Traffic Flow Analysis of a Multi-hop Wireless Sensor Network Subject to Node Failure
Shensheng Tang
Missouri Western State University, Department of Engineering Technology, 4525 Downs Dr., St. Joseph, MO 64507, USA
stang@missouriwestern.edu

Abstract: We develop an analytic model for traffic flow analysis in a multi-hop wireless sensor network subject to node failure through a queueing theoretic framework. The dynamics of traffic flow in the source-to-sink path is modeled by a number of single-server queues with finite capacity linked in tandem. Both the blocking effect of tandem queues due to limited buffer and the impact of node failure due to limited node power are taken into account during modeling. The tandem-node path is analyzed by decomposing it into individual nodes with modified arrival and service processes as well as queue capacities. An iterative algorithm is developed for evaluating the performance metrics. One metric, i.e., source-to-sink delay, is used for studying QoS (quality of service) control. A simple QoS control scheme is proposed and its effectiveness is validated by simulations.

Keywords: Wireless sensor network, Tandem queueing network, Traffic modeling, Source-to-sink delay (SSD), Node failure.

1. Introduction
In the past a few years, the research of wireless sensor networks (WSNs) has gained increasing significance due to their enormous potential applications, such as civilian surveillance, environment monitoring, biological detection and situational awareness in the battlefield [1]. A typical WSN consists of a large amount of such sensors that are capable of probing the environment and reporting the collected data to a command center (data sink) in wireless manner. In some applications, several sets of sensors are grouped to form clusters for performing sensing tasks and data transmission in a geographic area of interest. Using a clustering approach, sensors can be managed locally by a cluster head (CH), a node that is responsible for managing the cluster and relaying data to other CHs or the sink. In addition, clustering provides inherent optimization capabilities at CHs, such as data pre-processing.

Most applications in WSNs involve battery-powered nodes with limited energy. Their batteries may not be convenient for recharging or replacing. When a node exhausts its energy, it cannot sense or relay data any more. Thus, much research on sensor networks focused on protocols related to energy efficient mechanisms [2][3]. However, with the development of a wide range of sensor applications, especially the real-time applications by employing imaging and video sensors, the consideration of QoS (quality of service) issues has been required for the design of WSNs. In this case, multiple types of traffic (e.g., real-time and non-real-time traffic) may coexist in the network. So a service differentiation mechanism is needed in order to guarantee the reliable delivery of the real-time data [4]. Moreover, in sensor networks different data packets might have different importance. For example, in the forest fire alarming system, the packet containing the 1000F temperature is much more important than the packet containing 60F temperature, since the former case could probably mean a forest fire [5].

Service differentiation is a basic way to provide QoS by giving one priority over another. QoS parameters for WSN applications may include bandwidth, packet delay, packet loss rate, jitter, reliability, coverage, power, etc. A few research works have been done focusing on different aspects of QoS issues [4][5][6][7]. In [4], an energy-aware QoS routing protocol was proposed to find a least cost and energy efficient path that meets certain end-to-end delay for real-time data. The link cost captures node’s energy reserve, transmission, error rate and other communication parameters. In [5], the concept of service differentiation based on prioritization was addressed, and an adaptive forwarding scheme (AFS) was proposed to control the reliability of communications. In [6], a QoS routing protocol called SPEED was presented for sensor networks. An end-to-end soft real-time communication was achieved by maintaining a desired delivery speed across the sensor network via a combination of feedback control and non-deterministic geographic forwarding. In [7], a QoS supporting scheme was developed for dynamic traffic conditions, and an explicit solution to the energy allocation for different clusters was obtained based on an optimal allocation criterion.

We have noticed that, the QoS solutions described above do not consider the impact of unreliable nodes or links, which is one of the main features in infrastructure-less networks (e.g., ad hoc, sensor networks). In [8], a systematic medium-scale measurement of packet delivery for dense wireless sensor networks in three different environments, an indoor, a habitat, and an open parking lot, indicates that the packet delivery performance in these environments is fairly pessimistic.

