

## Article

# Machine-Learning Based Channel Quality and Stability Estimation for Stream-Based Multichannel Wireless Sensor Networks

Waqas Rehan <sup>1,\*</sup>, Stefan Fischer <sup>1</sup> and Maaz Rehan <sup>2</sup><sup>1</sup> Institute of Telematics (ITM), University of Luebeck, 23562 Luebeck, Germany; fischer@itm.uni-luebeck.de<sup>2</sup> COMSATS Institute of Information Technology (CIIT), Quaid Avenue, 47040 Wah Cantt, Pakistan; maazrehan@ciitwah.edu.pk

\* Correspondence: rehan@itm.uni-luebeck.de; Tel.: +49-451-3101-6413

Academic Editor: Ana M. Bernardos

Received: 26 April 2016; Accepted: 8 September 2016; Published: 12 September 2016

**Abstract:** Wireless sensor networks (WSNs) have become more and more diversified and are today able to also support high data rate applications, such as multimedia. In this case, per-packet channel handshaking/switching may result in inducing additional overheads, such as energy consumption, delays and, therefore, data loss. One of the solutions is to perform stream-based channel allocation where channel handshaking is performed once before transmitting the whole data stream. Deciding stream-based channel allocation is more critical in case of multichannel WSNs where channels of different quality/stability are available and the wish for high performance requires sensor nodes to switch to the best among the available channels. In this work, we will focus on devising mechanisms that perform channel quality/stability estimation in order to improve the accommodation of stream-based communication in multichannel wireless sensor networks. For performing channel quality assessment, we have formulated a composite metric, which we call channel rank measurement (CRM), that can demarcate channels into good, intermediate and bad quality on the basis of the standard deviation of the received signal strength indicator (RSSI) and the average of the link quality indicator (LQI) of the received packets. CRM is then used to generate a data set for training a supervised machine learning-based algorithm (which we call Normal Equation based Channel quality prediction (NEC) algorithm) in such a way that it may perform instantaneous channel rank estimation of any channel. Subsequently, two robust extensions of the NEC algorithm are proposed (which we call Normal Equation based Weighted Moving Average Channel quality prediction (NEWMAC) algorithm and Normal Equation based Aggregate Maturity Criteria with Beta Tracking based Channel weight prediction (NEAMCBTC) algorithm), that can perform channel quality estimation on the basis of both current and past values of channel rank estimation. In the end, simulations are made using MATLAB, and the results show that the Extended version of NEAMCBTC algorithm (Ext-NEAMCBTC) outperforms the compared techniques in terms of channel quality and stability assessment. It also minimizes channel switching overheads (in terms of switching delays and energy consumption) for accommodating stream-based communication in multichannel WSNs.

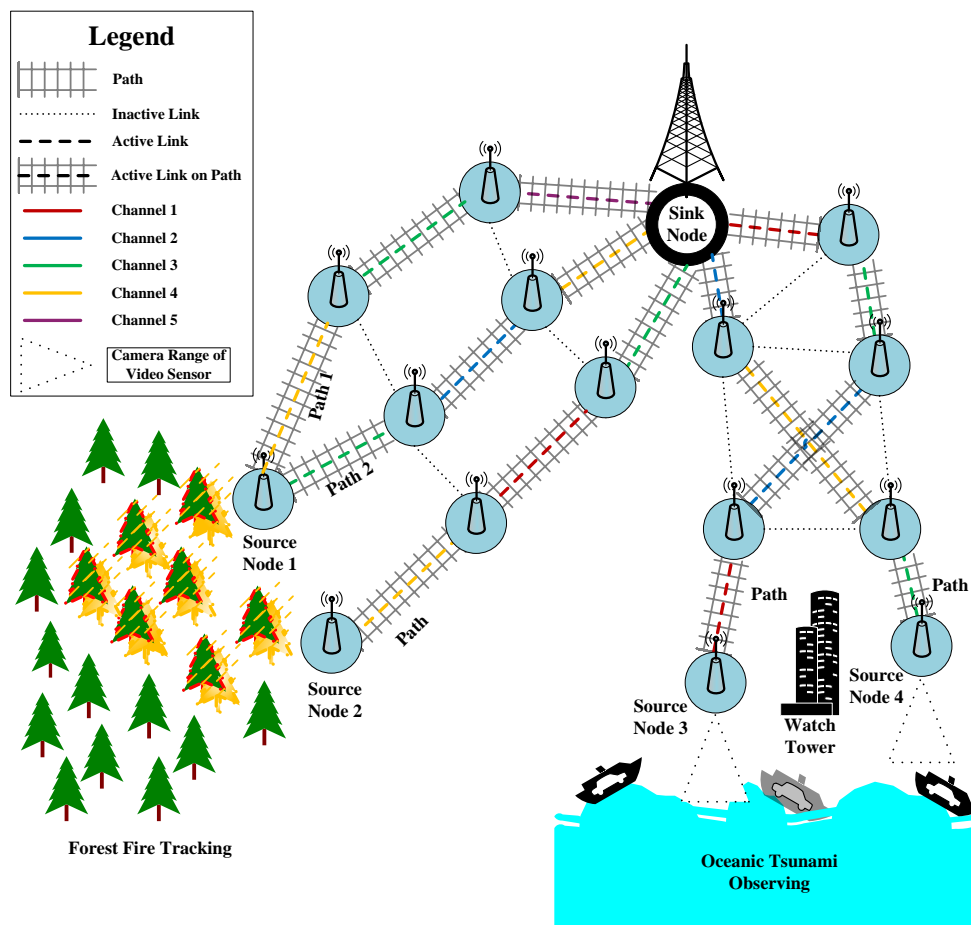
**Keywords:** multichannel; multi-radio; channel quality prediction; wireless sensor networks; machine learning

## 1. Introduction

Wireless sensor networks (WSNs) consist of tiny devices, which have limited energy, memory, sensing/processing unit and transmission capability [1]. Such networks often also have the capacity of self-organization in ad hoc mode [2,3]. Traditionally, energy efficiency has always been a key issue in WSNs, whereas throughput, fairness and delay were not so much the focus of research [4].

However, technological advancement has shifted this focus and enabled sensor nodes to transmit large volumes of scalar environmental data, e.g., concerning temperature, humidity or pressure. Meanwhile, camera and microphone-enabled sensor nodes have emerged; networks build from such nodes are called wireless multimedia sensor networks (WMSNs) [5,6] and have the ability to transmit very large volumes of delay-sensitive multimedia data to sink nodes. Additionally, a variety of multipath and multichannel approaches have been proposed by researchers whose purpose is to further improve the performance of WSNs.

Figure 1 shows a sample scenario using multiple paths and channels. Source Node 1 and the sink node are connected by two paths, whereas the remaining sources can communicate with the sink using only a single path. Any two neighbor nodes on a path have an active link between them, which resides on the best among all available channels.



**Figure 1.** Surveillance using a multichannel/multipath WSN.

Unlike sensor nodes in conventional networks, nodes in multichannel sensor networks additionally have to deal with a potentially large number of channels for data routing, i.e., they have to decide which channel to use for transmission; the better the choice the better the overall system performance. Therefore, an appropriate channel quality/stability assessment is needed. Since the behavior of wireless channels is probabilistic in nature (which makes channel quality assessment a recurrent task), any instantaneous channel quality assessment cannot, by nature, provide an adequate measure of the channels' health. In [7], the authors have realized that combining current and past channel quality assessments at the receiver may help to predict the channel quality indicator (CQI), which may guide the transmitter to adapt the transmission parameters accordingly and improve the performance of the wireless communication systems.

A large number of multichannel protocols for WSNs uses single radio per node [8–12]. If transmitting and receiving frequencies of sensor nodes are different, then such sensor networks may suffer from channel switching delays and additional power consumption [12]. When the data rate is very high, frequent channel switching may result in data loss [12], which may adversely affect the system performance. Therefore, for accommodating multimedia traffic, it is more efficient and cost-effective to aim for a stream-based channel assignment rather than to do this on a per-packet base [13,14]. However, channel behavior is not deterministic in nature; therefore, reserving a channel for a whole data stream requires knowledge about average channel response, which may be assessed in advance through channel quality and stability assessment. To the best of our knowledge, there are no multichannel protocols in WSNs that embody any mechanism of channel quality and stability assessment for supporting stream-based channel allocation in multichannel WSNs.

The quality of a channel in a neighborhood can be assessed on the basis of link quality in that vicinity. In [15], a link quality-based channel selection approach is proposed, which ranks channels into different categories and selects good quality channels for improving system performance. Moreover, for measuring link quality, different authors have introduced distinct metrics, such as [16–19]. The author in [20] has realized that good, intermediate and bad links can be discriminated on CC2420 using the standard deviation of RSSI ( $std(RSSI)$ ) and the average of link quality indicator (LQI) ( $avg(LQI)$ ), as shown in Table 1. The advantage of using  $std(RSSI)$  and  $avg(LQI)$  is that they may measure performance and central tendency, respectively, of link quality in a better manner. On the other hand, the average of RSSI ( $avg(RSSI)$ ) and the standard deviation of LQI ( $std(LQI)$ ) are not good estimators of link quality because they may cause overlap of bad with intermediate quality links and intermediate with good quality links, respectively [20]. Since various link quality metrics have their own limitations [21], therefore no consensus has been developed among the research community for the most suitable link quality metric [17]. Consequently, a hybrid metric may ultimately be used for accurately accessing link quality [21].

