Information Technology Journal 10 (2): 358-366, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.358.366

A Framework for Dynamically Selecting Gain in LEDBAT Algorithm for High Speed Networks

Amuda James Abu and Steven Gordon
Sirindhorn International Institute of Technology, Thammasat University,
Rangsit, P.O. Box 22, Pathumthani 12121, Thailand

Abstract: Low Extra Delay Background Transport (LEDBAT) congestion control algorithm has been developed as an alternative for Internet applications that use multiple TCP connections. To allow efficient data transfer when no other traffic exists, a LEDBAT source saturates a bottleneck link while maintaining the access router queue delay at or below a pre-defined target. The source rapidly reduces its sending rate upon the arrival of a new traffic. This paper analyses the relationship between LEDBAT performance and the congestion window gain in high-speed access networks. We develop mathematical models, validated with simulations, which show high values of gain are necessary otherwise a LEDBAT source will take a long time to reach optimal sending rate (steady state), underutilizing the bottleneck link. However our analysis also shows that, once in steady state, with a high gain a LEDBAT source can induce jitter into real-time applications and also achieve sub-optimal sending rates. Therefore, we propose a novel framework for dynamically selecting a gain during a LEDBAT connection. Unlike LEDBAT using a fixed gain, our framework achieves a better trade-off between throughput for the LEDBAT source and fairness with other sources. It also provides the ability to tune LEDBAT depending on requirements (throughput or fairness).

Key words: LEDBAT, congestion control, high-speed networks, real-time applications, peer-to-peer applications

INTRODUCTION

The Low Extra Delay Background Transport (LEDBAT) congestion control algorithm has been developed as an alternative for Internet applications that use multiple TCP connections (Shalunov, 2010). By establishing multiple TCP connections, an application from one customer can induce significant queue delays in the ISP's access router, severely degrading the performance of voice, video and gaming applications of other customers. Therefore, a LEDBAT source adjusts its sending rate to maintain the queue delay in the access router at or below a pre-defined target. To achieve high throughput for applications, a LEDBAT source aims to saturate the bottleneck link in the path to the destination. A competing aim is also to yield quickly when other (TCP or UDP) sources start sending over the bottleneck link.

LEDBAT is designed to provide a lower-than-best-effort service for background file transfer applications, especially P2P file sharing. It operates under the assumptions that the queue delay at the access router of the bottleneck link will be the primary varying contributor to end-to-end delay and the access router does not employ active queue management. This is typical in numerous ISP networks (Akella *et al*., 2003). To measure the queue delay a LEDBAT source adds a timestamp to each data packet sent; the receiver calculates the one-way delay, returning the value in acknowledgement packets. The source then estimates the queue delay from the difference between the current one-way delay and a base one-way delay calculated during the start of the connection. The LEDBAT source controls its sending rate by updating its congestion window, w_t, for every ACK received according to Eq. 1 where, $\hat{D}$ is the current estimated queue delay, T is the target delay and G_0 is the gain. As with TCP, the LEDBAT sending rate and congestion window are approximately proportional (Kelly, 1999). T and G_0 (both constants) are two key parameters that influence how well LEDBAT achieves its aims of saturating the bottleneck and yielding quickly to other traffic:

$$w_{t+1} = \begin{cases} w_t + \dfrac{G_0 \times |T - \hat{D}|}{w_t} & \text{if } \hat{D} < T \\ w_t - \dfrac{G_0 \times |T - \hat{D}|}{w_t} & \text{if } \hat{D} > T \\ w_t & \text{if } \hat{D} = T \\ \dfrac{1}{2} w_t & \text{on packet loss} \end{cases} \quad (1)$$

Corresponding Author: Amuda James Abu, Sirindhorn International Institute of Technology, Thammasat University, Rangsit, P.O. Box 22, Pathumthani 12121, Thailand

In this study, we analyse the relationship between gain, G_0 and LEDBAT performance. We develop mathematical models, validated with simulations, that show high values of gain (>1 packet per RTT) are necessary, especially with high-speed access networks, otherwise a LEDBAT source will take a long time to reach optimal sending rate (steady state), underutilizing the bottleneck link. However our analysis also shows that, once in steady state, with a high gain a LEDBAT source can induce jitter into real-time applications and also achieve sub-optimal sending rates. Therefore, we propose a novel framework for dynamically selecting a gain during a LEDBAT connection. Compared to using a fixed gain, our framework achieves a better trade-off between throughput for the LEDBAT source and fairness with other sources. It also provides the ability to tune LEDBAT depending on requirements (throughput or fairness).

There are numerous transport layer congestion control algorithms in use and proposed in the literature (Mohan and Ravichandran, 2007; Sasipraba and Srivatsa, 2006; Jasem *et al*., 2010). They can be classified as: loss-based, delay-based, rate-based and low-priority. LEDBAT exhibits characteristics of delay-based and low-priority algorithms. However, it differs from many such algorithms including TCP-Vegas (Brakmo and Peterson, 1995), TCP-NICE (Venkataramani *et al*., 2002) and TCP-LP (Kuzmanovic and Knightly, 2006), in that it aims at minimizing queue delay in a network to a defined value and the magnitude of its congestion window is a function of the estimated queue delay. Welzl and Ros (2010) provides a survey of these and other low-priority congestion control algorithms.

