

DYNAMIC BANDWIDTH ALLOCATION SCHEME IN LR-PON WITH PERFORMANCE MODELLING AND ANALYSIS

Tony Tsang¹

¹Department of Computer Science and Computer Engineering,
La Trobe University, Melbourne, Australia

ABSTRACT

We consider models of telecommunication systems that incorporate probability, dense real-time and data. We present a new formal abstraction method for computing minimum and maximum reachability probabilities for such models. Our approach uses strictly local formal abstract steps to reduce both the size of abstract specifications generated and the complexity of operations needed, in comparison to previous approaches of this kind. A selection of large case studies are implemented the techniques and evaluate, which include some infinite-state probabilistic real time models, demonstrating improvements over existing tools in several cases. The capacity of metro and access networks are extended the reach and split ratio of the conventional Long - Reach Passive Optical Networks (LR-PONs). The efficient solutions of LR-PONs are appeared in feeder distances around 100km and high split ratios up to 1000-way . Among many existing approaches, one of the most effective options to improve network performance in LR-PONs are the multi-thread based dynamic bandwidth allocation (DBA) scheme where several bandwidth allocation processes are performed in parallel is considered. Without proper intercommunication between the overlapped threads, multi-thread DBA may lose efficiency and even perform worse than the conventional single thread algorithm. Real Time Probabilistic Systems are used to evaluate a typical PON systems performance. This approach is more convenient, flexible, and lower cost than the former simulation method, which do not need develop special hardware and software tools. Moreover, how changes in performance depend on changes in the particular modes can be easily analysis by supplying ranges for parameter values. The proposed algorithm with traditional DBA is compared, and shows its advantage on average packet delay. The key parameters of the algorithm are analysed and optimized, such as initiating and tuning multiple threads, inter -thread scheduling, and fairness among users. The algorithms advantage in numerical results are decreased the average packet delay and improve network throughput under varying offered loads.

KEYWORDS

Long- reach Passive Optical Networks; Dynamic Bandwidth Allocation; Real Time Probabilistic Systems; Performance Analysis;

1. INTRODUCTION

Formal Modelling is a highly successful approach to the performance analysis of complex infinite-state systems. The basic idea is to construct a sequence of increasingly precise abstractions of the system to be verified, with each abstraction typically over-approximating its behaviour. Through a process of refinement which terminates once the abstraction is precise enough to verify the desired property of the system under analysis, construct the successive abstractions. Formal techniques have also been used to verify probabilistic systems, including those with real-time characteristics and continuous variables. Frequently, though, the high

complexity of both the abstractions involved and the operations needed to construct and refine, which hinder the practical implementations of these techniques.

In this article, the performance analysis of systems is targeted, whose behaviour incorporates both probabilistic and real-time aspects, and which include the manipulation of (potentially infinite) data variables. Real Time Probabilistic Systems (RTPS) are modelled, whose semantics are defined as infinite-state Markov decision processes (MDPs) [1]. A formal abstraction procedure is introduced for computing minimum and maximum reachability probabilities in RTPS. This provides outer bounds on reachability probabilities (i.e., a lower bound on the minimum probability or an upper bound on the maximum). In addition, the inner bounds and based on a stepwise concretization of adversaries of this abstract MDP are computed, yielding upper and lower bounds on minimum and maximum probabilities, respectively. Untimed models used concretization. The key difference in our work is kept the abstraction small by using local refinement and simplification operations, so as reducing the need for expensive operations such as Craig interpolation.

The formal abstraction loop repeated to attempt constructing a concrete adversary of the RTPS. Based on the exploration of the part of the state space, which is the current abstract adversary can be concretized. In each exploration step, an inconsistency is encountered, in which case we derive a formal operation and restart. Otherwise, the constructed adversary is numerically solved, giving inner bounds on the desired probability values. The difference between upper and lower bounds is smaller than a specified threshold is terminated the formal abstraction loop. The formal abstraction approach are implemented; deploy it on various large case studies, and compare to the probabilistic verification tools PRISM [2], are illustrated to improve performance in many cases. Real Time Probabilistic Systems enable to verify and containing both real-time behaviour and infinite data variables, which this tool can be handle.

The high bandwidth of high-capacity communication systems (e.g., optical fibre) are bring closer to the remote end users with affordable costs, the long-reach passive optical network (PON) was introduced and studied over the past few years. The LR-PON are developed coinciding with the momentum in the integration of metro and access networks, as well as the integration of wire line and wireless networks. The hierarchical architecture of the telecommunication network are simplified and can significantly reduce both capital expenditure (CapEx) and operational expenditure (OpEx) by lowering the total number of active sites, such as points of presence (PoPs) and local exchanges (LXs). Three parts are consisted in a typical PON: the optical line terminal (OLT) at the telecom central office (CO), optical network units (ONUs) located at end users' premises and remote nodes (RNs) in between, as shown in Figure 1. To provide high bandwidth to a large number of users in an LR-BAN, hybrid architecture should be deployed, exploiting both time-division multiplexing (TDM), where a single wavelength channel is shared among multiple users and wavelength-division multiplexing (WDM), which supports multiple wavelengths to increase capacity.

However, the traditional PONs are relatively small span and inflexible topology of restrict a Broadband Access Networks from reaching remote users. Extend the coverage of PONs in both directions are do the research. In the network direction, the long-reach PON (LR-PON) [3] it is no longer a passive structure has been proposed to extend the coverage from 20 to 100 km by exploiting advanced technologies such as reflective semiconductor optical amplifiers (R-SOAs) and burst-mode receivers. Instead of copying the traditional "tree-and-branch" topology in PONs, researchers are also investigating a "ring-and-spur" topology for LR-PONs [3], as shown in Figure 2, where a PON segment consists of the OLT and a set of ONUs operating on a wavelength set; each PON segment exhibits a logical tree topology; several such PON segments coexist; and the OLT and ONUs are connected through a fibre ring, and RNs are deployed on the

ring. Although, additional RNs are introduced increasing the overall complexity of the network, an advantage of this design is that the metro synchronous optical network / digital hierarchy (SONET/SDH) ring networks with fibre can be reused already deployed by substituting the RN equipment and thus achieve great savings in CapEx. This ring-and-spur topology can not only cover a larger area than the traditional tree topology, but also provide the broadband access networks two-dimensional coverage for failure protection, which is very important today as strict quality of service (QoS) may be specified in a user's service level agreement (SLA). The introduction of the paper has been explained the nature of the problem, previous work, purpose, and the contribution of the paper. The contents of each section may be provided to understand easily about the paper.

