Abstract—We seek to understand proportional fairness in equilibrium in the presence of both applications with multiple flows (MFAs) and traditional applications with single flows (SFAs). Proportional fairness is shown to be applicable to TCP variants such as TCP Reno, TCP Vegas, and FAST TCP. We consider the competing scenario where an SFA with proportionally fair sharing TCP is competing with an MFA that invokes a number of parallel flows, also with proportional fair sharing TCP. We present analytical results for a linear network, and show the extent of the quantitative gain of an MFA can over an SFA regardless of whether it is connecting the far end of a linear network or just the adjacent nodes.

I. INTRODUCTION

Most current Internet applications use a single TCP flow for connections. On the other hand, there has been a growing interest in applications with multiple parallel TCP flows or applications that can use multipath TCP [4], [8].

In this work, we seek to understand the impact of applications with multiple flows (‘Multiple-Flow-based Application,’ or MFA, in short) when competing with traditional applications with single flows (‘Single-Flow-based Application,’ or SFA, in short). It is commonly believed that MFAs have an unfair advantage over SFAs. Here, we seek to answer the question: what is the extent of MFAs unfair advantage over SFAs? We use the following simplifying assumption to understand this issue: 1) An MFA has the same source and destination, 2) Both MFAs and SFAs are either based on TCP Vegas [2], FAST TCP [15], or TCP Reno [6].

To elaborate, in the first assumption, we assume that MFAs have the same source and destination routers for a particular session. For example, they could be connecting the same end hosts. It is also possible that a client host is connected to a data center through the same source and destination routers, while the data center may have multiple hosts serving the MFA to the client. Clearly, there may be internal performance differences between different hosts within a data center serving an MFA.

However, we consider the performance differences (such as latency) within the data center to be negligible compared to the end-to-end performance for the application between the client and the data center.

Our work focuses on proportional fairness; the notion of proportional fairness in the log-form was presented in [7], while the same notion in the product form was presented earlier in [11]. The second assumption is related to proportional fairness of TCP. There are several variants of TCP. We assume that both MFAs and SFAs follow either TCP Vegas [2] or FAST TCP [15], or TCP Reno [6]. We restrict this analysis to these variants of TCP because they have been shown to be proportionally fair [9], [15]. Proportional fairness has also been analyzed in [3], [12], [13] for SFAs. Without specifically focussing on TCP, the impact of fairness was addressed in [1].

We focus our work for a linear network that avoids any house effects due to routing [14], as there is only a single path between a source and a destination. Thus, an application with multiple parallel flows will take the same path as an SFA does. It is often common in TCP studies to consider the well-known dumbbell topology (without routing). A linear network makes the problem tractable, and it also allows us to gain insights on the effects due to short-distance-flows and long-distance-flows (these two terms will be discussed later) for both SFAs and MFAs.

In this work, we present a formal analysis on proportional fairness in the equilibrium of SFAs competing with MFAs in a linear network. We quantify and obtain insights on the extent of the unfair gain by MFAs over SFAs and show the asymptotic behavior. There are a number of works that consider the gain due to parallel flows [5], [10]. To our knowledge, none have presented analytical results on MFAs’ gains when competing with SFAs.

II. REVIEW: PROPORTIONAL FAIRNESS

It has been shown that TCP Vegas [9] and FAST TCP [15] are proportionally fair in the equilibrium, and have been verified through simulation. Therefore, banking on their verification through simulation, our work focuses on the behavior at equilibrium through an analytical framework. The optimization problem is to maximize the utility function $U(x_j)$ of each session $j$ where $x_j$ denotes the rate of session $j$

$$\max_{x \geq 0} \sum_{j \in J} U(x_j) = \alpha d_j \log x_j \tag{1a}$$

subject to

$$\sum_{j \in J} \delta_{\ell j} x_j \leq c_{\ell}, \quad \ell \in \mathcal{L} \tag{1b}$$
where $\delta_{ij}$ is the indicator that takes the value 1 if session $j$ uses link $\ell$, 0 otherwise.