**Table 1.** Demarcation link types. LQI, link quality indicator.

Link Type	$std(RSSI)$	$avg(LQI)$
Good	<4	>104
Intermediate	4–10	70 to $\leq 104$
Bad	>10	<70

Most of the multichannel protocols in WSNs do not consider any scheme of channel quality assessment before channel assignment. For example, tree Construction and Channel Allocation algorithm (CCA) [22], Iterative Channel Adjustment Data Aggregation Routing algorithm (ICADAR) [23], Lagrangean Relaxation algorithm (LGR) [24,25] do not employ any mechanism to differentiate between good and bad quality channels. On the other hand, there are some multichannel protocols in WSNs that employ different mechanisms for measuring channel quality, such as Efficient Multichannel MAC protocol (EM-MAC) [26], Decentralized Optimization for Multichannel Random Access (DOMRA) [27], Multi-radio Multi-channel Opportunistic Cooperative Routing algorithm (MMOCR) [28], Distributed Routing and Channel Selection scheme (DRCS) [12], Regret Matching based Channel Assignment algorithm (RMCA) [29,30]. However, to the best of our knowledge, there is no multichannel protocol in WSNs that can make channel quality and stability assessment on the basis of both current and past channel quality data.

In this paper, we focus on accomplishing channel quality and stability assessment for accommodating stream-based communication in WSNs. In this context, the main contributions of our work can be summarized as follows:

- Formulating a composite channel rank measurement (CRM) metric that is used to create a dataset for training the normal-equation-based predictor.

- Employing a normal-equation-based supervised machine learning algorithm (NEC algorithm) and training it using a channel quality-based generated dataset in such a way that it performs channel rank estimation ( $CRE_t^i$ ) of any channel  $i$  based on only instantaneous values of  $std(RSSI_t^i)$  and  $avg(LQI_t^i)$  of received packets.
- Extending the NEC algorithm by devising the NEWMAC algorithm, which employs a weighted moving average-based criterion for predicting the final channel rank estimation ( $\phi_{t,NEWMAC}^i$ ) of any channel  $i$  based on both past and current values of channel quality prediction.
- Devising the NEAMCBTC algorithm as an extension of NEC algorithm, which employs an adaptive weighting procedure by considering past and current channel quality predictions for estimating the final channel rank estimation ( $\phi_{t,NEAMCBTC}^i$ ) of any channel  $i$  and also promptly tracking channel quality degradations/upgradations. For more robustness, devising an extended version, entitled as Ext-NEAMCBTC algorithm that may perform both channel quality and stability assessment as a composite task.

The rest of the paper is structured as follows. In Section 2, we discuss the related work already mentioned above in a more detailed manner and draw our conclusions for the motivation of our research. Section 3 explains the underlying system model. Section 4 discusses the channel quality assessment metric. In Section 5, we elaborate the proposed supervised machine learning-based algorithms along with their problem statements. Section 6 presents and discusses the detailed performance evaluations of those algorithms along with their pros and cons. Section 7 sums up the overall presented work and draws conclusions.

## 2. Related Work and Motivation

The objective of this paper is to propose a robust multichannel algorithm that may perform both channel quality and stability assessment to support stream-based communication in WSNs. For this purpose, we have studied a large number of multichannel techniques in WSNs and found a limited number of protocols that embody some channel quality assessment mechanism for achieving high performance.

Tang et al. [26] have proposed the EM-MAC algorithm where the channel quality assessment criterion is maintained through the Clear Channel Assessment (CCA) technique. When a congested/interfered channel is encountered, it is marked as blacklisted and avoided till the end of the current session. In this way, bad quality channels are differentiated from good ones. Similarly, Jingrong et al. [28] have introduced a new channel quality-based metric called channel interference strength (CIS), which is based on a composite metric of power received  $P_r$  and Signal-to-Interference-plus-Noise Ratio (SINR). It allows sensor nodes to select the channel with the smallest CIS for future communication. The CIS-based metric does not consider previous channel quality and focus on instantaneous measures.

Khan et al. [29] have proposed a game theory-based multichannel protocol for WMSNs where a channel is selected by a cluster head for the next round when that channel is not selected by its neighboring clusters in the previous round. Otherwise,  $avg(LQI)$  is used as a metric for channel selection; however, the approach does not outline how it does that, and it does not consider past channel quality in future channel quality assessment. Similarly, Pal et al. [12] have utilized node energy and an expected number of transmissions (ETX)-based channel quality assessment metric for performing future communication. Although this ETX-based metric is reliable, it may require probing packets and, consequently, is costly to carry out. Moreover, it only considers instantaneous channel quality and does not regard past channel quality and stability assessment for measuring the final channel quality estimation. Likewise, Yu et al. [30] have presented a multichannel protocol where each node maintains a utility function and a past information-based performance matrix that helps to predict future network topology/flows and actions of neighbor nodes. Subsequently, channels are assigned accordingly. The protocol makes future channel assignment on the basis of predictions based on past knowledge only and does not consider current channel quality assessment.

Finally, to the best of our knowledge, we can conclude the related work as outlined in the Table 2, where the summarized results clearly show that the multichannel protocols in WSNs are either considering as instantaneous channel quality assessment or selecting the final channel on the basis of some past knowledge. Therefore, we can conclude that no multichannel protocol is available in WSNs that considers both current and past channel quality estimation to predict final channel rank estimation in WSNs.

In addition to the above discussion, we have noted earlier that  $std(RSSI)$  and  $avg(LQI)$  of received packets may discriminate links into different categories [20], as shown in detail in Table 1. Since each link in the neighborhood of a node may use a specific frequency channel for transmission as shown in Figure 1, the quality of these links in the neighborhood of a node may determine the overall quality of a channel in this neighborhood. Consequently we have formulated the CRM metric that is used to train a normal equation-based predictor for executing channel rank estimation ( $CRE_t^i$ ) of any channel  $i$  at instant  $t$  using  $std(RSSI_t^i)$  and  $avg(LQI_t^i)$  of received packets.

Our second observation from the literature review is that there is no multichannel protocol in WSNs that employs channel quality and stability assessment, using present and past channel knowledge to accommodate stream-based applications. Two of the three algorithms we present in this paper, namely NEWMAC and NEAMCBTC, are closing this gap. We also believe that this is the first work that employs a normal equation-based supervised machine learning algorithm for channel quality approximation in multichannel WSNs.

**Table 2.** Summary of related protocols reviewed. ETX, expected number of transmissions.

Protocol	Field	Current Knowledge	Past Knowledge
RMCA [30]	Multichannel	—	Regret matching based
EM-MAC [26]	Multichannel	Interference based	—
DRCS [12]	Multichannel routing	Battery power and ETX based	—
[29]	Multichannel routing	—	Game-theory based
MMOCR [28]	Multichannel Routing	RSSI and SINR based	—

### 3. System Model

We model a WMSN as a directed graph  $G(S, E)$  where the set of vertices  $V$  represent  $N$  multimedia-enabled sensor nodes, i.e.,  $V = \{n_i | i = 1, 2, 3, \dots, N\}$ . The sensor nodes are randomly distributed in the sensing field and may be static or dynamic in nature. There is a bidirectional edge  $e \in E$  between any two neighboring vertices  $n_i$  and  $n_j$ , which may enable them to perform channel negotiation with each other.

The physical layer model allows each sensor node to compute  $std(RSSI_t^i)$  and  $avg(LQI_t^i)$  of received packets on any channel  $i$  at instant  $t$ . Afterwards, machine learning-based technology is employed to estimate the quality of the corresponding channel.

The MAC layer model allows each sensor node to sense available channels in the neighborhood and perform channel negotiation with the preferred neighbor node on a path in a manner that the available channel of highest quality is negotiated first, then the one with the second-highest quality, and so on. Once channel handshaking has been performed and the best channel is agreed on, both sender and receiver jump to the desired channel for performing stream-based communication and stay there till the data stream ends. All channels are of equal bandwidth and orthogonal in nature. Additionally, it is assumed that all channels are not jammed or degraded simultaneously, and therefore, some channels of good quality are always available for performing stream-based communication.

For simplicity, we assume that the quality of all links on a particular frequency channel in a neighborhood is the same and may reflect channel quality in the corresponding locality. Otherwise, each sensor node may have to record the quality of all available channels for each link in a neighborhood separately, which may increase system complexity accordingly.

#### 4. Channel Rank Measurement

We noted earlier that  $std(RSSI)$  and  $avg(LQI)$  of received packets are useful parameters to describe the link quality [20] and, thus, also the channel quality. In this section, we will discuss a mathematical formulation of our channel rank measurement  $CRM$  metric, which is used to create the dataset for training our normal-equation-based channel quality predictor.