Several works have analysed the performance of LEDBAT in simulation and real-life environments. Rossi *et al*. (2010a) evaluated LEDBAT performance in a controlled testbed and Internet experiment. They found out that TCP traffic on the unrelated backward path is capable of causing LEDBAT to significantly underutilize the link capacity in the forward path. Rossi *et al*. (2010b) identified potential fairness issues in LEDBAT while Carofiglio *et al*. (2010b) proposed that random drops of LEDBAT sender window and multiplicative decrease are promising solutions to the problem of LEDBAT latecomer advantage. The proposed solutions are not without a performance trade-off between link utilization and fairness. Comparative analysis of LEDBAT with other low priority protocols (TCP-NICE and TCP-LP) in the presence of TCP showed that LEDBAT achieves the lowest priority (Carofiglio *et al*., 2010a). The authors further showed via sensitivity analysis that unfairness exists between two LEDBAT flows with different delay targets or different network conditions and LEDBAT is aggressive with TCP in the case of LEDBAT misconfigurations. From a performance evaluation of a Python implementation of LEDBAT in real and emulated network environment, there exists a large computational overhead with the implementation resulting in underutilizing the available network bandwidth more than TCP (Andreica *et al*., 2010).

Two key parameters, namely target (T) and gain (G_0), are used in Eq. 1. Although, the value of T depends on the delay tolerance of real-time applications and the operating systems' accuracy in timestamping packet, Shalunov (2010) requires target to be 25 msec (as at the time this work was carried out). This is because Shalunov (2010) observes that the performance of most real-time applications (e.g., voice, video, etc) degrades as they experience queue delay greater than 25 m sec in most access networks of an ISP. For LEDBAT not to increase its congestion window faster than TCP, the value of gain must be carefully chosen (Shalunov, 2010). Although, a gain of 1 packet per Round Trip Time (RTT) (Rossi *et al*., 2010b; Shalunov, 2009) and 10 packets per RTT (Shalunov, 2009) have been suggested, only the work of Abu and Gordon (2010) has analysed the performance impact of different values of gain on LEDBAT. They used simulations to illustrate the impact of high and low gain on LEDBAT performance. Our paper extends this by providing a theoretical framework for analysing gain and offering new insights into the impact of LEDBAT gain. As a result of our analysis, in this paper we also develop a new approach for calculating the optimal value of gain for LEDBAT and then give practical guidelines for implementation. Finally, we demonstrate the performance advantages of a dynamic gain LEDBAT with respect to throughput, fairness and responsiveness.

SYSTEM MODEL

To analyse LEDBAT a dumbbell topology is used (Fig. 1). This represents an ISP network with multiple customers (traffic sources) sending data to various destinations via a shared bottleneck link. The traffic sources may use LEDBAT (e.g., P2P file sharing applications uploading data) or UDP (e.g., real-time voice or video applications). TCP applications are not considered in this paper. It has been shown that in the presence of TCP, LEDBAT will revert to its minimum sending rate (Carofiglio *et al*., 2010a), at which point the gain has minimal impact on throughput and fairness. As uploads using TCP will often be short (compared to LEDBAT uploads), we consider the case when TCP upload is absent.

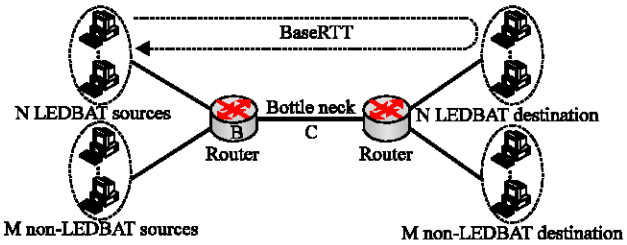

Fig. 1: Network topology

The round trip time (RTT) is assumed to be the sum of the fixed link delays and the variable queue delay for data traffic at the access router. ACKs do not experience queue delay at the access router. All routers use FIFO drop-tail queues with maximum queue length of B packets.

Simulations are conducted using ns-2.34 (ns-2), modified to support LEDBAT congestion control with T = 25 msec and default G_0 = 40. The LEDBAT source runs an FTP application sending 1500 byte packets. The UDP source runs a constant bit rate application sending 500-byte packets at 625 pkt sec^{-1}. The bottleneck link has capacity, C, of 1250 pkt sec^{-1} and 50 msec delay. All other links have capacity of 12,500 pkt sec^{-1} and 5 msec delay. Hence, the base RTT (RTT_{base}) is 120 msec. The access router buffer size, B, is 2000 packets. In this study results from three simulation scenarios are presented:

- LEDBAT source starts at 0s and runs until the end of simulation at 120 sec. A UDP source starts at 10 sec, runs for 30 sec, then after a 5 sec break the next UDP source starts with the same timing and so on
- LEDBAT source starts at 0 sec and finishes at 100 sec. No UDP traffic source
- LEDBAT sources starts at 0 sec and finishes at 600 sec. A UDP source starts at 100 sec and finishes at 600 sec

IMPACT OF GAIN ON LEDBAT THROUGHPUT

What is an appropriate value of gain for LEDBAT? We will show that a high gain reduces the time spent in the LEDBAT start-up phase. This is beneficial because during the start-up phase LEDBAT is not saturating the bottleneck bandwidth. However during the steady state phase, we show that a lower gain is desirable so as to minimize the variation of LEDBAT congestion window (Δw).