In Problem (1), constraints (1b) say that rates $x_j$ due to all sessions $\mathcal{J}$ using link $\ell$ do not exceed capacity $c_{\ell}$ of link $\ell \in \mathcal{L}$. In the objective function (1a), $d_j$ corresponds to propagation delay of session $j$, and $\alpha_j$ is based on an interpretation to be discussed shortly. To be precise, due to these parameters, Problem (1) is said to address $(\alpha_j, d_j)$-weighted proportional fairness; in other words, the solution (equilibrium) $x^*$ to (1) is $(\alpha_j, d_j)$-weighted proportionally fair.

As discussed in [9], there are two valid implementations of TCP Vegas that arose out of different interpretations found in [2]. The first one (based on the actual code) is based on bytes per round trip time, while the second one (based on the text described in [2]) is based on bytes per second. The former corresponds to $\alpha_j = \alpha/d_j$, while the latter corresponds to all sessions having the same $\alpha$, i.e., $\alpha_j = \alpha$. In terms of the objective function, the former reduces to

$$U(x_j) = \alpha \log x_j,$$

while the latter becomes

$$U(x_j) = \alpha d_j \log x_j.$$  

It is easy to see that the first situation does not give any weights to the sessions; the solution or equilibrium in this situation is (standard) proportionally fair. The latter situation gives weights to the propagation delay $d_j$ for session $j$, i.e., the solution is $d_j$-weighted proportionally fair. For brevity, we will refer to the former situation as the fixed delay situation and the latter as the delay-weighted situation. It has been shown in [15] that the solution in FAST TCP is $d_j$-weighted proportionally fair, i.e., it falls into the delay-weighted category. Without loss of generality, we set $\alpha = 1$ in the rest of the paper.

The log function for proportional fairness can also be written in the product form. Thus, for the fixed-delay situation (2), we seek to

$$\max_{x \geq 0} \prod_{j \in \mathcal{J}} x_j$$

whereas for the delay-weighted situation (3), we seek to

$$\max_{x \geq 0} \prod_{j \in \mathcal{J}} x^{d_j}_j$$

subject to (1b). In the rest of the paper, we will use the product form. We will discuss TCP Reno separately in Section V.

III. PROPORTIONAL FAIRNESS IN A LINEAR NETWORK: APPLICATIONS WITH SINGLE FLOWS

In this work, we focus on a linear network with $n$ nodes where all links have capacity 1, i.e., $c_{\ell} = 1, \ell \in \mathcal{L}$. First, we consider all applications to be SFAs. Within this, we consider two types of sessions: one between adjacent nodes (short-distance-flow) and the other between the far end nodes (long-distance-flow). We number the $n$ nodes in the linear network as $1, 2, \ldots, n$ in the order they are connected to each other in a linear fashion. A session between two adjacent nodes $i$ and $i+1$ is a short-distance-flow, with rate $x_{i,i+1}$. The session connecting the far end nodes 1 and $n$ is a long-distance-flow, with rate $x_{1,n}$. This is pictorially depicted in Fig. 1.

In each link that connects node $i$ and $i+1$, there are two SFAs, one for the adjacent short-distance-flow $(x_{i,i+1})$ and the other for the long-distance-flow $(x_{1,n})$. Thus, constraints (1b), with the capacity set to 1, reduces to

$$x_{i,i+1} + x_{1,n} \leq 1, \quad i = 1, 2, \ldots, n-1.$$

A. SFA: Fixed Delay Situation

The utility objective for the fixed-delay situation (4) for a linear network with $n$ nodes becomes

$$U^SFA_{FD}(x) = x_{1,n} \cdot \prod_{i=1}^{n-1} x_{i,i+1}.$$ 

Therefore, the optimization problem for the fixed-delay situation for proportional fairness with all sessions being SFAs can be stated as

$$\max_{x \geq 0} U^SFA_{FD}(x) = x_{1,n} \cdot \prod_{i=1}^{n-1} x_{i,i+1}$$

subject to

$$x_{i,i+1} + x_{1,n} \leq 1, \quad i = 1, 2, \ldots, n-1.$$  

To solve (6), we rely on the key result that at the optimal solution, the constraints (6b) must be active for every link (the proof is shown in the Appendix). This implies that

