# An Adaptive Fuzzy Control Double Leaky Bucket Using Backoff Time and Backpressure in Policing Mechanism Schemes over VDSL Network

**Somchai Lekcharoen**[1], Non-member

**ABSTRACT**

When congestion occurs in VDSL networks, it will drop incoming frames because the buffer fills up frames until it overflows. In backpressure algorithm suitable for low traffic [1], it prevents buffer overflows. In backoff time in leaky bucket over traffic policing mechanism suitable for high traffic [2], Backoff time computations are wildly used in order to avoid the impact on the network performance. In this paper, we propose a fuzzy control double-leaky bucket using backpressure and backoff time detects violations in parameter negotiation. We evaluate and compare the performance of a fuzzy control backpressure and backoff time in double-leaky bucket as a policing mechanism (FBBD), a leaky bucket (LB), a jumping window (JW), a triggered jumping window (TJW) and a double-leaky bucket (DLB). The performance of fuzzy control backpressure and backoff time in double-leaky bucket was investigated through fluctuations in telecommunications traffic streams (burst/silent type). Simulation results show that in VDSL frames, the fuzzy logic control system helps improve the performance of our fuzzy control double-leaky bucket using backpressure and backoff time schemes as a policing mechanism much better than conventional policing mechanism by about 25 % in terms of conforming and non-conforming frames once various types of burst/silence traffic are generated.

**Keywords**: Double leaky bucket, fuzzy control, congestion, policing mechanism

## 1. INTRODUCTION

When a VDSL network device receives more frames than it can send to a destination, the receiver begins to drop frames. This leads to network congestion because the source device retransmits frames and this causes a degraded throughput.

In other words, congestion in VDSL networks causes a decrease in conforming frames or more dropped frames occurring in a shorter period of time. The solution to this problem is to take precautions before the congestion occurs in the network. The network must retain the traffic parameters that contact the network defined earlier, such as data rate, peak rate, burst/silence, etc. The mechanism used to take care of this problem is the policing mechanism system. It can be classified in two groups. The first group is the window-based mechanism and the second group is the leaky bucket system.

Window-based mechanisms have difficulties and is complicated to implement; however, the leaky bucket system is easy to implement and can be applied to the network [3-5].

An alternative approach is to perform hop-by-hop flow control, or backpressure at every link. Hop-by-hop flow control combines two mechanisms. The first is a strict portioning of the available buffer space among active flows, and selective backpressure on a per flow basis. The second is fair queuing to all resources shared by flows within a single network device. The result is twofold: frames are never dropped due to congestion within the network, and the available bandwidth is fairly divided up among all competing active flows. With no contention for priorities, each flow has access to the entire bandwidth on each link; but when there is contention, each flow obtains a fair share of the bandwidth [6-8].

An another approaches are backoff computation whereby each source will delay the message whenever the transmission to the next service fails. Backoff algorithms have introduced a variety of other techniques such as exponential backoff, random backoff, linear backoff and quadratic backoff as described in many papers [9-11].

In this paper, we apply fuzzy control backpressure and backoff time in double leaky bucket as policing mechanism concepts to the management of network queues and we also evaluate the performance based on VDSL network model. There are many previous studies involving backpressure [6-7] and exponential backoff time [9], however fuzzy control backpressure and bakoff time in double leaky bucket has not been reported before. We therefore carried out a study on the performance comparison between fuzzy control backpressure and exponential backoff time verses the traditional leaky bucket, double leaky bucket, jump-

Manuscript received on August 6, 2007 ; revised on November 15, 2007.
[1] The author is with Faculty of Information Technology Rangsit University Thailand, Emails: somchai@yahoo.com

ing window and triggered jumping window in VDSL network.

The paper is organized as follows. In section 2, an overview of the most significant policing mechanism schemes already proposed in literature is given. Section 3 defines backoff time and backpressure algorithm. Section 4 defines the model of a fuzzy backpressure and exponential backoff time applied double leaky bucket in traffic policing schemes. Section 5 defines the simulation model and section 6 contains a performance evaluation of the proposed solution and comparison between the fuzzy backpressure and backoff time (FBBD), traditional leaky bucket (LB), jumping window (JW), triggered jumping window (TJW) and double leaky bucket (DB). In section 7, the conclusions and recommendations for future research are presented.

## 2. DESCRIPTION AND MODELING OF POLICING MECHANISM

Policing mechanism allows us to control the peak rate of traffic sent or received on an interface during the entire active phase. The mechanisms have been proposed which are described in the following sections.

### 2.1 Requirement for policing mechanism

The policing mechanism system can monitor traffic parameters as defined earlier such as peak rates (Fig. 1). When an arriving frames are detected as being in the receiving system, the network monitor can drop the non-conforming frames. The policing mechanism scheme prevents excessive data-rate connections from bottleneck the source network transmission, and significantly improves the quality of service (QoS).