On the basis of Table 1, it is clear that the values of  $std(RSSI)$  and  $avg(LQI)$  have different ranges and spreads; therefore, for getting the benefits of both worlds, we first have to bring these channel quality metrics into a common scale for calculating the channel quality measurement ( $CQM$ ) metric. In this way, the final impact of  $std(RSSI)$  and  $avg(LQI)$  is approximately equalized, and therefore, a clear boundary can be drawn between good, intermediate and bad quality links. The  $CQM$  metric is calculated as:

$$CQM = [scale(LQI) + scale(RSSI)] \quad (1)$$

where

$$scale(LQI) = \left[ \frac{avg(LQI) - min(LQI)}{\gamma} \right] \quad (2)$$

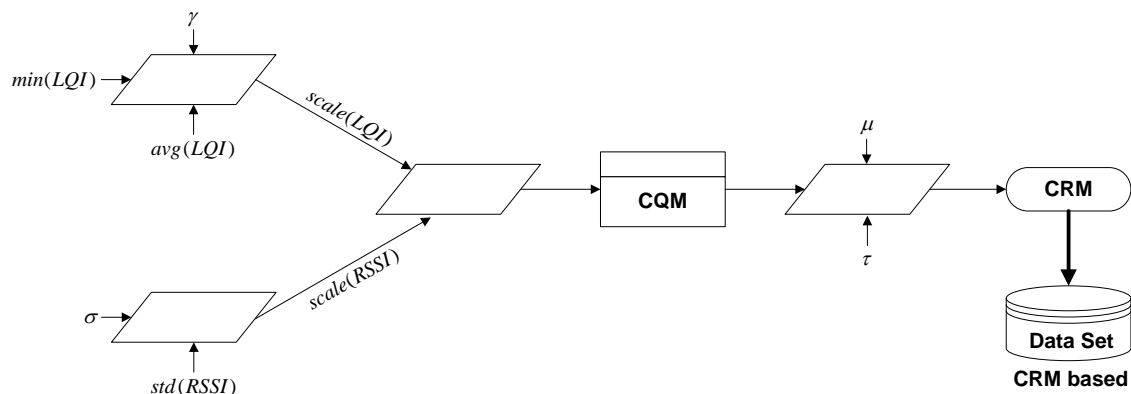
$$scale(RSSI) = [\sigma - std(RSSI)] \quad (3)$$

where, following [20,31,32],  $min(LQI) = 50$ . We choose the values of the scaling parameters  $\sigma = 15$  and  $\gamma = 4.0$  in order to bring  $std(RSSI)$  and  $avg(LQI)$  into a common scale.

Our  $CRM$ , then, is given by:

$$CRM = \left[ \frac{CQM \times \tau}{\mu} \right] \quad (4)$$

where  $\tau = 3.5$  and  $\mu = 100$  are adjustment parameters that constrain the values of the  $CRM$ -based channel quality training dataset in the range  $[0, 1]$  as shown in Table 3. Thus, channel rank estimation ( $CRE$ ) will fall into the unit interval and, subsequently, channel manipulation of all quality levels is in the range  $[0, 1]$ . The detailed representation of  $CRM$  based metric is shown in Figure 2, while channel classification into different categories is explained in Table 3. In the next section, we will explain how the  $CRM$ -based dataset may be used to perform  $CRE$ .



**Figure 2.** Diagrammatic representation of channel rank measurement ( $CRM$ ) metric-based training dataset generation.



**Table 3.** Channel rank measurement metric.

Channel Type	$std(RSSI)$	$avg(LQI)$	$scale(RSSI)$	$scale(LQI)$	CRM
Good	<4	>104	11< to ≤15	13.52 to 15	0.82≤ to ≤1.00
Intermediate	4–10	70≤ to ≤104	5 to 11	5 to 13.50	0.33≤ to <0.82
Bad	>10	<70	0≤ to <5	0 to 4.97	0≤ to <0.33

## 5. Supervised Machine Learning-Based Prediction Algorithms

Since each sensor node has limited energy, it is important to employ cost-effective algorithms in WSNs. For this purpose, a dataset is generated on the basis of calculations made in Section 4 and is used to train our basic normal-equation-based machine learning algorithm called normal equation-based channel quality prediction (NEC). Afterwards, we propose two more sophisticated extensions of the basic NEC algorithm, namely normal equation-based weighted moving average channel quality prediction algorithm (NEWMAC) and normal equation-based aggregate maturity criteria with beta tracking-based channel weight prediction algorithm (NEAMCBTC), which consider both instantaneous and past values of channel quality for making final channel quality prediction. We will discuss the mathematical background of normal equation-based prediction in Section 5.1 and the proposed algorithms in Sections 5.2–5.4, and we also elaborate on their pros and cons.

### 5.1. Normal Equation-Based Prediction

The normal equation-based prediction is more feasible and cost effective when the number of features is small. Since, in this work, we are considering two features i.e.,  $std(RSSI_t^i)$  and  $avg(LQI_t^i)$ , it is more cost effective to use normal equation-based channel quality prediction rather than employing a gradient descent algorithm for making channel quality assessment.

Consider an over-determined system where  $m$  is the number of training examples (corresponding to  $m$  linear equations) and  $n$  is the number of features (corresponding to  $n$  unknown coefficients, i.e.,  $\theta_1, \theta_2, \theta_3, \dots, \theta_n$ ) with  $m > n$ , then the system can be expressed as [33]:

$$\sum_{j=1}^n X_{ij}\theta_j = y_i, \quad \text{where } i = 1, 2, 3, \dots, m \quad (5)$$

In the matrix form [33], we can write:

$$\mathbf{X}\boldsymbol{\theta} = \mathbf{y} \quad (6)$$

where  $\mathbf{X}$  is the feature matrix,  $\boldsymbol{\theta}$  is the learning coefficients vector and  $\mathbf{y}$  is the output vector given by:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & x_{m3} & \dots & x_{mn} \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Now, by solving the quadratic minimization problem,  $\theta$  is given by [33]:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \mathcal{O}(\theta) \quad (7)$$

Here,  $\mathcal{O}$  is the objective function and is given by [33]:

$$\mathcal{O}(\theta) = \sum_{i=1}^m |y_i - \sum_{j=1}^n X_{ij}\theta_j|^2 = \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2 \quad (8)$$

$$= (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = \mathbf{y}^T \mathbf{y} - \boldsymbol{\theta}^T \mathbf{X}^T \mathbf{y} - \mathbf{y}^T \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T \mathbf{X}^T \mathbf{X}\boldsymbol{\theta} \quad (9)$$

$$= \mathbf{y}^T \mathbf{y} - 2\boldsymbol{\theta}^T \mathbf{X}^T \mathbf{y} + \boldsymbol{\theta}^T \mathbf{X}^T \mathbf{X} \boldsymbol{\theta}, \quad (\text{as } \boldsymbol{\theta}^T \mathbf{X}^T \mathbf{y} = \mathbf{y}^T \mathbf{X} \boldsymbol{\theta}) \quad (10)$$

Taking the derivative of the above equation with respect to  $\boldsymbol{\theta}$  and equating it to zero, we get the normal equation [33] as shown below:

$$\boldsymbol{\theta} = (\mathbf{X}^T \mathbf{X})^{-1} + \mathbf{X}^T \mathbf{y} \quad (11)$$

The above equation clearly indicates that  $\mathbf{X}^T \mathbf{X}$  is an  $[n \times n]$  matrix. Therefore, the cost of inverting this matrix is  $O(n^3)$ . However, it is affordable in our case, because the number of features are only two, i.e.,  $std(RSSI)$  and  $avg(LQI)$ . The training of the system with the dataset is carried out only once, at the start of the system execution phase, and the learning coefficients vector  $\boldsymbol{\theta}$  is calculated using Equation (11). Since we are dealing with only two features, therefore  $\boldsymbol{\theta}$  and  $\mathbf{y}$  would be  $[(2 + 1) \times 1]$  and  $[m \times 1]$  vectors, respectively. Moreover  $\mathbf{X}$  and  $\mathbf{X}^T$  would be  $[m \times (2 + 1)]$  and  $[(2 + 1) \times m]$  matrices, respectively as given below:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_{m1} & x_{m2} \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

## 5.2. Normal Equation-Based Channel Quality Prediction Algorithm

The NEC algorithm can perform channel rank estimation  $CRE_t^i$  of any channel  $i$  on the basis of current values of  $std(RSSI_t^i)$  and  $avg(LQI_t^i)$  of received packets on the corresponding channel, as shown in Figure 3. Since it employs two input features, it is more efficient to use the normal equation-based machine learning algorithm for solving  $\boldsymbol{\theta}$ . Henceforth, the hypothesis function for estimating  $CRE_t^i$  is as follows:

$$h_{\boldsymbol{\theta}}(X) = \theta_0 X_0 + \theta_1 X_1 + \theta_2 X_2 \quad (12)$$

where  $h_{\boldsymbol{\theta}}(X) = CRE_{t,NEC}^i$ ,  $X_0 = 1$ ,  $X_1 = std(RSSI_t^i)$  and  $X_2 = avg(LQI_t^i)$ . Moreover, the leaning coefficients  $\theta_0, \theta_1$  and  $\theta_2$  are measured on the basis of available dataset.