$$x_{i,i+1} = 1 - x_{1,n}, \quad i = 1, 2, \ldots, n-1.$$ 

Plugging $x_{i,i+1}$ in (6a), the objective function reduces to

$$U^SFA_{FD}(x_{1,n}) = x_{1,n} \cdot (1 - x_{1,n})^{n-1}.$$  

Clearly, this is a function of a single variable. By differentiating (7) and setting the resulting expression to zero, we find the equilibrium solution for the long-distance-flow with (standard) proportional fairness to be

$$x^*_{1,n} = \frac{1}{n}.$$ 

Then, the solution for short-distance-flows is

$$x^*_{i,i+1} = 1 - \frac{1}{n}, \quad i = 1, 2, \ldots, n-1.$$ 

B. SFA: Delay-Weighted Situation

We now turn to the delay-weighted situation. The objective for the delay-weighted situation (5) for the linear network
scenario shown in Fig. 1 becomes
\[
U^{SFA}_{DW}(x) = x_{1,n}^d_1 \cdot \prod_{i=1}^{n-1} x_{i,i+1}^{d_{i,i+1}}.
\] (8)

In order to consider the propagation delay, we make a simplifying assumption. Assume that the propagation delay between adjacent nodes is normalized to \(1 = d_{i,i+1}\). Then, the long-distance-flow that connects the far end nodes \(1 \) and \(n\) will face propagation delay, \(d_{1,n} = n - 1\) for the number of links traversed. Thus, the optimization problem for the delay-weighted case of proportional fairness with all sessions being SFAs can be stated as
\[
\max_{x \geq 0} U^{SFA}_{DW}(x) = x_{1,n}^{n-1} \cdot \prod_{i=1}^{n-1} x_{i,i+1}^{1}.
\] (9a)

subject to
\[
x_{i,i+1} + x_{1,n} \leq 1, \quad i = 1, 2, ..., n - 1.
\] (9b)

Similar to the fixed-delay situation, at optimality for (9), constraints (9b) must be active for all links, which implies that \(x_{i,i+1} = 1 - x_{1,n}, \quad i = 1, 2, ..., n - 1\). Plugging \(x_{i,i+1}\) in the objective function (9a), we obtain
\[
U^{SFA}_{DW}(x_{1,n}) = x_{1,n}^{n-1} \cdot (1 - x_{1,n})^{n-1}.
\] (10)

Again, this is a function of a single variable. By differentiating (10) and setting the resulting expression to zero, we find the equilibrium solution to be
\[
x^{*}_{i,i+1} = x^{*}_{1,n} = \frac{1}{2}.
\] (11)

From the above, we can make an important observation. For the delay-weighted case with weighted-proportional fairness in a linear network, all SFAs receive an equal rate at the equilibrium, regardless of the number of nodes. On the other hand, in the fixed-delay case, the long-distance-flow receives \(\frac{1}{n}\)-th of the bandwidth of the link.

IV. PROPORTIONAL FAIRNESS IN A LINEAR NETWORK: APPLICATIONS WITH MULTIPLE FLOWS

We next move on to see what happens when SFAs are competing with MFAs. We consider two possible cases. In the first case, all adjacent nodes have MFA sessions while the far end nodes have an SFA session. In the second case, all adjacent nodes run SFAs while the far end nodes run an MFA.

A. MFAs: The Adjacent Node Case

In Fig. 2, we illustrate the case of adjacent nodes running MFAs and the far end nodes running an SFA. In general, we assume that each of the MFAs runs \(k\) parallel flows (shown for \(k = 2\) in Fig. 2). For brevity, we refer to this situation as MFA/A. We wish to consider this problem again for the two situations: fixed-delay and delay-weighted. We first start with the fixed-delay situation.