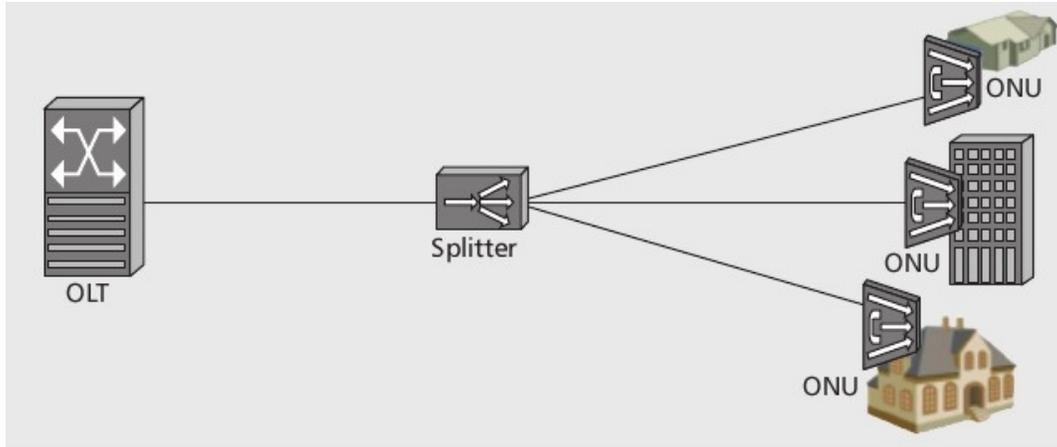


Figure 1. PON Architecture

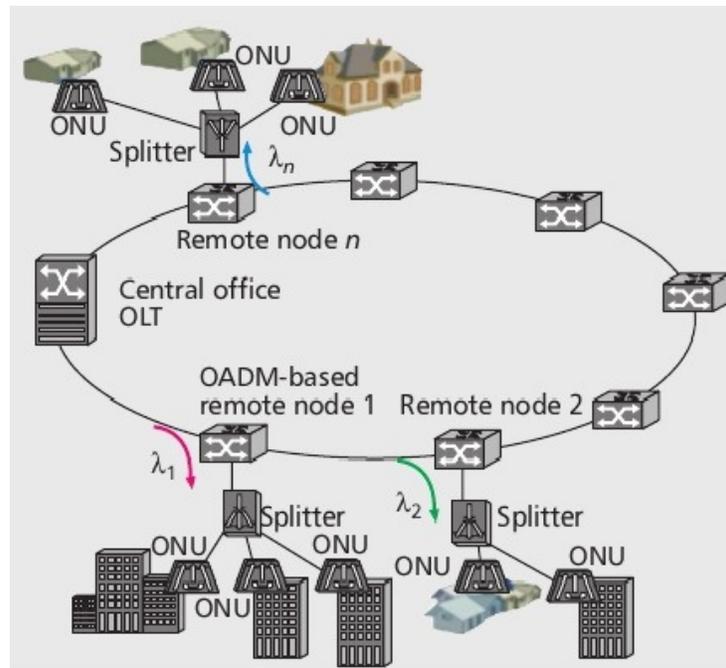


Figure 2. Ring and Spur LR-PON

The PON standards have adopted Time-division multiple access (TDMA), Ethernet PON (EPON) and Gigabit PON (GPON), to share the optical capacity among subscribers by assigning

different timeslots for each user. Centralized dynamic bandwidth allocation (DBA), in which the OLT at the CO arbitrates time-division access to the shared upstream channel. The round-trip time (RTT) are depended on performance of centralized allocation, since it imposes a delay on the OLT-ONUs bandwidth allocation control loop. This delay is not as significant in traditional PONs as it is in LR-PONs, where the reach extension may cause the RTT to grow from today's 200 μ s (20 km reach) to 1 ms (100 km reach). The performance of centralized DBA is ultimately degraded with this is increased.

multi-thread polling and inter-thread scheduling are reviewed and analysed, which is a recently proposed bandwidth allocation algorithm for LR-PONs, and traditional interleaved polling is compared. Unexpectedly, the overall packet delay performance of the latter is found better than the recently proposed multi-thread polling scheme. Our investigation highlights how multi-thread polling looks compelling for LR-PONs compared to single thread polling only when the latter is assumed to be scheduled offline. The offline scheduling is used a single thread leaves a long idle period that the multi-thread scheme utilizes by creating additional threads. However, while traditional inter-thread polling is a single-thread scheme, online scheduling is exploited with extremely small idle times. The demonstration of key results are that inter-thread polling with online scheduling better reduces the overall packet delay in addition to better utilizing the upstream channel. In the paper, some inaccuracies are pointed out in the multi-thread algorithm, and suggest some modifications.

The rest of the paper is organized as follows. In Section II we give a brief introduction to Real Time Probabilistic Systems. Next, we review PRISM Probabilistic Model Checker in Section III, explaining how its performance analysis is highly affected by extending the network span. In Section IV, we review multi-thread polling and discuss the different aspects of its algorithm, highlighting its major differences from enhanced inter-thread scheduling. Section V presents illustrative numerical results, and Section VI concludes the study.

2. REAL TIME PROBABILISTIC SYSTEMS

The definition of real time probabilistic system is reviewed in this section, both nondeterministic and stochastic behaviour are exhibited in the modelling framework for real-time systems. The extending classical timed automata with discrete probability distributions over edges derive the formalism. First, standard notation is introduced for clocks and zones of timed automata, and then we proceed to the definition of probabilistic timed automata [4, 5, 6]. At the end of this section, Probabilistic Timed Computation Tree Logic (PTCTL) is introduced as a probabilistic timed temporal logic for the specification of properties of probabilistic timed automata.

2.1. Clocks and Zones

Let X is a finite set of variables called clocks which take values from the time domain $\mathfrak{R}_{\geq 0}$ (non-negative reals). A function $v: X \rightarrow \mathfrak{R}_{\geq 0}$ is referred to as a clock valuation. The set of all clock valuations is denoted by $\mathfrak{R}_{\geq 0}^X$. For any $v \in \mathfrak{R}_{\geq 0}^X$ and $t \in \mathfrak{R}_{\geq 0}$, we use $v + t$ to denote the clock valuation defined as $(v + t)(x) = v(x) + t$ for all $x \in X$. We use $v[X := 0]$ to denote the clock valuation obtained from v by resetting all of the clocks in $X \subseteq X$ to 0, and leaving the values of all other clocks unchanged; formally, $v[X := 0](x) = 0$ if $x \in X$ and $v[X := 0](x) = v(x)$ otherwise.

The set of zones of X , written $Zones(X)$, is defined inductively by the syntax:

$$\zeta ::= x \leq d \mid c \leq x \mid x + c \leq d \mid \neg \zeta \mid \zeta \vee \zeta$$

Where $x, y \in X$ and $c, d \in N$. As usual, $\zeta_1 \wedge \zeta_2 = \neg(\neg\zeta_1 \vee \neg\zeta_2)$ and strict constraints can be written using negation, for example $x > 2 = \neg(x \leq 2)$.