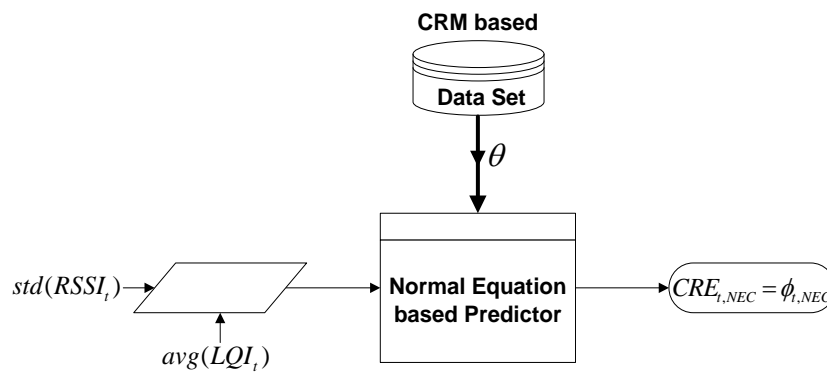


Figure 3. Data flow diagram of the NEC algorithm.

### 5.2.1. Problem Definition

Let  $\phi_{t,NEC}^i$  be the measure of channel quality, i.e.,  $(CRE)_{t,NEC}^i$  of any channel  $i$  at the instant  $t$ . Let  $C$  denote a set of all channels in the neighborhood and  $Z$  denote a set of all channels, except  $i$ , i.e.,  $C = Z + i$ . Then, the objective function  $F$  is to select a channel  $i$  at instant  $t$  that exhibits the maximum quality:

$$\text{Maximize:} \quad F(\phi_{t,NEC}^i), \quad i = 1, 2, 3 \dots C \quad (13)$$



$$\text{Subject to: } (CRE)_{t,NEC}^i \geq (CRE)_{t,NEC}^Z, \quad C = \{i + Z/i \notin Z\} \quad (14)$$

where:

$$0 \leq (\phi_{t,NEC}^i, (CRE)_{t,NEC}^i) \leq 1.0$$

The above constraint (Equation (14)) elaborates that if any channel  $i$  has the highest channel rank estimation  $(CRE)_{t,NEC}^i$  at instant  $t$ , then it would be more suitable to accommodate stream-based data communication. Since the NEC algorithm accesses multichannel quality on the basis of instantaneous channel knowledge only; therefore, it may not be suitable to address stream-based communication that requires average channel knowledge at a particular epoch for accommodating the whole data stream afterwards.

### 5.3. Normal Equation-Based Weighted Moving Average Channel Quality Prediction Algorithm

The NEWMAC algorithm employs a simple weight moving average-based criterion where the final channel rank estimation is calculated by assigning equal weights to current and past channel quality predictions. Here, current channel quality is predicted by employing the same mechanism as used in the NEC algorithm, whereas the past channel quality is based on the weighted moving average outcome in the previous iteration, as shown in Figure 4. Since the final channel quality assessment is based on past and current channel quality predictions, the NEWMAC algorithm has the ability to accommodate stream-based communication.

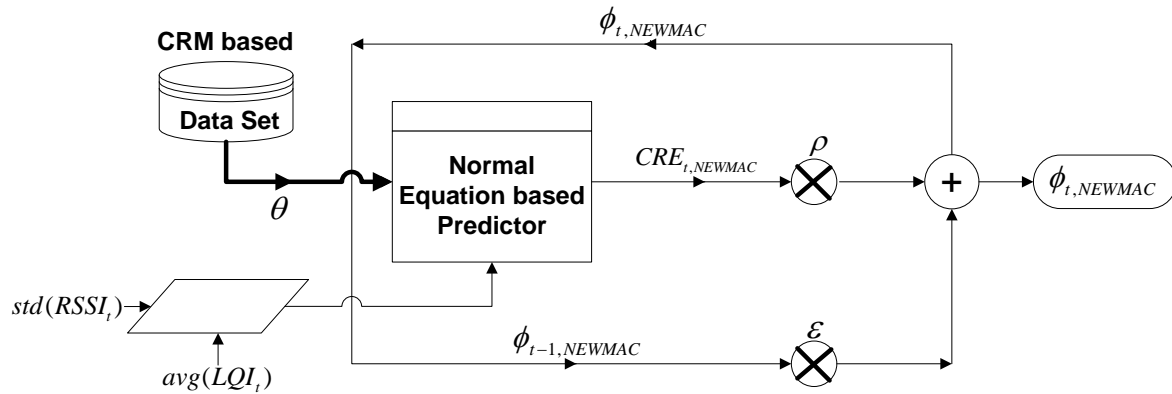


Figure 4. Data flow diagram of NEWMAC algorithm.

#### 5.3.1. Weight Moving Average-Based Channel Quality Prediction Mechanism

Let  $\phi_{t-1,NEWMAC}^i$  denote the past channel quality prediction and  $(CRE)_{t,NEWMAC}^i$  denote the current channel rank estimation. Let  $\epsilon$  and  $\rho$  denote the weights of past and current channel quality predictions, respectively, which have been assigned equal value in this calculation. Then, the moving average-based final channel quality prediction is given by:

$$\phi_{t,NEWMAC}^i = \epsilon \times \phi_{t-1,NEWMAC}^i + \rho \times (CRE)_{t,NEWMAC}^i \quad (15)$$

with

$$\epsilon + \rho = 1 \quad (16)$$

#### 5.3.2. Problem Definition

Let  $\phi_{t,NEWMAC}^i$  denote the final quality of any channel  $i$  at time  $t$ , being measured by taking the moving average of NEWMAC-based past channel quality prediction and normal equation-based current channel rank estimation. Let  $C$ ,  $Z$  and  $i$  be defined as above, again with  $C = Z + i$ . Then, the objective function  $F$  is to select a channel  $i$  at instant  $t$  that exhibits the maximum quality:

$$\text{Maximize: } F(\phi_{t,NEWMAC}^i), \quad i = 1, 2, 3 \dots C \quad (17)$$

$$\text{Subject to: } \phi_{t,NEWMAC}^i \geq \phi_{t,NEWMAC}^Z, \quad C = \{i + Z/i \notin Z\} \quad (18)$$

where,

$$0 \leq (\phi_{t,NEWMAC}^i) \leq 1.0$$

The above constraint (Equation (18)) says that the sensor node would select such a channel  $i$  for stream-based data communication at a particular epoch that exhibits the maximum quality. While the moving average-based NEWMAC algorithm predicts the average channel behavior by considering the current and past channel quality predictions, it is rather slow in tracking channel quality degradations/upgradations at a particular epoch (as discussed in performance evaluation Section 6.1.2), which may adversely affect system performance. Additionally, the NEWMAC algorithm embodies no mechanism for performing channel stability assessment (as discussed in the performance evaluation Section 6.1.3).

#### 5.4. Normal Equation-Based Aggregate Maturity Criteria with Beta Tracking Based Channel Weight Prediction Algorithm

The NEAMCBTC algorithm has the ability to accommodate stream-based communications, because it estimates the long-term average channel quality on the basis of current and past channel rank estimations, as shown in Figure 5. It employs a dynamic channel maturity criterion as a measure of quality-stability of a channel discussed in Section 5.4.1. Moreover, it has the ability to immediately track any change in channel quality as explained in Section 5.4.2.

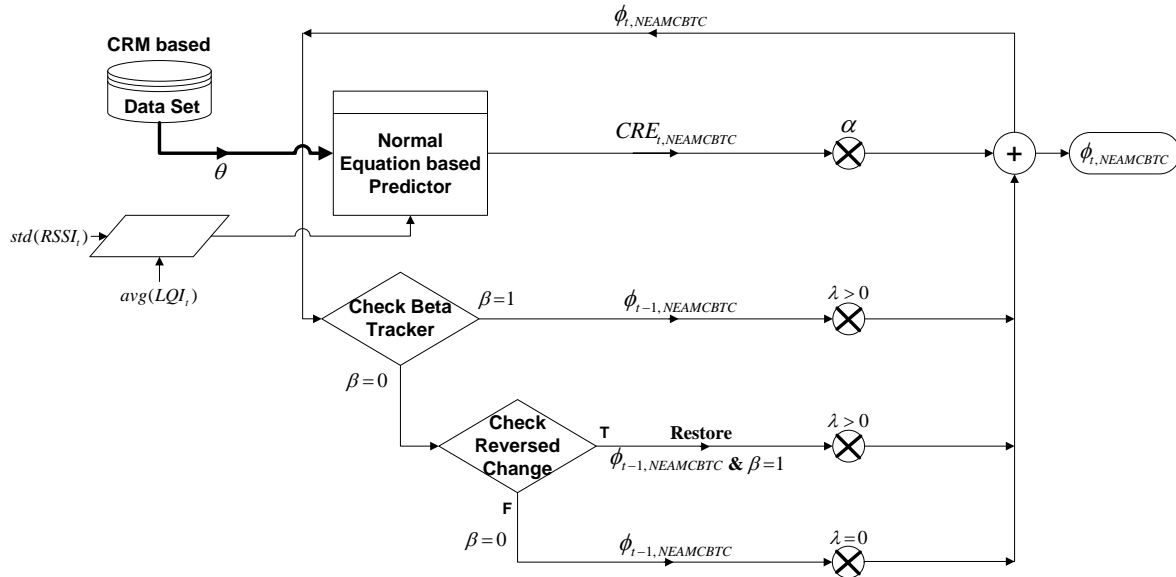


Figure 5. Data flow diagram of NEAMCBTC algorithm.