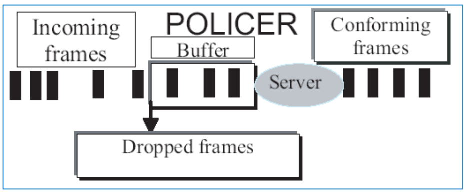


***Fig.1:*** *Policing mechanism.*

The policing mechanism system allows us to control the maximum rate of traffic sent and received during the active phase and can be operated in real time. These mechanisms have been proposed which are described in following sections.

#### 2.1.1 Traffic source models

In our simulation, the traffic source model is based on Burst/Silence traffic stream. Burst-periods and Silence-periods are strictly alternating as described in [12-14].

In our simulation, a burst traffic stream from a single source is modeled as a burst/silence traffic stream.

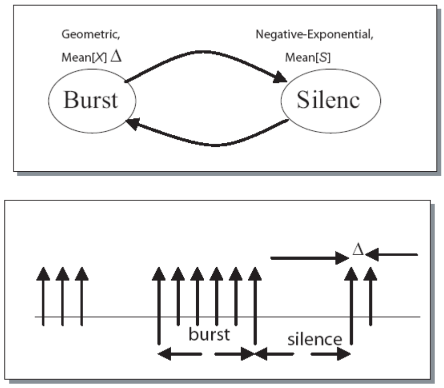


***Fig.2:*** *The burst silence traffic model used in this study.*

The burst/silence ratio is strictly alternating. The number of frames per burst is assumed to have a geometric distribution with mean $E[X]$; the duration of the silence phases is assumed to be distributed according to a negative-exponential distribution with mean $E[S]$; and inter-frames arrival time during a burst is given by $\Delta$. With

$$\begin{aligned} \textit{mean burst duration} &= E[X]\Delta \\ \textit{mean silence duration} &= E[S] \\ \textit{mean cycle duration} &= E[X]\Delta + E[S] \end{aligned}$$

### 2.2 Policing mechanism models [4,15,16]

The policing mechanism schemes are done at the edges of the network for frame-based traffic. This mechanism decides whether to accept a unit of incoming frames or drop frames. This study selected five policing mechanisms, including the Leaky Bucket (LB), the jumping window (JW), the triggered jumping window (TJW), the Double-Leaky Bucket (DLB) and the Fuzzy-Control, Double-Leaky Bucket (FBBD).

#### 2.2.1 Leaky bucket process model

The VDSL network must provide a large bandwidth and handle the quality of service (QoS) guarantees. The Leaky Bucket (LB) mechanism (see Fig. 1) ensures that the source traffic does not exceed the negotiated rate. The bucket-size can be represented as a buffer with capacity $N$. If the buffer is filled up with frames until it is overflowing, then the frames are dropped. The server generates at a specific data rate, $R$. The LB is a commonly used for traffic control in high-speed network.

#### 2.2.2 The double leaky bucket mechanism (DLB)[17,18]

The policing mechanism system can be used a double-leaky bucket algorithm that requires a high-speed network. As indicated by the name, the behavior of a leaky bucket is similar to a bucket with

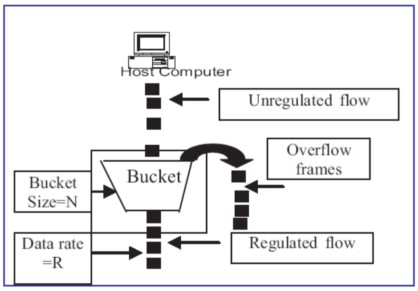


***Fig.3:*** *Leaky bucket mechanism.*

a hole in its bottom. If data flows into the bucket faster than it flows out of the bucket, then the bucket overflows. This causes data to be dropped until there is enough room in the buffer to accept new data. A leaky bucket uses two parameters to control the flow of traffic:

- Data rate – the number of frames per second that leak from the leaky bucket; permitting data to enter the network.
- Burst size – the number of groups of frames that are allowed to accumulate in the bucket.

Note: this algorithm drops the entire frame if it does not fit into the available buffer space in the leaky bucket. It is important to ensure that the data rate is appropriate for the frame interface. The use of double-leaky buckets as a traffic monitor is shown in Fig. 4.

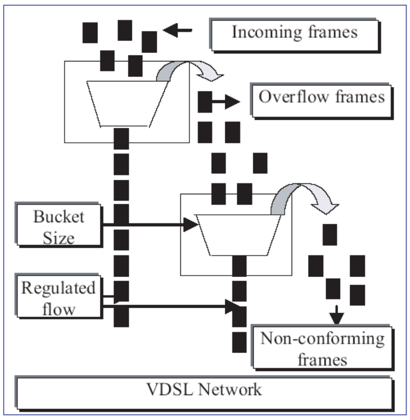


***Fig.4:*** *Double leaky bucket traffic monitor.*

The insured rate bucket in Fig. 4 determines whether an incoming unit of data can be accepted by the bandwidth for this network connection. The parameters for this double leaky bucket are the data rate and burst size for the VDSL network.