The clock valuation v satisfies the zone ζ , written $v \triangleright \zeta$, if and only if ζ resolves to true after substituting each clock x with the corresponding clock value $v(x)$ from v . Intuitively, the semantics of a zone is the set of clock valuations (subset of $\mathfrak{R}_{\geq 0}^X$) which satisfy the zone. Note that more than one zone may represent the same set of clock valuations (for example, $(x \leq 2) \wedge (y \leq 1) \wedge (x \leq y + 2)$ and $(x \leq 2) \wedge (y \leq 1) \wedge (x \leq y + 3)$). We hence forth consider only canonical zones, which are zones for which the constraints are as tight as possible. For any valid zone $\zeta \in ZonesX$, there exists a $O(X)$ algorithm to compute the (unique) canonical zone of ζ . This enables us to use the above syntax for zones interchangeably with semantic, set-theoretic operations.

We require the following classical operations on zones [4, 6]. For zones $\zeta, \zeta' \in Zones(X)$ and subset a of clocks $X \subseteq X$, let:

$$\zeta \prec \zeta' = \{v \mid \exists t \geq 0. (v + t \triangleright \zeta \wedge \forall t' \leq t. (v + t' \triangleright \zeta' \vee \zeta'))\}$$

$$[X := 0]\zeta = \{v \mid v[X := 0] \triangleright \zeta\}$$

$$\zeta[X := 0] = \{v[X := 0] \mid v \triangleright \zeta\}.$$

The zone $\zeta \prec \zeta'$ contains the clock valuations that can, by letting time pass, reach a clock valuation in ζ and remain in ζ' until ζ is reached. The zone $[X := 0]\zeta$ contains the clock valuations which result in a clock valuation in ζ when the clocks in X are reset to 0. The zone $\zeta[X := 0]$ contains the clock valuations which are obtained from clock valuations in ζ by resetting the clocks in X to 0.

2.2. Syntax and Semantics of Real Time Probabilistic Systems

We now present the formal syntax of Real Time Probabilistic Systems.

Timed Action

A tuple $\langle \alpha, \lambda, T \rangle$ of timed action consisting of the type of the action α , the rate of the action λ and temporal constraint of the action T . The kind of action is denoted the type, such as transmission of data packets, while the rate indicates the speed at which the action occurs from the view of an external observer. The random variables rate use to denoted specifying the duration of the actions. The different types of probability distribution function define the actions such as Exponential, Poisson, Constant, Geometric and Uniform distribution. Moreover, each transition is also bounded by a temporal constraint. In this section, Real Time Probabilistic Systems briefly introduce some basic notations and operation semantics. The syntax of Real Time Probabilistic Systems is defined as follows:

$$P ::= \text{stop} \mid \langle \alpha, \lambda, T \rangle . P \mid P + Q \mid P \oplus_r T Q \mid P \triangleright \langle L, T \rangle Q \mid P \nabla \Delta P, T Q \mid P/L \mid A$$

The conventional stochastic process algebra operators and the additional operations are described in the following:

- stop is an inactive process.
- $\langle \alpha, \lambda, T \rangle . P$, which stands for a prefix operator, where the type of the action is a probability distribution function (pdf) type α , with the activity rate denoted by λ , and the temporal constraint of component is T .
It subsequently behaves as P . Sequences of actions can be combined to build up a time constraint for an action. The time constraint T is defined as above.
- $P + Q$ is choice combinatory capturing the possibility of competition or selection between different possible Activities. It represents a system which may behave either as P or as Q . All the current actions P and Q are enabled. The first action to complete distinguishes one of the processes. The other process of the choice is discarded. The system will then behave as the derivative resulting from the evolution of the chosen process.
- $P \oplus_r T Q$ denotes the probabilistic choice with the conventional generative interpretation, thus with probability r the process behaves like P and with probability $1 - r$ it behaves like Q bounded with the time constraint T .
- $P \triangleright \langle L, T \rangle Q$ is a cooperation, in which the two actions P and Q are parallel, synchronizing on all activities whose type is in the cooperation set L of action types. The lifetime of two actions is the time constraint T . These two actions are disabled when the time constraint expires. Any action whose type is not in L will proceed independently. As a syntactic convenience the parallel combinator is defined by $\triangleright \langle \square, T \rangle$, where the cooperation set L is empty and the lifetime of two actions is T .
- $P \nabla \Delta P, T Q$ is a unary operator which returns the set of actions that meet the temporal predicate condition specified by T . P consists of several predicates combined with the boolean connectives: 'And', 'Or', Exclusive-Or (EXOR) and 'Not'. $\nabla \Delta \text{And}, T$ means both actions can occur during the interval T . $\nabla \Delta \text{Or}, T$ means that one or both actions can occur during the interval T . $\nabla \Delta \text{EXOR}, T$ means that one of these actions occurs; it immediately determines whether P or Q can subsequently occur during the triggered interval T . $\nabla \Delta \text{Not}, T$ means that both actions do not occur during the interval T .
- P/L is a hiding operation, where the set L of visible action types identifies those activities which are to be considered internal or private to the component. These activities are not visible to an external observer, nor are they accessible to other components for cooperation.
- $A := P$ is a countable set of constants.

2.3. Probabilistic Timed Computation Tree Logic

We now describe Probabilistic Timed Computation Tree Logic (PTCTL) which can be used to specify properties of probabilistic timed automata. This logic is a combination of two extensions of the temporal logic CTL [7], the timed logic TCTL [8] and the probabilistic logic PCTL [9]. The logic TCTL employs a set of formula clocks, Z , disjoint from the clocks X of the probabilistic timed automaton under study. Formula clocks are assigned values by formula clock valuations $\square \in RZ \geq 0$. The logic TCTL can express timing constraints and includes the reset

quantifier $z. \square$, used to reset the formula clock z so that the formula \square is evaluated from a state at which $z = 0$. PTCTL is obtained by enhancing TCTL with the probabilistic quantifier $P \sim \lambda[\cdot]$ from PCTL and removing the path quantifiers \exists and \forall .