Let  $(CRE)_{t,NEAMCBTC}^i$  and  $\phi_{t-1,NEAMCBTC}^i$  denote the current and past channel quality predictions of a channel  $i$ . Let  $\eta_t^i$  and  $f(c)$  represent dynamic weights assigned to current and past channel quality assessment, while  $\beta$ -tracker is the measure of channel quality tracking. Then, NEAMCBTC-based channel quality assessment is given by:

$$\phi_{t,NEAMCBTC}^i = \left( \beta - tracker \right) \times \left( f(c) \right) \times \phi_{t-1,NEAMCBTC}^i + \left( \frac{1}{\eta_t^i} \right) \times (CRE)_{t,NEAMCBTC}^i \quad (19)$$

It is clear from Equation (19) that NEAMCBTC-based channel quality assessment is performed with the help of four interconnected procedures, i.e., channel maturity criterion  $\eta$ , channel stability criterion  $\psi$ , channel tracking criterion  $\beta$ -tracker and a circular function  $f(c)$ , as explained below.

#### 5.4.1. Channel Maturity Criterion ( $\eta$ )

Unlike the NEWMAC algorithm, the NEAMCBTC algorithm employs a dynamic channel weighting procedure controlled by  $\eta_t^i$ , which adaptively assigns weights to past and current channel quality predictions. If the channel quality level is sustained, then the value of  $\eta_t^i$  is matured (incremented) with time using general-stability criterion  $\psi_t^i$  (Equation (20)) and subsequently increases confidence on past channel quality prediction as determined by circular function  $f(c)$ , unless either the maximum maturity limit (here  $\eta_t^i = 10$ ) is obtained or the channel quality is majorly/minorly degraded/upgraded and the channel maturity procedure is resumed again (here  $\eta_t^i = 1$ ). Mathematically, we can write:

$$\eta_t^i = \begin{cases} \psi_t^i, & \text{if } \psi_t^i \leq 10 \\ 10, & \text{Otherwise} \end{cases} \quad (20)$$

and:

$$\lambda_t^i = f(c) = \begin{cases} \left( \frac{\eta_t^i - 1}{\eta_t^i} \right), & \text{if } \eta_t^i > 1 \\ 0, & \text{Otherwise} \end{cases} \quad (21)$$

#### 5.4.2. Channel Tracking Criterion ( $\beta$ -Tracker)

The  $\beta$ -tracker continuously monitors channel quality levels  $Q(ch)$  shown in Table 4 and makes the appropriate decision in the case of any degradation/upgradation in channel quality, as outlined in Table 5. As a result, it may adjust the channel stability criterion accordingly (Equation (25)). Moreover, when a channel enters into new quality level, then both  $\beta$ -tracker and circular function  $f(c)$  nullify the past channel rank estimation ( $\phi_{t-1, NEAMCBTC}^i$ ). The NEAMCBTC algorithm also employs a mechanism for handling any abnormal channel quality degradation/improvement, as discussed in Section 6.1.2.

**Table 4.**  $\beta$ -tracker based channel quality level  $Q(ch)$  assignment with  $q_1 = 0.3$ ,  $q_2 = 0.2$ ,  $q_3 = 0.1$ .

Serial No.	Channel Type	CRE	$Q(ch)$
1.	Good	$0.82 \leq$ to $\leq 1.00$	$q_1$
2.	Intermediate	$0.33 \leq$ to $< 0.82$	$q_2$
3.	Bad	$0 \leq$ to $< 0.33$	$q_3$

**Table 5.**  $\beta$ -tracker-based channel decision making.

Serial No.	$Q(ch_{t-1}^i)$	$Q(ch_t^i)$	$\beta_t^i$ -Tracker Decision Making	Channel Quality Explanation
1	$q_1$	$q_1$	1	Maintaining Good Quality
2	$q_2$	$q_2$	1	Maintaining Intermediate Quality
3	$q_3$	$q_3$	1	Maintaining Bad Quality
4	$q_1$	$q_2$	0	Minor Change (to Intermediate Quality)
5	$q_2$	$q_1$	0	Minor Change (to Good Quality)
6	$q_2$	$q_3$	0	Minor Change (to Low Quality)
7	$q_3$	$q_2$	0	Minor Change (to Intermediate Quality)
8	$q_1$	$q_3$	0	Major Change (to Bad Quality)
9	$q_3$	$q_1$	0	Major Change (to Good Quality)

The  $\beta$ -tracker is dependent on the current and previous values of channel quality level estimations  $Q(ch_{t,t-1}^i)$  and helps to track any change in channel quality level as calculated below:

$$\beta_{t,NEAMCBTC}^i = \left\langle \frac{\text{MIN}(Q(ch)_{t,NEAMCBTC}^i, Q(ch)_{t-1,NEAMCBTC}^i)}{\text{MAX}(Q(ch)_{t,NEAMCBTC}^i, Q(ch)_{t-1,NEAMCBTC}^i)} \right\rangle \quad (22)$$

Rewriting Equation (19) in simplified form, we get:

$$\phi_{t,NEAMCBTC}^i = \beta_t^i \times \lambda_t^i \times \phi_{t-1,NEAMCBTC}^i + \alpha_t^i \times (CRE)_{t,NEAMCBTC}^i \quad (23)$$

with:

$$\alpha_t^i + \lambda_t^i = 1 \quad (24)$$

#### 5.4.3. Channel General-Stability Criterion ( $\psi$ )

The channel general-stability criterion  $\psi$  is the measure of time since when a channel resides in a specific quality level (i.e., good/intermediate/bad). In this respect, a channel is considered more stable if it maintains a particular quality level for a prolonged interval. When a channel shifts to a new quality level, then the channel stability criterion  $\psi$  is resumed and incremented on each interval as long as the channel remains in that particular quality level as shown in the following equation:

$$\psi_t^i = \begin{cases} \psi_{t-1}^i + 1, & \text{if } \beta_t^i = 1 \\ 1, & \text{Otherwise} \end{cases} \quad (25)$$

#### 5.4.4. Problem Definition

Since, the behavior of a channel varies with time, the channel quality assessment has to be made repeatedly. For stream-based transmission on a channel, the quality assessment becomes even more critical because it involves sending more volume of data and reserving the channel for a prolonged interval. Therefore, if the channel quality assessment is done appropriately, it results in selecting an appropriate channel for performing future data transmission and routing. As a consequence, higher throughput and better system reliability may be achieved.

Let  $\phi_{t,NEAMCBTC}^i$  denotes the final quality of any channel  $i$  at time  $t$ , which is determined through NEAMCBTC-based past channel quality prediction and normal equation-based current channel rank estimation. Let  $C$ ,  $Z$  and  $i$  be defined as above with  $C = Z + i$ . Then, the objective function  $F$  is to select a channel  $i$  at instant  $t$  that exhibits maximum quality:

$$\text{Maximize: } F(\phi_{t,NEAMCBTC}^i), \quad i = 1, 2, 3 \dots C \quad (26)$$

$$\text{Subject to: } \phi_{t,NEAMCBTC}^i \geq \phi_{t,NEAMCBTC}^Z, \quad C = \{i + Z / i \notin Z\} \quad (27)$$

where,

$$0 \leq (\phi_{t,NEAMCBTC}^i) \leq 1.0$$

The above constraint (Equation (27)) states that the channel  $i$  with the highest quality will be selected for performing stream-based data communication at instant  $t$ . Since the NEAMCBTC algorithm also employs the channel maturity criterion ( $\eta$ ) and the channel tracking criterion ( $\beta$ -tracker), therefore it has the capability to perform limited quality stability and to track instantaneously any major/minor channel in channel quality, respectively. That is why the NEAMCBTC algorithm is more suitable to accommodate stream-based data communication than the NEWMAC algorithm.