High gain for fast start-up: Figure 2 approximates the LEDBAT congestion window (w) and queue delay at access router when no other traffic is present. w_t, w_0, w_{sat} and w_{stdy} represent the instantaneous, initial, bottleneck

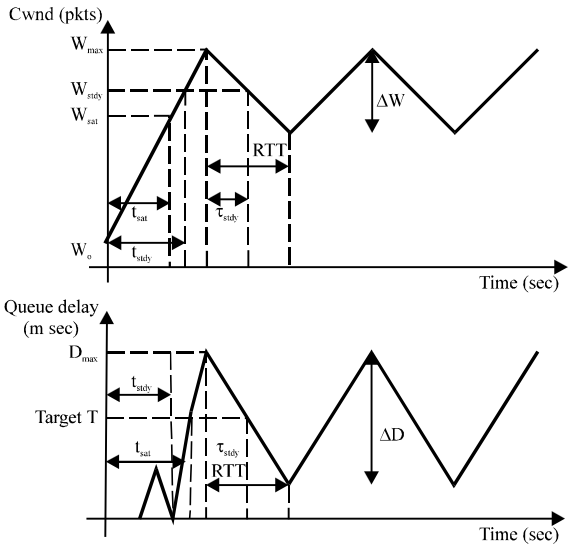

Fig. 2: Evolution of LEDBAT congestion window and access router queue delay

saturated and steady state congestion windows of LEDBAT respectively. Using Fig. 2, we will derive the time LEDBAT spends in the start-up phase (t_{stdy}).

The start-up phase can be divided into two periods: from the start (t = 0) until the bottleneck link is fully saturated (t_{sat}) and then until steady state is reached (t_{stdy}). In the first period, as the input sending rate is less than the bottleneck link capacity, the average queue delay ($\hat{D}_{ave}$) is assumed to be 0. Therefore, w_t is increased by G_0T/w_t every ACK received, or approximately G_0T every RTTs. As the queue delay is 0, RTT = RTT_{base} increase in this period is therefore, G_0T/RTT_{base}. Alternatively, from Fig. 2, the rate of increase in the first period is $(w_{sat}-w_0)/t_{sat}$. That is:

$$\frac{G_0T}{RTT_{base}} = \frac{w_{sat} - w_0}{t_{sat}} \quad (2)$$

During the second period, $\hat{D}$ linearly increases from 0 to T. Therefore, $\hat{D}_{ave} = T/2$. From Eq. 1, w_t is increased by $G_0(T-\hat{D})$ every RTT. Assuming the average values of $\hat{D}$

and RTT, RTT_{ave} = RTT_{base} +T/2 and the rate of increase in this period is G_0 (T-T/2)/(RTT_{base}+T/2). Similar to above, considering the congestion window size from Fig. 2, it can be shown that:

$$\frac{G_0(T/2)}{RTT_{base}+T/2}=\frac{w_{stdy}-w_{sat}}{t_{stdy}-t_{sat}} \quad (3)$$

where, μ_{led} and μ_{udp} respectively represent the available bottleneck bandwidths for LEDBAT and UDP traffic, where, μ_{led} = C-μ_{udp}. w_{stdy} and w_{sat} are proportional to the available bandwidth-delay products where the delay component includes and excludes the LEDBAT delay target respectively. That is, w_{stdy} = μ_{led} (RTT_{base}+T) and w_{sat} = μ_{led}×RTT_{base}. Substituting for w_{stdy} and w_{sat} and re-arranging, we have Eq. 4 from Eq. 2 and 3:

$$t_{stdy}=\frac{\frac{RTT_{base}(\mu_{led}\cdot RTT_{base}-w_0)}{T}+\mu_{led}(2RTT_{base}+T)}{G_0} \quad (4)$$

For example, a LEDBAT flow obtaining 100 Mb sec^{-1}, with RTT_{base} = 200 m sec, w_0 = 2 pkts, T = 25 m sec and G_0 = 40, would take 7 min to reach steady state. Even for a long connection (1 h), this represents a significant time at which the LEDBAT source is not sending at the optimal rate. Setting G_0 = 400 would reduce the time to reach steady state to 42 sec.

Figure 3 presents theoretical results from Eq. 4 and from simulation of the first scenario, showing how high gain takes less time to reach steady state (t_{stdy}) with increasing C than low gain (where, μ_{led} = C in this case). The larger discrepancy between theoretical and simulation results for lower G_0 is because more time is spent during the period when $0<\hat{D}\le T$ with G_0 = 40 than other values of G_0, as we assume $\hat{D}_{ave}=T/2$ in the analytical model.

Figure 4 shows the normalized throughput ($\eta_{startup}$) of a simulated LEDBAT source running a 30 sec active session application with different values of gain and bottleneck capacity. The decreasing throughput especially for G_0 = 40 as C increases is due to LEDBAT not reaching steady state long before 30 sec. This illustrates the need for high values of gain (G_0) in LEDBAT, especially for high-speed access networks and short-session applications.

Low gain for high utilization in steady state: Having shown the benefit of using high gain in LEDBAT startup phase, we now show how low gain can minimize the variation of LEDBAT congestion window (Δw) during the steady state phase.

As the congestion window varies between w_t-Δw/2 and w_t+Δw/2 in Fig. 2, minimizing Δw has two benefits.