In this paper, we develop an analytic traffic model with service differentiation for a cluster based multi-hop WSN subject to node failure. The collected data traffic is gathered from an area of interest and relayed from one cluster to another until to the sink. The contribution of the work is summarized as follows.
• The dynamics of traffic flow at each node in the source-to-sink path is modeled and analyzed by considering the impact of the node’s upstream and downstream nodes, not just that node itself.
• The impact of node failure is combined into the traffic model to adapt to more realistic situations in WSNs.
• A single-node decomposition method is developed for the traffic flow analysis of an individual node in the tandem-node path. An iterative algorithm is proposed for computing performance metrics.
• A simple QoS control scheme is proposed using a derived QoS metric called source-to-sink delay (SSD).
The goal of the scheme is to guarantee the QoS requirement within a predefined range under dynamic load conditions.

The remainder of the paper is organized as follows. Section 2 presents the problem formulation and traffic modeling of the WSN. Section 3 analyzes the traffic model by a single-node decomposition method. A critical performance metric called SSD is derived. Based on which, a QoS control scheme is developed in Section 4, and its effectiveness is validated by simulations. Finally, the paper is concluded in Section 5.

2. Problem Formulation and Modeling

Consider a class of wireless sensor networks, where the monitored target or sensing task is pre-determined and a set of clusters are thus built between the area of interest and the command center (sink). The formed cluster-based WSN may be an independent network or a virtual network built over a larger sensor network. The latter is like a virtual private network (VPN) over a larger network such as Internet [9].

In the cluster-based WSN, the sensor nodes are grouped into clusters controlled by the sink. The clusters include the sensing cluster and relaying clusters. The sensing cluster may comprise multiple types of sensors depending on application requirements, such as temperature-type, humidity-type, pressure-type, acoustic-type, image-type, video-type, etc. Hence, multiple types of traffic are generated from the monitored area and transferred to the sink through a number of relaying clusters. These traffic types can be distinguished by different criticality or priority according to the application requirements. A relaying cluster may include just a few sensors for the purpose of traffic forwarding and CH substitution. For simplicity, we assume that the relaying cluster only contributes to the aforementioned pre-determined target/task monitoring.

In each cluster, there is a cluster-head (CH) that manages nodes in that cluster. Nodes and CHs are assumed to be stationary or limited moving inside respective clusters. A CH is located within the communication range of all the nodes of its cluster and can communicate with its neighbor CHs. Since a CH forwards traffic to other CHs with longer distance as compared to the sensing nodes, in some applications it is designed more powerfully in terms of energy, bandwidth and memory [10][11], while others select CHs from the deployed sensors [12]. When a CH is under failure condition due to insufficient power, another CH will be selected among the existing nodes (Suppose a node can adjust its transmission power to assume the role of CH; the details on transmission power control issues is referenced to [13]). It is noteworthy that, in our study, the sensing cluster is assigned much more sensor nodes than a relaying cluster, since the former needs to perform the monitoring task in the area of interest while the latter mainly performs the relaying task. This cluster-based architecture raises many interesting issues such as cluster formation (e.g., considering the different characteristics of sensing cluster and relaying cluster), CH selection, and cluster maintenances (e.g., CH replacing under failure). Here we only focus on modeling the traffic flow behavior along with the source-to-sink path under possible failures.

The system model consists of three parts: 1) sensing part, the sensors around the target area are responsible for probing the target/event and sending the collected data to their CH (called sensing CH); 2) relaying part, the collected data are relayed from the sensing CH to other CHs (called relaying CHs) until to the sink; 3) sink, the sink performs system-level data analysis and processing for an overall situation awareness. Therefore, the collected data from the sensing CH is transferred to the 1st relaying CH, then to the 2nd relaying CH, and so forth until it reaches the sink. Note here we focus our analysis on the data traffic flow while the control signaling between upstream and downstream nodes is assumed to be ideal.