When an MFA has \(k\) parallel flows between two adjacent nodes \(i\) and \(i + 1\), then each flow can be identified by \(x_{i,i+1}^f,\) for \(f = 1, 2, ..., k\). Along with the single SFA between the far end nodes, the objective function (4) for the fixed-delay situation becomes
\[
U^{MFA/A}_{FD}(x) = x_{1,n} \cdot \prod_{i=1}^{n-1} x_{i,i+1}^k.
\]

In this case, we have the following link constraints
\[
\sum_{f=1}^{k} x_{i,i+1}^f + x_{1,n} \leq 1, \quad i = 1, 2, ..., n - 1.
\]

It is easy to see that each of the \(k\) flows for each MFA will get the same treatment. Thus, we can simplify as \(x_{i,i+1} = x_{i,i+1}^f, f = 1, 2, ..., k\). Therefore, the optimization problem for the fixed-delay situation becomes
\[
\max_{x \geq 0} U^{MFA/A}_{FD}(x) = x_{1,n} \cdot \prod_{i=1}^{n-1} x_{i,i+1}^k
\] (12a)

subject to
\[
kx_{i,i+1} + x_{1,n} \leq 1, \quad i = 1, 2, ..., n - 1.
\] (12b)

Similar to our argument before, the capacity constraints (12b) are active at optimality. This means that
\[
x_{i,i+1} = \frac{1 - x_{1,n}}{k}, \quad i = 1, 2, ..., n - 1.
\] (13)

Plugging this in the objective function (12a), it reduces to
\[
U^{MFA/A}_{FD}(x_{1,n}) = x_{1,n} \cdot \left(\frac{1 - x_{1,n}}{k}\right)^{(n-1)k}.
\] (14)

Again, this is a function of a single variable. By differentiating (14) and setting the resulting expression to zero, we find the solution for the SFA serving the far end nodes to be
\[
x^{*}_{1,n} = \frac{1}{k(n-1) + 1}.
\] (15)

Then, the solution for each flow of an MFA is
\[
x^{*}_{i,i+1} = \frac{n - 1}{k(n - 1) + 1}.
\] (16)

Since each MFA has \(k\) flows, then at the equilibrium, the rate for each MFA between adjacent nodes is
\[
k \cdot x^{*}_{i,i+1} = \frac{k(n - 1)}{k(n - 1) + 1}.
\] (17)

The analysis for the delay-weighted situation is similar. We need to account for the propagation delay of \(n - 1\) for the SFA connecting the far end nodes, while the adjacent nodes have
the propagation delay normalized to 1. The link constraints remains the same as (12b). However, the objective function takes the following form

$$U_{\text{DW},A}^{\text{MFA}}(x) = x_{1,n}^{n-1} \prod_{i=1}^{n-1} x_{i,i+1}^k.$$ We can again substitute, as above, to reduce the problem to a single variable, resulting in

$$U_{\text{DW},A}^{\text{MFA}}(x_1, n) = x_{1,n}^{n-1} \left(\frac{1-x_{1,n}}{k}\right)^{(n-1)k}.$$ The rate at the equilibrium for the SFA connecting the far end nodes for the delay-weighted situation is

$$x_{1,n}^* = \frac{1}{k+1},$$ whereas the rate for each MFA is

$$k \cdot x_{i,i+1}^* = \frac{k}{k+1}. \quad (19)$$ It is important to observe that the solution is independent of the number of nodes for the delay-weighted situation. However, the number of parallel flows ($k$) of an MFA influences the effective rate in the case of delay-weighted proportional fairness.

**B. MFA: The Far End Case**

This time we assume that the far end nodes are running an MFA session with $k$ parallel flows, while all the adjacent nodes are running SFAs. This is illustrated in Fig. 3 for $k = 2$. We again start with the fixed-delay situation. If each of the flows for the MFA for the far end is identified by $x_{1,n_f}, f = 1, 2, ..., k$, then the objective function (4) becomes

$$U_{\text{FD}}^{\text{MFA/FE}}(x) = \prod_{f=1}^{k} x_{1,n_f} \prod_{i=1}^{n-1} x_{i,i+1},$$ and the link constraint is

$$x_{i,i+1} + \sum_{f=1}^{k} x_{1,n_f} \leq 1, \quad i = 1, 2, ..., n-1.$$ Again, due to symmetry, we can write $x_{1,n_f} = x_{1,n}, f = 1, 2, ..., k$. Thus, the optimization problem reduces to

