Z_2)\mu$ then $||q(t+1) - q^*(t+1)||^2 \leq ||q(t+1) - q^*(t+1)||^2$ and if $||q(t) - q^*(t)|| < Z_1 + Z_2$ then $||q(t+1) - q^*(t+1)||^2 \leq ||q(t) - q^*(t)||^2 + Z_1 + Z_2$. The result follows. □

LEMMA 15. $q(t+1)\cdot\nu \geq q(t)\cdot\nu$. *Moreover, if* $q(t+1)\cdot\nu > q(t)\cdot\nu$ *then* $q(t+1)\cdot\nu \geq q(t)\cdot\nu + \alpha$.

PROOF. Follows from the definition of C, the definition of α and the fact that

$$\begin{aligned} q(t+1)\cdot\nu &= (q(t) + a - \omega(t))\cdot\nu \\ &= q(t)\cdot\nu + (a - \omega(t))\cdot\nu. \end{aligned}$$

□

Following [19] we now define another useful quantity, the *invariant point*. This is denoted by $q^{**}(t)$ and is defined to be the vector q that minimizes $||q||^2$ subject to the constraints $q\cdot\gamma \geq q(t)\cdot\gamma$ for all $\gamma \in C$.

LEMMA 16. $q^{**}(t) = q^*(t)$.

PROOF. If $q(t)$ lies in C then the result is immediate. If $q(t) \notin C$ then the result follows from the fact that for any $\gamma \in C$, $(q(t) - q^*(t))\cdot(\gamma - q^*(t)) \leq 0$. In particular, this property implies that $q^*(t)$ does satisfy all constraints $q\cdot\gamma \geq q(t)\cdot\gamma$, and for any vector q satisfying these constraints, $q\cdot q^*(t) \geq q(t)\cdot q^*(t) \geq \|q^*(t)\|^2$. This in turn implies that $||q||^2 \geq ||q^*(t)||^2$. □

LEMMA 17. *If* $q^*(t+1) \neq q^*(t)$ *then* $q(t+1)\cdot\nu \geq q(t)\cdot\nu + \alpha$.

PROOF. If $q^*(t+1) \neq q^*(t)$ then since $q^*(t) = q^{**}(t)$, there must exist some $\gamma \in C$ such that $q(t+1)\cdot\gamma > q(t)\cdot\gamma$. Recall that because ν is in the relative interior of C, if $\nu\cdot(a - \omega(t)) = 0$ then $\gamma\cdot(a - \omega(t)) = 0$ which implies that $q(t+1)\cdot\gamma = q(t)\cdot\gamma$. Therefore, we must have that $q(t+1)\cdot\nu > q(t)\cdot\nu$ and so by Lemma 15, $q(t+1)\cdot\nu \geq q(t)\cdot\nu + \alpha$. □

We can now prove Theorem 8. By an argument almost identical to the proof of Lemma 13, $||q(t+1)||^2 \leq ||q(t)||^2 + (Z_1 + Z_2)$. Therefore $||q(t)|| \leq \sqrt{t(Z_1 + Z_2)}$ which implies that $q(t)\cdot\nu \leq \sqrt{t(Z_1 + Z_2)}$ since ν is a unit vector.

Let S be the maximum number of integer lattice points that can lie in an n^2 dimensional ball of radius $(Z_1 + Z_2 + (Z_1 + Z_2)^2/\mu^2)^{1/2}$. Let T be an integer such that $T\alpha/S > \sqrt{T(Z_1 + Z_2)}$, e.g. $T = 1 + \lceil S^2(Z_1 + Z_2)/\alpha^2\rceil$. Since $q(T)\cdot\nu \leq \sqrt{T(Z_1 + Z_2)}$ and whenever $q(t)\cdot\nu$ increases it must increase by at least α, there must be a subinterval of length S in the interval $[0, T]$ during which $q(t)\cdot\nu$ remains constant. By Lemma 17, $q^*(t)$ remains constant during this interval. However, since $q(t)$ remains within a ball of radius $(Z_1 + Z_2 + (Z_1 + Z_2)^2/\mu^2)^{1/2}$ centered at $q^*(t)$ during this interval, and this ball has at most S integer points, the $q(t)$ vector must repeat itself during the interval. Since the system is deterministic, the vector must continue to cycle within this ball. Hence for all t the queue vector satisfies $||q(t)|| \leq B := \sqrt{T(Z_1 + Z_2)}$. This in turn implies that $q^t_{v,d} \leq B$ for all t, v, d.