Without loss of generality, we assume that the traffic flow is transferred through M intermediate nodes, which are modeled by M finite-capacity queues linked in tandem. We classify all the sensing traffic types into two categories in accordance with their criticality: high priority (HP) traffic and low priority (LP) traffic. The packets of each traffic class are labeled, and are checked on each CH by a classifier and sent to their corresponding buffers (thus there are two buffers at each CH). The server on each CH provides service to each traffic class according to a proportion factor β, which is an adjustable control parameter, defined as the percentage of the service rate at the node for HP traffic.

The server on each CH is subject to failure due to the nature of sensors. We define a power threshold for a CH which corresponds to the minimum power for it to transmit data to the next CH. When the CH’s power is consumed to reach the predefined threshold, the CH is said to be under failure condition (i.e., the CH cannot transmit data any more) and a CH selection procedure starts. Note that the CH in the defined failure condition is not completely down and can still do some auxiliary work, such as switching its remaining data to the new CH or receiving data from its upstream CH if required. When an HP data packet completes its service at the ith CH, it proceeds to the (i+1)st CH if the corresponding buffer is not full. If, however, the buffer is full, the HP packet is blocked and forced to wait at the ith CH until a departure of HP packet occurs from the (i+1)st CH. During this time, the ith CH can still provide service to the LP packet since each type of traffic has its own buffer. A similar situation holds for a LP packet.

3. Traffic Flow Analysis

The above problem can be translated into a modified “queueing network with blocking” problem (plus node failure conditions). In general, queueing networks with blocking do not have exact product-form solutions [16]. Exact solutions can only be obtained by numerical techniques [17], or under special constraints [18].

1 However, the relaying node can be extended to forward traffic from other paths (if any) not belong to proposed WSN, where some constraints would be introduced, e.g., the merged traffic should not interfere with the operations of existing traffic, or the existing traffic must have a preemptive priority over the merged traffic.

2 This scenario is reasonable since in the wireless communications manner, receiving packets usually consumes less energy than transmitting packets [14][15].

3 Here we assume each packet at a CH is blocked only by its immediate downstream CH; “blocked by multiple downstream CHs” would lead to more complicated analysis. Note also that this type of blocking should not be confused with the notion of a call getting blocked in a cellular system, which means that the call gets lost if it arrives to find that the cell is full.
Consequently, much work in the literature has been devoted to approximate solutions [19]. Most of the approximation methods are to decompose the queueing network into a set of smaller subsystems, such as single-node or two-node decomposition.

We shall analyze the above traffic flow model by using a single-node decomposition method, i.e., by decomposing the tandem queueing network into individual nodes with modified arrival and service processes and modified queue capacities. The impact of the individual node’s upstream and downstream nodes is taken into account.

Assume that the HP traffic and LP traffic are generated independently and the arrival process to each individual CH (node) is approximated as Poisson. The service time of the ith CH is exponentially distributed with parameter $\mu_i$. We notice that the same assumption has been widely used to keep the tractability of analytical models [20][21]. The ith CH, $i = 1, 2, ..., M$, has two buffers: one with size of $K_i$ (including the one in service) for HP traffic and the other with size of $L_i$ (including the one in service) for LP traffic. The service rate at the ith CH is $\mu_i$ for HP traffic and $(1-\beta)$ $\mu_i$ for LP traffic. Packets in each node are served in a FCFS (first come first served) manner. When a node is processing data, it may fail due to insufficient power. The time to failure is assumed to be exponentially distributed with parameter $\beta_i$. When a node fails, a recovery process starts by selecting another node and the remaining packets in the old node are switched to the new one. Assume the recovery process of the ith node is exponentially distributed with parameter $r_i$. Next, we analyze the performance for HP traffic and the same procedure can be followed for LP traffic.