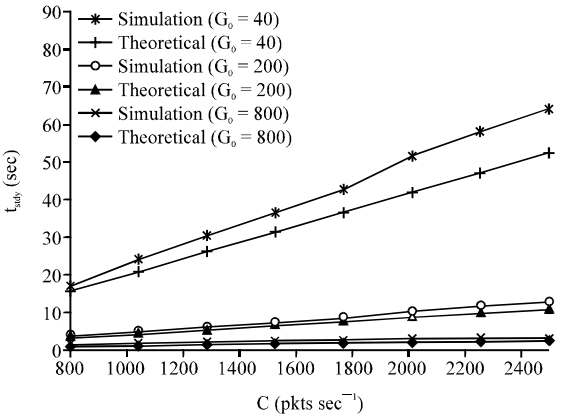

Fig. 3: Time for LEDBAT to reach steady state in the absence of UDP

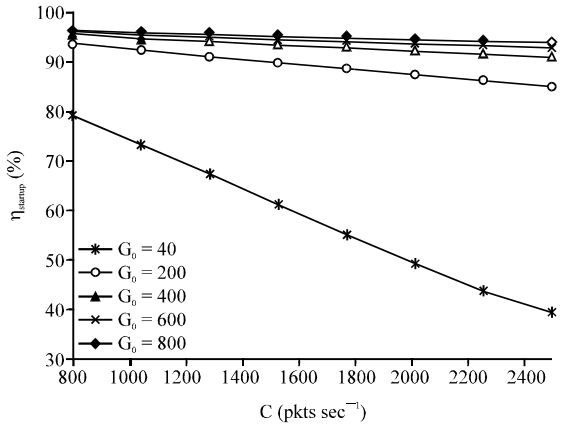

Fig. 4: Normalized throughput for a 30 sec active session application using LEDBAT in the absence of UDP

Firstly, a window of w_t-Δw/2 may be less than w_{stdy} representing significant underutilization of the available bottleneck link capacity. Secondly, a window of w_t+Δw/2 means the LEDBAT source will be contributing more than the target delay to the queue delay if w_t+Δw/2>w_{stdy}, resulting in brief increases in delay experienced by other applications. This introduces jitter, which can degrade the performance of real-time applications. Therefore, we will derive an expression for Δw in LEDBAT steady state.

In Fig. 2, we represent Δw as the change in LEDBAT congestion window every RTT in steady state. From Eq. 1, Δ_w = G_0×(D_{max}-T) because $\hat{D}=D_{max}>T$. Δw is given by Eq. 5 because D_{max} = T+ΔD/2, where ΔD is the variation in the access router queue delay (Fig. 2):

$$\Delta w=\frac{1}{2}G_0\cdot\Delta D \quad (5)$$

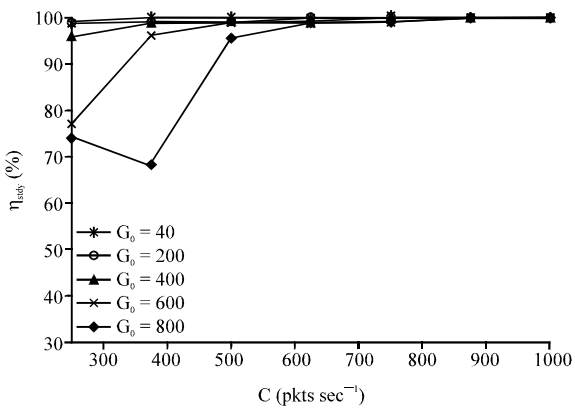

Fig. 5: Normalized LEDBAT steady state throughput in the presence of UDP

D_{max} depends only on Δw when no other non-LEDBAT traffic exists otherwise it depends also on other factors such as the arrival rate of other traffic.

We now present two practical cases where G_0 will and will not have significant impact on Δw. The change in LEDBAT congestion window every ACK received is simply $\Delta w/w_t$ where $w_t = \mu_{led}(RTT_{base}+D_t)$. Therefore, the fraction $\Delta w/w_t$ decreases as μ_{led} and RTT_{base} increase and vice versa. This means when little or no other traffic co-exists with LEDBAT in the same high speed access network, there will be little or no impact of G_0 on Δw in LEDBAT steady state. On the other hand, when there is an increasing arrival rate of UDP traffic in the high speed access network, low gain becomes desirable as μ_{led} decreases and consequently decreasing w_t.

In the scenario under consideration, we have no solution for D_{max} at the moment because its value is not only determined by LEDBAT but also by UDP. Therefore only simulation results are presented to show the impact of gain on Δw and consequently on throughput. Figure 5 shows results from the third scenario. μ_{led} is varied by increasing the arrival rate of UDP traffic at the bottleneck for each value of G_0. The benefit of using low gain in steady state is shown in Fig. 5 as higher normalized steady state throughput (η_{stdy}) of LEDBAT is observed with low gain than higher ones. This is due to increasing variation of LEDBAT congestion window as μ_{led} decreases and G_0 increases (Fig. 8).

LEDBAT WITH DYNAMIC GAIN

There are two conflicting requirements in selecting G_0 for LEDBAT: maximize G_0 to reduce the time spent in the start-up phase (when sending rate is sub-optimal) and minimize G_0 to reduce the variation of LEDBAT congestion window in the steady state phase (when bottleneck link is underutilized). This therefore motivates the need to include a dynamic gain in the LEDBAT algorithm, i.e. one gain during start-up and another in steady state. To do this we present in this section a framework for: calculating the gain; collecting information needed to calculate the gain and knowing when to switch from one phase to another by a LEDBAT source.

Calculating the gain: To calculate the gain in start-up phase, we use Eq. 4. For the gain in steady state, we need to derive two expressions which are defined as functions of G_0 ($f(G_0)$ $h(G_0)$).