#### 5.4.5. Extended-NEAMCBTC Algorithm (with General Stability Assessment)

When channel quality is majorly/minorly degraded/upgraded, then the NEAMCBTC algorithm resumes the channel maturity criterion from scratch. This may result in preferring a channel that has just attained the best quality, although it may have suffered from instability in the recent past (e.g., see the behavior of Channel 7 between interval [36, 39], as discussed in Section 6.1.1). This issue can be handled if we consider the general stability criterion  $\psi_t^i$  as a metric in the final channel quality estimation  $\phi_{t,NEAMCBTC}^i$  of any channel  $i$ . The resulting metric ( $\xi_{t,Ext-NEAMCBTC}^i$ ) would be more robust as given below:

$$\xi_{t,Ext-NEAMCBTC}^i = \phi_{t,NEAMCBTC}^i + \psi_t^i \quad (28)$$

Rewriting Equation (23), we get:

$$\xi_{t,Ext-NEAMCBTC}^i = \left( \beta_t^i \times \lambda_t^i \times \phi_{t-1,NEAMCBTC}^i + \alpha_t^i \times (CRE)_{t,NEAMCBTC}^i \right) + \psi_t^i \quad (29)$$

The  $\xi_t^i$  metric may enable sensor nodes to predict both quality and stability and make a better choice among available channels for performing stream-based communications. The data flow diagram of the Ext-NEAMCBTC algorithm is shown in Figure 6.

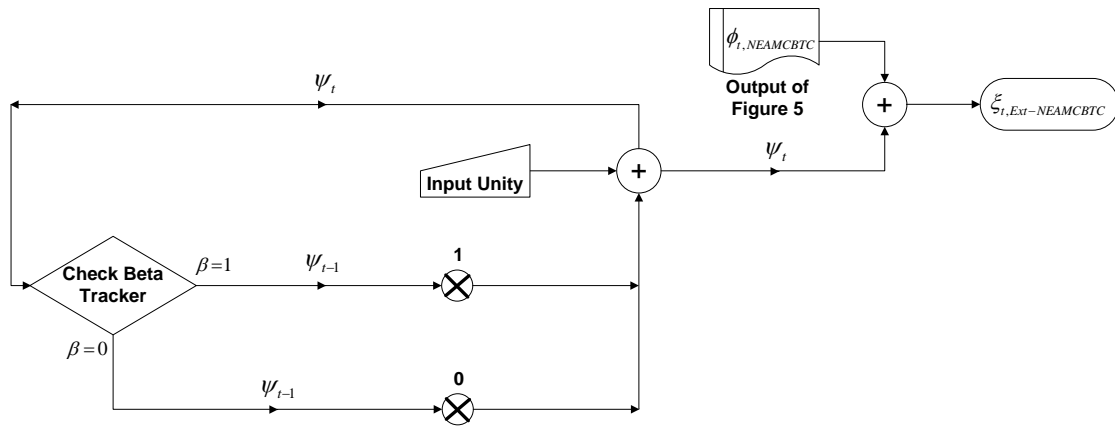


Figure 6. Data flow diagram of the Ext-NEAMCBTC algorithm.

## 6. Performance Evaluation

For measuring the performance of the proposed machine learning-based algorithms, we have conducted extensive simulations in MATLAB. In this respect, the experimental section can be divided into two main sub-sections:

- In the first portion of simulations, we discuss channel quality and stability assessment of our machine learning-based algorithms on the basis of randomly-generated samples of  $std(RSSI_t^i)$  and  $avg(LQI_t^i)$  in various channel quality ranges, as shown in Table 6, and representing the quality of seven channels. For this purpose, we have assumed that sensor nodes have some inherent mechanism for calculating  $std(RSSI_t^i)$  and  $avg(LQI_t^i)$  on the basis of received packets. The concluding remarks of this section are outlined in Section 6.1.4 and Table 7.
- In the second portion of simulations, we perform channel switching energy  $Energy_{Ch-Switch}$  and channel switching delay  $Delay_{Ch-Switch}$  related measurements of all discussed algorithms and compare them accordingly. Since, to the best of our knowledge, there is no multichannel scheme for WSNs similar to our work, therefore we have compared the performance of our schemes to the following approaches.

- i. Random selfish approach: In this technique, the channel is selected randomly among all of the available channels. Afterwards, the sensor node communicates on the selected channel as long as the channel quality is either good ( $Ch_{good}$ ) or intermediate ( $Ch_{inter}$ ).
- ii. EM-MAC-based approach: It follows the pseudo-random order-based frequency hopping mechanism of EM-MAC [26], whereby channels of acceptable quality are hopped only in a pseudo-random manner, while bad quality channels ( $Ch_{bad}$ ), once identified, are marked as blacklisted for a specific time interval.

Table 6. Simulation parameters.

Symbol	Description	Value
$N$	Number of channels	7
$\theta_0$	Machine learning based weight of parameter $X_0$	0.0824
$\theta_1$	Machine learning based weight of parameter $X_1$	−0.0333
$\theta_2$	Machine learning based weight of parameter $X_2$	0.0083
$std(RSSI)_{good}$	Standard deviation RSSI of good quality channel	<4 [20]
$avg(LQI)_{good}$	Average LQI of good quality channel	>104 [20]
$std(RSSI)_{inter}$	Standard deviation RSSI of intermediate quality channel	4–10 [20]
$avg(LQI)_{inter}$	Average LQI of intermediate quality channel	70 to ≤104 [20]
$std(RSSI)_{bad}$	Standard deviation RSSI of bad quality channel	>10 [20]
$avg(LQI)_{bad}$	Average LQI of bad quality channel	<70 [20]
$Ch_{good}$	Quality range of good rank channel	$0.82 \leq$ to $\leq 1.00$
$Ch_{inter}$	Quality range of intermediate rank channel	$0.33 \leq$ to $< 0.82$
$Ch_{bad}$	Quality range of bad rank channel	$0.0 \leq$ to $< 0.33$
$T_s$	Sampling Time Interval	$1 \times 10^2$ ms
$Delay_{Ch-Switch}$	Overall channel switching delay	50 ms (approx) [34]
$Delay_{CbRX}$	Delay in calibrating receiver	22.08 ms [34]
$Delay_{CbTX}$	Delay in calibrating transmitter	23.44 ms [34]
$Delay_{T_{restart}}$	Delay in restarting radio after calibration	4.32 ms [34]
$Energy_{Ch-Switch}$	Total energy consumption in channel switching	1940 nJ (approx) [34]
$Energy_{CbRX}$	Energy consumption for calibrating receiver	1005.05952 nJ [34]
$Energy_{CbTX}$	Energy consumption for calibrating transmitter	838.42536 nJ [34]
$Energy_{T_{restart}}$	Energy consumption for restarting radio after calibration	96.95376 nJ [34]

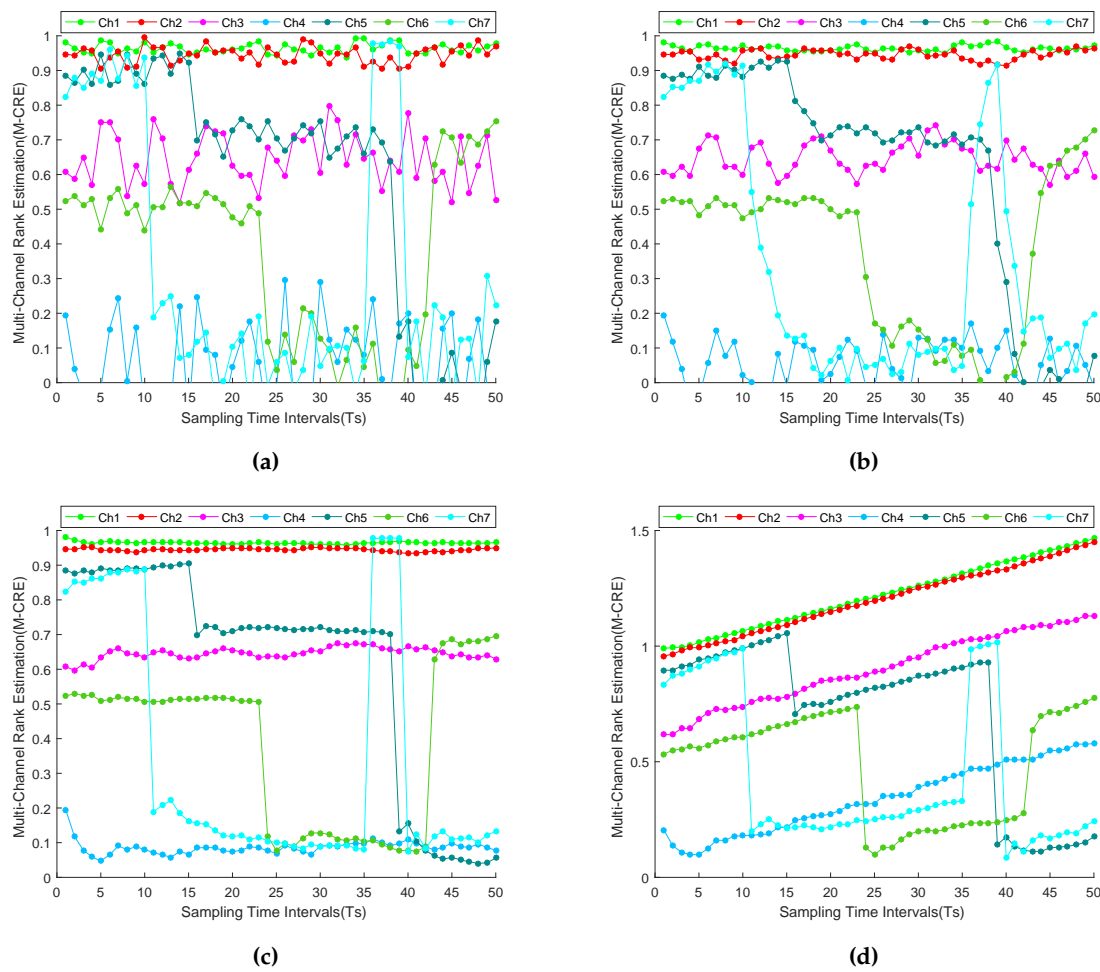
### 6.1. Channel Quality and Stability Assessment Using the Proposed Machine Learning-Based Algorithms

In this section, we simulate and discuss the functionality of proposed algorithms for accommodating stream-based communication. For this purpose, the proposed algorithms are evaluated using three main factors as given below.