It is comprising of two leaky buckets $LB_1$ and $LB_2$. This DLB can function as explained in previous section. The capacity of each bucket can be summarized as follows. Let $\delta_{12}$ denote the capacity coefficient of these two buckets and $R_1$ is an output data rate. Let $M$ and $N$ be the bucket size of $LB_1$ and $LB_2$ respectively. $Max\_Size$ is the number of frames. $I$ is an input data rate. The total capacity of DLB (BKT) can be computed by

$$BKT = M + \delta_{12} N \tag{1}$$

where

$$\delta_{12} = \begin{cases} 1, & I > R_1 \ and \ N < Max\_Size \\ 0, & elsewhere \end{cases} \tag{2}$$

#### 2.2.3 The Jumping Window Mechanism (JW)

The JW mechanism limits the maximum number of frames accepted from a source within a fixed time interval (window) to a maximum number $N$. The new interval starts immediately at the end of the preceding interval (jumping window) and the associated counter is restarted again with an initial value of zero. Therefore, the time interval during which a specific frame is influencing the counter value varies from zero to the window width. The implementation complexity of this mechanism is comparable to the complexity of the LB mechanism. Counters are needed to measure the interval $T$ and to count the number of arrivals, and variables are needed for the counter limit and the interval length $T$. The probability that policing actions must be taken on a frame can be computed by using the counting process for the frame arrivals which characterizes the number of arriving frames in an arbitrary time interval. For example, the counting process for negative-exponential inter-arrival times is a Poisson process.

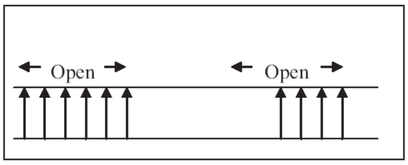


***Fig.5:*** *Jumping window traffic monitor.*

#### 2.2.4 The Triggered Jumping Window Mechanism (TJW)

The time window is not synchronized with source activity in the JW mechanism. To avoid the ambiguity problems arising from that fact, the *triggered jumping window* mechanism has been proposed, where the time windows are not consecutive but are triggered by the first arriving frame.

The implementation complexity for this mechanism is comparable to the complexity of the mechanisms described above.

## 3. BACKOFF TIME AND BACKPRESSURE ALGORITHM

Backoff computation, each source will delay the message whenever the transmission to next service fails. Backoff algorithms have introduced many techniques such as exponential backoff, random backoff,

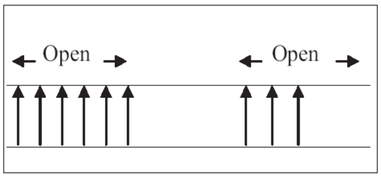


***Fig.6:*** *Trigged jumping window traffic monitor.*

linear backoff and quadratic backoff as described in many papers [2]. In this paper, we apply backoff concepts to waiting time in the queue with policing mechanisms and evaluate the performance using a high speed network model.

### 3.1 Backoff algorithm

Many papers study the backoff algorithms in terms of their effect on network performance as the offered load increases. However, simplification or modification of backoff algorithm can lead to very different analytical results [2,3]. Many backoff schemes have been proposed and studied.

#### 3.1.1 Pseudorandom backoff

In pseudorandom backoff (PB) scheme, none of the computation is applicable but queue disciplines. They are FIFO, LIFO and priority. In this paper, the FIFO and the maximum queue size are preset.

#### 3.1.2 Exponential backoff

Exponential backoff (EB) is an algorithm being widely used in traffic offered load. In EB, each node doubles the backoff time after each retry occurs ($2x$) but not above the maximum value ($B_{max}$), and decreases the backoff interval to the minimum value ($B_{min}$) after a successful retry. We summarize EB by the following set of equations:

$$\begin{aligned} x &\leftarrow min(2x, B_{max}), \ \ upon\ retry\ and \\ x &\leftarrow B_{min}, \ \ upon\ successful\ transmission. \end{aligned}$$

The $x$ is the backoff interval value. The values of the $B_{max}$ and $B_{min}$ are predetermined, based on the possible range of number of active nodes and the traffic load of a network. For example, $B_{max}$ and $B_{min}$ are usually set to 1024 and 2, respectively. Although some researchers found that the channel throughput in the network will be degraded as the backoff interval does not correctly represent the actual contention of the channel [2,11] but we experience somehow the EB can help improve the performance of the system regarding to the fluctuation of telecommunication traffic.

#### 3.1.3 Random backoff

Another approach is the use of the random backoff (RB) technique. In order to avoid repeated retry by one particular node based upon the detection of non availability of transmission, the sender is required to wait for a random period of time before next retry. This random period is referred to as *retry delay* or simply *backoff*. Backoff algorithms, which usually adaptively change the retry delay according to the traffic load, are implemented to address the dynamic network conditions and to improve the performance of such system. In a backoff algorithm, the duration of the backoff is usually selected randomly in the range of 0 and some maximum time duration, which we refer to as the *backoff interval* ($\tau$).