Firstly, we derive an expression that presents Δw as a function of G_0, $h(G_0)$. From Eq. 5 and Fig. 2, replacing ΔD by $2(D_{max}-T)$ we have $f(G_0)$ in Eq. 6.

$$f(G_0) = \Delta w = G_0(D_{max} - T) \tag{6}$$

Secondly, we derive an expression for τ_{stdy} as a function of G_0, $h(G_0)$. Let τ_{stdy} be the time it takes LEDBAT to decrease its congestion window from w_{max} to w_{stdy}. Note that the smaller the value of τ_{stdy} the faster the responsiveness of LEDBAT in yielding to newly arriving traffic. Suppose it takes LEDBAT one RTT and RTT/2 to decrease w_{max} by $2(w_{max}-w_{stdy})$ and $w_{max}-w_{stdy}$, respectively ($RTT/2 \equiv \tau_{stdy}$, Fig. 2). Similar to the derivation of Eq. 2 and 3, we have Eq. 7 having substituted $RTT_{base}+D_{max}$ for RTT:

$$\frac{RTT_{base} + D_{max}}{2(w_{max} - w_{stdy})} = \frac{\tau_{stdy}}{w_{max} - w_{stdy}} \tag{7}$$

Substituting $2(w_{max}-w_{stdy})$ for Δw in Eq. 5 we have $\Delta D = 4(w_{max}-w_{stdy})/G_0$. Using this in the expression for D_{max} we have Eq. 8 from Eq. 7:

$$h(G_0) = \tau_{stdy} = \frac{w_{max} - w_{stdy}}{G_0} + \frac{RTT_{base} + T}{2} \tag{8}$$

Equation 6 and 8 show that the calculation of gain in steady state presents an optimization problem. The objective of the problem is to:

$$\underset{G_0}{\text{argmin}}\ f(G_0) \text{ and } h(G_0) \tag{9}$$

In Eq. 9 there exists no single solution because there are multiple conflicting objectives (Eq. 6 and 8). Introducing more constraints such that the objective becomes:

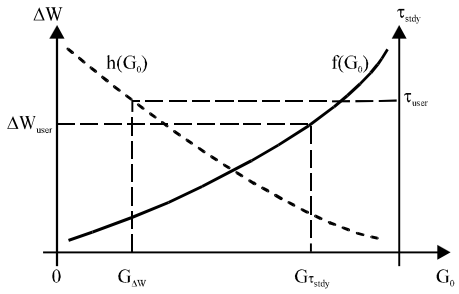

Fig. 6: Optimization problem in calculating LEDBAT gain in steady state

$$\underset{G_0}{\text{argmin}}\ h(G_0): f(G_0) \le \Delta w_{user} \tag{10}$$

$$\underset{G_0}{\text{argmin}}\ f(G_0): h(G_0) \le \tau_{user} \tag{11}$$

where, Δw_{user} and τ_{user} are design parameters which respectively give the maximum tolerable values of Δw and τ_{stdy}. Δw_{user} limits the impact of G_0 on LEDBAT steady state throughput and access router queue delay variation (jitter) while τ_{stdy} limits the impact of G_0 on the rate at which LEDBAT yields to newly arriving traffic. The optimization problem in Eq. 10 either has no solution (for the case when Δw_{user} is too small) or a single solution. In a similar fashion to Eq. 10, either no solution or a single solution exists for Eq. 11.

Let $G\tau_{stdy}$ and $G_{\Delta w}$ be the G_0 obtained from Eq. 8 and 6, respectively. Figure 6 further elucidates the optimization problem. We consider the gain in steady state (G_{stdy}) to be bounded by $G_{\Delta w}$ and $G_{\tau stdy}$, where, $G_{\Delta w}$ represents the lower bound while $G\tau_{stdy}$ the upper bound. To calculate the gain in steady state (G_{stdy}), we optimize G_{stdy} for high utilization of the available bottleneck bandwidth and fast responsiveness to newly arriving traffic by LEDBAT in steady state. To do this we present G_{stdy} in Eq. 12 as the average value of $G_{\Delta w}$ and $G\tau_{stdy}$:

$$G_{stdy} = \frac{\Delta w_{user} + 2(w_{max} - \mu_{led}(RTT_{base} + T))}{2(2\tau_{user} - (RTT_{base} + T))} \tag{12}$$

Practical considerations: For a LEDBAT source to calculate the gain in start-up and steady state phases, it needs information about: the values of μ_{led}, RTT_{base}, t_{stdy}, w_0, w_{max} Δw_{user} and when to switch between Eq. 4 and 12.

The LEDBAT source maintains the minimum one-way delay in its path. Twice the value of this could be RTT_{base} if symmetric links existed between the source and destination. Other option is to keep the minimum RTT from the start of the connection (suitable for asymmetric links). The value of w_0 can be obtained from the initial value of the source congestion window (typically w_0 = 2 pkts). W_{max} is simply the highest congestion window before the calculation of gain switches from Eq. 4 to 12. The LEDBAT source estimates μ_{led} in its path using an efficient method for estimating the available network bandwidth in high speed networks proposed by Man *et al*. (2008). As reported by Guerrero and Labrador (2010), the promising method by Man *et al*. (2008) has been shown to provide available bandwidth estimates in high-speed networks with zero overhead because it uses data instead of probing (UDP) packets.

