

Research Article

Dynamic Architectural Reconfiguration Algorithms and Transmission Protocols for Clustered Sensor Network Topologies with Prioritized Data

Fatma Salem,¹ A. T. Burrell,² and P. Papantoni-Kazakos¹

¹ *Electrical Engineering Department, University of Colorado Denver, Denver, CO 80217, USA*

² *Computer Science Department, Oklahoma State University, Stillwater, OK 74078, USA*

Correspondence should be addressed to P. Papantoni-Kazakos, titsa.papantoni@ucdenver.edu

Received 14 May 2012; Accepted 4 June 2012

Academic Editors: A. Alomainy and S. Ozdemir

Copyright © 2012 Fatma Salem et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The objective of a sensor network is the execution of specific signal processing functions on data that are collected in a distributed fashion. The transmission of the data is facilitated by protocols whose operations may be constrained by physical limitations of the network units, while their performance must simultaneously comply with the performance requirements of the deployed signal processing operations. At the same time, the network architecture affects the performance of both the signal processing and the data transmission operations, while some of the sensors may generate high-priority data. In this paper, we consider clustered sensor network topologies deploying a specific stable random access transmission algorithm per cluster, which facilitates high-priority data. We then introduce a dynamic architectural reconfiguration algorithm which controls individual cluster rates for optimal overall network performance. The latter algorithm is facilitated by a high-level traffic rate monitoring protocol.

1. Introduction

Wireless sensor networks satisfy signal processing objectives; their performance metrics are thus determined by those of the latter objectives [1]. When time constraints are imposed on high-accuracy signal processing operations, the consequence is increased required overall data rates. At the same time, in wireless sensor networks, observation data are transmitted via appropriate multiple access protocols [2–5], whose performance is a function of their input data rates, and are collected and processed by life-limited nodes, whose life span is a function of the data rates they process, [6–9]. Thus, required overall data rates, in conjunction with rate-dependent transmission protocol performance and node life spans, necessitate network-architecture and network-operations adaptations, so that the nodes' survivability limitations do not interfere with the required network overall performance [10, 11]. Since the network-architecture and network-operations adaptations are functions of the acting data rates, it is eminent that data rates be monitored and that rate changes be detected accurately and rapidly [12].

The distinguishing feature in wireless sensor networks is limited life spans of the nodes, induced by energy consumption. Interesting results focusing on energy consumption have been obtained by several researchers: Bounds on energy conservation techniques have been derived in [13], role assignments targeting energy conservation have been developed in [10], energy conservation routing techniques have been proposed in [7, 8, 14], and issues arising due to energy conservation have been discussed in [14]. In addition, topological and node-cooperation issues have been included in [11, 15], while approaches to performance monitoring have been presented in [16]. An interesting rate allocation algorithm has been presented in [9], which is based on a modification of the max-min routing in [17] and the lexicographic linear programming approach in [18]. In [12], a dynamic rate allocation methodology has been presented that is facilitated by a powerful data rate monitoring algorithm. In [19], energy-efficient clustering algorithms are proposed, including a discussion on LEACH approaches, where energy consumption is assumed to be strictly a function of geographical distances and where transmission collisions

are completely ignored. Considering transmission protocols, those proposed or partially implemented (in ZigBee, IEEE-802.15.4, etc.) within the random access class are ALOHA-based and are characterized by well-known instabilities that pull the throughput down rapidly to zero, as the user population increases.

In this paper, we consider clustered sensor network topologies containing many randomly distributed sensors some of which may generate high-priority data. We deploy specific stable random access transmission protocols per cluster, that facilitate high-priority data. Data rates are time-varying in such a network, mainly due to expiring life-limited nodes. We focus on the problem of dynamic architectural reconfigurations, facilitated by a data rate monitoring higher-level protocol. The organization of the paper is as follows. In Section 2, we present the system model and the problem formalization. In Section 3, we propose an architectural reconfiguration algorithm. In Section 4, we describe and evaluate a data transmission random access protocol that we recommend for deployment in each cluster. In Section 5, we summarize the data rate monitoring protocol used to facilitate the architectural reconfiguration process and discuss its performance. In Section 6, we discuss overall system issues. In Section 7, we include numerical results. In Section 8, we draw conclusions.

2. System Model and Problem Formalization

We consider a clustered network architecture, as in [9, 12], shown in Figure 1. The components in the architecture are microsensors, microsensor clusters, and a backbone network of cluster heads and a fusion center. In [9, 12], the microsensors, the cluster heads, and the fusion center have been, respectively, termed microsensor nodes (MSNs), aggregation and forwarding nodes (AFNs), and base station (BS). We will use the same terminology for consistency. Given a pre-determined signal processing objective, the MSNs, AFNs, and BS perform the following functions. (a) The MSNs are grouped into distinct clusters, where each cluster contains a single AFN. Each MSN collects local data and transmits them to its local AFN, via some multiple access transmission protocol implemented on a distinct channel associated with the AFN, that is, the overall network encompasses distinct separate channels (e.g., distinct frequency bands), each associated with a single AFN. Since the identities of the sensors contained in each cluster are time-varying due to possible sensor expirations and to possible subsequent architectural reconfigurations, the unknown user population model per cluster arises dictating the deployment of multiple access protocols from the random access class (RAs) [2–5, 16]. We will further elaborate upon this concept in Section 4. The AFNs and the BS are connected via separate channels (or frequencies) comprising the backbone network. Commands regarding routing adaptations in the latter network and system architectural reconfigurations propagate through the backbone network and do not interfere with the MSN transmissions. Possible reconfiguration broadcasts to MSNs are implemented via their local AFN, either on separate

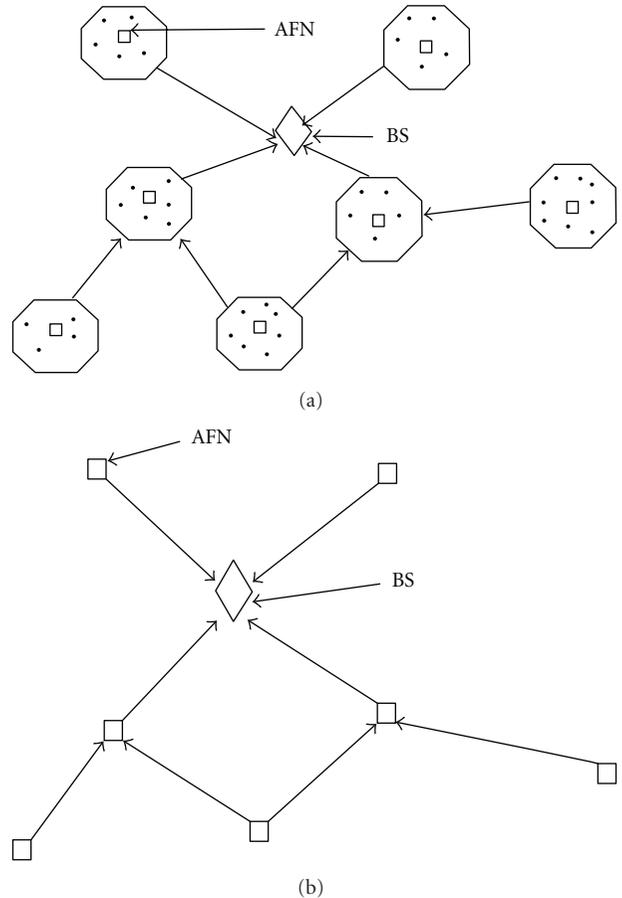


FIGURE 1: (a) Physical topology. (b) Backbone network.

channels (frequencies) or on dedicated minislots of the MSN transmission channel(s) and do not interfere with the MSN transmissions. The MSNs are of low cost and low energy; thus, short-life devices, and some of them generate high-priority data that must be received by the corresponding AFNs in relatively short time. (b) Each AFN collects the data sent by its local MSNs and processes them, using an operation determined by the network signal processing objective [1]; it also receives processed data sent by other neighboring AFNs. The AFN then processes the compounded processed data, utilizing an operation that is determined by the network signal processing objective [1], and transmits the outcome to selected neighboring AFNs or the BS. The AFNs have processing capabilities and are devices with energy and life spans that are much higher than those of the MSNs; their life spans and energy are still limited, however. (c) The BS fuses data transmitted to it by neighboring AFNs, utilizing an operation that is determined by the network signal processing objective [1]. The BS has practically unlimited life span and processing power.

The MSNs transmit to their assigned AFN via an RA protocol. Operations are performed at all nodes of the backbone network: at the AFNs and the BS. The nature of the operations is determined by the network signal processing objective, the environment that generates the data, and the

data rates [1]. At the same time, the energy consumption of the AFNs is a function of the data rates they receive and produce, and the complexity of the operations they perform. Each AFN performs operations on its input data rates, to produce the data rates it outputs to neighboring AFNs and/or the BS. Let us define the notation shown in Table 1.

We will assume that the MSNs per cluster are spatially randomly and independently distributed, and their number is large. In addition, their life span is short; thus, each MSN basically transmits a short-duration data message/or signal and expires. This, gives rise to a limit Poisson data packet user model per cluster [3], that is, the cluster user model reflects infinitely many independent Bernoulli users, each generating a single data packet, where an overall Poisson data packet traffic is induced.

2.1. The Static Problem. We initially consider the static problem stated in [9, 12]. The network signal processing objective is specified, the network architecture is fixed, the data environments and the RA transmission protocols are known and unchanged, the operations performed by the AFNs are determined and the data rates $\{\lambda_{Ci}\}$ generated in the clusters are fixed. Given, in addition, the time constraint T , for the completion of the signal processing objective, the performance of the network will be maximized when the maximum number of data is fused in time length T [9]. Since the fusion operation is performed by the BS and since the maximum number of data in T corresponds to the maximum attainable data rate, maximum network performance is then attained when the data rate μ in Table 1 is maximized. A generally nonlinear dynamic programming problem is then formalized to determine the optimal routing protocol in the backbone network comprised of the AFNs and the BS. This is known as the *static rate allocation* problem [12, 20].

2.2. The Dynamic Problem. The network is required to complete its signal processing objective within T time units. During this time period, some MSNs generally expire, since their life spans are only fractions of the time period T . This causes changes in the cluster data rates $\{\lambda_{Ci}\}$, and thus induces dynamics, first in the rate allocation problem, and then in network architectural configurations. Specifically, the rates $\{\lambda_{Ci}\}$ may generally change during the T time period, when the network operates towards the satisfaction of its objective. For rate changes that are relatively modest, a dynamic rate allocation problem arises: if such changes of the rates $\{\lambda_{Ci}\}$ can be detected, then the constants of the static rate allocation problem will be adjusted accordingly, and the new data rate allocations will be then dictated by the solution of the adjusted problem. For more dramatic detected rate changes, the satisfaction of the overall system performance requirements may necessitate network architectural reconfigurations. To detect changes in cluster data rates, an algorithm must be devised, that detects changes accurately and rapidly. Such algorithm will be deployed at the AFNs, as an upper level protocol, it will detect changes and will communicate them across the AFNs. Then, either a static rate allocation algorithm on the unchanged network architecture

TABLE 1: Notation.

λ_{Ci} :	The data rate from the MSNs in the cluster of the i th AFN to the AFN, where $\lambda_{Ci} = 0$ if all the MSNs in the cluster have expired
T :	The time constraint imposed on the network for completing the designated signal processing operation
μ :	The data rate collected by the BS

will be initiated that uses the newly detected cluster rates as constants, or an architectural network reconfiguration will be implemented with a matching rate allocation technique. The data rates required for the communication among the AFNs will be a fixed algorithmic characteristic, they will be added as a permanently fixed constants in the functions that interrelate network data rates [12] and will not interfere with the dynamics of the resulting rate allocation and architectural reconfiguration problems. The important performance characteristics of the rate monitoring algorithm are accuracy, speed, and stability: the detection time induced by the algorithm (convergence rate) compounded by the time to solve the static rate allocation and architectural reconfiguration problems must be shorter than the time taken for actual changes across the network (rate changes), while false detections must simultaneously occur infrequently.