Subcritically loaded case. As mentioned in the Introduction, bounds on queue size in the case that the network is subcritically loaded are well-known. We now give an explicit bound that will be useful in the next section and which can be derived using an argument similar to the proof of Lemma 1.

LEMMA 18. *Suppose that the network is subcritically loaded, i.e.* $\varepsilon > 0$. *Then under* MAX-WEIGHT *the maximum queue size is at most* $Z_3(\varepsilon) := 2n(Z_1 + Z_2)/\varepsilon$.

7. APPLICATIONS TO MULTICOMMODITY FLOW

We now consider the multicommodity flow problem presented at the beginning of Section 6 and show how the stability of MAX-WEIGHT in static networks when $\varepsilon = 0$ implies a running time for fixed instances that is an improvement on the analysis given by Awerbuch and Leighton in [7, 8]. Let λ^* be the optimal value of problem. If λ^* is known they show how to use the MAX-WEIGHT protocol to obtain a solution of value at least $(1-\varepsilon)\lambda^*$ in time $O(1/\varepsilon^2)$. We improve this running time to $O(1/\varepsilon)$. Moreover, we then show that by using binary search we can find a solution of value at least $(1-\varepsilon)\lambda^*$ in time $O(\frac{1}{\varepsilon})$, even if we do not know the value of λ^*. We remark that many of the well-known iterative algorithms for multicommodity flow (e.g. Garg-Könemann [11], Klein et al. [13] and references therein) have a running time of the form $\Omega(1/\varepsilon^2)$. Moreover, Klein and Young showed in [14] that for a large class of such algorithms, the running time is $\Omega(1/\varepsilon^2)$. However, their example is not directly applicable to our result since the size of their instances grows with ε. We are more interested in the dependence of the running time on ε for fixed instances. In [9], Bienstock and Iyengar present an iterative algorithm with a running time of $O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$. However, their algorithm is more complex than the MAX-WEIGHT protocol that is the focus of our paper.

THEOREM 19. *Suppose that λ^* is known. Then the* MAX-WEIGHT *protocol finds a solution to the MCF problem of value at least $(1-\varepsilon)\lambda^*$ in time $O(1/\varepsilon)$.*

PROOF. We run the MAX-WEIGHT protocol with injection rate $\rho_i = \lambda^* f_i$ for all i. The analysis in Section 6 implies that $q^t_{v,d} \leq B$ for all t, v, d. By Lemma 7 after n^2B/ε steps we have a solution to the MCF problem of value at least $(1-\varepsilon)\lambda^*$. □

We remark that although the bound B does not depend on ε, it is difficult to calculate explicitly since it depends on the quantities α and μ. However, we do not need to calculate B in advance. After each time step we can calculate the value of the current solution since it depends solely on how much data has reached its destination. Therefore the "stopping condition" for the algorithm is when we have a solution of value at least $(1-\varepsilon)\lambda^*$. The implication of Theorem 19 is that this condition is satisfied in time $O(1/\varepsilon)$.

Now suppose that λ^* is not known *a priori*. In this case we can still obtain a solution of value at least $(1-\varepsilon)\lambda^*$ in time $O(1/\varepsilon)$. Recall that L is the number of bits required to represent the MCF problem. In the full version of the paper we show,

THEOREM 20. *We can construct a binary search procedure using the* MAX-WEIGHT *protocol that determines λ^* in time $O(LZ_3(2^{-L}))$. By Theorem 19 this implies that for a fixed instance of the MCF problem, the* MAX-WEIGHT *protocol can find a solution of value at least $(1-\varepsilon)\lambda^*$ in time $O(1/\varepsilon)$.*