The syntax of PTCTL is defined as follows:

$$\square ::= a \mid \zeta \mid \neg \square \mid \square \vee \square \mid z. \square \mid P \sim \lambda[\square \cup \square] \mid P \sim \lambda[\square \vee \square]$$

where $a \in AP$, $\zeta \in \text{Zones}(X \cup Z)$, $z \in Z$, $\sim \in \{\leq, <, >, \geq\}$ and $\lambda \in [0, 1]$. We use the abbreviations $\diamond \square$ and $2\square$ for true $U\square$ and false $V\square$ respectively. In PTCTL we can express properties such as:

- with probability strictly greater than 0.96, the system delivers packet 1 within 5 time units and does not try to send packet 2 in the meantime, which is represented by $z.P > 0.96[\text{packet2unsent} U(\text{packet1delivered} \wedge (z < 5))]$;
- with probability at least 0.93, the system clock x does not exceed 3 before 8 time units elapse, which is represented as $z.P > 0.93[(x \leq 3) U (z = 8)]$;
- the system remains up after the first 60 time units have elapsed with probability greater than 0.97, represented as $z.P > 0.97[2(\text{system up} \vee (z \leq 60))]$.

Next, we define the semantics of PTCTL. We write v, δ to denote the composite clock valuation in obtained from $v \in \mathfrak{R}_{\geq 0}^X$ and $\delta \in \mathfrak{R}_{\geq 0}^Z$. Given a state and formula clock valuation pair (l, v) , δ , zone ζ and duration t , by abuse of notation we let $(l, v), \delta \triangleright \zeta$ denote $v, \delta \triangleright \zeta$, and $(l, v) + t$ denote $(l, v + t)$.

3. PRISM PROBABILISTIC MODEL CHECKER

Model checking for several types of probabilistic models are provided in Prism [2]: discrete and continuous time Markov chains and Markov decision processes, as well as a modelling language in which to express them. The tool has been widely taken up and used for quantitative verification in a broad spectrum of application domains, from wireless communication protocols to quantum cryptography to systems biology.

3.1. Overview of Functionality

We begin with a brief overview of the current functionality of the prism tool. Items are the new or improved features in version 4.0, which are described in more detail in the remainder of the paper.

- many types of probabilistic models are modelled and constructed, now including probabilistic timed automata; all can be augmented with costs or rewards, in the case of PTAs yielding the model of priced probabilistic timed automata;
- a wide range of quantitative properties are the model checking, expressed in a language that subsumes the temporal logics PCTL, CSL, LTL and PCTL*, as well as extensions for quantitative specifications and costs/rewards;

- both symbolic (BDD-based) and explicit- state multiple are model checking engines; and a variety of probabilistic verification techniques, such as symmetry reduction and quantitative abstraction refinement;
- with support for statistical model checking methods are the discrete-event simulator, including confidence-level approximation and acceptance sampling;
- Model import options, e.g. From SBML (systems biology mark-up language);
- Optimal adversary/strategy generation for nondeterministic models;
- With model editor is a GUI, simulator and graphing, or command-line tool;
- Probabilistic models and associated properties in benchmark suite.

4. MULTI-THREAD SCHEDULING SCHEME OF DYNAMIC BANDWIDTH ALLOCATION

In this section, we briefly describe two different inter-thread scheduling algorithms presented in [10, 11, 12] in Figure 3. The one proposed in [12] is referred to as newly arrived plus (NA+), which aims to overcome the over granting problem in multithread schemes. We name the reviewed algorithm (i.e., in [10]) as subsequent requests plus (SR+), since it is able to reduce grant delay by utilizing the information regarding the requests received in the overlapped subsequent DBA processes. Furthermore, we integrate the key advantages of these two existing approaches and propose an enhanced inter-thread scheduling (EIS) algorithm in order to further improve the DBA performance. All the notations used for the description of the algorithms are listed as follows:

M : Total number of active ONUs

N : Total number of threads

$B_{i,n}$: Actual bandwidth demand for ONU_i in DBA process n

B_i^{min} : Predefined guaranteed minimum bandwidth for ONU_i

$R_{i,n}$: Bandwidth reported from ONU_i in DBA process n (i.e., the buffer status of ONU_i at time $t_{i,n}$)

$R'_{i,n}$: OLT based on $R_{i,n}$ are recalculated bandwidth request, and the associated inter-thread scheduling schemes

$G_{i,n}$: Resulting grant to ONU_i responding to $R_{i,n}$

$g_{i,n}$: Actual bandwidth for data transmission between $t_{i,n-1}$ and $t_{i,n}$ at ONU_i

$A_{i,n}$: New traffic that arrived to ONU_i between $t_{i,n-1}$ and $t_{i,n}$

$U_{i,n}$: Unused time slots occurrence during the nth DBA process at ONU_i

$C_{i,n}$: A term to compensate backlogged traffic

The overall system specification becomes ONU and OLT:

OLT := ONU₀ < λ_1 > ONU₁ < λ_2 > ONU₂ < λ_3 > ••••

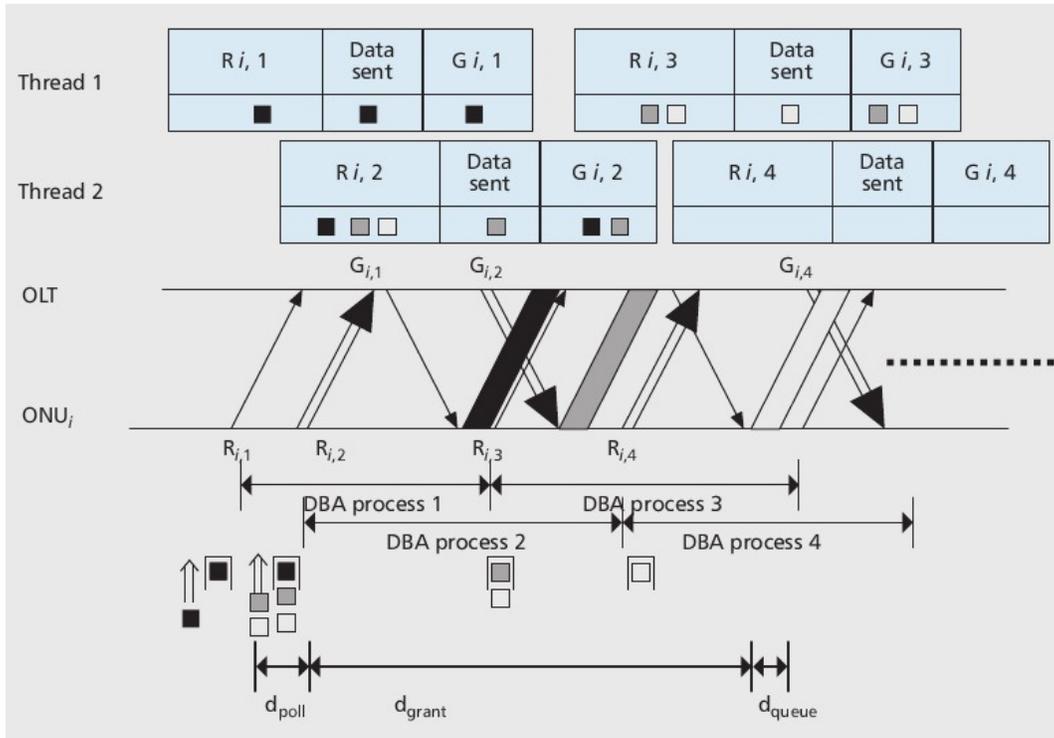


Figure 3. Multi-Thread DBA processes between an OLT and an ONU.