$$\max_{x \geq 0} U_{\text{FD}}^{\text{MFA/FE}}(x) = x_{1,n}^k \prod_{i=1}^{n-1} x_{i,i+1} \quad (20a)$$ subject to

$$x_{1,i+1} + k \cdot x_{1,n} \leq 1, \quad i = 1, 2, ..., n-1. \quad (20b)$$ Since at optimality, all constraints are active, we can write

$$x_{1,i+1} = 1 - k \cdot x_{1,n}, \quad i = 1, 2, ..., n-1. \quad (21)$$ Plugging this in (20a), we arrive at

$$U_{\text{FD}}^{\text{MFA/FE}}(x_1, n) = x_{1,n}^k \cdot (1-kx_{1,n})^{n-1},$$ which leads to the optimal solution

$$x_{1,n}^* = \frac{1}{n+k-1}. \quad (22)$$

$$x_{i,i+1}^* = \frac{n-1}{n+k-1}. \quad (23)$$ Since the MFA connecting the far end nodes has $k$ parallel flows, its collective rate at the equilibrium is

$$k \cdot x_{1,n}^* = \frac{k}{n+k-1}. \quad (24)$$ Finally, we consider the delay-weighted case. Due to the propagation delay of $n-1$ for the MFA between the far end nodes, the objective function takes the form

$$U_{\text{DW},A}^{\text{MFA/FE}}(x) = x_{1,n}^{(n-1)k} \prod_{i=1}^{n-1} x_{i,i+1},$$ subject to the same constraints (20b) as the fixed-delay situation. Substituting the same way as before, the objective function reduces to

$$U_{\text{DW},A}^{\text{MFA/FE}}(x_1, n) = x_{1,n}^{(n-1)k} \cdot (1-kx_{1,n})^{n-1}.$$ By solving, we obtain

$$x_{1,n}^* = \frac{1}{k+1}. \quad (25)$$ Since in this case, the MFA connecting the far end nodes has $k$ parallel flows, its collective rate is

$$k \cdot x_{1,n}^* = \frac{k}{k+1}. \quad (26)$$ while the SFA’s equilibrium rate is $1/(k+1)$.

**V. TCP Reno**

The utility function (1a) for TCP Reno is given by

$$U(x_f) = \frac{d_i}{1-\beta} x_f^{1-\beta} \quad (27)$$ with $\beta = 2$ [15], which also satisfies the proportional fairness property [12]. Then, for the case of the MFAs between adjacent nodes and the SFA between the far end nodes, we can use (13) along with $\beta = 2$ to reduce the objective function (27) to

$$U_{\text{Reno}}^{\text{MFA/A}}(x_{1,n}) = -\frac{n-1}{x_{1,n}} - \frac{(n-1)k^2}{1-x_{1,n}}. \quad (28)$$ Interestingly, this one has the same optimal solutions as given by (18) and (19) for the MFA/A, delay-weighted case.
Similarly, for the case of the MFA between the far end nodes and the SFAs between the adjacent nodes using (21), the objective function (27) becomes

$$U_{Reno}^{MFA/FE}(x_{1,n}) = \frac{(n-1)}{x_{1,n}} - \frac{n-1}{1-kx_{1,n}}. \quad (29)$$

For this one, the optimal solutions are the same as given by (25) and (26) for the MFA/FE, delay-weighted case.

**VI. Results**

We will now illustrate a number of results based on the analytical solutions that were derived above. We break this discussion into two categories: the fixed delay situation and the delay-weighted situation.

**A. Fixed Delay**

First, we consider the case when the MFAs are between adjacent nodes and the SFA is between the far end nodes. The equilibrium solutions are given by (15) for the SFA’s rate and by (17) for the MFAs’ rates, which are both dependent on $n$ and $k$. We illustrate for two values of $n$: $n = 3$ and $n = 10$, which are shown in Fig. 4(a) and Fig. 4(b), respectively, where we varied the value of $k$, the number of parallel flows for MFAs. Note that $k = 1$ falls back to both types of sessions being SFAs. We can clearly see that as $k$ increases, MFAs occupy higher bandwidth, leading to starvation for the SFA that connects the far end nodes. For a short linear network, i.e., $n = 3$, MFAs have the most gain by increasing their shares by 43% by increasing the number of flows from $k = 1$ to $k = 10$. From (15) and (17), we see that keeping $n$ fixed and letting $k \rightarrow \infty$, we observe the following asymptotic behavior:

$$\text{SFA: } x_{i,n}^* \rightarrow 0; \quad \text{MFA: } k \cdot x_{i,i+1}^* \rightarrow 1. \quad (30)$$