The values of t_{stdy}, Δw_{user}, τ_{user} depend on the user preference and the type of application. For long session applications (e.g., seeding large files to remote peers) the value of t_{stdy} may not necessarily be small. However, for short session applications (e.g., web browsing) we recommend relatively small value of t_{stdy}. The available bottleneck bandwidth for LEDBAT (μ_{led}) will be underutilized if the change in the LEDBAT congestion window (Δw) is greater than or equal to twice the number of packets backlogged in the queue of the access router. We therefore recommend Δw_{user} to be in the range {0, $2\mu_{led}\times T$}. τ_{stdy} cannot be less than $RTT_{base}+T$ because the source needs to receive acknowledgement before updating its congestion window. We consider the additional time spent by LEDBAT to reduce its sending rate until the queue delay remains approximately at the target to be no greater than the LEDBAT target delay. As a result we suggest τ_{user} to be defined in the range {$RTT_{base}+T$, $RTT_{base}+2T$].

A LEDBAT source is in steady state when it estimates the queue delay in its path to be greater than or equal to the delay target otherwise it is out of steady state. In the LEDBAT source algorithm with the proposed dynamic gain framework, the source uses Eq. 4 to calculate the gain at the start of its connection because $\hat{D} < T - \delta$ where, δ represents a threshold value. The source rapidly increases its sending rate until $\hat{D} \ge T + \delta$ (steady state) where it calculates the gain using Eq. 12. If the source detects again that $\hat{D} < T - \delta$ (out of steady state) due to the departure of other traffic from the access network, the calculation of the gain reverts to Eq. 4. In the same manner, the source quickly increases its sending rate until $\hat{D} \ge T + \delta$ where Eq. 12 is used again. We define δ in the range [0, 0.2T]. In the framework, we avoid a case of gain calculated from Eq. 12 to be greater than the gain

from Eq. 4 by using the minimum value of G_{stdy} from previously calculated gain in steady state (the minimum value is reset to the gain in start-up phase when the source is no longer in steady state).

PERFORMANCE EVALUATION

We modified the LEDBAT source algorithm to include our proposed dynamic gain framework in ns-2.34 while ensuring all constraints associated with the framework. Several simulations, based on the model described in the section on system model, were run for the values of G_0 and μ_{led} while other parameters assume their default values. We use Δw_{user} = 0.5 pkts, τ_{user} = 170 msec and δ = 0. Results showing performance improvement of LEDBAT with dynamic gain (DG-LEDBAT) over LEDBAT with fixed gain (FG-LEDBAT) are presented in the sections on throughput analysis (for the third scenario), fairness analysis (for the third scenario) and responsiveness analysis (for the first scenario). We omit the results of the performance evaluation of DG-LEDBAT for the second scenario because DG-LEDBAT and FG-LEDBAT differ only in steady state phase (Fig. 10)

Throughput analysis: Figure 7 shows the steady state normalized throughput η_{stdy} of DG-LEDBAT and FG-LEDBAT with different values of G_0 and μ_{led}. Ideally, a LEDBAT traffic source will fully utilize the available bottleneck bandwidth in steady state when no other traffic exists. The results from FG-LEDBAT show that η_{stdy}<100% when μ_{led}<900 pkts/s for fixed gains of 600 and 800 while η_{stdy} for DG-LEDBAT remains at 100% for all values of μ_{led}. This is because the mean deviation of FG-LEDBAT congestion window (MD_{cwnd}) with fixed gains of 600 and 800 for μ_{led}<900 pkts/s is greater than or equal to the number of packets backlogged in the bottleneck buffer (Fig. 8). Note that Δw is twice MD_{cwnd}. On the other hand, results from DG-LEDBAT show that full utilization of the available bottleneck capacity is achieved for all values μ_{led} with fast startup phase, shown in Fig. 7. This is because MD_{cwnd} for DG-LEDBAT in Fig. 8 is less than $\mu_{led}\times T$.

Fairness analysis: LEDBAT assumes to be fair with other traffic when the access router queue delay is less than or equal to the target. Therefore we use the average value of the difference between the queue delay above the target and the target delay T, represented by $\delta_{D\geq T}$, to quantify the fairness of FG-LEDBAT and DG-LEDBAT with real-time UDP traffic. That is:

$$\delta_{D\geq T} = \frac{\sum_{i=1}^{j}(D_i - T)}{j} \quad (13)$$

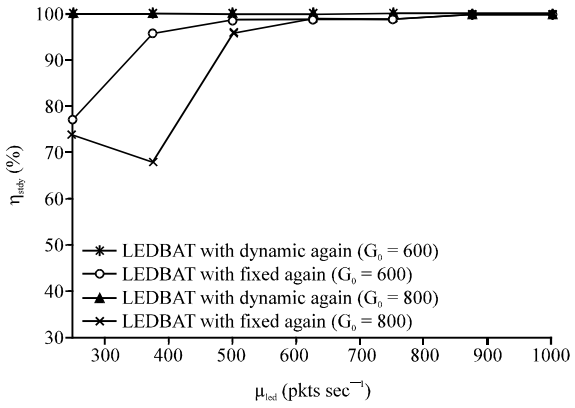

Fig. 7: Normalized steady state throughput for LEDBAT with fixed and dynamic gains in the presence of UDP

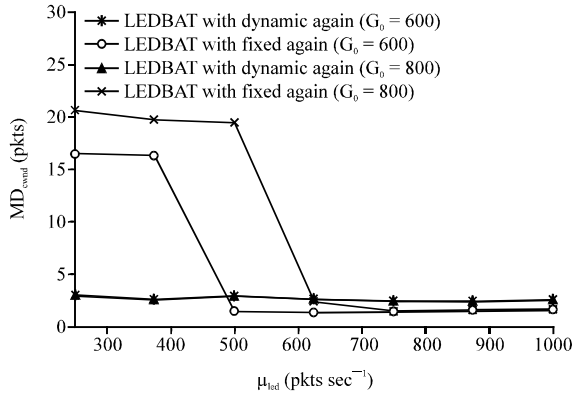

Fig. 8: Mean deviation of the congestion windows of LEDBAT with fixed and dynamic gains in the presence of UDP

where, D_i is the access router queue delay of the ith packet. We interpret increasing value of $\delta_{D\geq T}$ as decreasing fairness and vice versa.