3. An Architectural Reconfiguration Algorithm

The allowable values of the cluster data rates $\{\lambda_{Ci}\}$ in the network are determined by the throughput/delay characteristics of the transmission random access (RA) algorithms deployed in the clusters. Specifically, given the cluster transmission RA algorithm, the highest allowable cluster data rate should be such that the induced delays and retransmissions are nondetrimental to the network mission, in the sense of causing the expiration of an unacceptable percentage of MSNs. We will elaborate upon this concept in Section 4. We note that algorithmic delays induced by RAs are dependent on their algorithmic throughput. Thus, the cluster data rates $\{\lambda_{Ci}\}$ are all bounded from above by bounds determined by the characteristics of the per cluster deployed transmission protocols from the MSNs to the corresponding AFN. In addition, as induced by the deployed transmission algorithm, when the data rate in a cluster drops below a certain level, the existence of the cluster becomes wasteful, due to the then unnecessary induced increase in the size of the backbone network. Assuming that the transmission RA protocol per cluster is fixed, identical for all clusters and known, let us denote by ν_l and ν_u the common determined lower and upper bounds to each cluster data rate, respectively. If the aggregate data rate from all MSNs in the overall network is denoted by λ_C , the smallest possible number of clusters in the network is then $\lceil \lambda_C/\nu_u \rceil$, while the largest such number is then $\lfloor \lambda_C/\nu_l \rfloor$.

We will assume that a rate monitoring algorithm is deployed at the AFNs. When the data rates in all clusters remain within the (ν_l, ν_u) range, then detected data rate changes dictate only rate reallocations in the backbone

network of the overall structure, without any imposed architectural reconfigurations, as in [12]. When, instead, some cluster data rates fall outside the (ν_l, ν_u) range then architectural reconfigurations are first imposed, dictated by an architectural reconfiguration algorithm, followed by data rate reallocations on the new backbone network topology, as dictated by the deployed routing algorithm. In this section, we focus on the architectural reconfiguration algorithm.

3.1. The Algorithm. The original overall network architecture is based on the aggregate network data rate λ_C and the upper and lower rate bounds ν_u and ν_l determined by the transmission protocols in the clusters. In particular, assuming that transmission range is not a limiting factor in the geographical region covered by the network, $\lceil \lambda_C / \nu_u \rceil$ AFNs are originally deployed, each heading a cluster of MSNs with equal cluster data rates $\{\lambda_{C_i}\}$. The latter number of AFNs selected minimizes the size of the backbone network, subject to the rates per cluster remaining within the target range (ν_l, ν_u) ; thus, it minimizes routing complexities and delays as well as network propagation delays. When the data rate monitoring algorithm detects that one or more cluster data rates fall outside the (ν_l, ν_u) range, the architectural reconfiguration algorithm is initiated.

Let $\{R_i\}_{i \geq 0}$ denote the sequence of time instants when network architectural reconfigurations are initiated, where R_0 denotes the time when the original network architecture is deployed, based on the principles stated in the above paragraph. Let $\{\lambda_C^{(i)}\}_{i \geq 0}$ be the aggregate network data rates at the instants $\{R_i\}_{i \geq 0}$, respectively, where $\lambda_C^{(0)} \equiv \lambda_C$. Assuming no MSN replacement during the time when the network objective is satisfied, the sequence of rates in $\{\lambda_C^{(i)}\}_{i \geq 0}$ is monotonically nonincreasing with increasing index i . According to the above notation, R_i denotes the i th time when the data rate monitoring algorithm detects that one or more cluster data rates fall outside the (ν_l, ν_u) range. At R_i , the architectural reconfiguration algorithm implements the following steps.

Step 1. The aggregate network data rate $\lambda_C^{(i)}$ and subsequently $\lceil \lambda_C^{(i)} / \nu_u \rceil$ are computed.

Step 2. (a) If $\lceil \lambda_C^{(i)} / \nu_u \rceil = \lceil \lambda_C^{(i-1)} / \nu_u \rceil$, then, as compared to the network architecture at time R_{i-1} , the architecture of the backbone network remains unchanged, while the MSNs are reallocated to formulate rate-equivalent clusters. The reallocation is initiated by the BS, broadcast to the MSNs by the AFNs, via logically separate channels than those used for MSN transmissions, and is implemented by each MSN independently via equiprobable selection among the active clusters. For the high-priority MSNs, the equiprobable selection will be among cluster groups, instead, as explained in Section 4. We note that MSN reallocation may be judged as inappropriate in cases where it may cause unacceptable increase in MSN extinction rate. This issue will be further discussed in Section 6.

(b) If $\lceil \lambda_C^{(i)} / \nu_u \rceil < \lceil \lambda_C^{(i-1)} / \nu_u \rceil$, then some clusters are eliminated. The clusters are first ranked according to their

aggregate local data rate, and $\lceil \lambda_C^{(i-1)} / \nu_u \rceil - \lceil \lambda_C^{(i)} / \nu_u \rceil$ clusters possessing the lowest such rates are eliminated. The survived AFNs are decided upon by the BS and become known to the MSNs via broadcasting. The MSNs are then reallocated among the survived clusters, for rate-equivalence. The implementation of the reallocation is as in (a).

4. The Per Cluster Random Access Transmission Protocol

Due to the dynamically changing architectural reconfigurations in the considered sensor topologies, the identities of all sensors are not known to the AFNs at all times. Thus, the unknown user population model arises, where the sensors are the users in this case: the identity of each sensor becomes known to the system, only after the sensor accesses successfully some AFN. The only possible class of transmission protocols for the unknown user population model is the random access (RA) class [21]. In addition, within the RA class of transmission protocols, the only implementable stable subclass is the class of limited sensing random access algorithms (LSRAAs), where it is required that upon generation of a message, each user start monitoring the channel feedbacks continuously, until this message is successfully accepted for transmission [2–5]. We emphasize that the ALOHA-based protocols currently proposed (e.g., ZigBee, IEEE802.15.4, etc.) are unstable, leading to rapidly decreasing throughputs with increasing MSN rates.

As stated in Section 2, we assume a limit Poisson data packet user model per cluster. We will also assume a synchronous system, where all AFN channels are slotted synchronously across all AFNs, and where a slot is the time corresponding to the transmission of a single packet. In this model, each packet represents a separate independent user (separate MSN here); thus, any number of packets may simultaneously attempt transmission, unless organized by some protocol; an LSRAA, in this case.

As stated in Section 2, the MSNs in a single cluster transmit through a single channel (to their AFN) and overall system connectivity is attained via the backbone network. Due to the possible MSN mobility or the dynamics of the architectural topology, some MSNs may move across clusters, while they have a packet to transmit. At cluster boundaries, such MSNs are exposed to feedbacks from more than one cluster channels and have the capability to transmit through anyone of those channels. The latter phenomenon can be exploited to avoid temporary isolation of users moving across clusters, and especially to reduce their transmission delays while in transit across clusters. In addition, some MSNs may generate high-priority data, requiring accelerated transmission. We encompass boundary MSNs into the high-priority category and name them all *high-priority* MSNs. We name non-high-priority MSNs, *regular* MSNs. The following approach can be then taken: to each regular MSN assign as a single channel for its transmissions that corresponded to its local AFN. To each MSN that is of high-priority, provide the capability to monitor and possibly transmit through a group of channels, instead, associated with the corresponding group of AFNs.

The MSNs can be then partitioned into two categories: the regular MSNs transmitting through a single local channel and the high-priority MSNs having access to several cluster channels. If we consider the environment where high-priority MSNs have access to M AFN channels, then incorporating the regular MSNs, $M + 1$ classes of MSNs arise: (a) M classes of regular MSNs, each class transmitting through one of the distinct M AFN channels and (b) the $(M + 1)$ th class of high-priority MSNs that has access to all the M AFN channels. The objective is the design of limited sensing random access algorithms (LSRAAs) for the system that requires no knowledge of the system state by the users and present the high-priority MSNs with a delay advantage (compared to the non-high-priority MSNs), even when the traffic rates in the system change dynamically, while keeping the delays of the regular MSNs as low as possible. When architectural reconfigurations, as in Section 3, are in place, each regular MSN selects one of the AFNs equiprobably, while each high-priority MSN selects one M -size group of AFNs, equiprobably among all possible such groups, for their transmissions.

For the multichannel system considered in the above paragraph, let the M channels be indexed from 1 to M . Then, the regular MSNs assigned to channel i deploy a limited sensing random access algorithm, named LSRAAi, for their transmissions. For each of these LSRAAs, we adopt the class of limited sensing random access algorithms in [2] due to their simple operational properties and their performance characteristics: the algorithms are easily implemented in the limited sensing environment, they operate with binary (collision versus noncollision) feedback, they have superior resistance to feedback errors and in the presence of the limit Poisson user model they attain a throughput of 0.43. Throughput is here defined as the highest traffic rate for which the algorithmic system is stable, where within the algorithm's stability region the traffic is maintained by the algorithm with finite delays. The high-priority MSNs in the system deploy the same LSRAA as that deployed by the local MSNs, in conjunction with a *selection policy* via which they choose their transmission channel. A selection policy may be either dynamic or probabilistic. The operations of a dynamic selection policy are dictated by the feedback sequences that the boundary MSNs observe. In contrast, a probabilistic selection policy is represented by a priori assigned probabilities. A probabilistic selection policy is simple, but it does not generally present the high-priority MSNs with significant advantage, especially in the presence of changing rates of the regular MSN traffics. In this paper, we propose a dynamic selection policy which does not require any a priori system state knowledge, and which indeed presents the high-priority MSNs with a significant delay advantage. The policy, which was first introduced in [4], induces light delay penalization for the regular MSNs, at the expense of a reduction in overall system throughput.

Considering an M -channel system as above, we assume that some synchronous LSRAA is deployed by all the MSNs in the system. In particular, (1) time is divided into slots of length equal to the duration of a packet, and the starting instants of the slots are identical in all channels. (2) Per

channel, feedback per slot exists and corresponds to the outcomes induced by the local LSRAA. This feedback is binary, Collision (C) versus NonCollision (NC). (3) Each regular MSN is required to monitor the feedbacks from its assigned channel continuously, from the time it generates a new packet to the time that this packet is successfully transmitted. We initially assume that no propagation delays and no forward or feedback channel errors exist in the system.

We also assume that each high-priority MSN receives the feedbacks from all channels correctly and without propagation delays. At the time when a high-priority MSN generates a new packet, it starts monitoring the feedbacks from all channels continuously, until it decides to join the operations of one of the LSRAAs for the transmission of his packet. Upon this decision, it maintains the continuous monitoring of only those feedbacks that correspond to the LSRAA it chose, until its packet is successfully transmitted.

The regular MSN populations for each channel and the high-priority MSN population are all modeled as limit Poisson. That is, it is assumed that the regular data packet traffic i , $i = 1, 2, \dots, M$, is a limit Poisson process with intensity λ_i , $i = 1, 2, \dots, M$, and that the data packet traffic generated by the high-priority MSNs is another independent limit Poisson process with intensity λ_{M+1} . As an interesting remark, for a large class of LSRAAs, including that considered in this paper, the limit Poisson user model provides a lower bound in performance within the class of identical and independent users whose packet generating process is memoryless.

4.1. The Algorithm for the Two-Channel System. We assume that the two LSRAAs in the two-channel system are identical and belong to the class of window algorithms in [2], all of which induce throughput 0.43 and operate with binary feedback.

Upon generation of a new packet, a high-priority MSN imagines itself belonging to the systems of both LSRAAs and follows their algorithmic steps until the first time that they enter a collision resolution event in one of them: the rules of this first entry are stated by the channel selection policy. Then, the MSN remains with the latter LSRAA system, until its packet is successfully transmitted.

Let time be measured in slot units, where slot t occupies the time interval $[t, t + 1)$. Let $x_t(j)$ denote the feedback that corresponds to slot t for channel j , $j = 1, 2$, where $x_t(j) = C$ and $x_t(j) = NC$ represent collision and noncollision slot t for channel j , respectively. The LSRAA for channel j is implemented independently by each MSN in the system; its steps are dictated by the feedback sequence $\{x_t(j)\}_t$. Each regular MSN is assigned one of the channels a priori and thus observes only the feedback sequence corresponding to the latter channel. Each high-priority MSN observes, instead, the feedback sequences of both channels until it selects the channel which it will transmit its packet through. Below, we first explain the algorithmic steps implemented by each local MSN, dropping the channel index j , for simplicity in notion.

Each algorithm in the class in [2] utilizes a window of size Δ as an operational parameter and induces a sequence of consecutive collision resolution intervals (CRIs).