Let $\lambda_i$ and $\bar{\lambda}_i$ be the overall arrival rate and the effective arrival rate of HP traffic to the ith node respectively. By the conservation of flow[22], the effective arrival rates of all nodes are the same in steady state, i.e., the source-to-sink throughput (SST) $\bar{\lambda}$. Due to the impact of both blocking and failure events, the effective service times are not exponentially distributed. Except for the last node (node M), the service mechanisms of the other (M-1) nodes need to be modified by properly adding exponential phases (with possible failures) to the existing exponential node. The resulting service time distribution is of phase-type (PH), which represents the original service a packet receives plus any possible delays it might undergo due to blocking and failure. Due to the nature of blocking, the server of node (i-1) acts as an additional space for node i during the period that it is blocked by node i, and the queue capacity of node i ($i \geq 2$) has to be augmented by 1. The capacity of the first node does not change, since it does not block any node. For the first node, a packet enters if the node is not full upon its arrival; otherwise, the packet is lost.

In the following, we proceed to modify the service mechanism and arrival process of each node in the tandem configuration from the last node. Therefore, node i, $i = 2, 3, ..., M$, can be modeled as an M/PH/1/K_i+1 queue except the first node, which is modeled as an M/PH/1/K_1 queue. The PH distribution $(a_i, T_i)$ shown in Fig. 1 reflects the effective service time a packet receives at the ith node (the dotted circle represents the node under failure condition), where

$$a_i = [1, 0, 0, 0]$$

and

$$T_i = \begin{bmatrix}
-\beta_i \mu_i - f_i & f_i & 0 & 0 \\
0 & 0 & -\beta_i \mu_i - f_i & f_i \\
0 & 0 & r_i & -r_i \\
\end{bmatrix}. \quad (1)$$

![Figure 1. The ith node in isolation](image)

It is noteworthy that the Mth node in isolation has just the second part of Fig. 1 since it is the last node. The PH distribution $(a_M, T_M)$ is represented as

$$a_M = [1, 0]$$

and

$$T_M = \begin{bmatrix}
-\beta_i \mu_M - f_M & f_M \\
0 & r_M \\
\end{bmatrix}. \quad (2)$$

The branching probabilities $a_{i,0}, a_{i,1}^0$ and $a_{i,1}^1$ can be calculated approximately using information of the (i+1)st node (Note that the analysis method is from the last node to the first node). Let $\pi_i$ be the steady-state conditional probability that upon a service completion at the ith node, the (i+1)st node is full. Let $w_{i,i+1,j}$ (respectively, $w_{i,i+1,j}^1$), $j = i+1, i+2$, be the steady-state probability that the (i+1)st node is at the jth phase of its service time with operation condition (respectively, failure condition), given that it is full. Then, we have

$$a_{i,1}^0 = \pi_i w_{i,i+1}^0 \quad \text{and} \quad a_{i,1}^1 = \pi_i w_{i,i+1}^1, \quad (2)$$

where $w_{i,i+1}^0 = \sum_j w_{i,i+1,j}^0$ and $w_{i,i+1}^1 = \sum_j w_{i,i+1,j}^1$, $j = i+1, i+2$, can be thought of as the equivalent probabilities of the (i+1)st node with operation and failure condition respectively, given that it is full.

Let $p_i(n,j,k)$ be the steady-state probability of the queue that there are n ($n \geq 1$) packets in the ith node, the node is at the jth ($j = i, i+1$) phase of service, and k is the indicator function which is equal to 0 if the node operates and 1 if it fails. The steady-state probability vector is

$$P_i = (p_i(0), p_i(1), \cdots, p_i(K_i+1)), \quad (3)$$

Note here the conservation of flow is valid, since the packets are only from the sensing cluster and not lost from a CH during its failure time (they are switched to the new CH).
where \( p_i(n) = p_i(n, j, 0), p_i(n, j, l), p_i(n, j+1, 0), p_i(n, j+1, l) \),
\( 1 \leq n \leq K_i + 1 \), and \( p_i(0) \) is the probability that the \( i \)-th node is empty. The overall arrival rate to node \( i \) is
\[
\lambda_i = \frac{\lambda}{1 - p_i(K_i + 1)e},
\] \( i=1,2,...,M \) (note that the external arrival rate is determined iteratively by the above fixed-point equation. That is, assuming an initial value for \( \lambda_i \), the queue-length distribution for the \( i \)-th node can be obtained, from which a new value for \( \lambda_i \) can be calculated. This procedure is iterated until the successive values of \( \lambda_i \) converge. Note that in the above procedure, \( \lambda \) will be determined through another iterative procedure after the analysis of all \( M \) nodes is finished. The specific algorithm is summarized in Fig. 2.