Next, we consider the case when the far end nodes are running an MFA while the adjacent nodes are running SFAs. The equilibrium solutions are given by (24) for the MFA’s rate and (23) for each SFA’s rate. From Fig. 5(a) ($n = 3$) and Fig. 5(b) ($n = 10$), we make an interesting observation. For a 10-node linear network, the MFA’s rate at the equilibrium is lower than that of the SFA’s rate when $k \leq 9$. However, as the number of parallel flows increases for the MFA, its rate eventually becomes more than that of the SFA. The cross-over point moves to the right as the number of nodes, $n$, increases. From (23) and (24), keeping $n$ fixed, and letting $k \rightarrow \infty$, we note the asymptotic behavior:

$$\text{SFA: } x_{i,i+1}^* \rightarrow 0; \quad \text{MFA: } k \cdot x_{n,n}^* \rightarrow 1. \quad (31)$$

**B. Delay-Weighted**

From the analytical results presented for the delay-weighted situation, we note that the rate at the equilibrium is independent of the number of nodes in a linear network. In particular, when all sessions are SFAs, we know from (11) that the rates are equal. In other words, weight-proportional fairness does not penalize sessions with long round trip delays.

For the case where the adjacent nodes have MFAs and the far end nodes are connected by an SFA, the equilibrium for the delay-weighted case is given by (19) for each MFA’s rate and (18) for the SFA’s rate. The solutions are plotted in Fig. 6(a) for $k = 1, \ldots, 10$. When $k = 1$ (i.e., both sessions are SFAs), the sessions share the rate equally. As the number of parallel flows for the MFA increases, they grab more bandwidth than the SFA connecting the far end nodes. In other words, the SFA suffers.

Next, consider the case where adjacent nodes are running SFAs, but the far end nodes are connected by an MFA. From Fig. 6(b), which is based on (25) and (26), we can see that the MFA connecting the far end nodes starts to grab more bandwidth while the SFAs connecting the adjacent nodes now suffer.

Based on the derivation in Section V, we know that the equilibrium solution for TCP Reno is the same as that for the delay-weighted situation. Thus, Fig. 6(a) and Fig. 6(b) are also applicable to TCP Reno.

**VII. Summary**

In this paper, we assessed the unfair advantage of MFAs over SFAs by presenting analytical results at the equilibrium for a linear network. We find that at the equilibrium, MFAs can take an undue share (taking away from the SFAs) depending on the number of parallel flows invoked. Our derivation results...
in a simple elegant solution that helps us gain insights on the asymptotic behavior.

ACKNOWLEDGEMENT

This work has been supported by National Science Foundation Grant # CNS-0916505.

REFERENCES


APPENDIX

Lemma: All constraints are active at optimality for Problem (6).

Proof: Suppose that for the optimal solution \( x^* \), the link connecting \( i \) and \( i + 1 \) is not active. This implies that \( x^*_{i,i+1} + x^*_{i,n} < 1 \). Assume that the gap is \( \epsilon > 0 \), i.e., \( x^*_{i,i+1} + x^*_{i,n} = 1 - \epsilon \). Now, set \( x^*_{i,i+1} = x^*_{i,i+1} + \epsilon \). Clearly, \( x^*_{i,i+1} \) with the rest of \( x^* \) is feasible. Since \( x^*_{i,i+1} < x^*_{i,i+1} \), we find that

\[
x^*_{i,n} \prod_{i=1}^{n-1} x^*_{i,i+1} < x^*_{i,n} x^*_{i,i+1} \prod_{i=1}^{n-1} x^*_{i,i+1},
\]

which contradicts the assumption that \( x^* \) is optimal. \( \blacksquare \)