Figure 9 shows that $\delta_{D\geq T}$ for FG-LEDBAT and DG-LEDBAT follow the same trend as MD_{cwnd} in Fig. 8. The increasing value of $\delta_{D\geq T}$ for FG-LEDBAT beyond some values of μ_{led} is due to the increasing variation of the congestion window and the arrival rate of UDP traffic. However, the results for DG-LEDBAT show that $\delta_{D\geq T}$ is greatly reduced in the worst case of FG-LEDBAT, minimizing the waiting time of real-time traffic in the access router queue with respect to the LEDBAT target delay.

Responsiveness analysis: Here, a comparative analysis of the rates at which FG-LEDBAT and DG-LEDBAT respond to the arrival and departure of UDP traffic is presented in Fig. 10 (with G_0 = 800). All UDP traffic sources send

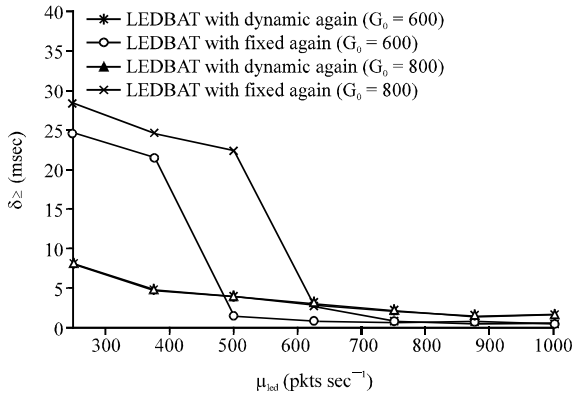

Fig. 9: Queue delay difference above the target for LEDBAT with fixed and dynamic gains in the presence of UDP

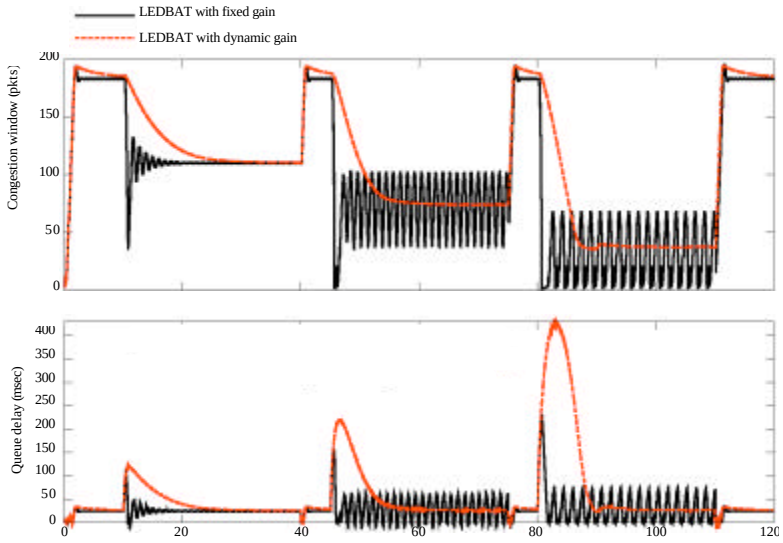

Fig. 10: Congestion windows and access router queue delays of LEDBAT with fixed and dynamic gains in the presence of three UDP traffic sources starting and stopping at different times and rates with 5s interval between successive UDP flows

packets at different rates such that μ_{led} takes different values in the presence of each UDP session (i.e., μ_{led} = 750, 500 and 205 packets per second).

Figure 10 shows that FG-LEDBAT and DG-LEDBAT reach steady state at the same time (at approximately 3s) because the two differ only in their steady states. In steady state, FG-LEDBAT and DG-LEDBAT stabilise their congestion windows (sending rates) and queue delays when $\mu_{led} \geq 750$ pkts sec^{-1}. However, variations of FG-LEDBAT congestion window and access router queue delay become noticeably high as $\mu_{led} \leq 500$ pkts sec^{-1} unlike DG-LEDBAT which achieves smooth sending rate and queue delay in the presence of UDP.

Observing the yielding rates of FG-LEDBAT and DG-LEDBAT upon the arrival of UDP traffic, FG-LEDBAT yields faster than DG-LEDBAT. This is because G_{stdy} takes its value between $G_{\Delta w}$ and $G\tau_{stdy}$ (Fig. 6). However, DG-LEDBAT can be configured to yield faster by using a value of G_{stdy} that is close to $G\tau_{stdy}$. Shortly after each UDP traffic stops after 30 sec of active session, FG-LEDBAT and DG-LEDBAT increase their congestion window at the same rate because DG-LEDBAT detects that it is no longer in steady state as queue delay is observed to be less than the target in Fig. 10, reverting to Eq. 4 for calculating the gain.