The window length Δ is subject to optimal selection for throughput maximization. Each CRI corresponds to the successful transmission of all packet arrivals within an arrival interval of length Δ . The length of the CRI is determined by the number of MSNs in the window Δ and the algorithmic steps of the collision resolution process. The placement of the Δ -size window on the arrival axis is determined asynchronously by the MSNs and is named the *examined interval* if the CRI. We will first describe the collision resolution process induced by the algorithm. Then, we will explain the process which determines the placement of the Δ -size window per CRI.

The algorithmic class contains algorithms whose collision resolution process can be depicted by a stack with finite number of cells. Let us consider this algorithm in the class which can be described by a K -cell stack. Then, in the implementation of the collision resolution process, each MSN utilizes a counter whose values lie in the set of integers, $[1, 2, \dots, K]$. We denote by r_t the counter value of some MSN at time t . The K different possible values of the counter place the user in one of the K cells of a K -cell stack. When its counter value is 1, the MSN transmits; it withholds at $K - 1$ different stages otherwise. When a CRI begins, all MSNs in the Δ -size window set their counters at 1; thus, they all transmit within the first slot of the CRI. If the window contains at most one packet, the first slot of the CRI is a noncollision slot and the CRI lasts one slot. If the window contains at least two packets, instead, the CRI starts with a collision which is resolved within the duration of the CRI via the following rules.

The MSN transmits in slot t if and only if $r_t = 1$. A packet is successfully transmitted in t if and only if $r_t = 1$ and $x_t = \text{NC}$.

The counter values transition in time as follows:

If $x_{t-1} = \text{NC}$ and $r_{t-1} = j$; $j = 2, 3, \dots, K$, then $r_t = j - 1$.

If $x_{t-1} = \text{C}$ and $r_{t-1} = j$; $j = 2, 3, \dots, K$, then $r_t = j$.

If $x_{t-1} = \text{C}$ and $r_{t-1} = 1$, then, $r_t =$

1; w.p. $1/K$

2; w.p. $1/K$

3; w.p. $1/K$

...

K ; w.p. $1/K$.

For any given K , the throughput of the algorithm is 0.43. This throughput is attained for different optimal window sizes Δ^* , as K varies. For $K = 2$, $\Delta^* = 2.33$. For $K = 3$, $\Delta^* = 2.56$.

From the above rules, it can be seen that a CRI that starts with a collision slot ends with K consecutive noncollision slots, an event which cannot occur at any other instant during the CRI. Thus, the observation of K consecutive noncollision slots signals the certain end of an CRI to all users in the system; it either signifies the end of an CRI that started with a collision or the end of a sequence of K

consecutive length-one CRIs. Therefore, a MSN packet that arrives in the system without any knowledge of the channel feedback history can synchronize with the system upon the observation of the first K -tuple of consecutive noncollision slots. This observation leads to the asynchronous by the MSNs generating of the size- Δ window placement on the arrival axis. Specifically, if a CRI ends with slot t , the window of the next CRI is selected with its right most edge $K - 1$ slots to the left of slot t and it contains those packets whose *updates* fall in the interval $(t - K + 1 - \Delta, t - K + 1)$. The *updates* $\{t^k\}$ of a packet are generated as follows. Let t_0 be the slot within which a packet is generated. Then define t^0 to be equal to t_0 . Starting with slot t^0 , the corresponding regular MSN senses continuously the channel feedbacks. It does so passively, until it observes the first K -tuple of consecutive NC slots, ending with slot t_1 . If $t^0 \in (t_1 - K + 1 - \Delta, t_1 - K + 1)$, the MSN participates in the CRI that starts with slot $t_1 + 1$. Otherwise, it updates its arrival instant to $t^1 = t^0 + \Delta$ and waits passively until the end of the latter CRI, ending with slot t_2 . If $t^1 \in (t_2 - K + 1 - \Delta, t_2 - K + 1)$, the MSN packet participates in the CRI which starts with slot t_2 ; otherwise, the MSN updates its arrival instant by Δ again and repeats the above process. In general, if $\{t_n\}_{n \geq 1}$ denotes the sequence of consecutive CRI endings since the first K -tuple of consecutive NC slots, the MSN packet participates in the k th CRI if $t^{k-1} \in (t_k - K + 1 - \Delta, t_k - K + 1)$ and $t^n \notin (t_{n+1} - K + 1 - \Delta, t_n - K + 1)$ for all $n \leq k - 2$. In the latter case, t_k denotes the beginning slot of the first after the packet's arrival CRI whose examined interval the packet's update falls into.

Each high-priority MSN observes both feedback sequences $\{x(1)\}_{t \geq t_1}$ and $\{x(2)\}_{t \geq t_1}$ and follows the evolution of both time sequences $\{t_i(1)\}_{i \geq 2}$ and $\{t_i(2)\}_{i \geq 2}$ and their corresponding sequences of updates. Given a high-priority packet, let $t_k(j)$; $j = 1, 2$ denote the beginning slot of the first after the packet's arrival CRI in channel j whose examined interval the channel j update of the packet falls into. Then, if $t_k(1) < t_k(2)$, the packet enters a CRI within the LSRAA of channel 1 in slot $t_k(1) + l$ and is successfully transmitted during its process. If $t_k(1) > t_k(2)$, instead, then the MSN packet joins a CRI in channel 2, in slot $t_k(2) + 1$. If $t_k(1) = t_k(2)$, then the MSN selects one of the local LSRAAs with probability 0.5. The evolution of the updates for each LSRAA and each high-priority MSN is exactly as those of the local MSNs. That is, the arrival sequences are always updated by Δ and the Δ -size window is equally divided between the two LSRAAs whenever $t_k(1) = t_k(2)$.

4.2. Expiration of MSNs during the Transmission Process. The MSNs that have a packet to transmit consume energy due to channel monitoring and due to retransmissions within the CRI which accommodates the transmission of the packet. If the energy consumed this way exceeds their limit, the MSNs expire and their generated packet is lost. We will assume that the energy consumed per single monitored channel slot is a constant, and so is the energy consumed per retransmission slot. Then, assuming the deployment of an M -channel

system LSRAA, we form a linear expiration expression, as explained below. Let us first define the following:

β : the amount of per MSN energy consumed by the monitoring of a single channel slot;

η : the amount of per MSN energy consumed by a single packet (re)transmission;

τ_1 : the number of slots during which an MSN monitors both channels in the system;

τ_2 : the number of slots during which an MSN monitors a single channel in the system;

m : the number of retransmissions during the CRI;

ξ : the total energy stored in a single MSN for packet transmission.

Using the above notation, we assume that the MSN expires, with subsequent loss of its generated packet, if:

$$\beta(M\tau_1 + \tau_2) + \eta m > \xi. \quad (1)$$

As first stated in Section 2, the MSN expiration, as given by expression (1), causes packet rejections and subsequent reduction of cluster traffic rates, where, as the number M of monitored by the high-priority MSNs channels increases, the life span of these MSNs decreases. Other causes of MSN expiration may be environmental hazards or random electronic failures. Regardless of the causes, MSN expirations may result in traffic rate reductions that dictate architectural reconfigurations.

4.3. Algorithmic Analysis of the Two-Channel System. We index the two channels from 1 to 2. Then, for convenience in notation, we will refer to the regular MSNs in channel 1, the regular MSNs in channel 2 and the high-priority MSNs as subsystem 1, subsystem 2, and subsystem 3, respectively. In the algorithmic analysis, we will assume that the three subsystem traffics are mutually independent and that the user traffic in subsystem j , $j = 1, 2, 3$, is limit Poisson with intensity λ_j .

Consider either one of the local LSRAAs, whose operations are described in Section 4.1. Let us select the 3-cell algorithm among those in the class in [2]. In Appendix A, the algorithmic analysis of the 2-channel system is presented. This analysis includes stability, which provides the maintainable by the system traffic rates. In Table 2, we include the maximum rate λ_2^* ; maintained by the system, for varying λ_1 and λ_3 values as well as the maximum maintained symmetric rate $\lambda^*(\lambda_3) = \lambda^* = \lambda_1^* = \lambda_2^*$ for varying λ_3 values. We also include the maximum rate λ_3^* ; maintained by the system, when $\lambda_1 = \lambda_2 = 0$. We note that due to the symmetries appearing in the system, for each fixed rate λ_3 we only need to compute λ_2^* for λ_1 values in the interval $[0, \lambda^*(\lambda_3))$. In Figure 2, we plot the boundaries of the (λ_1, λ_2) acceptable regions of maintainable traffic, parameterized by various λ_3 values. Those boundaries are clearly symmetric around the 45° straight line and determine the stability regions for the algorithmic system.

TABLE 2: Maximum maintainable Poisson rates. Window size: $\Delta = 2.5599$.

λ_3	λ_1	λ_2^*	$\lambda^* = \lambda_1^* = \lambda_2^*$
0	<0.43	0.43	0.43
	0.01	0.429	
	0.10	0.4288	
0.001	0.20	0.4285	0.4278
	0.30	0.4282	
	0.40	0.4280	
0.01	0.00	0.424	0.42
	0.20	0.423	
	0.40	0.422	
0.10	0.00	0.372	0.36
	0.10	0.370	
	0.20	0.368	
0.20	0.30	0.364	0.29
	0.00	0.32	
	0.10	0.31	
0.30	0.20	0.30	0.23
	0.30	0.28	
	0.00	0.260	
0.40	0.10	0.245	0.17
	0.15	0.240	
	0.20	0.232	
0.50	0.00	0.198	0.11
	0.05	0.90	
	0.10	0.180	
0.60	0.15	0.170	0.04
	0.00	0.132	
	0.05	0.12	
	0.10	0.11	
	0.00	0.062	
	0.01	0.06	
	0.02	0.05	
For $\lambda_1 = \lambda_2 = 0 : \lambda_3^* = 0.68$			

From Table 2 and Figure 2, we observe that whenever λ_3 is strictly positive, the sum $\lambda_1^* + \lambda_2^* + \lambda_3$ is strictly less than 0.86, that is, strictly less than twice the throughput of each local LSRAA. The reason for this is twofold: (a) when $\lambda_3 > 0$, some percentage of the traffic generated by the high-priority MSNs is always assigned for transmission to an LSRAA already overloaded by local MSN traffic. This is true because the expected CRI length is upper bounded by the window size Δ if the algorithm operates in its stability region, and it is slightly larger than Δ if the system is temporarily overloaded. Thus, if the rate of the regular local MSN traffic is close to the algorithmic throughput, then the presence of high-priority MSNs results in overloading of the LSRAA. (b) The rate λ_3 traffic sometimes uses the window size Δ that is optimal for the local LSRAAs, while sometimes it uses the window $\Delta/2$, instead. This causes throughput reduction even in the absence of regular local MSN traffics. In fact, in the latter case the maximum such reduction results because

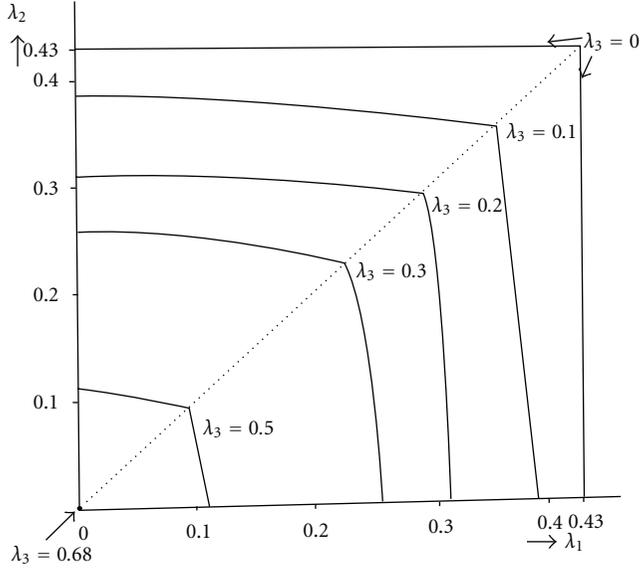


FIGURE 2: Maximum maintainable Poisson rates. Window size: $\Delta = 2.5599$.

a higher percentage of the overall system traffic uses then the suboptimal window size $\Delta/2$.

We point out that the stability regions in Table 2 and Figure 2 represent the case where the traffic is *maintained*, meaning that no MSNs expire. We also point out that when traffic is maintained, the collision resolution process induced by the deployed LSRAAs changes the traffic statistics, that is, when the input traffic is limit Poisson, the output traffic is not Poisson then, while it simultaneously maintains the input rate and can be closely approximated by a Poisson process.