#### 6.1.1. Channel Quality Assessment

Stream-based communication requires transmitting chunks of information from source to destination rather than performing packet-by-packet delivery of data. When the quality of available channels is overlapping, then those channel quality assessment (CQA) approaches that perform channel quality estimation on the basis of only instantaneous observation(s) of channel quality may suffer from frequent channel switching overheads. This is due to the fact that the sensor node may tend to occupy the best quality channel at each epoch, which may result in inducing frequent channel switchings at the corresponding epochs. Such frequent channel switching may be risky and, therefore, is avoided for high data-rate applications, such as stream-based communication, because it may result in additional energy consumption and data loss [12]. This can be seen, for instance, in Figure 7a, where an NEC algorithm-based sensor node mostly switches between Channels 1 and 2 and sometimes to Channel 7. In addition, at Time Instances 7, 19 and 33, the sensor node rapidly shifts between Channels 1 and 2, which increases the channel switching overhead. Therefore, the NEC algorithm is unable to perform channel quality assessment required for stream-based communication.





**Figure 7.** Channel quality (and stability) assessment results. (a) results for NEC; (b) results for NEWMAC; (c) results for NEAMCBTC; (d) results for Ext-NEAMCBTC (with stability).

Figure 7b shows the results for NEWMAC algorithm. Here, channel switching happens only between Channels 1 and 2 at Time Instances 12, 20, 30, 46 and 48, which may induce channel switching overhead in stream-based communication. In contrast, the NEAMCBTC algorithm increases the confidence on past channel rank estimation with time. As a result, it gives a better estimate of long-term average channel quality and handles any short-term channel quality degradations and upgradations. From Figure 7c, it is obvious that the NEAMCBTC algorithm clearly estimates the superiority of Channel 1 over Channel 2; therefore, no switching happens. Additionally, it suffers from only one round-trip channel switching overhead between Channels 1 and 7, and therefore, NEAMCBTC algorithm is more suitable to accommodate stream-based communication in WSNs than NEWMAC algorithm. From Figure 7d, it is evident that Ext-NEAMCBTC algorithm handles the round-trip channel switching overhead between Channels 1 and 7 in the Time Interval [36, 39], therefore Ext-NEAMCBTC algorithm performs better than the other devised algorithms.

### 6.1.2. Channel Quality-Tracking Assessment

Channel quality-tracking assessment (CQTA) is the measure of an algorithm's ability to promptly pursue any minor/major degradation/upgradation in channel quality. As shown in Table 5, minor degradation or upgradation happens when the quality of a channel changes from a higher to an adjacent lower level, or vice versa, whereas major degradation or upgradation in channel quality takes place when channel quality decreases from good to bad level, or vice versa. The CQTA of our algorithms is discussed below.

- NEC-based channel tracking: The NEC algorithm performs channel rank estimation on the basis of instantaneous channel quality observations(s); therefore, it provides prompt knowledge of a channel quality change, as shown in Figure 7a.
- NEWMAC-based channel tracking: Due to its moving average-based design, NEWMAC is unable to quickly respond to any change in channel quality. For example, looking at instantaneous knowledge based Figure 7a at Time Instant 11, Channel 7 is jammed. However, due to slow tracking ability, the NEWMAC algorithm still considers Channel 7 as of intermediate quality at Instant 11 and therefore prefers Channel 7 over Channel 6, as shown in Figure 7b. Consequently, this may result in system performance degradation. Similarly, when Channel 7 recovers from jamming at Time Instant 36, then the NEWMAC algorithm again takes some time in tracking the new quality of Channel 7. Hence, due to its poor tracking ability, the NEWMAC algorithm is not a good candidate for performing stream-based communication.
- NEAMCBTC-based channel tracking: Whenever any meaningful change in channel quality level is observed, then it is immediately tracked by  $\beta$ -tracker, which equalizes its channel quality tracking ability to that of NEC algorithm, as shown in Figure 7c. For example, as soon as Channel 7 suffers from jamming attack at Time Instant 11, then NEAMCBTC immediately tracks it, as shown in Figure 7c, and thereby, avoids Channel 7. Unlike NEWMAC, NEAMCBTC is able to promptly track recovery of Channel 7 from jamming, as shown in Figure 7c, which makes it a good candidate for accommodating stream-based communication. Since Ext-NEAMCBTC is based on NEAMCBTC, therefore it can promptly track any major/minor degradation/upgradation in channel quality at a particular epoch and can accommodate stream-based communication in WSNs.
- Channel abnormal behavior tracking and healing: Sometimes, instantaneous distortions in channel quality estimations crop up due to environmental factors, which may strongly effect the prediction capability of those memory-based systems that consider past knowledge in the future channel quality estimations. Unlike NEWMAC, NEAMCBTC has the ability to effectively handle any such instantaneous quality distortions. This capability of NEAMCBTC helps its channel maturity criterion to function properly and make better decisions for final channel rank estimations. For example, we have deliberately introduced short-term minor/major abnormalities in Channel 5 and drawn its graph using the NEC, NEWMAC and NEAMCBTC algorithms, as shown in Figure 8. Being instantaneous knowledge based, NEC is not affected by any such irregularities from the past. On the other hand, NEWMAC is strongly affected by those oddnesses, while NEAMCBTC employs an inherent mechanism for suppressing these instantaneous abnormal distortions.

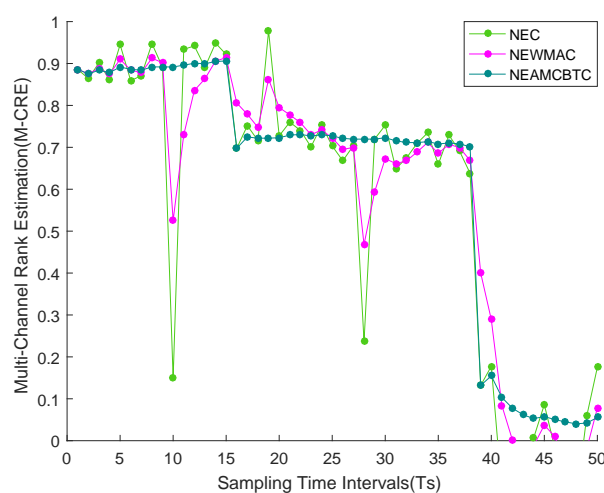


Figure 8. Channel abnormal behavior tracking and healing.

### 6.1.3. Channel Stability Assessment

The stability of a channel is the measure of time during which a channel occupies a particular channel quality level. When stability is also considered as a metric for determining the channel rank, then good quality stable channels are assigned more weight and preferred over good quality unstable channels for performing stream-based communication.

More specifically, channel quality assessment aims to select the best quality channel, whereas channel stability assessment focuses on selecting a channel whose quality may remain steady for a prolonged interval. The above discussion has realized the fact that although NEAMCBTC is superior to its counterparts; it embodies only partial quality stability. Therefore, it handles both good quality stable/unstable channels using a similar mechanism, as shown for instance in Figure 7c, where good quality unstable Channel 7 is preferred over good quality stable Channel 1 in the time interval [36, 39]. To bridge this gap, the extension Ext-NEAMCBTC incorporates a composite metric (Equation (29)), which considers both channel quality and stability. It enables sensor nodes to prefer Channels 1, 2 and 3 over Channel 7 for time interval [36, 39], as shown in Figure 7d, and thus, enhances the capability of our system to accommodate stream-based communication.

### 6.1.4. Concluding Remarks: A Brief Discussion

In this section, we will summarize the pros and cons of our algorithms and their appropriateness for performing stream-based data communication in WSNs, as outlined in Table 7.

**Table 7.** Feasibility of the proposed schemes for stream-based communication in multichannel WSNs. CQA, Channel Quality Assessment; CQTA, channel quality-tracking assessment; CSA, channel stability assessment.

Protocol	CQA	CQTA	CSA
NEC	—	✓	—
NEWMAC	✓	—	—
NEAMCBTC	✓	✓	Partial
Ext-NEAMCBTC	✓	✓	✓

Since NEC makes estimations on the basis of current channel quality observation(s) only, therefore it is computationally the lightest among the discussed algorithms and can figure out the best among the available channels at the current epoch, which enables it to perform channel quality tracking easily. On the other hand, it is unable to give any long-term/average prediction of the channel quality and may suffer from frequent channel switching overheads, which makes it a poor choice for accommodating stream-based communication.