CONCLUSIONS

This study has analysed the relationship between gain and LEDBAT performance in a high speed access network. Our results show that high values of gain are necessary in order to reduce the time it takes LEDBAT to

reach optimal sending rate. However, our analysis also shows that, once in steady state, with a high gain a LEDBAT source can induce jitter into real-time applications and also achieve sub-optimal sending rates. On this basis we propose a novel framework for dynamically selecting a gain during a LEDBAT connection. Performance evaluation of our proposed framework shows that a better trade-off between throughput for the LEDBAT source and fairness with other sources is achieved when compared to LEDBAT using a fixed gain. Additionally, the proposed framework also provides the ability to tune LEDBAT depending on requirements (throughput or fairness).

LEDBAT assumes queue delay is the only variable component of RTT. As future work, we will analyse the impact of other variable components of delay (e.g., re-routing, variable Internet/link delay) on LEDBAT performance.

ACKNOWLEDGMENTS

This research was conducted as part of A.J.A.'s M.Sc. studies, supervised by S.G., from November 2008 to September 2010 at Sirindhorn International Institute of Technology, Thammasat University, Thailand. A.J.A. acknowledges the financial support provided by SIIT Thammasat University Graduate Student Scholarship.

REFERENCES

Abu, A.J. and S. Gordon, 2010. A dynamic algorithm for stabilising LEDBAT congestion window. Proceedings of the 2nd International Conference on Computer and Network Technology, April 23-25, Bangkok, Thailand, pp: 157-161.

Akella, A., S. Seshan and A. Shaikh, 2003. An empirical evaluation of wide-area internet bottlenecks. Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, June 11-14, San Diego, CA, USA., pp: 316-317.

Andreica, M.I., N. Tapus and J. Pouwelse, 2010. Performance evaluation of a python implementation of the new LEDBAT congestion control algorithm. Proceedings of IEEE International Conference on Automation, Quality and Testing Robotics (AQTR), May 28-30, Cluj-Napoca, Romania, pp: 1-6.

Brakmo, L. and L. Peterson, 1995. TCP vegas: End to end congestion avoidance on a global internet. IEEE J. Select. Areas Commun., 13: 1465-1480.

Carofiglio, G., L. Muscariello, D. Rossi and C. Testa, 2010a. A hands-on assessment of transport protocols with lower than best effort priority. Proceedings of the 35th IEEE Conference on Local Computer Networks, Oct. 11-14, Denver, Colorado, pp: 1-8.

Carofiglio, G., L. Muscariello, D. Rossi and S. Valenti, 2010b. The quest for LEDBAT fairness. Proceedings of the IEEE Globeco, Dec. 6-10, Miami, FL. USA., pp: 1-6.

Guerrero, C.D. and M.A. Labrador, 2010. On the applicability of available bandwidth estimation techniques and tools. Comput. Commun., 33: 11-22.

Jasem, H.N., Z.A. Zukarnain, M. Othman and S. Subramaniam, 2010. The delay with new-additive increase multiplicative decrease congestion avoidance and control algorithm. Inform. Technol. J., 4: 1327-1335.

Kelly, F., 1999. Mathematical modelling of the internet. Proceedings of the 4rth International Congress on Industrial and Applied Mathematics, July 5-9, Edinburgh, Scotland, pp: 105-116.

Kuzmanovic, A. and E.W. Knightly, 2006. TCP-LP: Low-priority service via end-point congestion control. IEEE/ACM Trans. Networking, 14: 739-752.

Man, C.L.T., G. Hasegawa and M. Murata, 2008. Inline bandwidth measurement techniques for gigabit networks. Int. J. Internet Protocol Technol., 3: 81-94.

Mohan, B.J. and V.C. Ravichandran, 2007. Cross layer optimization for congestion control in Ad hoc networks. Inform. Technol. J., 6: 283-287.

Rossi, D., C. Testa and S. Valenti, 2010a. Yes, we LEDBAT: Playing with the new bittorrent congestion control algorithm. Passive Active Measurement, 6032: 31-40.

Rossi, D., C. Testa, S. Valenti and L. Muscariello, 2010b. LEDBAT: The new bittorrent congestion control protocol. Proceedings of the International Conference on Computer Communication Networks, Aug. 2-5, Zurich, Switzerland, pp: 1-6.

Sasipraba, T. and S.K. Srivatsa, 2006. Network border patrol, a novel congestion avoidance mechanism for improving QOS in wireless networks. Inform. Technol. J., 5: 427-432.

Shalunov, S., 2009. Low extra delay background transport. Proceedings of the 75th IETF Meeting, July 26-31, Stockholm, Sweden, pp: 1-12.

Shalunov, S., 2010. Low Extra Delay Background Transport (LEDBAT): IETF internet draft. July 12, pp: 1-14. http://tools.ietf.org/id/draft-ietf-ledbat-congestion-01.txt.

Venkataramani, A., R. Kokku and M. Dahlin, 2002. TCP nice: A mechanism for background transfers. Proceedings of the 5th Symposium on Operating Systems Design and Implementation, Dec. 9-11, Boston, Massachusetts, pp: 329-343.

Welzl, M. and D. Ros, 2010. A survey of lower-than-best effort transport protocols. IETF Internet Draft, March 1, pp: 1-14. http://tools.ietf.org/html/draft-ietf-ledbat-survey-00.