The 2-channel algorithmic system induces regenerative points within its stability region. Thus, the regenerative theory, as applied in [4, 5], is applicable here as well, for the computation of the expected delays in each of the three subsystems. The details of this approach can be found in [4]. In Figures 3 and 4, we plot expected delays for two symmetric systems, with the systems' 1 and 2 rates being equal to 0.1 and 0.3, respectively. From these figures, we observe that the delays of the high-priority MSNs are very little affected by their traffic rate and remain low, while the same delays for the regular MSNs are significantly increasing as the rate of the high-priority traffic increases. From the latter figures, we decide that: (a) below the $(\lambda_1 = \lambda_2 = 0.1; \lambda_3 = 0.1)$ rate region, the system is wasteful, and (b) above the $(\lambda_1 = \lambda_2 = 0.3; \lambda_3 = 0.05)$ rate region, the delays of the regular MSNs are unsatisfactory. Rate region $(\lambda_1 = \lambda_2 = 0.1; \lambda_3 = 0.2)$ reflects 0.15 rate per cluster and 33% of the total MSN population being high-priority MSNs. Rate region $(\lambda_1 = \lambda_2 = 0.3; \lambda_3 = 0.05)$ reflects 0.325 rate per cluster and 7.6% of all MSNs being high-priority MSNs. Referring to the architectural reconfiguration algorithm in Section 3 and the notation there, we may initially subsequently select $\nu_l = 0.15$ and $\nu_u = 0.325$, between 0% and 10% of total traffic generated by high-priority MSNs.

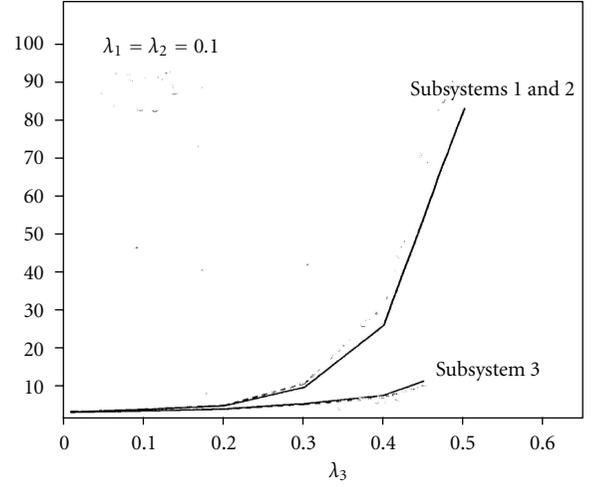


FIGURE 3: Expected delays. Symmetric 2-channel system with regular MSN rates equal to 0.1.

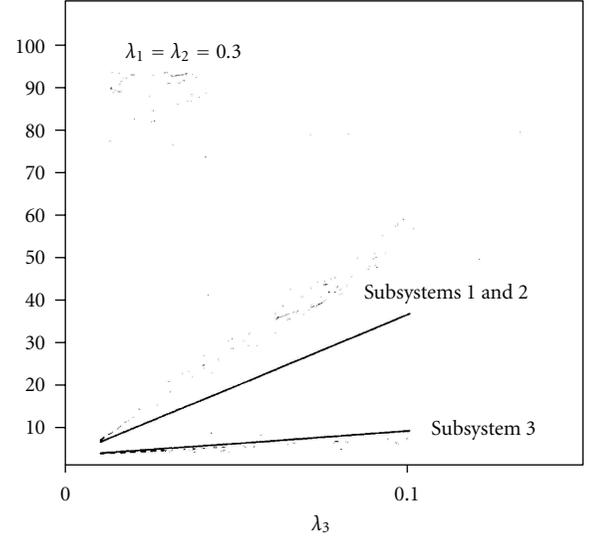


FIGURE 4: Expected delays. Symmetric 2-channel system with regular MSN rates equal to 0.3.

As we will further discuss in Section 5, the selection of the ν_l and ν_u values should be finally tuned depending on the MSN expiration constants in (1).

4.4. The M -Channel System. In the M -channel system, with $M > 2$, the algorithms operate as in Section 4.1, with the following simple modification: when a high-priority MSN with a packet to transmit observes simultaneous beginnings of N CRLs and its updates are within the examined intervals of all of them, it selects each one of the N LSRAAs with probability $1/N$. The stability conditions for the M -channel case are along the same lines as in Theorem 4 in Appendix A. However, they induce interactions among the M LSRAAs, and the evolving recursions for their evaluation give rise to multi-dimensional couplings. It is possible to devise a simplified

recursive methodology which will basically first decompose the M -channel system into lower-dimensionality subsystems, and then use these subsystems as units towards an overall system evaluation. Specifics on this methodology and throughput/delay results can be found in [4]. For $M > 2$, the delays of the high-priority users decrease, while, via expression (1) in Section 4.2, their life span simultaneously decreases. The latter tradeoff may weigh favorably towards the 2-channel system selection, for a considerable range of values for the constants involved in expression (1).

5. The Data Rate Monitoring Protocol

As stated in Section 2, the architectural reconfiguration algorithm in Section 3 is facilitated by a high-level data rate monitoring protocol. In this section, we present the highlights of the algorithm that implements the latter protocol, as presented and fully analyzed in [12]. Isolating a single cluster in the system, we first denote by λ the data rate accessing the AFN of the cluster: this data rate encompasses MSN expirations, whose occurrence is explained in Section 4.2. We subsequently decide on the ν_l and ν_u rate values to be monitored, where 0.15 and 0.325 have been respectfully selected in Section 4.3. The objective is then to decide rate shifts from ν_u to ν_l . Below, we will summarize the generalized algorithm from [12], which monitors shifts among a set $\{\lambda_i\}$ of rates.

The proposed algorithm is sequential, with its operational values updated at discrete, equally spaced, time instants $\{T_i\}_{i \geq 0}$, where the time interval between consecutive T_i is a fraction of the life expectancy of MSNs. We name the time interval between two consecutive T_i 's, *frame*. Let us denote by $\{S_i\}_{i \geq 0}$ the subsequence of the sequence $\{T_i\}_{i \geq 0}$ such that $T_0 = S_0$ is the beginning of time when the system starts operating, and S_i is the i th after S_0 time instant (corresponding to the beginning of some frame) at which the monitoring algorithm decides that a shift in the acting rate has just occurred. Let us assume that, given any fixed rate λ_k , the cluster data arrival process is stationary, and let $f_k(n_1, \dots, n_q)$ denote its q -dimensional distribution in frame lengths, that is, $f_k(n_1, \dots, n_q)$ is the probability that, in a sequence of q consecutive frames, n_1 data arrivals occur in the first frame, and so on, with n_q data arrivals finally occurring in the q th frame, given that the active rate is λ_k throughout. Let us assume that the distributions $\{f_k(n_1, \dots, n_q)\}$ are well known for all k 's (or all λ_k 's). Then, we propose the following sequential data rate monitoring algorithm, from [12], which traces consecutive data rate shifts.

5.1. The Algorithmic System. (1) The monitoring algorithmic system is designed at rates $\{\lambda_k\}_{1 \leq k \leq n}$. (2) Let at some S_i the rate λ_k be decided as just starting, by the monitoring algorithmic system. Then, immediately (at S_i), a set of $n - 1$ parallel algorithms starts operating. The j th such algorithm is monitoring a possible shift from the rate λ_k to the rate λ_j , where $\lambda_k \neq \lambda_j$, sequentially, with adaptations occurring at beginnings of frames. The $n - 1$ parallel algorithms use a reflective threshold at zero and a common decision threshold,

η_k . Let $V_{kj}(0)$ denote the operating value of the algorithm monitoring $\{\lambda_k \rightarrow \lambda_j\}$ shift at S_i , when it starts operating; let $V_{kj}(r)$ denote its operating value at its r th adaptation step. Then, the operating values of the algorithm are sequentially updated as follows:

$$V_{kj}(0) \triangleq 0,$$

$$V_{kj}(r+1) = \max\left(0, V_{kj}(r) + \log \frac{f_j(n_{r+1} | n_1, \dots, n_r)}{f_k(n_{r+1} | n_1, \dots, n_r)}\right). \quad (2)$$

S_{i+1} is then the first time after S_i that one of the above $n - 1$ algorithms *first* crosses the common decision threshold η_k . If the latter algorithm is the one which monitors a $\lambda_k \rightarrow \lambda_p$ shift, it is decided that the latter shift occurred and a set of $n - 1$ parallel algorithms, each monitoring a shift from λ_p to one of the remaining rates, is immediately initialized. The latter algorithmic system has operational characteristics as those stated above, utilizing a common decision threshold η_p .

A summary of the performance characteristics of the algorithm, taken from [12], is included in Appendix B. The algorithm induces correct decisions as well as false alarms whose relative relationship is controlled by the value of the selected decision threshold.

When the data rate monitoring algorithm is deployed for both architectural reconfiguration purposes and dynamic routing adaptations, as in [12], then, multiple rates are monitored and multiple parallel algorithms are simultaneously operating, as stated above. When focusing solely on the architectural reconfiguration problem, the data rate monitoring per cluster is implemented by a single algorithm as in (2) that monitors ν_u to ν_l possible shifts.

5.2. The Algorithm for the Poisson Model. As explained in Section 2, the data generated by the MSNs in cluster i may be modeled as cumulatively comprising a homogeneous limit Poisson process, with rate λ_{Ci} bits/time unit. As stated in Section 4.4, the traffic accessing the AFN of the cluster, while not Poisson then, it can be closely approximated by a Poisson process. The Poisson arrival process is memoryless: the conditioning in the log-ratio updating step of the algorithm in (2) then drops. If d denotes the length of a frame in time units, the $\{\lambda_k \rightarrow \lambda_j\}$ monitoring algorithm in (2) takes the following form:

$$V_{kj}(0) \triangleq 0,$$

$$V_{kj}(r+1) = \max\left(0, V_{kj}(r) + d \left[\lambda_k - \lambda_j + n_{r+1} \log \left(\frac{\lambda_j}{\lambda_k} \right) \right] \right), \quad (3)$$

where n_{r+1} denotes the number of transmitted data within the $(r+1)$ th frame from the beginning of the $\{\lambda_k \rightarrow \lambda_j\}$ monitoring algorithm.

Let us define the following:

$$\zeta(\lambda_k, \lambda_j) \triangleq [\lambda_k - \lambda_j] \left[\log \left(\frac{\lambda_k}{\lambda_j} \right) \right]^{-1}, \quad (4)$$

where $\min(\lambda_k, \lambda_j) < \zeta(\lambda_k, \lambda_j) < \max(\lambda_k, \lambda_j)$. We may select $\{\lambda_i\}$ rates such that $\zeta(\lambda_k, \lambda_j)$ are rational numbers for all k and j , and we then define the integers t_{kj} and s_{kj} , $t_{kj} < s_{kj}$, as follows:

$$\zeta(\lambda_k, \lambda_j) = \frac{t_{kj}}{s_{kj}}. \quad (5)$$

The algorithmic thresholds can then all be selected as positive integers, and without lack in generality, the algorithmic adaptation in (3) can be transformed as follows:

$$V_{kj}(r+1) = \max\left\{0, V_{kj}(r) + (-1)^{\text{ime}(k,j)} [s_{kj}n_{r+1} - dt_{kj}]\right\}, \quad (6)$$

where

$$\text{ime}(k, j) = \begin{cases} 0; & \text{if } \lambda_j > \lambda_k; \\ 1; & \text{if } \lambda_j < \lambda_k. \end{cases} \quad (7)$$

6. Overall System Issues

As induced by the architectural reconfiguration algorithm in Section 3, when a $\nu_u \rightarrow \nu_l$ shift is detected by at least one of the cluster data rate monitoring high-level protocols, the overall system reconfigures itself, where all MSNs reselect the AFN(s) that they will be connected to for their transmissions. This reselection halts the transmission process, until reconfiguration of the clusters is completed, causing interruptions of those transmissions that are in progress at the time of the halting. The latter transmissions need to be restarted by the corresponding MSNs, resulting in additional delays and retransmissions that may exceed the limit imposed by the low energy MSN characteristic, (see Section 4.3). Thus, architectural reconfigurations may cause increased MSN expiration rates. The so caused MSN expiration rate increase may be reduced if, Step 2(a) of the architectural reconfiguration algorithm in Section 3 is modified so that no cluster reselection by the MSNs occurs then, that is, in the case where the $\nu_u \rightarrow \nu_l$ rate shift in one or a few clusters does not change the number of clusters in the system, the MSNs remain, instead, with the AFN(s) they were associated with before data rate shifts were detected, and do not interrupt any transmissions in progress. In the latter case, however, the overall system is not balanced: MSNs that are located in relatively overloaded clusters will suffer higher delays in their future transmissions, since they are not allowed to reallocate, at the risk of exhausting their life span. In summary, maintaining Step 2 in the Architectural Reconfiguration Algorithm, as described in Section 3, versus modifying it to exclude MSN reallocations induces a tradeoff between MSN expirations due to transmission interruptions and MSN expirations due to delays in relatively overloaded clusters. Given a specific clustered sensor environment, this tradeoff needs to be numerically evaluated, before a final decision regarding MSN reallocations in Step 2 of the architectural reconfiguration algorithm is made.