#### 3.1.4 Backoff interval time essentials

The backoff interval is dynamically controlled by the backoff algorithms as described above. Setting the length of the backoff interval is, however, not a trivial task. On one hand, with a fixed number of ready nodes, small backoff intervals do not help reduce the correlation among the retrying nodes to any appropriate low levels. These results are moreover raising too high number future retries, lowering the channel throughput. On the other hand, large backoff intervals introduce unnecessary idle time on the channel (waiting time in queue), increase the average packet delay and unneeded preparation of buffer to handle the size of queue, also eventually would degrade the system's performance.

### 3.2 Waiting time in the queue

In this section, we describe our proposed queue policy, called threshold-based queue management in details. The average waiting time of lots in the buffer of single-machine station can be approximated by :

$$\phi_q = \left(\frac{C_a^2 + C_e^2}{2}\right)\left(\frac{u}{1-u^{t_e}}\right) \tag{3}$$

where
$C_a$ = the coefficient of variation of the arrival times
$C_e$ = the corresponding coefficient of variation
$t_e$ = the mean processing time of station
$u$ = utilization

Concepts of waiting time in the queue have been introduced and developed by [9].

The behavior of backoff concept applicable to waiting time in the queue with policing mechanisms is nevertheless investigated. In this paper, we proposed comparisons of the performance between a fuzzy control backpressure and backoff time in double-leaky bucket as a policing mechanism (FBBD), a leaky bucket (LB), a jumping window (JW), a triggered jumping window (TJW) and a double-leaky bucket (DLB).

### 3.3 Backpressure algorithm

With backpressure algorithm, it works like XON /XOFF techniques with an objective to prevent buffer overflows and transient congestion in the network. Congestion control is invoked by triggering XON/XOFF flow control messages. The XOFF flow control message is sent to the upstream when buffer exceeds the upper threshold. When a server receives an XOFF signal, it will paused sending frames until it received an XON signal from the same server or until the time in XOFF message expired. The XON signal is triggered when the buffer at the congested server has descended below the lower threshold.

When frames arrive at the server's buffer, the backpressure algorithm is activated. If the queue is not full and is below the threshold, the server then send a message to the source hob so that it can double the transmission rate, and the frame is transmitted without delay. If the buffer exceeds the upper threshold, then the destination hop sends a message to the source hop so that it will reduce transmission rate to half. The Backpressure algorithm is illustrated with the pseudo code as follows.

//The Backpressure Algorithm
//The goal is to provide flow control at every link.
//If the buffer exceeds the upper threshold then
//the destination hop sends the message to source
//hop and reduce to half transmission rate.

Start Check:
**IF** simulation time reached MAX
**THEN** GOTO Stop
**ELSE** {
  **IF** $Q_{DESTN} < Q_{THRESHOLD}$
  **THEN** send feedback message and source increases input rate to double
    GOTO Start Check
  **ELSE** send feedback and source decreases input rate to half
    GOTO Start Check
}
Stop:

## 4. FUZZY CONTROL PRE-BUFFER

In this section, we will first describe a new fuzzy control pre-buffer as the monitor, which meets the requirements of performance implementation of VDSL networks. Concepts of fuzzy sets, fuzzy logic, and fuzzy logic control have been introduced and developed in this system.

### 4.1 Fuzzy control mechanism

Fuzzy logic was first introduced by L. Zadeh in the 1960s. Fuzzy control is a system that employs fuzzy logic as a model of the uncertainty of a given situation. It has been widely used for control systems and for decision making systems. The basic concept of a fuzzy controller is shown in Fig. 7. There are four parts: fuzzifier, fuzzy inference engine, rules based and a defuzzifier [20-22].

- Fuzzifier: The fuzzifier must convert the input parameters into suitable values that are needed in the inference engine.
- Fuzzy Rule: The fuzzy rule-based system, contains a set of fuzzy control rules. It comes from the researcher's experience.
- Inference Engine: The inference engine infers the fuzzy control rules and the related input parameters.
- Defuzzifier: The defuzzifier must convert the inferred fuzzy control action into a non-fuzzy control action.

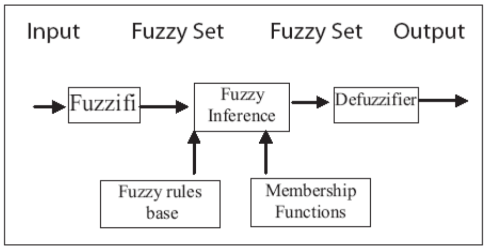


***Fig.7:*** *A model of a fuzzy controller.*

### 4.2 Regulator input fuzzification

Input variables are transformed into fuzzy sets (fuzzification) and manipulated by a collection of IF-THEN rules as shown in the figure below.