PROOF. By the remarks at the beginning of this section, we can assume that λ^* is an integer multiple of four lying between 0 and 2^L. We run a binary search procedure to find the maximum value of $k \in \mathbb{Z}$ such that if we run the MAX-WEIGHT protocol with injection rates $\rho_i = (4k+2)f_i$, after time $Z_3(2^{-L})$ we have found a solution to the MCF problem of value at least $(1-2^{-L})(4k+2)$. Let k^* be the maximum such k. Since we have found a solution of value $(1-2^{-L})(4k+2) \geq 4k+1$, we know that $\lambda^* \geq 4k^*+4$ by our assumption that λ^* is a multiple of four. However, if $\lambda^* \geq 4k^*+8$ then by Lemma 18 if we run the MAX-WEIGHT protocol with injection rates $\rho_i = (4(k+1)+2)f_i$, after time $Z_3(2^{-L})$ we should have found a solution to the MCF problem of value at least $(1-2^{-L})(4(k+1)+2)$. This contradicts the optimality of k^*. Hence $\lambda^* = 4k^*+4$. By the properties of binary search the number of times that we must run the MAX-WEIGHT protocol to find λ^* is $O(L)$ and so the total running time for this procedure is $O(LZ_3(2^{-L})$.

By Theorem 19 we now can run the MAX-WEIGHT protocol with injection rates $\rho_i = \lambda^* f_i$ to obtain a solution to the MCF problem of value at least $(1-\varepsilon)\lambda^*$ in time $O(1/\varepsilon)$. For a fixed MCF instance, $LZ_3(2^{-L})$ is a constant and so the total running time taken to find the solution is $O(1/\varepsilon)$. □

8. CONCLUSIONS

In this paper we have shown stability of the MAX-WEIGHT routing and scheduling protocol for the case of adversarial multicommodity traffic in dynamic networks at subcritical loads and the case of static multicommodity traffic in static networks at critical loads. As far as we are aware, the stability of MAX-WEIGHT has not been previously addressed in these contexts. A number open questions remain. First, is the MAX-WEIGHT protocol stable at critical loads when either the network or the traffic patterns change over time? The analysis of Section 6 cannot be applied here since the rate region V and the normal cone C could change over time. Second, for static networks at critical loads, can we obtain a bound on queue size that is polynomial in the size of the network. (Note that the bound obtained in Section 6 depends on the quantity S that is exponential in the size of the network.) We believe that it is of interest to resolve this question since it will determine the extent to which our $O(1/\varepsilon)$ bound on the time required to solve the multicommodity flow problem is better in practice than the previous $O(1/\varepsilon^2)$ bound due to Awerbuch and Leighton.

9. REFERENCES

[1] W. Aiello, E. Kushilevitz, R. Ostrovsky, and A. Rosen. Adaptive packet routing for bursty adversarial traffic. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 359 – 368, Dallas, TX, May 1998.

[2] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting. CDMA data QoS scheduling on the forward link with variable channel conditions. *Bell Labs Technical Memorandum*, April 2000.

[3] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting. Providing quality of service over a shared wireless link. *IEEE Communications Magazine*, February 2001.

[4] M. Andrews and L. Zhang. Scheduling over a time-varying user-dependent channel with applications to high speed wireless data. In *Proceedings of the 43nd Annual Symposium on Foundations of Computer Science*, Vancouver, Canada, November 2002.

[5] E. Anshelevich, D. Kempe, and J. Kleinberg. Stability of load balancing algorithms in dynamic adversarial systems. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, Montreal, Canada, May 2002.

[6] B. Awerbuch, P. Berenbrink, A. Brinkmann, and C. Scheideler. Simple routing strategies for adversarial systems. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, pages 158 – 167, Las Vegas, NV, October 2001.

[7] B. Awerbuch and T. Leighton. A simple local-control approximation algorithm for multicommodity flow. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, pages 459–468, 1993.

[8] B. Awerbuch and T. Leighton. Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 487–496, 1994.

[9] D. Bienstock and G. Iyengar. Solving fractional packing problems in $O^*(1/\varepsilon)$ iterations. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 146–155, Chicago, IL, 2004.