The false alarms induced by the data rate monitoring algorithm in Section 5 may occasionally dictate erroneous

cluster reconfigurations and subsequent backbone network reductions. In particular, the monitoring algorithm may erroneously lead to the Step 2(b) case of the architectural reconfiguration algorithm in Section 3, imposing elimination of some clusters, while the actual overall network data rate λ_C has not been sufficiently reduced to necessitate such reduction. Implementation of cluster elimination will then result in traffic overloading of some clusters, with subsequent increase of the MSN expiration rate in them. The latter expiration rate increase must be evaluated against such increase due to the excessive routing and the propagation delays in the nonreduced backbone network, for the subsequent adjustment of the data rate monitoring decision threshold, in harmony with the per cluster boundary rates (ν_l, ν_u) .

Given a member of the LSRAA class in Section 4, given the upper bound ν_u of the maintained data rate per cluster, given the MSN-expiration formula in (1), a specific value of the MSN expiration rate is induced. When the latter value is subtracted from ν_u , the result should be bounded from below by the data rate lower bound ν_l . In addition, if it is desirable that the MSN expiration rate, solely due to (1), be upper bounded, then, an upper bound to the value of ν_u is induced. As a conclusion, the pair (ν_l, ν_u) of rate values are dependent on the MSN-expiration formula in (1).

7. Numerical Results

We first simulated a two-cluster network, where the high-priority MSNs may transmit via either one of the two cluster channels. We assumed limit Poisson traffics generated by both the regular and the high-priority MSN populations, and we considered high-priority traffics from 0% to 10% of the regular traffics. We used the expression in (1) to model MSN expiration, with various values of the β , η , and ξ constants, while we adopted the LSRAA transmission protocol for two channels in Section 4. We simulated the two-cluster system and computed the resulting MSN expiration rates as well as the delay distributions of the successfully transmitted packets. In accordance with the results and the subsequent discussion in Section 4, we selected $\nu_l = 0.15$ and $\nu_u = 0.325$, with 0% to 10% high-priority MSNs. We plot some of our results in Figures 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16. We observe that, as expected, for fixed constant values ξ , β , and η , the expiration rates of all three subsystems converge to identical values, while the nonexpired nodes of subsystem 3 are presented with a substantial delay advantage. When the constants β and η increase, the rejection rates increase uniformly, at the benefit of substantial decrease in the delays of the nonexpired MSNs. We note that within our ξ , β , and η value selections, the only case where the effective transmission rate falls below the 0.15 per cluster value is that corresponding to $\xi = 100$, $\beta = 10$, and $\eta = 50$.

We simulated the two channel system for the total per cluster traffic rate being the as selected above $\nu_u = 0.325$ and for the high-priority traffic being between 0% and 10% of the per cluster regular traffic. We monitored the input to the two AFNs traffics deploying the high-level protocol in Section 5 and implemented the architectural reconfiguration

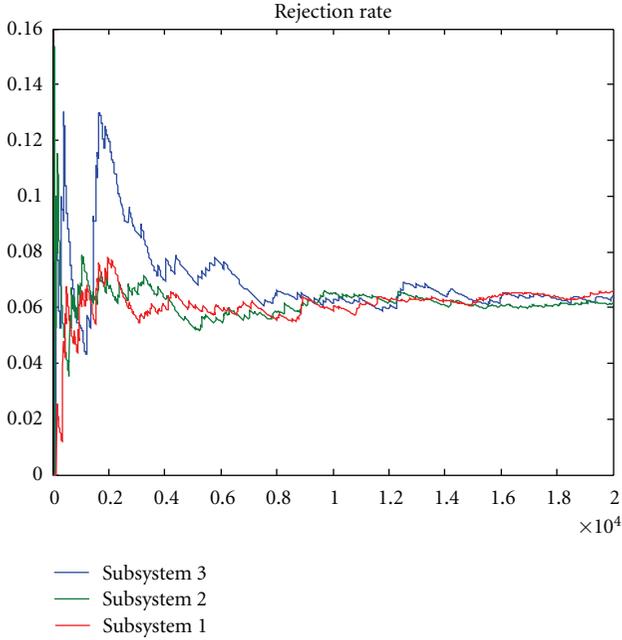


FIGURE 5: Rejection rate for $\xi = 100$, $\beta = 5$, and $\eta = 10$.

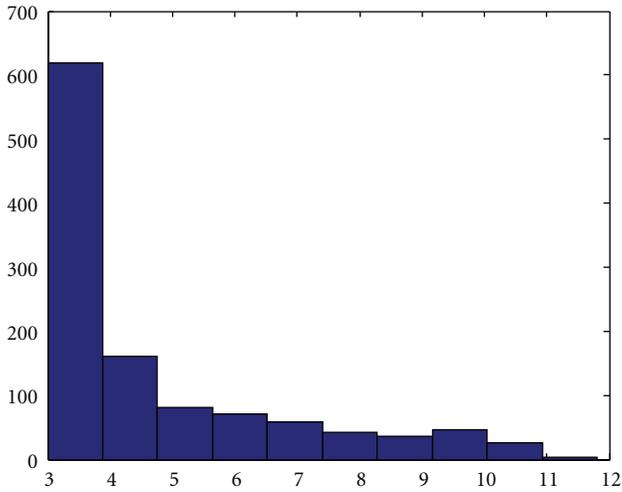


FIGURE 6: The delay distribution of nonexpired nodes for the subsystem 3 for $\xi = 100$, $\beta = 5$, and $\eta = 10$.

algorithm in Section 3. We selected frame length equal to 11 units and we chose the integers t_{kj} and s_{kj} in (5) equal to 11 and 50, respectively. We selected the threshold value of the monitoring algorithm equal to 1000, via inspection of the power and false alarm curves induced by various threshold values. For the ξ , β , and η constant values used in Figures 5–16, we simulated the monitoring algorithm. In all cases, the monitoring algorithm detected the rate change correctly. In particular, it detected such a change only when it actually occurred, in the case of the $\xi = 100$, $\beta = 10$, and $\eta = 50$ constants selection. In the latter case, we computed the distribution of the decision delay for the $\nu_u = 0.325$ to $\nu_l = 0.15$ change, as induced by the monitoring algorithm.

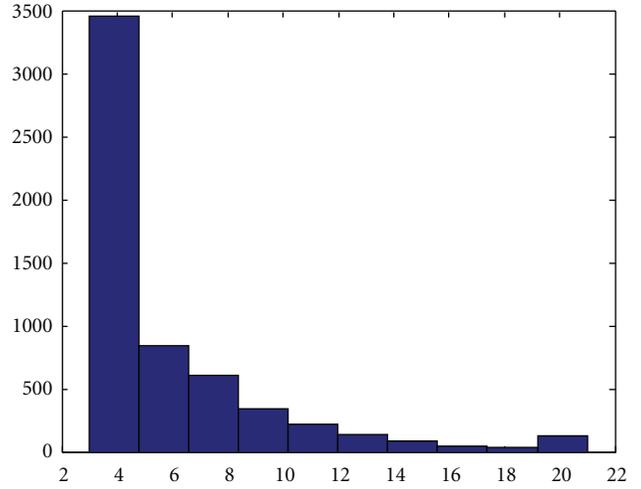


FIGURE 7: The delay distribution of nonexpired nodes for the subsystems 1 and 2 for $\xi = 100$, $\beta = 5$, and $\eta = 10$.

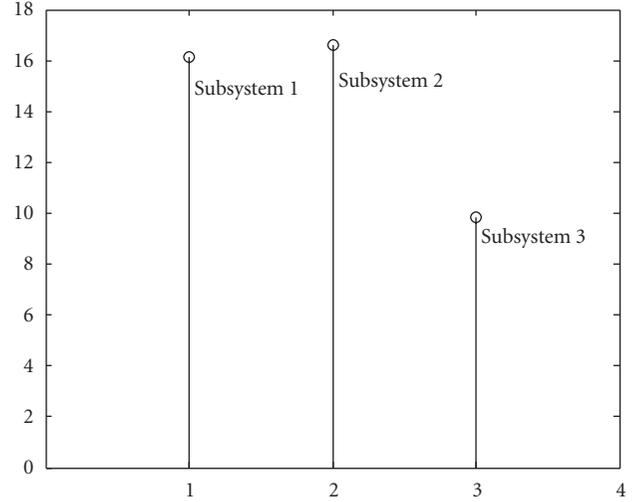


FIGURE 8: The expected transmission delays for $\xi = 100$, $\beta = 5$, and $\eta = 10$.

The result is exhibited in Figure 17. From the latter figure, we observe the effective response of the monitoring system.

8. Conclusions

We presented an algorithm for the architectural reconfiguration of clustered wireless sensor networks, where power consumption per sensor is modeled as a linear function of retransmission and channel monitoring periods imposed upon it. The algorithm is based on per cluster cumulative rate control, in contrast to other existing cluster formation schemes (LEACH-based) that ignore the composite rate factor; it is facilitated by a distributed and highly efficient rate monitoring algorithm. We evaluated the algorithm for a system deploying a powerful stable random access transmission protocol that accommodates high-priority data

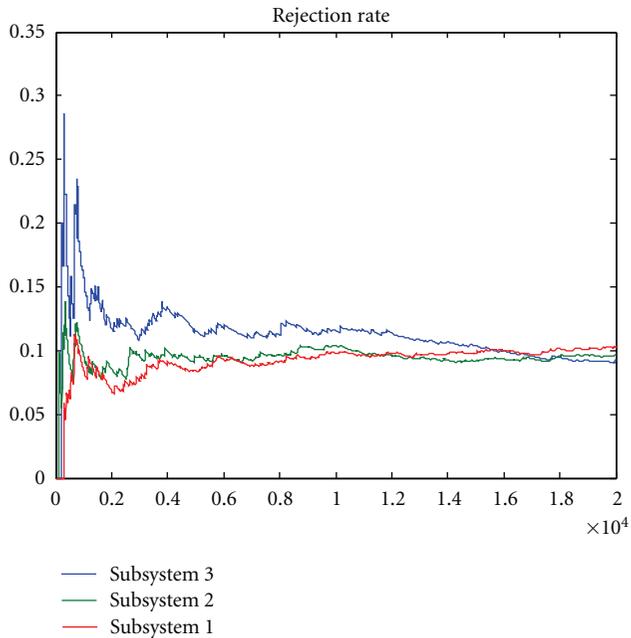


FIGURE 9: Rejection rate for $\xi = 200$, $\beta = 10$, and $\eta = 30$.

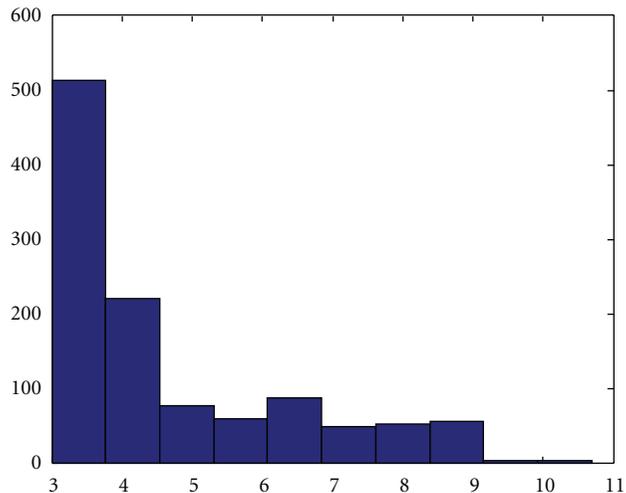


FIGURE 10: The delay distribution of nonexpired nodes for the subsystem 3 for $\xi = 200$, $\beta = 10$, and $\eta = 30$.

transmissions. The latter protocol is limited sensing and stable, reducing effectively the high-priority delays, while simultaneously accommodating low-priority delays satisfactorily. In contrast, the existing alternative proposals for transmission protocols (e.g., ZigBee, IEEE802.15.4) are unstable with rapidly decreasing throughput, while they also do not accommodate high-priority data. The overall system proposed in this paper is robust and highly effective, controlling expiration rates and transmission delays with high efficiency.