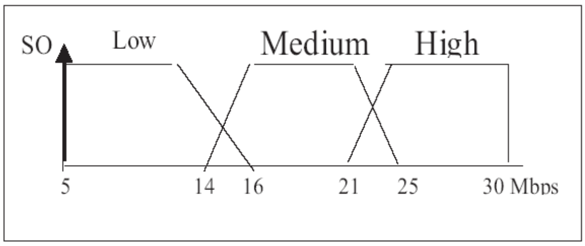


***Fig.8:*** *Membership function of an SO input variable.*

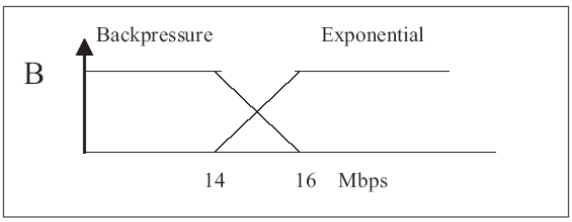


***Fig.9:*** *Membership function of B output variable.*

### 4.3 Inference, Fuzzy Rules and Defuzzification

Fuzzy control is involved only in rule bases. There is no separate defuzzification step. Based on our defined input variables and their membership functions, the fuzzy control is described by IF-THEN rules, each of which locally represents a linear input-output relation for the regulator. Fig. 7 shows the simple fuzzy rules used in the experiment.

**IF** So is Low (L) **THEN** B is Backpressure
**ELSE IF** So is Medium (M) or High (H)
**THEN** B is Exponential

***Fig.10:*** *Some fuzzy rules.*

Fig. 8 and 9 respectively show the membership functions of the linguistic values the input variables SO and the output variables B. Fig. 8 illustrates that when the source traffic is lower than 16 Mbps and the membership function is less than 0.5, backpressure scheme is applied. If analysis of the fuzzy system rules (Fig. 10) shows that if the source is low then traffic uses backpressure. On the other hand, if source is above medium then maximum waiting time in queue uses backoff schemes.

Set of rules as show in Fig. 8, 9 and 10. The selection of rules base is based on our experiment and beliefs on how the system should behave. Input traffics comprise of burst traffic stream (burst/silent stream) to fluctuate the VDSL network can be controlled by a fuzzy controller.

## 5. SIMULATION MODEL

Fig. 11 shows a simulation model used in this paper. It consists of a link processor, which transmits frames into the link and a finite buffer to hold frames. Frame arrival process is random and the inter-arrival time between frames follow a particular distribution function. Furthermore, the arriving frame is given to the link processor, which immediately starts transmitting the frame, while the buffer is not empty, the link processor retrieves a frame at a time from the buffer in a first-in-first-out (FIFO) order and transmits it onto the link. The link processor is never idle when there has been a frame waiting in the buffer.

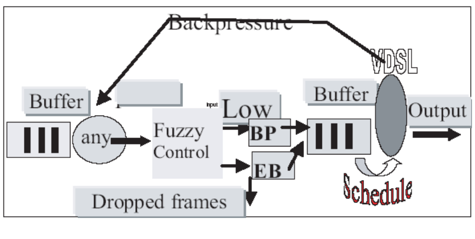


***Fig.11:*** *Simulation model.*

### 5.1 Input traffic

This research confines the discussion to main data only. Data source are generally bursty in nature whereas voice and video sources can be continuous or bursty, depending on the compression and coding techniques used

### 5.2 Characteristics of a queuing network model

Here are three components with certain characteristics that must be examined before the simulation models are developed.

#### 5.2.1 Characteristics of queuing network model

There are three components with certain characteristics that must be examined before the simulation models are developed.

**1) Arrival characteristics**

The pattern of arrivals input traffic mostly is characterized to be Poisson arrival processes. The probability of the inter-arrival time between event t, is defined by the inter-arrival time probability density function (pdf). The following formulae give the resulting probability density function (pdf), which the inter-arrival time $t$ is larger than some value $x$ when the average arrival rate is $\lambda$ events per second:

$$f_X(t) = \begin{cases} e^{-\lambda t}, for & t \geq 0 \\ 0, for & t < 0 \end{cases} \tag{4}$$

$$F_X(t) = P(X \leq t) = \int_0^t e^{-\lambda x} dx \tag{5}$$

In this paper, we adopt the ON/OFF bursty model [4].

#### 5.2.2 Service facility characteristics

In this paper, a server used service time based on randomly distribution with the exponential probability distribution. The arrival rates of frames are Poisson distributed. In order to examine the traffic congestion as the output of VDSL downstream links (15Mbps) [23], the service time in the simulation model was specified by the speed of an output link, given that the service time is 216 $\mu$s per frame where the frame size is 405 bytes [24].