[10] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length based scheduling and congestion

control. In *Proceedings of IEEE INFOCOM '05*, Miami, FL, March 2005.

[11] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 300 – 309, Palo Alto, CA, November 1998.

[12] M. Goemans. Lecture notes on linear programming.

[13] P. Klein, S. Plotkin, C. Stein, and E. Tardos. Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts. *SIAM Journal on Computing*, 23(3):466–487, 1994.

[14] P. Klein and N. Young. On the number of iterations for Dantzig-Wolfe optimization and packing-covering approximation algorithms. In *Integer Programming and Combinatorial Optimization*, pages 320–327, 1999.

[15] N. W. McKeown, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. In *Proceedings of IEEE INFOCOM '96*, pages 296 – 302, San Francisco, CA, March 1996.

[16] S. Muthukrishnan and R. Rajaraman., An adversarial model for distributed dynamic load balancing. In *Proceedings of the 10th ACM Symposium on Parallel Algorithms and Architectures*, pages 47–54, 1998.

[17] M. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. In *Proceedings of IEEE INFOCOM '05*, Miami, FL, March 2005.

[18] M. Neely, E. Modiano, and C. Rohrs. Power and server allocation in a multi-beam satellite with time varying channels. In *Proceedings of IEEE INFOCOM '02*, New York, NY, June 2002.

[19] A. Stolyar. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Bell Labs Technical Memorandum*, 2002.

[20] A. Stolyar. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Annals of Applied Probability*, 14(1):1–53, 2004.

[21] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, 50(4):401–457, 2005.

[22] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936 – 1948, December 1992.

[23] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory*, 30:466 – 478, 1993.

Appendix: Proof of Lemma 1

LEMMA 1. *Consider the* MAX-WEIGHT(β) *protocol in adversarial dynamic graphs with multicommodity demands for* $\varepsilon > 0$. *Then for each injection of packet* p *at time* t *having size* ℓ_p, *we can associate this injection with packet movements along its corresponding simple path so that the sum of potential changes due to these movements are at least* $-\frac{\varepsilon}{1-\varepsilon}\ell_p(\beta+1)q^\beta + \ell_p O(q^{\beta-1})$, *where* q *is the height of the queue where the packet is injected. Moreover, all these changes occur in a window of size* ω. *Hence there is a constant* q^* *depending on* n, ω *and* ε, *so that if* $q \geq q^*$ *the sum of potential changes due to the injection is less than* $-\varepsilon\ell_p q^\beta$.

Recall that from the definition 1, the adversary can associate with each packet $p \in I^t$, a simple path Γ_p from s_p to d_p such that for all $e \in E$,

$$\sum_{p \in I^{[t,t+\omega-1]}, e \in \Gamma_p} \ell_p \leq (1-\varepsilon) \sum_{t'=t}^{t+\omega-1} c_e(t').$$

For each packet $p \in I^t$ we will associate some portion of capacities of edges in Γ_p, $c_p = \{d_{p,e}(t') | e \in \Gamma_p, t' \in [t-\omega+1, t+\omega-1]\}$ as follows. Let W be the time interval $[j\omega, (j+1)\omega - 1]$ for some integer $j \geq 0$. Let $p_1, \ldots p_m$ be the packets in I^W. We know that

$$\sum_{i=1, e \in \Gamma_{p_i}}^{m} \ell_{p_i} \leq (1-\varepsilon) \sum_{t' \in W_j} c_e(t').$$

First, for p_1 and for each $e \in \Gamma_{p_1}$, we can define $d_{p_1,e}(t')$ for each $t' \in W_j$ so that

$$0 \leq d_{p_1,e}(t') \leq c_e(t') \text{ and } \sum_{t' \in W_j} d_{p_1,e}(t') = \frac{\ell_{p_1}}{1-\varepsilon}.$$

Then, for each $e \in E$ and $t \in W_j$, let $c_e^1(t') = c_e(t') - d_{p_1,e}(t')$. Then we obtain that

$$\sum_{i=2, e \in \Gamma_{p_i}}^{m} \ell_{p_i} \leq (1-\varepsilon) \sum_{t' \in W_j} c_e^1(t').$$