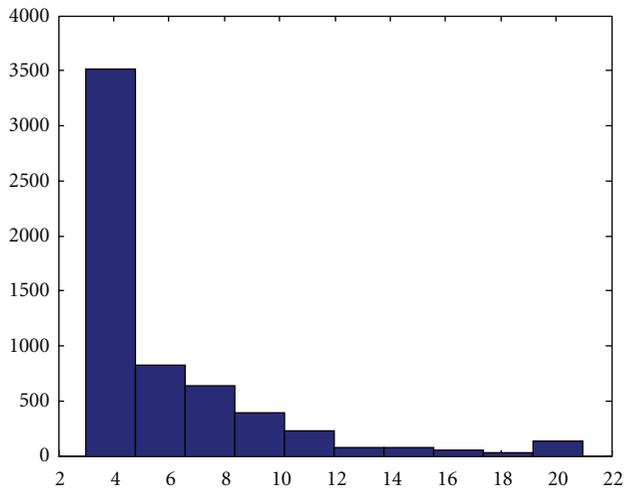


FIGURE 11: The delay distribution of nonexpired nodes for the subsystems 1 and 2 for $\xi = 200$, $\beta = 10$, and $\eta = 30$.

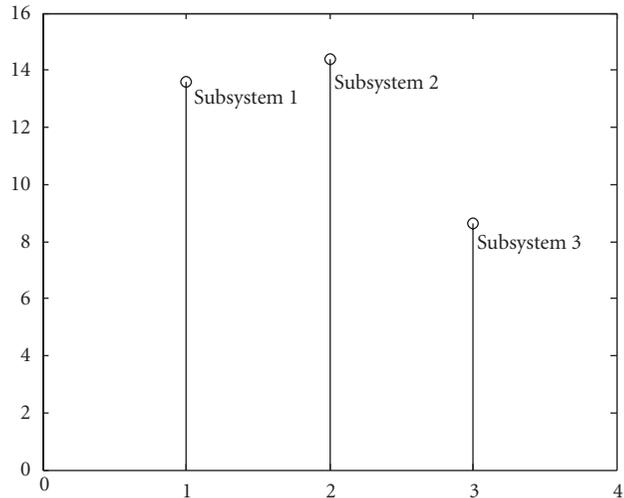


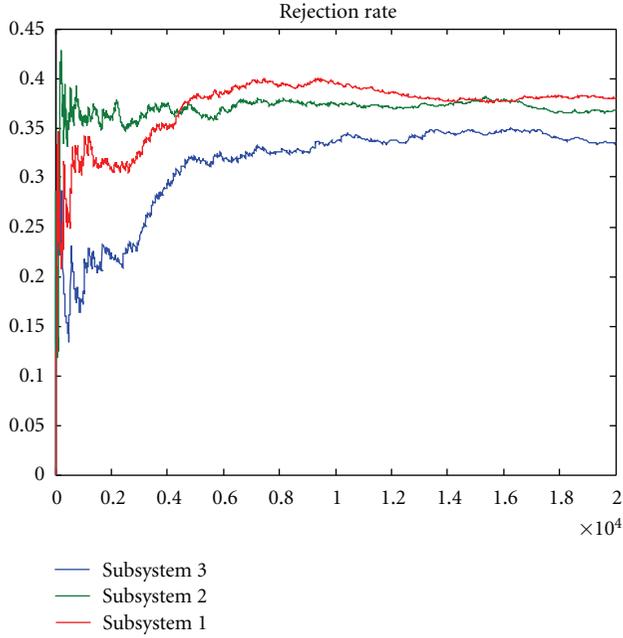
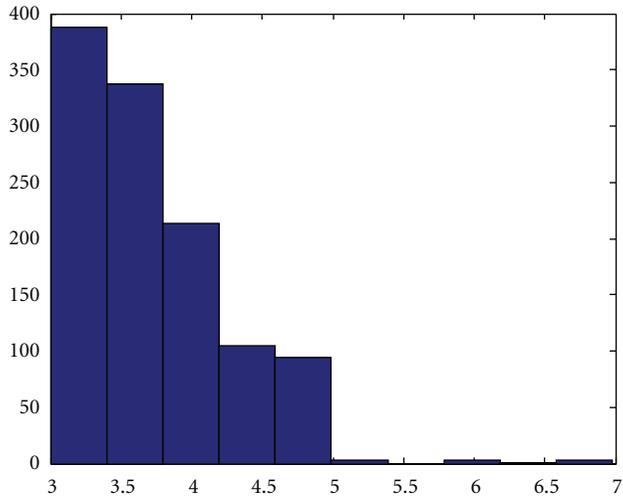
FIGURE 12: The expected transmission delays for $\xi = 200$, $\beta = 10$, and $\eta = 30$.

Appendices

A. Throughput Analysis of the Random Access Transmission Protocol

Consider either one of the local LSRAAs, whose operations are described in Section 4.1. Let us select the 3-cell algorithm among those in the class in [2]. Consider some CRI within the system of the LSRAA, which starts with transmissions from packet arrivals in an arrival interval of cumulative length u . We will call u , the “length of the examined interval.” Let us then define the following:

$E\{l \mid u\}$: given that the length of the examined interval is equal to u , the expected number of slots needed for its resolution, that is, for the successful


 FIGURE 13: Rejection rate for $\xi = 100$, $\beta = 10$, and $\eta = 50$.

 FIGURE 14: The delay distribution of nonexpired nodes for the subsystem 3 for $\xi = 100$, $\beta = 10$, and $\eta = 50$.

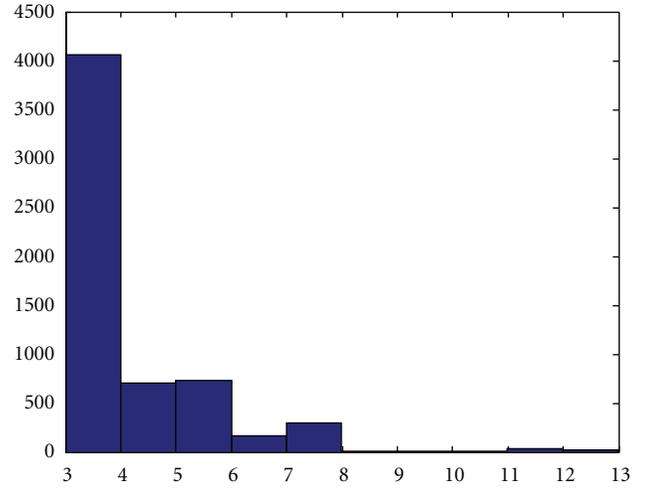
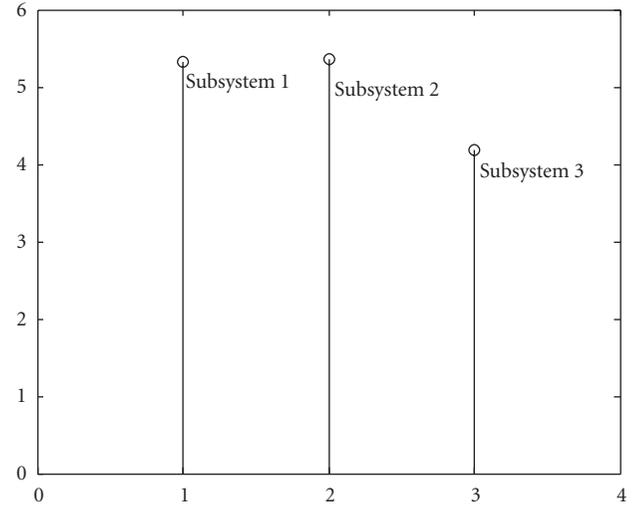
transmission of all the arrivals in the examined interval.

Let L_k denote the expected length of a CRI given that it starts with a collision of multiplicity k . We can then write the following:

$$E(l | u) = \sum_{k=0}^{\infty} E(l | u, k) e^{-\lambda u} \frac{(\lambda u)^k}{k!} = \sum_{k=0}^{\infty} L_k e^{-\lambda u} \frac{(\lambda u)^k}{k!}, \quad (\text{A.1})$$

since

$$E(l | u, k) = L_k \quad (\text{A.2})$$


 FIGURE 15: The delay distribution of nonexpired nodes for the subsystems 1 and 2 for $\xi = 100$, $\beta = 10$, and $\eta = 50$.

 FIGURE 16: The expected transmission delays for $\xi = 100$, $\beta = 10$, and $\eta = 50$.

depends only on k . In [2], and focusing on the 3-cell algorithm, we have shown that:

- (1) L_k ; $k \geq 0$ can be computed recursively;
- (2) L_k is quadratically upper bounded.

As found in [2], in the presence of the limit Poisson user model, each local LSRAA has then a throughput of 0.43, attained for window size $\Delta^* = 2.5599$. Thus, in view of the system considered in this paper, the Poisson intensities λ_j , $j = 1, 2, 3$ must satisfy the following necessary conditions, for overall system stability:

$$\lambda_1 < 0.43, \quad \lambda_2 < 0.43, \quad \lambda_1 + \lambda_2 + \lambda_3 < 2(0.43) = 0.86. \quad (\text{A.3})$$

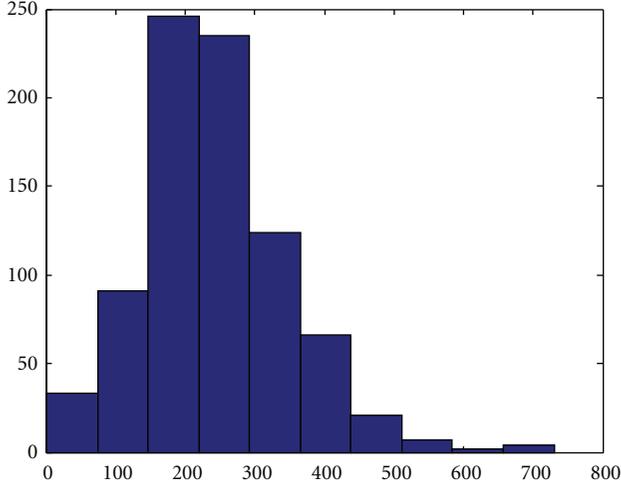


FIGURE 17: Distribution of the decision delay for threshold 1000 For $\xi = 100$, $\beta = 10$, and $\eta = 50$.

The necessary conditions in (A.3) determine a $\{\lambda_j, j = 1, 2, 3\}$ hyperplane, which generally contains the $\{\lambda_j, j = 1, 2, 3\}$ region that determines the system throughput. Tight bounds on the $\{\lambda_j, j = 1, 2, 3\}$ space, which provide the system throughput, will be attained via the system stability analysis below.

We consider the evolution of the algorithms in the two-channel system, and we assume that the system starts operating at time zero. Let us consider the sequence in time of the CRIs induced by the two LSRAAs in the system. Let the sequence $\{T_n\}_{n \geq 0}$ be such that: (a) for each n , T_n corresponds to the starting point of a slot which is the beginning of some CRI. We note that at T_n , two CRIs may simultaneously begin; one for each of the two LSRAAs in the system. (b) T_n is the first after T_{n-1} time instant which corresponds to the beginning of some CRI. (c) $T_0 = 2$, and at T_0 two CRIs begin; one for each of the two LSRAAs in the system.