#### 5.2.3 Source traffic descriptor

The parameters of the traffic source are characterized by the traffic that will be transmitted during the connection. The relation of each traffic parameter used in the simulation model is defined below:
$PFR$ ($peak\ frame\ rate$) $= \lambda a = 1/T$ in units of labels/second, where $T$ is the minimum inter-frame spacing in seconds. This research focused on:
Leaky bucket 1 has parameters as

$PFR = \lambda a = 10$ Mbps ($\sim$ 3,086 frames/s)
Hence, $T = 324$ sec.
Queue length = 15 frames
Leaky bucket 2 has parameters as
$PFR = \lambda a = 10$ Mbps ($\sim$ 3,086 frames/s)
Hence, $T = 324$ sec.
Queue length = 30 frames.

## 6. RESULTS AND ANALYSIS

The comparison between a fuzzy, backpressure and exponential in double- leaky bucket as the policing mechanism, a double-leaky bucket, leaky bucket, jumping window and triggered jumping window is shown in figures 12-16.

This section illustrates the simulation results from fuzzy control double-leaky bucket using backpressure and backoff in a policing system, double-leaky bucket and traditional policing mechanism, that are, fuzzy control double-leaky bucket (FBBD), double-leaky bucket (DB), leaky bucket (LB), jumping window (JW) and triggered jumping window (TJW) performance will be compared. The input frames (frame rates vary from 0 Mbps to 30 Mbps) with a burst/silence ratio of 100:100 provided simulation results as shown in Fig. 12. It clearly indicates that the fuzzy, backpressure and backoff time in double-leaky bucket is the best to guarantee the throughput. Throughput is one of factors of QoS that helps guarantee higher reliability of the network performance. In conclusion, the fuzzy, backpressure and backoff time in double-leaky bucket may assure higher reliability to handle multimedia applications such as multimedia traffic, when compared to traditional policing mechanism systems.

The results shown in Fig. 13 indicate that a fuzzy, backpressure and backoff time in double-leaky bucket will produce the lowest drop in frames compared to other policing schemes. It indicates that it has low retransmitting frames. It can handle multimedia traffic.

In Fig. 14, the results indicate that utilization of the traditional policing mechanism schemes provides the lowest utilization rate. From this perspective, the processing unit will be available for other sources in terms of sharing. The result is in the line of low processing power required by traditional policing mechanism schemes because they produces less conformance frames and higher dropped frames. Most frames are discarded before being transferred (into the network) to the entrance of the VDSL network. It seems that the traditional policing mechanism schemes cause less congestion but they will reflect a lower throughput in return.

In Fig. 15 shows that fuzzy control using backpressure and backoff time in double leaky bucket in policing mechanisms, all frames have to wait longer in the buffer next to the entrance of VDSL network. The consequence of long waiting hour is compatible to results shown in Fig. 16. It is due to less frames dropped and higher number of successful retry.

In conclusion, the fuzzy control, double-leaky bucket as the policing mechanism, performs better than traditional policing systems in forming conforming and non-conforming frames.

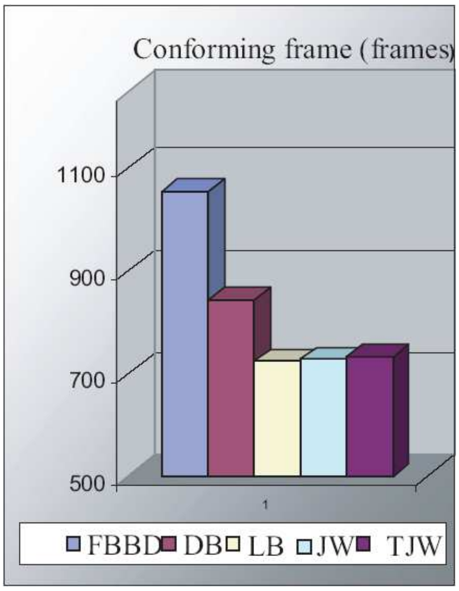


***Fig.12:*** *illustrates conforming frames compared with fuzzy, backpressure and exponential backoff in double-leaky bucket (FBBD), double-leaky bucket (DB), leaky bucket (LB), jumping window (JW) and triggered jumping window (TJW) systems as the policing mechanism at burst: silence rate of 100:100.*

## 7. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In this paper, we carried out a comparative study to investigate the performance of fuzzy control double leaky bucket using backpressure and exponential backoff time, in policing mechanism scheme with fixed types of traffic. The study was accomplished through simulation after developing an analytical queuing model.

We found that based on simulation results in general, the fuzzy control double leaky bucket using backpressure and exponential backoff time in policing mechanism scheme is the best in terms of maximizing the number of conforming frames, less non-conforming frames. It can be said that less non-conforming frames reduce re-transmission frames and high throughput. It is also believed that fuzzy control double leaky bucket using backpressure and exponential backoff time in policing mechanism scheme is suitable for data and multimedia.