Similarly for p_2 and for each $e \in \Gamma_{p_1}$ we can define $d_{p_2,e}(t')$ for each $t' \in W_j$ so that

$$0 \leq d_{p_2,e}(t') \leq c_e^1(t') \text{ and } \sum_{t' \in W_j} d_{p_2,e}(t') = \frac{\ell_{p_2}}{1-\varepsilon}.$$

By continuing this process, we can define $d_{p_i,e}(t')$ for each $e \in \Gamma_{p_i}$ and $t' \in W_j$ so that

$$0 \leq d_{p_i,e}(t') \leq c_e^{i-1}(t') \text{ and } \sum_{t' \in W_j} d_{p_i,e}(t') = \frac{\ell_{p_i}}{1-\varepsilon}.$$

Recall that $c_{\max}$ is the maximum possible edge capacity. Consider an edge $e = (v,u) \in E$ and suppose that e has capacity $c_e(t)$ at time t, and $q_{v,d}^t \geq q_{u,d}^t + c_e(t)$. Then the potential change due to transmission via e at time t is

$$(q_{u,d}^t + \ell_p)^{\beta+1} - (q_{u,d}^t)^{\beta+1} + (q_{v,d}^t - \ell_p)^{\beta+1} - (q_{v,d}^t)^{\beta+1}$$

$$= c_e(t)(\beta+1)\left((q_{u,d}^t)^\beta - (q_{v,d}^t)^\beta\right) + c_e(t) O\left((q_{u,d}^t)^{\beta-1} + (q_{v,d}^t)^{\beta-1}\right).$$

Note that this is also true when $|q_{u,d}^t - q_{v,d}^t| < c_e(t)$. Hence, when a $d_{p,e}(t)$ amount of edge capacity of e at time t is assigned to an injected packet p, we can consider that a $d_{p,e}(t)(\beta+1)\left((q_{u,d}^t)^\beta - (q_{v,d}^t)^\beta\right) + d_{p,e}(t) O\left((q_{u,d}^t)^{\beta-1} + (q_{v,d}^t)^{\beta-1}\right)$ amount of potential change is induced by the packet p.

Now, for $t, t' \in W$, $|q_{u,d}^t - q_{u,d}^{t'}| \leq n c_{\max}\omega$ since at each time slot, at most $c_{\max}$ amount of packet can move along an edge connecting u. Hence by considering ω and n and $c_{\max}$ as constants, we obtain that for any $t, t' \in W$,

$$(q_{u,d}^{t'})^\beta = (q_{u,d}^t)^\beta + O\left((q_{u,d}^t)^{\beta-1}\right).$$

Suppose that a packet p with capacity ℓ_p is injected at a node v_0 at time $t_0 \in W$. Let $v_0, v_1 \ldots v_m = d$ be the corresponding simple path given by adversary. Then, the increase of potential due to the direct injection of p is $\ell_p(\beta+1)(q_{v_0,d}^{t_0})^\beta + \ell_p O((q_{v_0,d}^{t_0})^{\beta-1})$. Hence the total change of potential induced by this injection is,

$$\ell_p(\beta+1)(q_{v_0,d}^{t_0})^\beta + \ell_p O\left((q_{v_i,d}^{t_0})^{\beta-1}\right) -$$

$$\sum_{i=0}^{m-1} \frac{\ell_p(\beta+1)}{1-\varepsilon} \left| (q_{v_i,d}^{t_0})^\beta - (q_{v_{i+1},d}^{t_0})^\beta + O\left((q_{v_i,d}^{t_0})^{\beta-1} + (q_{v_{i+1},d}^{t_0})^{\beta-1}\right) \right|$$

$$\leq -\frac{\varepsilon}{1-\varepsilon}\ell_p(\beta+1)(q_{v_0,d}^{t_0})^\beta + \ell_p O\left((q_{v_0,d}^{t_0})^{\beta-1}\right).$$

Hence there is a constant q^*, depending on n, ω and ε, so that if $q \geq q^*$ the sum of potential changes due to the injection is less than $-\varepsilon\ell_p q^\beta$.