Let $\{T_n^{(s)}\}_{n \geq 0}$ be the subsequence of sequence $\{T_n\}_{n \geq 0}$ which consists of those time instants when two CRIs begin simultaneously; one for each of the two LSRAAs in the system. Clearly, $T_0^{(s)} = T_0 = 2$. Let $D_{n,s}^{(j)} + 1, j = 1, 2, 3$ denote the total length of the unresolved arrival intervals in subsystem j , at the time instant $T_n^{(s)}$. $D_{n,s}^{(j)}$ is then called “the lag of subsystem j at time $T_n^{(s)}$.” From the algorithmic operations in the system, we conclude the following: (a) $D_{n,s}^{(j)} \geq 1$ and the values of $D_{n,s}^{(j)}$ are denumerable for all n and j . (b) $D_{0,s}^{(j)} = 1, j = 1, 2, 3$. (c) At time $T_n^{(s)}$, the LSRAA in cluster $k, k = 1, 2$, examines two arrival intervals: one from subsystem k which has length $\min(D_{n,s}^{(j)}, \Delta)$ and contains arrivals generated by a Poisson process with intensity λ_k , and one from subsystem 3 which has length $\min(D_{n,s}^{(3)}, \Delta)$ and contains arrivals generated by a Poisson process with intensity $0.5\lambda_3$. (d) The triple $(D_{n,s}^{(j)} + 1, j = 1, 2, 3)$ describes the state of the system at time $T_n^{(s)}$, and the sequence $\{S_n\}_{n \geq 0} \triangleq D_{n,s}^{(j)} + 1, j = 1, 2, 3$ is a three-dimensional irreducible and aperiodic Markov chain.

The stability of the system is represented by the ergodicity of the three-dimensional Markov chain $\{S_n\}_{n \geq 0}$, [20]. In addition, at time $T_n^{(s)}$, the backlogs of each of the two LSRAAs and of the overall system are all represented by the three lags, $D_{n,s}^{(j)}, j = 1, 2, 3$. Given $D_{n,s}^{(j)} = d_n^{(j)}, j = 1, 2, 3$, the expected system backlog at time $T_n^{(s)}$ is $\sum_{j=1}^3 \lambda_j d_n^{(j)}$, where $V(\bar{d}_n) = V(\{d_n^{(j)}\}_{1 \leq j \leq 3}) \triangleq \sum_{j=1}^3 \lambda_j d_n^{(j)}$ is a Lyapunov function of the three systems lags $\{d_n^{(j)}\}_{1 \leq j \leq 3}$. Let C denote the state space of the Markov chain $\{S_n\}_{n \geq 0}$, and let us then define the operator $AV(\bar{d}) = AV(\{d_n^{(j)}\}_{1 \leq j \leq 3})$ called a generalized drift, as follows:

$$\begin{aligned}
 AV(\bar{d}) &= AV\left(\{d^{(j)}\}_{1 \leq j \leq 3}\right) \triangleq E\left\{V\left(\{D_{n+1,s}^{(j)}\}_{1 \leq j \leq 3}\right)\right. \\
 &\quad \left.- V\left(\{D_{n,s}^{(j)}\}_{1 \leq j \leq 3}\right) \mid D_{n,s}^{(j)} = d^{(j)}, j = 1, 2, 3\right\} \\
 &= E\left\{\sum_{j=1}^3 \lambda_j D_{n+1,s}^{(j)} \mid D_{n,s}^{(j)} = d^{(j)}, j = 1, 2, 3\right\} \\
 &\quad - \sum_{j=1}^3 \lambda_j d^{(j)}; \quad \bar{d} = \{d^{(j)}\}_{1 \leq j \leq 3} \in C.
 \end{aligned} \tag{A.4}$$

Thus, the generalized drift in state \bar{d} is the expected change in backlog over one slot time, starting in state \bar{d} , and it can be used to establish necessary and sufficient conditions for the ergodicity of the Markov Chain $\{S_n\}_{n \geq 0}$ [20]. To see that, let us define the following:

$$\begin{aligned}
 p_{\bar{d},\bar{e}} &\triangleq P(\bar{d}_{n+1} = \bar{e} \mid \bar{d} = \bar{d}), \\
 F_{\bar{d},V}(z) &\triangleq (1-z)^{-1} \left[z^{V(\bar{d})} - \sum_{\bar{e} \in C} p_{\bar{d},\bar{e}} z^{V(\bar{e})} \right]; \\
 z &\in [0, 1), \quad \bar{d} \in C,
 \end{aligned} \tag{A.5}$$

where $F_{\bar{d},V}(z)$ is the generalized Kaplan function on the Lyapunov function V [22]. From the results in [1], we can then express the following proposition.

Proposition 1. *Assume that the following hold:*

- (i) $|AV(\bar{d})| < \infty, \quad \forall \bar{d} \in C;$ (A.6)
- (ii) *there exist finite subset $H_2 \in C$ and some nonnegative finite constant B such that*

$$F_{\bar{d},V}(z) \geq -B, \quad \forall z \in [0, 1), \quad \forall \bar{d} \in C - H_2. \tag{A.7}$$

Then,

- (A) *if there exists a finite subset H_1 of C , and some $\varepsilon' > 0$ such that*

$$AV(\bar{d}) < \varepsilon', \quad \forall \bar{d} \in C - H_1, \tag{A.8}$$

then the Markov chain $\{S_n\}_{n \geq 0}$ is ergodic;

(B) if H is a proper subset of C such that

$$\inf_{\bar{d} \in [C-H]} V(\bar{d}) > \sup_{\bar{d} \in H} V(\bar{d}), \quad (\text{A.9})$$

$$AV(\bar{d}) > 0, \quad \forall \bar{d} \in C - H,$$

then the Markov chain $\{S_n\}_{n \geq 0}$ is nonergodic.

Remark 2. Condition (A.7) is trivially satisfied for Markov chains, which are uniformly downward bounded with respect to the Lyapunov function V , that is, for Markov chains such that, $P(V(\bar{D}_{n+1}) < V(\bar{d}_n) - x) = 0$, for all $x > \alpha$, for some positive and finite constant α and for all \bar{d}_n in C . In the problem studied in this paper, when $\lambda_3 = 0$, two decoupled systems arise. Each of those systems corresponds to a local LSRAA, whose lags at collision resolution points form a uniformly downward bounded Markov chain; the Lyapunov function $V(\bar{d})$ reduces then to $d^{(j)}$, for the system in channel j , $j = 1, 2$, and the constant α equals then $[\Delta] = 2$.

We now express a lemma whose proof is as in Lemma 1 in [4].

Lemma 3. Consider the Markov Chain $\{S_n\}_{n \geq 0} = \{D_{n,s}^{(j)}, j = 1, 2, 3\}_{n \geq 0}$ and its Lyapunov function $(\bar{d}_n) = \sum_{1 \leq j \leq 3} \lambda_j d_n^{(j)}$, and let C be the state space of $\{S_n\}_{n \geq 0}$. Let $AV(\bar{d})$ and $F_{\bar{d},V}(z)$ be as in (A.4) and (A.5), respectively. Then,

(i)

$$|AV(\bar{d})| < \infty, \quad \forall \bar{d} \in C, \quad \forall \{\lambda_i\} \text{ such that } \sum_{j=1}^3 \lambda_i \leq 1; \quad (\text{A.10})$$

(ii) there exist a positive finite constant B and some finite subset H_2 of C such that

$$F_{\bar{d},V}(z) \geq -B, \quad \forall z \in [0, 1), \quad \forall \bar{d} \in C - H_2, \quad (\text{A.11})$$

$$\forall \{\lambda_j\} \text{ such that } \sum_{j=1}^3 \lambda_i \leq 1.$$

Due to Lemma 3, and in view of the proposition, the search for the ergodicity versus the nonergodicity of the Markov chain $\{S_n\}_{n \geq 0}$ reduces to the identification of appropriate properties for the satisfaction of conditions (A) and (B) in the proposition, respectively.

Consider $T_n^{(s)}$, and let the lags $\{D_{n,s}^{(j)}\}_{1 \leq j \leq 3}$ be sufficiently long, so that in the time interval $[T_n^{(s)}, T_{n+1}^{(s)}]$, each CRI from each of the two LSRAAs in the system, resolves an arrival interval of length Δ . Given system Poisson rates $\{\lambda_j\}_{1 \leq j \leq 3}$, and for $E_\lambda \{l | u\}$ as in (A.1), let us then define the following:

$i = 1, 2$; $N_i(\{\lambda_j\}, \Delta)$: given $\{\lambda_j\}$, the number of CRIs generated by the algorithm in channel i in $[T_n^{(s)}, T_{n+1}^{(s)}]$, when each CRI resolves arrival intervals of length Δ , from both its low- and high-priority users. The first such CRI starts at $T_n^{(s)}$, and the last such CRI ends at $T_{n+1}^{(s)}$.

$AV(\bar{d}_\Delta, \{\lambda_j\})$: The generalized drift in (2), for given Poisson rates $\{\lambda_j\}$, when, in $[T_n^{(s)}, T_{n+1}^{(s)}]$, each CRI generated by either algorithm in the system, resolves arrival intervals of length Δ .

$$i = 1, 2; \quad A^{(i)}V(\bar{d}_\Delta, \{\lambda_j\}) \stackrel{\Delta}{=} E_{\lambda_i + \lambda_{3/2}} \{l | \Delta\} - \Delta + [E\{N_i(\{\lambda_j\}, \Delta)\} - 1][E_{\lambda_i + \lambda_3} \{l | \Delta\} - \Delta]. \quad (\text{A.12})$$

Then, via (3), we easily derive the following expression:

$$AV(\bar{d}_\Delta, \{\lambda_j\}) = \left(\lambda_1 + \frac{\lambda_3}{2}\right)A^{(1)}V(\bar{d}_\Delta, \{\lambda_j\}) + \left(\lambda_2 + \frac{\lambda_3}{2}\right)A^{(2)}V(\bar{d}_\Delta, \{\lambda_j\}) - \frac{\lambda_3 \Delta}{2} [E\{N_1(\{\lambda_j\}, \Delta)\} + E\{N_2(\{\lambda_j\}, \Delta)\} - 2], \quad (\text{A.13})$$

where

$$E_{\lambda_i + \lambda_{3/2}} \{l | \Delta\} + [E\{N_1(\{\lambda_j\}, \Delta)\} - 1][E_{\lambda_i + \lambda_3} \{l | \Delta\}] = E_{\lambda_2 + \lambda_{3/2}} \{l | \Delta\} + [E\{N_2(\{\lambda_j\}, \Delta)\} - 1] \times [E_{\lambda_2 + \lambda_3} \{l | \Delta\}]. \quad (\text{A.14})$$

We note that $A^{(i)}V(\bar{d}_\Delta, \{\lambda_j\})$ in (A.12) is a generalized drift for the algorithm in cluster i . We now state a theorem whose proof is as in Theorem 1 in [4].

Theorem 4. (i) Let there exist some $\varepsilon > 0$ such that the two conditions below are satisfied:

$$A^{(1)}V(\bar{d}_\Delta, \{\lambda_j\}) < -\varepsilon, \quad (\text{A.15})$$

$$A^{(2)}V(\bar{d}_\Delta, \{\lambda_j\}) < -\varepsilon.$$

Then, the Markov chain $\{S_n\}_{n \geq 0}$ is ergodic at the Poisson rates $\{\lambda_j\}_{1 \leq j \leq 3}$.

(ii) Let at least one of the two generalized drifts, $A^{(i)}V(\bar{d}_\Delta, \{\lambda_j\})$, $i = 1, 2$, be nonnegative.

Then, the Markov chain $\{S_n\}_{n \geq 0}$ is nonergodic at the Poisson rates $\{\lambda_j\}_{1 \leq j \leq 3}$.

The conditions (A.15) in the theorem define lower bounds on the $\{\lambda_j\}$ regions for which the Markov chain $\{S_n\}_{n \geq 0}$ is ergodic. Since the constant ε in (A.15) can be arbitrarily small, and in combination with the statement in part (ii) of the theorem, we conclude that the ergodicity bounds determined by (A.15) are tight.

In [4], a methodology regarding the computation of the $\{\lambda_j\}$ values, for which the Markov chain $\{S_n\}_{n \geq 0}$ is ergodic, is described, leading to the system stability regions.

B. Performance of the Data Rate Monitoring Protocol

In this appendix, we summarize the performance of the data rate monitoring high-level protocol.