4.1. Newly Arrived Plus

Over-granting can be a serious performance issue in a typical LR-PON scenario when a multi-thread DBA scheme is used. To overcome this problem, Newly Arrived Plus (NA+) has been proposed in the literature [12]. The main idea of NA+ is that each thread is mainly responsible for the bandwidth allocation for the traffic that has arrived since the initiation of the last DBA process. In this way load is evenly distributed among the different threads. NA+ can compensate for the backlogged traffic at a specific ONU_i in DBA process *n* using the information gathered from some previously completed DBA process. A compensation mechanism for backlogged traffic is important because a DBA allocation process may not be able to cater for all the requested bandwidth in a given DBA process. Furthermore, EPON does not support frame fragmentation, and unused time slots (UTS) may occur. NA+, presented in [12], also includes a compensation mechanism for the UTS in EPONs. The effectiveness of NA+ can be clearly observed from [12], where an over-reporting problem is eliminated by allocating bandwidth only once. For instance, the first arrived packet (i.e., the black packet) is reported in both $R_{i,1}$ and $R_{i,2}$, but the corresponding bandwidth granting is only performed once (i.e., in $G_{i,1}$) because $R'_{i,2}$ excludes the bandwidth of the black packet, which has already been granted in $G_{i,1}$. This allows improving network throughput as well as delay and jitter. The operation in Newly Arrive Plus multi-thread DBA Scheme for ONU_i specified as follows:

ONU_i := < sent, λ_1 > . Report($R_{i,1}$) \triangleright L1 < get, λ_1 > . OLT ;

ONU_i := < poll, delay > . Report($R_{i,2}$) \triangleright L2 < get, λ_2 > . OLT ;

$ONU_i := \langle \text{grant}_{NA+}, \text{delay}_{NA+i} \rangle . \text{Data}_i \triangleright \langle L3 \langle \text{get}, \lambda_3 \rangle . \text{OLT} ;$

$ONU_i := \langle \text{sent}, \text{delay}_i \rangle . \text{Report}(R)_{i,3} \triangleright \langle L4 \langle \text{get}, \lambda_4 \rangle . \text{OLT} ;$

$ONU_i := \langle \text{queue}, \text{delay}_i \rangle . \text{Data}_i \triangleright \langle L5 \langle \text{get}, \lambda_5 \rangle . \text{OLT} ;$

The operation in DBA Scheme for OLT specified as follows:

$OLT := \langle \text{sent}, \text{delay}_1 \rangle . \text{Result}(G)_{i,1} \triangleright \langle L1 \langle \text{get}, \lambda_1 \rangle . ONU_1 ;$

$OLT := \langle \text{sent}, \text{delay}_2 \rangle . \text{Result}(G)_{i,2} \triangleright \langle L2 \langle \text{get}, \lambda_2 \rangle . ONU_2 ;$

$OLT := \langle \text{sent}, \text{delay}_3 \rangle . \text{Result}(G)_{i,3} \triangleright \langle L3 \langle \text{get}, \lambda_3 \rangle . ONU_3 ;$

4.2. Subsequent Requests Plus

In Subsequent Requests Plus (SR+) [10, 11], an inter-threading mechanism is introduced that allows bandwidth allocation even for the packets that arrived at an ONU queue after issuing the report for the current DBA process. Therefore, this mechanism to perform bandwidth allocation for subsequent requests can reduce packet delay noticeably in many cases. [10] gives an example of SR+ employed in a two-thread DBA. Before calculating the bandwidth allocation for the $R_{i,1}$, $R_{i,2}$ is also available. So some packets queuing at the ONU_i after the transmission of $R_{i,1}$ do not have to wait for the receipt of $G_{i,2}$ to be transmitted because they have been considered in $R'_{i,1}$ before $G_{i,2}$ is issued and hence could be transmitted as soon as $G_{i,1}$ is received. This clearly reduces the d_{grant} for the light grey packet shown in Fig. 3b. It should be noted that two packets are reported in $R_{i,2}$, but there is still a risk that not both of them can be transmitted after receiving $G_{i,1}$, because the bandwidth request cannot always be satisfied. Furthermore, the over-granting problem cannot be avoided in SR+. For instance, the black packet is considered twice in $R'_{i,1}$.

The operation in Subsequent Requests Plus multi-thread DBA Scheme for ONU_i specified as follows:

$ONU_i := \langle \text{sent}, \lambda_1 \rangle . \text{Report}_{i,1} \triangleright \langle L1 \langle \text{get}, \lambda_1 \rangle . \text{OLT} ;$

$ONU_i := \langle \text{poll}, \text{delay}_i \rangle . \text{Report}_{i,2} \triangleright \langle L2 \langle \text{get}, \lambda_2 \rangle . \text{OLT} ;$

$ONU_i := \langle \text{grant}_{SR+}, \text{delay}_{SR+i} \rangle . \text{Data}_i \triangleright \langle L3 \langle \text{get}, \lambda_3 \rangle . \text{OLT} ;$

$ONU_i := \langle \text{sent}, \text{delay}_i \rangle . \text{Report}_{i,3} \triangleright \langle L4 \langle \text{get}, \lambda_4 \rangle . \text{OLT} ;$

$ONU_i := \langle \text{queue}, \text{delay}_i \rangle . \text{Data}_i \triangleright \langle L5 \langle \text{get}, \lambda_5 \rangle . \text{OLT} ;$