**Figure 2.** The iterative algorithm for parameter evaluation

By using Neuts’ matrix-geometric procedure \([17]\), the queue-length distribution is obtained:
\[
p_i(n) = p_i(0)a_i(R_i)^n, \quad 1 \leq n \leq K_i,
\]
\[
p_i(K_i + 1) = p_i(0)a_i(R_i)^{K_i}[-\lambda_iT_i^{-1}].
\]

where \( R_i = \lambda_i[I - \lambda_i e a_i - T_i]^{-1} \), and \( I \) is an identity matrix with the same size as \( T_i \). The probability that the queue is empty is

\[
p_i(0) = \left\{ a_i \left( \sum_{n=0}^{K_i} (R_i)^n - \lambda_i (R_i)^{K_i} T_i^{-1} \right) \right\}^{-1}.
\]

After analyzing the \( i \)-th node, we can obtain the quantities \( \pi_{i+1} \) and \( w_{i,j} \) (\( j = i, i+1 \)), which will be used for analyzing the \((i-1)\)-st node. From basic probability theory, \( w_{i,j}^0 \) and \( w_{i,j}^1 \) are obtained by
\[
w_{i,j}^0 = \frac{p_i(0)a_i(R_i)^{K_i} e_i^1}{p_i(K_i)e_i}, \quad w_{i,j}^1 = \frac{p_i(0)a_i(R_i)^{K_i} e_i^0}{p_i(K_i)e_i},
\]
where \( p_i(K_i)e_i \) is the probability that the \( i \)-th node contains \( K_i \) packets, and \( e_i^0 \) (respectively, \( e_i^1 \) \( j = i, i+1 \)) is a column vector with all elements equal to zero except the element corresponding to state \((K_i, j, 0)\) (respectively, state \((K_i, j, 1)\)) which is equal to 1. By applying Little’s Law to the \((K_i+1)\)-st position of the \( i \)-th node, we can determine \( \pi_{i+1} \) by
\[
(\lambda \pi_{i+1})_i = -w_i T_i^{-1} e_i = p_i(K_i + 1)e_i,
\]
where \( w_i = (w_{i,j}^0, w_{i,j}^1, w_{i-1,j}^0, w_{i-1,j}^1, \ldots, w_{1,j}^0) \) (Note that for the \( M \)-th node, \( w_M = (w_{M,M}^0, w_{M,M}^1) \)), and \( p_i(K_i + 1) \) has been obtained from the analysis of the \( i \)-th node. The quantity in the brackets is the expected remaining service time at the \( i \)-th node at the moment the \((i-1)\)-st node gets blocked. Therefore, we have
\[
a_{i,j}^0 = \pi_{i+1} w_{i,j}^0 \text{ and } a_{i,j}^1 = \pi_{i+1} w_{i,j}^1,
\]
where \( w_j = \sum_{j} w_{i,j}^0 \) and \( w_j = \sum_{j} w_{i,j}^1 \), \( j = i, i+1 \). Now the \((i-1)\)-st node is ready to be analyzed. The analysis procedure proceeds until the first node.