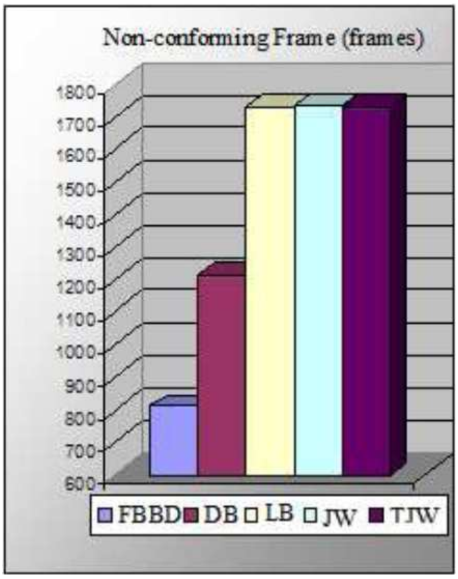


***Fig.13:*** *illustrates non-conforming frames compared with fuzzy, backpressure and exponential backoff in double-leaky bucket (FBBD), double-leaky bucket (DB), leaky bucket (LB), jumping window (JW) and triggered jumping window (TJW) systems as the policing mechanism at burst: silence rate of 100:100.*

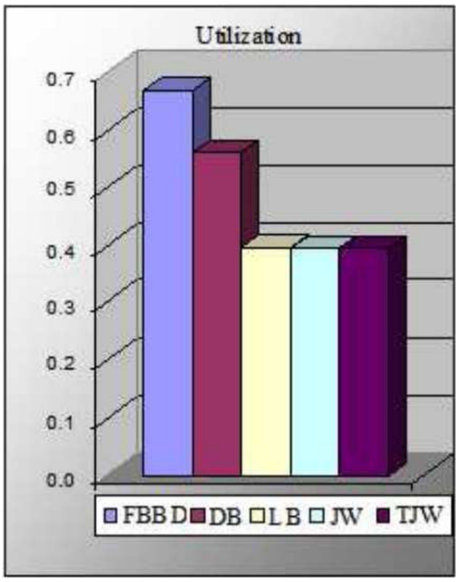


***Fig.14:*** *illustrates the utilization comparison between fuzzy, backpressure and exponential backoff in double-leaky bucket (FBBD), double-leaky bucket (DB), leaky bucket (LB), jumping window (JW) and triggered jumping window (TJW) systems as the policing mechanism at burst: silence rate of 100:100.*

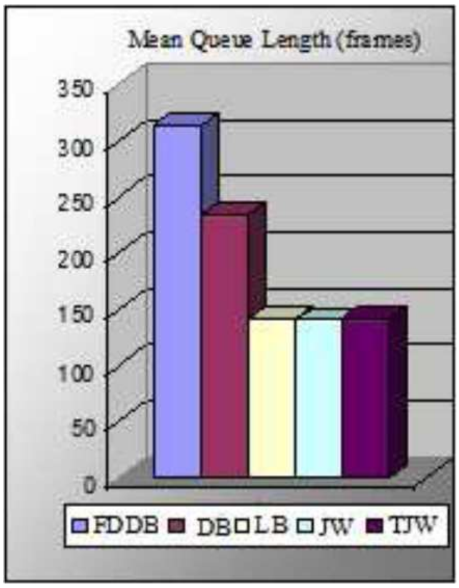


***Fig.15:*** *illustrates the mean queue length comparison between fuzzy, backpressure and exponential backoff in double-leaky bucket (FBBD), double-leaky bucket (DB), leaky bucket (LB), jumping window (JW) and triggered jumping window (TJW) systems as the policing mechanism at burst: silence rate of 100:100.*

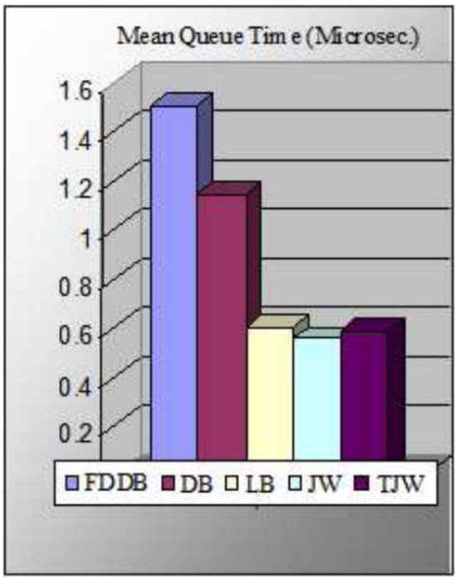


***Fig.16:*** *illustrates the mean queue time comparison between a fuzzy, backpressure and exponential backoff in double-leaky bucket (FBBD), double-leaky bucket (DB), leaky bucket (LB), jumping window (JW) and triggered jumping window (TJW) systems as the policing mechanism at burst: silence rate of 100:100.*

## References


[1] S. Lekcharoen and C. Jittawiriyanukoon, "Scheduling and qos for backpressure over telecommunications network," in *International Conference on Telecommunications and Computer Networks (TCN'05)*, pp. 56–60, 2005.