4.3. Enhanced Inter-Thread Scheduling

Our proposed enhanced inter-thread scheduling (EIS) is a hybrid scheme capturing key ideas from both NA+ and SR+ to improve DBA performance in a multi-thread environment. For bandwidth granting, we consider a newly arrived packet as in NA+ to effectively eliminate the over-reporting problem, while bandwidth requested in the subsequent DBA processes is also taken into account as in SR+ if it arrives before the calculation of the bandwidth granted in the current DBA process. Pseudo code for the EIS

scheme is outlined in [11]. Benefits of employing EIS can be seen clearly from [11] where the grant of the packet with the white colour can be significantly reduced compared to NA+ and SR+. This is because $R_{i,1}$ can correctly reflect the current bandwidth request. The traffic reported in $R_{i,2}$ can be taken into account when issuing $G_{i,1}$. Furthermore, by eliminating the over-reporting for the black packet (which has been reported twice), both packets that arrived after $R_{i,1}$ at ONU_i can be transmitted after receiving $G_{i,1}$. Performance in a PON system is sensitive to DBA computation time, particularly in a multi-thread environment where several threads in parallel increase the complexity. In this regard, our proposed EIS scheme does not require any additional inter-thread communication mechanism compared to NA+ and SR+. In other words, there is no explicit control information that needs to be exchanged between the threads (except for the usual report messages), so the DBA complexity grows linearly relative to a single-thread implementation. However, due to DBA control messages in each thread, there is still a linear increase in overhead with respect to the employed number of threads. Consequently, it may lead to degraded performance by increasing the number of threads beyond a certain level, as the added control messages introduced by the extra threads could decrease the bandwidth utilization and affect the other performance measures such as delay and jitter. As shown in the next section, in most of the evaluated cases applying two or three threads is optimal to gain the maximum benefit depending on the specific reach scenario as well as the employed multi-thread algorithm. The operation in Enhanced Inter-Thread Scheduling for ONU_1 specified as follows:

$ONU_i := \langle \text{sent}, \lambda_1 \rangle . \text{Report}_{i,1} \triangleright \langle L_1 \langle \text{get}, \lambda_1 \rangle . \text{OLT} ;$
 $ONU_i := \langle \text{poll}, \text{delay}_i \rangle . \text{Report}_{i,2} \triangleright \langle L_2 \langle \text{get}, \lambda_2 \rangle . \text{OLT} ;$
 $ONU_i := \langle \text{sent}, \text{delay}_i \rangle . \text{Report}_{i,3} \triangleright \langle L_3 \langle \text{get}, \lambda_3 \rangle . \text{OLT} ;$
 $ONU_i := \langle \text{grantEIS}, \text{delayEIS}_i \rangle . \text{Data}_i \triangleright \langle L_4 \langle \text{get}, \lambda_4 \rangle . \text{OLT} ;$
 $ONU_i := \langle \text{waitEIS}, \text{delayEIS}_i \rangle . \text{Data}_i \triangleright \langle L_5 \langle \text{get}, \lambda_5 \rangle . \text{OLT} ;$
 $ONU_i := \langle \text{queue}, \text{delay}_i \rangle . \text{Data}_i \triangleright \langle L_6 \langle \text{get}, \lambda_6 \rangle . \text{OLT} ;$

5. PERFORMANCE EVALUATION

Several well-known benchmarks for the cyclic redundancy check problem are evaluated on the procedure. First, a parametrisable case study is used to present evidence that the algorithm gives correct answers and then we present systematic comparisons with PRISM. The PRISM simulation framework is extended for sampling purposes. Because we use the same input language as PRISM, many off-the-shelf models and case studies can be used with our approach.

5.1. Performance Result

We simulate a LR-PON with M-ONUs. Single or multiple wavelengths are used to carry the upstream traffic, depending on the aggregated user requests. The problem is made clear and focused on packet delay, we assume that the M-ONUs share the same upstream channel (one upstream wavelength). If more users join the network, a new upstream

channel can be assigned to them and the same polling scheme is implemented on the new channel independently.

From the access side, an ONU are arrived packets from a user connected to the ONU. The ONU buffered the packets until the ONU is allowed to transmit them to the OLT. In our model, we consider RD to be the data rate of the access link from a user to an ONU, and RU to be the data rate of the upstream channel from ONU to OLT. Note that, if $RU < M \times RD$, there are exist the bandwidth utilization problem, because the system capacity is greater than the aggregated load from all ONUs. (The steady-state case is, but during temporary overload situations due to burst traffic, the load may be higher than capacity.) So, we choose $M = 16$ and RU and RD to be 1 Gbps and 100 Mbps, respectively.

Packets are generated in the form of Ethernet frames (64 to 1518 bytes) and they arrive at each ONU from the end user. The property of Internet traffic is reflected, the generated user traffic is self-similar by aggregating multiple sub-streams, each consisting of alternating Pareto-distributed on/off periods, with a Hurst Parameter of 0.8. The buffer size at each ONU is limited to 10 Mbytes.

The performance of our multi-thread polling algorithm are studied, the LR-PON are simulated in two different scenarios, where ONUs is 20 km and 100 km from the OLT, respectively. 0.5s sets in the initial interval of threads for the 100-km scenario and 0.3 s for 20-km scenario, which represent maximum cycle duration (RTT); the tuning threshold θ tune is set to be 5 in the simulation. The benefit of multi-thread is highlighted polling on the average packet delay, the constraints are dropped on single-thread polling by allowing it to ignore the fairness issue, and thereby the idle time is not counted.

From the results shown in Figure 4, high load (0.8) scenarios can observe ; delay performance improvement is tangible in EIS compared to NA+ and SR+. The performance can be noted in hit due to multi-threading through the increased overhead associated with report and grant messages. For this reason, all the multi-thread schemes perform worse when tested in the three thread case than in the two thread one. Increasing the number of threads by more than two affects the DBA performance are introduced extra control overhead. This performance degradation is more severe in SR+ than the other two schemes where the over-granting problem is well treated. Besides, in NA+ and EIS, the performance three thread cases become less obvious when the reach increases. Because of the longer reach, the increased overhead incurred by a three thread implementation is compensated more by the gain of introduction of one more thread than in the two thread case. In particular for the reach of 100 km and beyond for both EIS and NA+ three thread implementations provides better performance. High load (0.8) is true for the statement as similar behaviour for the different schemes can be seen in the high load situation (0.8) as well, although the absolute values for delay are larger because higher load leads to much longer average round-trip time (RTT) per thread. EIS still performs best, while NA+ performs the worst in the high load scenario. According to the results shown in Figure 4, three is generally the optimal number of threads are found for the schemes presented here in order to obtain the best delay performance except for reach of 100 km and beyond, where a three thread implementation performs better, as explained above, for both EIS and NA+. Furthermore, this value is also dependent on system

characteristics (number of ONUs, line rate, buffer size, etc.) and traffic profile (load, burstiness, etc.).

The different schemes of jitter performance are shown in Figure 5 for high (0.8) load, respectively. EIS can saw that has the lowest jitter for three thread cases. NA+ shows decent jitter performance but at the expense of worse delay performance, as shown earlier. SR+ with jitter performance is very dependent on the number of threads employed. The best performance is shown the three thread implementation at high load but it gets significantly worse for the three thread implementation where the over-granting problem is more intense. Apart from this observation, jitter performance for different schemes is close at high loads.