It is worthy to note that, after the above iterative procedures stop, the actual probability that there are \( K_i \) packets in node \( i \), \( i=2,3,...,M \), is obtained as the sum of \( p_i(K_i)e_i \) and \( p_i(K_i + 1)e_i \), since the input buffer remains full while blocking the \((i-1)\)-st node. Therefore, the mean number of packets in the tandem queueing network, \( E[L] \), is determined as
\[
E[L] = \sum_{i=1}^{M} \left( \sum_{n=0}^{K_i} n p_i(n) e_i + K_i p_i(K_i + 1)e_i \right).
\]

The steady-state SST of the tandem network, \( \bar{\lambda} \), can be determined by the algorithm provided in Fig. 2. By Little’s law, the mean sojourn time, i.e., the source-to-sink delay (SSD) for a packet in the tandem queueing system is obtained by
\[
SSD = \frac{E[L]}{\bar{\lambda}} = \frac{1}{\bar{\lambda}} \sum_{i=1}^{M} \left( \sum_{n=0}^{K_i} n p_i(n) e_i + K_i p_i(K_i + 1)e_i \right).
\]

To study the impact of \( \beta \) on the SSD performance, we present some numerical computation. The multi-hop WSN is configured with \( M = 4 \) intermediate CHs between a source-destination path. The CHs are set to have different capacities, different service rates, different failure rates and different recovery rates to show the flexibility of the
proposed model. We randomly choose \([\mu_1, \mu_2, \mu_3, \mu_4] = [15, 16, 15, 14], [K_1, K_2, K_3, K_4] = [30, 28, 29, 28], [T_1, T_2, T_3, T_4] = [0.05, 0.08, 0.09, 0.06], [\beta_1, \beta_2, \beta_3, \beta_4] = [0.001, 0.0015, 0.002, 0.001] \) \( \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta = 0.5, 0.6, 0.7, \) respectively. The HP traffic arrival rate from the source \( \lambda_{HP} = \lambda_1 \) is set to be changed from 1 to 13. In Fig. 3, we observe that the SSD decreases as \( \beta \) is increased. This is reasonable as larger \( \beta \) leads to larger processing rate for HP traffic, and thus leads to a decreased SSD.

![Figure 3. The impact of \( \beta \) on the SSD of HP traffic](image)

### 4. QoS Control

The traffic load in sensor networks often shows dynamic characteristics due to the applications of image/video sensors and/or the different forms of data-reporting (event-triggered, periodical, etc.). A proper queue management scheme should be employed in WSNs to adapt to the dynamic load and to maintain the QoS requirements. As an example, we choose the metric SSD obtained in Section 3 as the QoS parameter to study QoS control. The goal is to control the SSD of HP traffic within a predefined range through adjusting the value of \( \beta \).

Therefore, when the SSD goes above a threshold (high threshold, \( T_1 \)), a request for increasing \( \beta \) is generated. Since the total processing rate of each CH is constant, keeping large \( \beta \) for low-density HP traffic is a waste of resource, too. When the SSD drops below another threshold (low threshold, \( T_2 \)), a request for decreasing \( \beta \) will be generated. In this way, the assigned service rate for LP traffic is automatically increased under low-density HP traffic condition and the performance of LP traffic is thus improved. In order to protect both classes of traffic, the \( \beta \) for each CH is limited between a maximum value, \( \beta_{max} \), and a minimum value, \( \beta_{min} \), respectively, \( 0 \leq \beta_{min} \leq \beta \leq \beta_{max} \leq 1 \). That is, the value of \( \beta \) should not be greater than \( \beta_{max} \) when it is increased, and not be less than \( \beta_{min} \) when it is decreased.

A simple QoS control scheme is formed from the idea: if the SSD of HP traffic is larger than the high threshold \( T_1 \), the CHs in the path will be instructed (by the sink) to increase their respective \( \beta \) values by a predefined value \( \delta_1 \) to release some resource (for LP traffic use). Through setting different amounts of \( \delta_1 \) and \( \delta_2 \), the scheme can be made to adapt to various situations, such as fast-increase and fast-decrease, fast-increase and slow-decrease, slow-increase and fast-decrease, and slow-increase and slow-decrease. The QoS control algorithm is summarized in Fig. 4.