B.1. Performance Characteristics of a Simple Algorithmic System. Let us consider the distinct rates $\lambda_1, \dots, \lambda_n$, and the corresponding arbitrary dimensionality distributions f_1, \dots, f_n of the arrivals, as defined in the beginning of this section. For given rates $\lambda_1, \dots, \lambda_n$, let the latter distributions be well known, and consider then the algorithmic system of $(n-1)$ algorithms, each monitoring a change from the rate λ_k to one of the remaining rates. The $\{\lambda_k \rightarrow \lambda_j\}$ algorithm utilizes the observed per frame number of arrivals and the distributions f_k and f_j in its updating log-ratio steps (see (2) in Section 5), while the parallel algorithms utilize a common threshold η_k . We now state a lemma that is included in [12].

Lemma 5. *Let the distributions f_1, \dots, f_n satisfy ϕ -mixing conditions as and consider the system of the $(n-1)$ parallel algorithms $\{\lambda_k \lambda_\rho\}; \rho \neq k$, from the time instant when they start operating. Let the rate λ_i be acting throughout the total observation time interval, and let then $N_{\{\lambda_k \rightarrow \lambda_j\}}^{\lambda_i}$ denote the stopping variable of the $\{\lambda_k \rightarrow \lambda_j\}$ algorithm, given the decisive common threshold η_k , that is, $N_{k_j}^i$ is the random variable denoting the number of frames needed for the $\{\lambda_k \rightarrow \lambda_j\}$ algorithm to first cross the threshold η_k , when arrivals are generated by the distribution/process f_i throughout. Then,*

$$\lim_{\eta_k \rightarrow \infty} E \left\{ N_{\{\lambda_k \rightarrow \lambda_j\}}^{\lambda_i} \right\} \approx \begin{cases} \frac{\eta_k}{2}; & \text{if } K\left(\frac{f_i}{f_k}\right) - K\left(\frac{f_i}{f_j}\right) < 0; \\ \frac{\log(\eta_k)}{K(f_i/f_k) - K(f_i/f_j)}; & \text{if } K\left(\frac{f_i}{f_k}\right) - K\left(\frac{f_i}{f_j}\right) > 0, \end{cases} \quad (\text{B.1})$$

where $K(f_p/f_i)$ is the Kullback-Leibler number [1] of the distribution/process of f_p , with respect to the process f_i , subject to the existence of the latter number.

As a direct consequence of (4), when the rate λ_i ($\lambda_i \neq \lambda_k$) acts throughout, then, asymptotically (for $\eta_k \rightarrow \infty$), the $\{\lambda_k \rightarrow \lambda_j\}$ algorithm stops first, in the expected stopping time sense. That is, λ_i acting throughout, and $\lambda_i \neq \lambda_k$:

$$\lim_{\eta_k \rightarrow \infty} E \left\{ N_{\{\lambda_k \rightarrow \lambda_i\}}^{\lambda_i} \right\} \approx \frac{\log(\eta_k)}{K(f_i/f_k)} < \lim_{\eta_k \rightarrow \infty} E \left\{ N_{\{\lambda_k \rightarrow \lambda_i\}}^{\lambda_j} \right\}; \quad \forall \lambda_j \neq \lambda_i; \lambda_j \neq \lambda_k. \quad (\text{B.2})$$

If λ_k acts throughout, on the other hand, the expected stopping time of each $\{\lambda_k \rightarrow \lambda_j\}; j \neq k$ algorithm is asymptotically ($\eta_k \rightarrow \infty$) of the same order with that of the threshold value.

The implementation of the algorithm is clearly determined by its nonasymptotic performance: its performance for finite values of the threshold η_k . Given a threshold value η_k , the performance of the $(\lambda_k \rightarrow \lambda_j)$ monitoring algorithm is basically characterized by two time curves: the power and false alarm curves, denoted, respectively, by $\beta_{kj}(r)$ and $\alpha_{kj}(r)$, where r denotes the time instant T_r , as in (B.1), and where

$\beta_{kj}(r)$: the probability that the $\lambda_k \rightarrow \lambda_j$ monitoring algorithm crosses threshold η_k before or at time T_r , given that the acting rate is λ_j throughout, named power curve;

$\alpha_{kj}(r)$: the probability that the $\lambda_k \rightarrow \lambda_j$ monitoring algorithm crosses threshold η_k before or at time T_r , given that the acting rate is λ_k throughout, named false alarm curve.

B.2. Reinitialization Performance. Above, we summarized the performance of a simple single-shot system of parallel algorithms, that monitor a shift from a given rate λ_k to any one of the remaining $n-1$ rates. Incorporating the common threshold η_k utilized by this system, we denote it $\{\lambda_k \rightarrow \lambda_j; \text{for all } j \neq k, \eta_k\}$. The algorithmic system in ([19, Section 5]) reinitializes each time it makes a decision. When the algorithm $\{\lambda_k \rightarrow \lambda_j; j \neq k, \eta_k\}$ decides in favor of a rate λ_ρ , a new algorithm $\{\lambda_\rho \rightarrow \lambda_j; j \neq \rho, \eta_\rho\}$ immediately starts and this process continuous on. The latter reinitialization process gives rise to stability issues: descriptively, the rate of the actual rate changes cannot be faster than the rate of algorithmic decisions. Below, we define stability in precise mathematical terms.

Let $C_{ei}(\lambda_i); i = 0, 1, \dots, (n-1)$ denote disjoint rate regions, respectively, centered around the rates $\lambda_i, i = 0, 1, \dots, (n-1)$. Let the traffic monitoring systems be designed at the latter central rates. Let $\lambda'_i; i = 0, 1, \dots, (n-1)$ denote the acting rates, where $\lambda'_i \in C_{ei}(\lambda_i)$ for all i . Let us assume that the arrival processes are all memoryless, and let the process which generates the actual rate shifts, from some λ'_i to some λ'_j , be homogeneous Markov. Let $\lambda_I < \lambda_{I+1}$ for all i . Let us then define the following:

$E\{l_i\}$: the expected length, in time units, during which the rate λ'_i is continuously acting;

$\bar{\lambda}'$: the vector $[\lambda'_0, \lambda'_1, \lambda'_2, \lambda'_3, \dots, \lambda'_{n-1}]$.

Consider the sequence $\{S_m\}$ of time instants when shifts in rates are declared by the algorithmic system. Let λ'_{im} denote the acting data rate, and let λ_{im} denote the rate decided by the rate monitoring system at S_m . Let X_m be the random variable denoting the data backlog at S_m . Let W_m be the random variable which denotes the distance from S_m of the instant when λ'_{im} started acting. Then, due to the memoryless and homogeneous Markov assumptions given above, $\{\lambda'_{im}, \lambda_{im}, W_m, X_m\}$ forms a Markov chain. System stability corresponds, then, to ergodicity of the latter Markov chain [20], which by the way, is clearly uniformly downward bounded [20]. We now express a Lemma, stated in [12].

Lemma 6. *Let the data arrival processes be memoryless, and let the processes that generate the actual rate shift be homogeneous Markov. Let the algorithmic thresholds $\{\eta_k\}_{0 \leq k \leq n-1}$ be such that $\eta_k \rightarrow \infty$; for all k . Then, the following sufficient and necessary conditions for the algorithmic stability hold, where $K(\lambda'_i/\lambda_m)$ is the Kullback-Leibler number as in Lemma 1, where*

$$K_{\max}(\lambda_m) \triangleq \max_{\lambda'_i} K\left(\frac{\lambda'_i}{\lambda_m}\right), \quad (\text{B.3})$$

$$\text{ind}(\lambda'_m, \lambda_m) \triangleq \begin{cases} m-1; & \text{for } \lambda'_m < \lambda_m; \\ m; & \text{for } \lambda'_m \geq \lambda_m. \end{cases}$$

Sufficient condition:

$$\sum_{m=0}^{n-1} \left\{ \frac{\log(\eta_m)}{K_{\max}(\lambda_m)} \sum_{k=0}^{\text{ind}(\lambda'_m, \lambda_m)} (\lambda'_k - \lambda_m) E\{l_k\} + \frac{\eta_m}{2} \sum_{k=\text{ind}(\lambda'_m, \lambda_m)+1}^{n-1} (\lambda'_k - \lambda_m) E\{l_k\} \right\} \leq 0. \quad (\text{B.4})$$

Necessary condition:

$$\sum_{m=0}^{n-1} \left\{ \frac{\eta_m}{2} \sum_{k=0}^{\text{ind}(\lambda'_m, \lambda_m)} (\lambda'_k - \lambda_m) E\{l_k\} + \frac{\log(\eta_m)}{K_{\max}(\lambda_m)} \sum_{k=\text{ind}(\lambda'_m, \lambda_m)+1}^{n-1} (\lambda'_k - \lambda_m) E\{l_k\} \right\} \leq 0. \quad (\text{B.5})$$

The sufficient and necessary conditions for system stability in the above Lemma are asymptotic. In general, both asymptotic and nonasymptotic conditions relate rates of actual data rate changes to rates of convergence of the algorithmic monitoring system, for stability, where stability translates to maintaining the data traffics with finite delays (nonincreasing expected backlogs). All conditions basically require that acting high rates be bursty.

References

- [1] D. Kazakos and P. Papantoni-Kazakos, *Detection and Estimation*, Computer Science Press, New York, NY, USA, 1990.
- [2] A. T. Burrell and T. P. Papantoni, "A class of limited sensing random access algorithms with resistance to feedback errors and effective delay control," *Journal of Communications and Networks*, vol. 8, no. 1, pp. 21–27, 2006.
- [3] P. Papantoni-Kazakos, "Multiple-access algorithms for a system with mixed traffic: high and low priority," *IEEE Transactions on Communications*, vol. 40, no. 3, pp. 541–555, 1992.
- [4] P. Papantoni-Kazakos, H. Delic, M. Paterkis, and M. Liu, "Transmission algorithms for a multi-channel packet radio system with priority users," *International Journal of Digital and Analog Communication Systems*, vol. 6, no. 4, pp. 193–212, 1993.
- [5] M. Paterakis and P. Papantoni-Kazakos, "Simple window random access algorithm with advantageous properties," *IEEE Transactions on Information Theory*, vol. 35, no. 5, pp. 1124–1130, 1989.
- [6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [7] J. H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '00)*, pp. 22–31, March 2000.
- [8] Y. T. Hou, Y. Shi, and H. D. Sherali, "On node lifetime problem for energy-constrained wireless sensor networks," *Mobile Networks and Applications*, vol. 10, no. 6, pp. 865–878, 2005.
- [9] Y. T. Hou, Y. Shi, and H. D. Sherali, "Rate allocation and network lifetime problems for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 321–334, 2008.
- [10] M. Bhardwaj and A. P. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," in *Proceedings of the IEEE Infocom*, pp. 1587–1596, June 2002.
- [11] R. Wattenhofer, L. Li, P. Bahl, and Y. M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1388–1397, April 2001.
- [12] T. Papantoni-Kazakos and A. T. Burrell, "The implementation of dynamic rate allocation in sensor networks," *Journal of Intelligent and Robotic Systems*, vol. 58, no. 3–4, pp. 211–238, 2010.
- [13] D. M. Blough and P. Santi, "Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks," in *Proceedings of The 8th Annual International Conference on Mobile Computing and Networking*, pp. 183–192, September 2002.
- [14] Y. T. Hou, Y. Shi, J. H. Reed, and K. Sohrawy, "Flow routing for variable bit rate source nodes in energy-constrained wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '05)*, pp. 3057–3062, May 2005.
- [15] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, "Cooperation in wireless ad hoc networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, pp. 808–817, April 2003.
- [16] A. T. Burrell and P. Papantoni-Kazakos, "Performance monitoring in sensor networks," in *Proceedings of the 6th International Conference on Information Technology: New Generations (ITNG '09)*, pp. 607–612, Las Vegas, Nev, USA, April 2009.
- [17] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, Englewood Cliffs, NJ, USA, 1992.
- [18] H. Luss and D. R. Smith, "Resource allocation among competing activities: a lexicographic minimax approach," *Operations Research Letters*, vol. 5, no. 5, pp. 227–231, 1986.
- [19] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, pp. 1713–1723, April 2003.
- [20] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*, Springer, 1993.
- [21] A. Papoulis and S. U. Pillai, *Random Variables and Stochastic Processes*, McGraw-Hill, New York, NY, USA, 4th edition, 2002.
- [22] M. Kaplan, "A sufficient condition for nonergodicity of a Markov chain," *IEEE Transactions on Information Theory*, vol. 25, no. 4, pp. 470–471, 1979.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