The last set of results shown in Figure 6 is for the throughput. EIS has the highest throughput under both low and high load conditions. As noted earlier for reach of 100 km and beyond, a three thread implementation is the most suitable option to achieve the best performance for the proposed EIS scheme. Here it is worth mentioning that the maximum throughput can be slightly higher than 0.497 (Fig. 6) since the exact value for the offered load is 0.495 under the high load scenario. Furthermore, packet loss can be minimized (or eliminated) by employing a larger buffer at the ONUs, which may significantly degrade the delay and jitter performance.

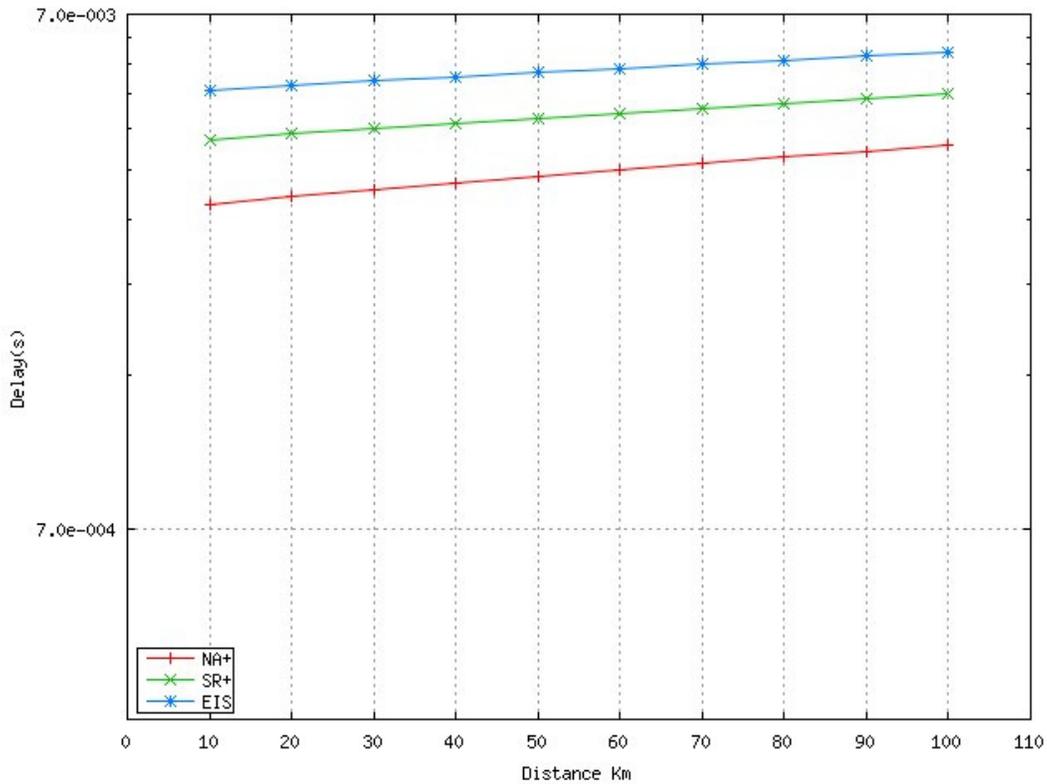


Figure 4. Packet Delay vs. Distance

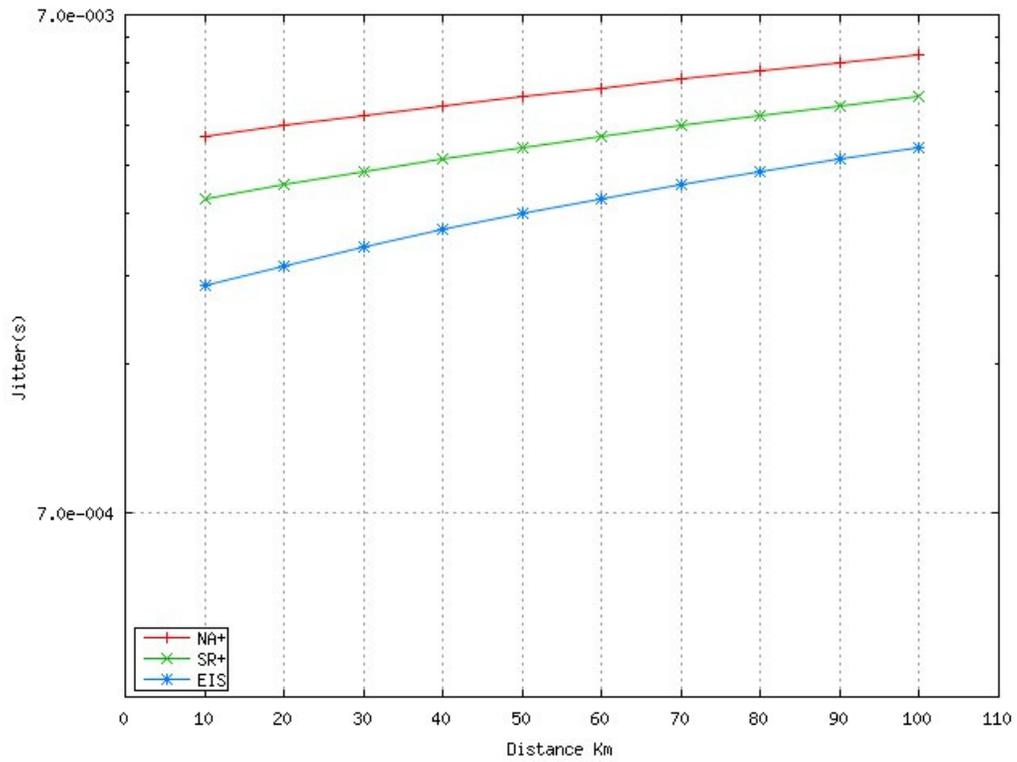


Figure 5. Jitter vs. Distance

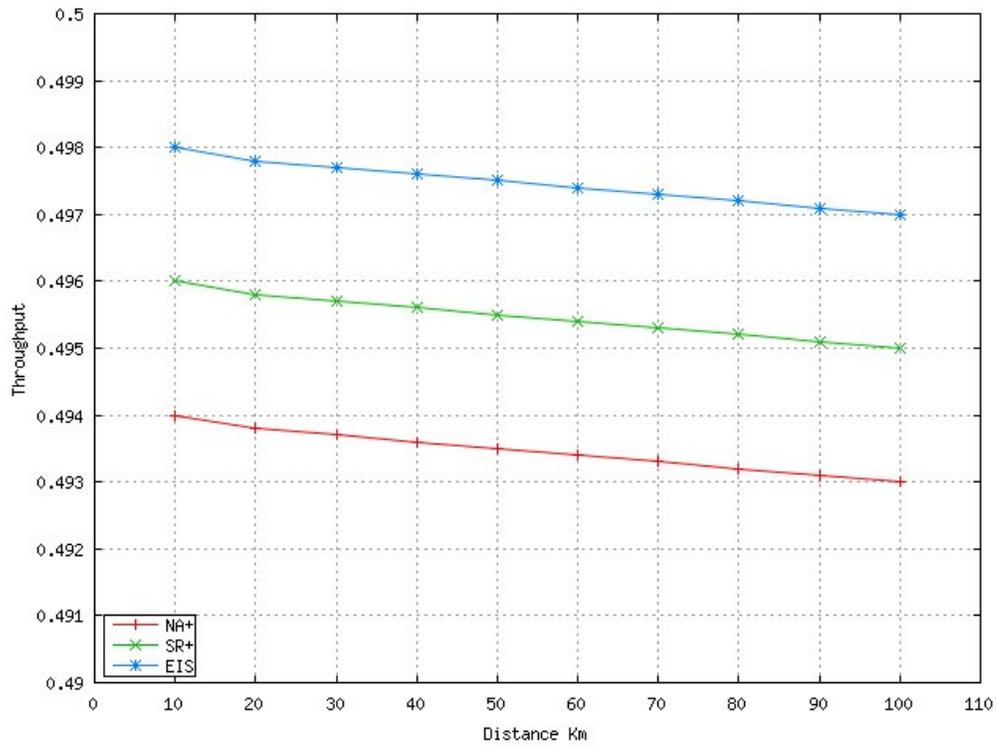


Figure 6. Throughput vs. Distance

6. CONCLUSIONS

Our approach tends to perform well, generating comparatively small abstract models using a relatively large number of (small, local) refinement steps. When the number of formal abstract models becomes particularly high, performance is degraded but, in our experiments, the tool still outperformed the game-based abstraction of PRISM. This gain derives from the use of simple operations on smaller abstractions and a reduction in the amount of numerical computation that needs to be performed. Perhaps unsurprisingly, our approach performs better on real-time models, than on the versions that have been discretised. We have presented a novel formal abstraction approach for the performance analysis of probabilistic, real-time systems with potentially infinite data variables. This approach uses local formal abstract steps, which results in more compact abstractions than alternative abstraction refinement techniques.

The network operators have LR-PON exhibits tangible advantages in terms of cost savings and hence have been identified as a promising candidate for future broadband access. One of the most important challenges in LR-PON is to deal with the significant network performance degradation because of the increased propagation delay. An efficient way to mitigate this performance degradation has advocated multi-thread-based DBA approach. However, this article shows that without a proper inter-communication mechanism among the overlapped bandwidth allocation processes, the efficiency of a multi-thread DBA approach could be affected drastically. In this regard, we concentrate on the multi-thread DBA approach, and highlight the issues arising due to the lack of inter-thread scheduling. After making an assessment of the currently available solutions, we have proposed a new approach named enhanced inter-thread scheduling (EIS), which integrates key features from the existing inter-thread scheduling algorithms.

We pointed out some inaccuracies in the multi-thread allocation algorithm and suggested some modifications to ensure that an ONU is not granted more than requested, by efficiently distributing the excess bandwidth among heavily loaded ONUs, and also to enable enhanced inter-thread scheduling to fully utilize a thread without disturbing the thread tuning. Simulation results show that enhanced inter-thread polling succeeds in decreasing reporting and queuing delays, whereas online inter-thread polling has a lower grant delay and therefore achieves a better overall delay performance. Online inter-thread polling also achieves a higher throughput, since multi-thread polling use more bandwidth for report messages and guard intervals.