![The Simple QoS Control Algorithm](image)

**The Simple QoS Control Algorithm**

\[
\text{Begin}
\]
\[
\text{Predefine } \delta_1 \text{ and } \delta_2; \text{ Computing the SSD in the sink through the procedure developed in Section III;}
\]
\[
\text{while } (\text{SSD} > T_1) \quad \text{Send command of increasing } \beta \text{ to each CH in the path, i.e., } \beta = \min{\beta_{max}, \beta + \delta_1}; \quad \text{Computing the SSD in the sink;}
\]
\[
\text{end}
\]
\[
\text{while } (\text{SSD} < T_2) \quad \text{Send command of decreasing } \beta \text{ to each CH in the path, i.e., } \beta = \max{\beta_{min}, \beta - \delta_2}; \quad \text{Computing the SSD in the sink;}
\]
\[
\text{end}
\]
\[
\text{while } (T_2 \leq \text{SSD} \leq T_1) \quad \text{No adjustment;}
\]
\[
\text{end}
\]

This scheme is simple but easy to implement. It is worthy to note that the above algorithm is executed in the command center (sink). The CHs only need to receive the commands from the sink and then perform relevant operations (increase or decrease).

Next, we perform simulations to validate the effectiveness of the algorithm. In order to see how the QoS control algorithm behaves under different traffic conditions, we perform the simulations under light and heavy traffic load respectively.

The simulations are performed under the same configuration as that in Section 3 and consider the two extreme traffic load conditions, i.e., \( \lambda_{HP} = 1 \) and \( \lambda_{HP} = 13 \). The QoS requirements are set as \( T_1 = 0.9 \) and \( T_2 = 0.8 \). The maximum and minimum values of \( \beta \) are set as: \( \beta_{max} = 0.9 \) and \( \beta_{min} = 0.1 \). The predefined delta values are given as 0.1, 0.05, 0.02 and 0.01 respectively. If SSD > \( T_1 \) or SSD < \( T_2 \), the associated algorithm will be triggered and the value of \( \beta \) will be adjusted up or down, respectively. Continuous adjustments can be performed automatically until the SSD enters the predefined QoS range. The performance is studied based on the number of adjustments.

Fig. 5 shows how the algorithm performs at heavy traffic load (\( \lambda_{HP} = 13 \)). In the beginning, the SSD of HP traffic is set to a large value, say 2.12, which is much larger than the high threshold (\( T_1 = 0.9 \)). The simulation is started by the proposed algorithm with different deltas. We observe that for \( \delta = 0.01 \), 15 adjustments are required for the SSD to enter the QoS range; for \( \delta = 0.02 \), 8 adjustments are required; for \( \delta = 0.05 \), 3 adjustments is required; and \( \delta = 0.1 \) does not work at this condition, since the SSD oscillates between the two points outside the QoS range but never enters that range. Fig. 6 shows the results at light traffic condition (\( \lambda_{HP} = 1 \)). In the beginning, the SSD of HP traffic is set to a small value, say 0.37, which is much lower than the low threshold.
After the simulation starts, we can observe that for $\delta = 0.01$ and $0.02$, 26 and 13 adjustments are required respectively for the SSD to enter the QoS range; and $\delta = 0.05$ and $0.1$ do not work due to oscillations. Thus, by appropriate selection of deltas, the proposed scheme can achieve expected goal.

**Figure 5.** The algorithm simulation process at heavy load

**Figure 6.** The algorithm simulation process at light load

5. **Conclusions**

We developed an analytic model for traffic flow analysis in a multi-hop wireless sensor network subject to node failure through a queueing theoretic framework. The dynamics of traffic flow in the source-to-sink path was modeled by a number of single-server queues with finite buffers linked in tandem and analyzed by decomposing the tandem queueing network into individual nodes with modified arrival and service processes as well as queue capacities. In the individual node modeling, both the blocking effect of tandem queues and the impact of node failure were taken into account. An iterative algorithm was developed for evaluating the performance metrics. A simple QoS control scheme was developed and its effectiveness was validated by simulations.

**References**