[2] S. Lekcharoen and C. Jittawiriyanukoon, "Deadlock of avoidance backo schemes in policing mechanisms," in *IEE Proceedings of the International Mobility Conference (MobiCon '05)*, pp. 1–5, 2005.

[3] L. Ma and W. Nutt, "Frequency operators for condensative queries over data streams," in *IEE International Conference on e-Business Engineering (ICEBE2005)*, pp. 518–525, 2005.

[4] A. AlWehaibi, A. Agarwal, S. AlWakeel, N-E. Rikli, and A. ElHakeem, "Performance behavior evaluation of internet congestion control policing mechanisms," in *IEEE Canadian Conference on Electrical and Computer Engineering (IEEE CCECE2003)*, vol. 2, pp. 1017–1020, 2003.

[5] H. H. Takada and A. Anzaloni, "A multifractual traffc policing mechanism," in *IEEE Communications Letters*, vol. 10, pp. 120–122, 2006

[6] L. Bajzik, L. Korossy, K. Veijalainen, and C. Vulkan, "Cross-layer backpressure to improve hsdpa performance," in *IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–5, 2006.

[7] G. Sapountzis and M. Katevenis, "Benes switching fabrics with o(n)-complexity internal backpressure," in *Workshop on High Performance Switching and Routing (HPSR 2003)*, pp. 11–16,2003.

[8] A.Zolfaghari and S. J. Golestani, "Performance study of backpressured packet networks using a soft fluid flow model," in *Canadian Conference on Electrical and Computer Engineering*, pp. 55–60, 2005.

[9] S. Lekcharoen and C. Jittawiriyanukoon, "Qos sensitive fuzzy backo schemes in policing mechanisms," in *International Conference on Mobile Technology, Applications and Systems*, 2005.

[10] K. Takahashi, "Performance analysis of a backo algorithm using finish tags in ieee 802.11 networks," in *Conference Asia-Pacific*, pp. 600–604.

[11] Byung-Jae Kwak, Nah-Oak Song, and Leonard E. Miller, "Performance analysis of exponential backoff," in *IEEE/ACM Transactions on Networking(TON)*, pp. 343–355.

[12] Sebastia Galms and Ramon Puigjaner, "The response time distribution of a discrete-time queue under a generalized batch arrival process," in *Proceedings of the 3rd international IFIP/ACM Latin American conference on Networking (LANC)*, pp. 31–39.

[13] L. Zhang, A. Vaidhiyanathan, and K. R. Subramanian, "Characterisation of singapore one atm network," in *IEE Proceedings Communications.*

[14] Sebastia Galms and Ramon Puigjaner, "An algorithm for computing the mean response time of a single server queue with generalized on/off trac arrivals," in *ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS '03).*

[15] Vijay Raghunathan, Saurabh Ganeriwal, Mani Srivastava, and Curt Schurgers, "Energy ecient wireless packet scheduling and fair queuing," in *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 13, pp. 3–24.

[16] M. Salamah and H Lababidi, "A novel trac policing mechanism for atm networks," in *International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 411–415.

[17] Martin A. Brown, "Double leaky bucket meter," http://linuxip.net/gl/tcng/node55.html.

[18] Cisco, "Cisco ons 15454 sonet/sdh ml−series multilayer ethernet card software feature and configuration guide, release 4.6," http://www.cisco.com/univercd/cc/td/doc/product/ ong/15400/r46docs/r46mlios/qos_cf.htm.

[19] S. Lekcharoen and C. Jittawiriyanukoon, "Bandwidth contention with backpressure scheme over vdsl," in *Network. International Journal of Computer Science and Network Security*, pp. 12–20.

[20] P. Siripongwutikorn, S. Banerjee, and D. Tipper, "Fuzzy-based adaptive bandwidth control for loss guarantees," in *IEEE Transactions on Neural Networks*, pp. 1147–1162.

[21] Rose Qingyang Hu and David W. Petr, "A predictive self-tuning fuzzy-logic feedback rate controller," in *IEEE/ACM transactions on networking*, pp. 697–709.

[22] Chung-Ju Chang, Terng-Yuan Liu, and Fang-Ching Ren, "Fuzzy/neural congestion control for integrated voice and data ds-cdma/frma cellular networks," in *IEEE Journal on Selected Areas in Communications*, pp. 283–293.

[23] D-Link Corporation/D-Link Systems, "D-link dev3001 datasheet," http://www.dlink.pl/docs/datenblatt/DEV3001_datasheet_en.pdf.

[24] ETSI, "The european telecommunications standards institute," http://www.etsi.org.

**Somchai Lekcharoen** received the M.Sc. degree in Information Technology from King Mongkut's University of Technology, Thonburi Campus, Thailand. He is currently a Ph.D. candidate in Computer Science at Assumption University of Thailand. His research interests include VDSL network, policing mechanisms, backoff time, fuzzy logic and network of queue.