REFERENCES

- [1] Tony Tsang, "Performance Analysis for QoS-Aware Two-Layer Scheduling in LTE Networks", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Vol. 2, Issue 2, July-August 2013.
- [2] Marta Kwiatkowska, Gethin Norman and David Parker, "PRISM 4.0: Verification of Probabilistic Real-time Systems", In Proceeding 23rd International Conference on Computer Aided Verification (CAV'11), volume 6806 of Lecture Notes of computer Science, pages 585-591, Springer, July 2011.
- [3] Song H., Kim B.W., and Mukherjee B., "Long-Reach Optical Access Networks: A Survey of Research Challenges, Demonstrations, and Bandwidth Assignment Mechanisms", IEEE Communications Surveys & Tutorials, vol. 12, no. 1, pages 112-123, 2010
- [4] Jensen, H.: "Model checking probabilistic real time systems", In Proceeding 7th Nordic Workshop on Programming Theory, Report 86, Chalmers University of Technology, pages 247-261, 1996.
- [5] Kwiatkowska, M., Norman, G., Segala, R., Sproston, J., "Automatic verification of real-time systems with discrete probability distributions", Theoretical Computer Science (TCS)282, pages 101-150, 2002.
- [6] Beauquier D., "Probabilistic timed automata", Theoretical Computer Science (TCS)292(1), pages 65-84, 2003.

- [7] Clarke E., Emerson E., Sistla A., “Automatic verification of finite-state concurrent systems using temporal logics”, *ACM Transactions on Programming Languages and Systems* 8 (2) pages 244V263, 1986.
- [8] Alur R., Courcoubetis C., Dill D., “Model checking in dense real time”, *Information and Computation* 104 (1) pages 2V34, 1993.
- [9] Hansson H., Jonsson B., “A logic for reasoning about time and reliability”, *Formal Aspects of Computing* 6 (4) pages 512V535, 1994.
- [10] Song H., Kim B W., and B. Mukherjee B., “Multi-Thread Polling: A Dynamic Bandwidth Distribution Scheme in Long-Reach PON”, *IEEE Journal Selected Area Communications (JSAC)*, vol. 27, pages 134-142, 2009
- [11] Helmy A., Fathallah H., and Mouftah H., “Interleaved Polling Versus Multi-Thread Polling for Bandwidth Allocation in Long-Reach PONs”, *IEEE/OSA Journal Optical Communication and Networks*, vol. 4, no. 3, pages 210V18, March 2012.
- [12] Skubic B. et al., “Dynamic Bandwidth Allocation for Long- Reach PON: Overcoming Performance Degradation”, *IEEE Communications Magazine*, vol. 48, pages 100V108, Nov. 2010.

Authors

Tony Tsang received the BEng degree in Electronics & Electrical Engineering with First Class Honours in U.K., in 1992. He received the Ph.D. from the La Trobe University (Australia) in 2000. He was awarded the La Trobe University Post graduation Scholarship in 1998. He is a Lecturer at the Hong Kong Polytechnic University. Prior to join in the Hong Kong Polytechnic University, Dr. Tsang earned several years of teaching and researching experience in the Department of Computer Science and Computer Engineering, La Trobe University. His research interests include mobile computing, networking, protocol engineering and formal methods. Dr. Tsang is a member of the ACM and the IEEE.