The moving average-based NEWMAC algorithm estimates channel quality on the basis of both past channel knowledge and current channel quality observation(s). Thus, it can predict the average behavior of a channel and may avoid frequent channel switching overheads, which makes it a better candidate for performing stream-based communication. Being based on the moving average approach, it requires more memory and processing power than NEC and exhibits a slow growth rate, which is unsuitable for promptly tracking any major/minor changes in channel quality. Hence, in an environment where channels are suffering from rapid quality degradations and/or upgradations, NEWMAC may not provide suitable knowledge of channel quality for accommodating stream-based communication.

NEAMCBTC, finally, estimates channel quality using an adaptive channel maturity criterion that dynamically assigns weight to past channel knowledge and current channel rank observations(s). Thus, it gets long-term average channel behavior required for handling channel switching overheads and accommodating stream-based communication. The approach also embodies a robust channel tracking mechanism, which may accurately track any major/minor change in channel quality. Due to increased

functionality, the algorithm may require more memory and processing capability as compared to the already discussed algorithms.

The NEAMCBTC algorithm, however, suffers from preferring those unstable channels that may exhibit better quality than stable channels, even for short intervals of time. This may result in inducing limited channel switching overheads, which may limit the performance of NEAMCBTC for accommodating stream-based communication. The extension, Ext-NEAMCBTC algorithm, solves this outstanding issue and encourages justice between stable channels and good quality unstable channels.

## 6.2. Measurement of Channel Switching Overhead

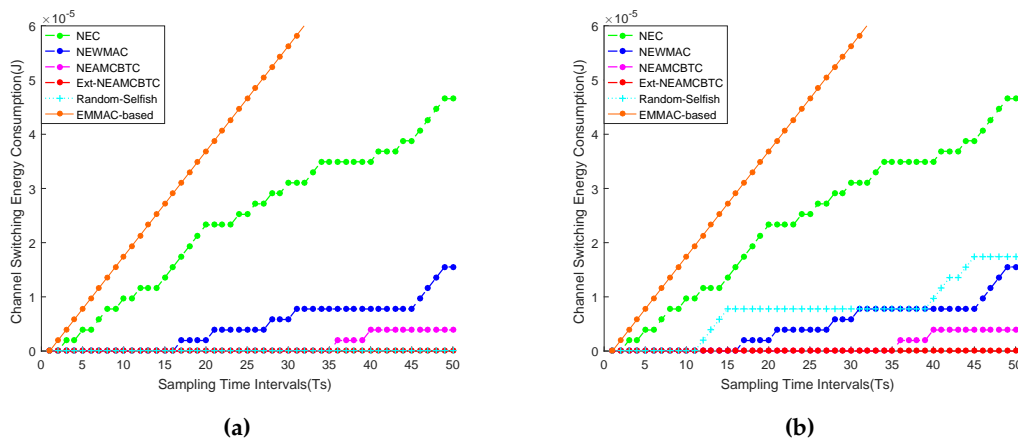
This section discusses channel switching overheads in terms of switching delay and energy consumption of relevant algorithms, which may help to figure out their efficiency and suitability for accommodating stream-based communication. For more realistic calculations, we have utilized channel switching delay and energy consumption values outlined in [34].

### 6.2.1. Channel Switching Energy Overhead

Channel switching energy overhead is the amount of energy consumed by a sensor node when it jumps from one channel to another. The total energy consumed in channel switching is the sum of the energy consumed in calibrating (RX,TX) and restarting the radio afterwards [34]:

$$Energy_{Ch-Switch} = [Energy_{Cb_{RX}}] + [Energy_{Cb_{TX}}] + [Energy_{T_{restart}}] \quad (30)$$

It is clear from the Figure 9a,b that Ext-NEAMCBTC performs better as compared to other algorithms because it allows sensor nodes to jump to the channel of best quality and stability. Moreover, the sensor node resides on the best available channel as long as the quality of that channel is better than the other available channels. Since Ext-NEAMCBTC helps the sensor node to minimize channel switching overhead as much as possible by taking preemptive channel assessment measures, it encounters the least channel switching energy consumption among the compared techniques. On the other hand, the performance of the EM-MAC-based approach is the lowest among all available approaches because it follows a procedure with frequent frequency hopping, which allows it to switch from one channel to another.



**Figure 9.** Channel switching energy overhead. (a) Case I: the channel switching energy consumption of the compared techniques shows that Ext-NEAMCBTC and random selfish (in the above scenario) are the best due to no switching energy overhead, while the EM-MAC-based approach behaves the worst among all. (b) Case II: the channel switching energy consumption of the compared techniques shows that Ext-NEAMCBTC is superior while random selfish (in the above scenario) performs worse than the NEWMAC approach. Random channel selection gives varying behavior to the random selfish approach.

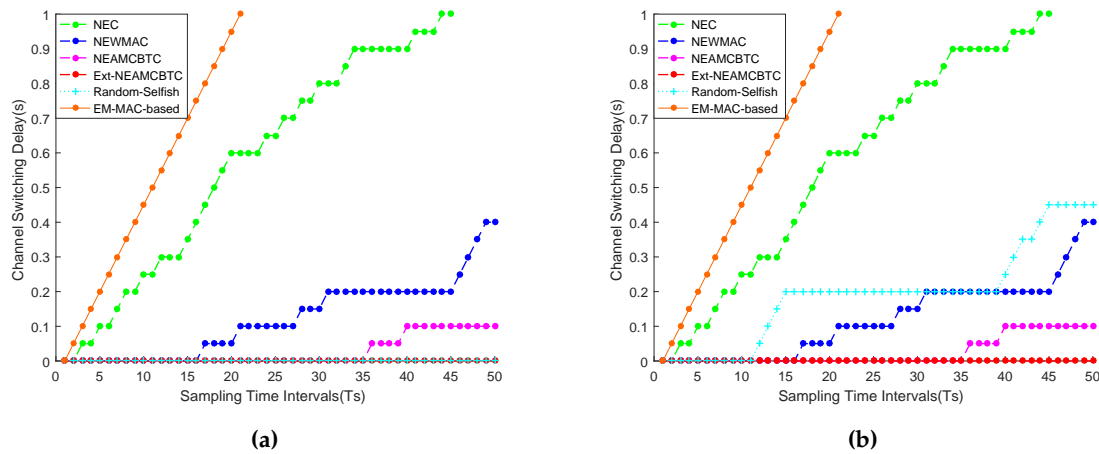
Additionally, the random selfish approach consumes  $0\text{--}1.8 \times 10^{-5}$  J energy by making 0–9 channel switchings, respectively, in the current scenario, as is evident from Figure 9a,b. This is due to the fact that the random selfish approach hops channels randomly and stays on a channel as long as the channel quality is good. Therefore, if it randomly hops to a channel of good quality and stability, it suffers from the least channel switching overhead, as shown in Figure 9a. On the other hand, if the random selfish approach selects channels that are either of bad quality or their quality is degraded soon, then it may suffer from frequent channel switchings, resulting in an increase in channel switching energy budget, as shown in Figure 9b. Consequently, such random behavior may cause performance issues, which may be considered preemptively while employing random-based approaches.

### 6.2.2. Channel Switching Delay Overhead

The switching delay is the summation of latency experienced by a sensor node while shifting from one channel to other. It is the measure of delay experienced in calibrating (RX,TX) and restarting the radio afterwards [34]. Mathematically, we can write:

$$Delay_{Ch-Switch} = [Delay_{Cb_{RX}}] + [Delay_{Cb_{TX}}] + [Delay_{T_{restart}}] \quad (31)$$

It is clear from Figure 10a,b that Ext-NEAMCBTC suffers from the least channel switching delay among the compared approaches because it has the ability to categorize channels on the basis of its inherent channel quality and stability assessment mechanism. As a result, it avoids channel switching delays as much as possible, which may result in timely delivery of data-stream and ensuring system reliability through avoiding switching-oriented data losses. On the other hand, frequent channel hopping in EM-MAC-based may cause the most persistent channel switching delays and data losses, which makes it a poor candidate for accommodating stream-based communication.



**Figure 10.** Channel switching delay overhead. (a) Case I: the channel switching delay measurement of discussed algorithms where Ext-NEAMCBTC and random selfish (in the above scenario) are behaving the best owing to no switching delay overhead, whereas the EM-MAC-based approach behaves the worst due to frequent channel hopping; (b) Case II: the channel switching delay measurement of discussed algorithms where Ext-NEAMCBTC behaves the best while random selfish (in the above scenario) behaves worse than NEWMAC technique. The varying behavior of random selfish approach is due to random channel selection.

Moreover, the random selfish-based approach shows a mixed trend, as evidenced from Figure 10a,b, whereby it may either perform as good as Ext-NEAMCBTC in case it randomly occupies a channel of good quality and stability or it may perform worse than the NEWMAC algorithm in case it is unable to randomly hop to channels of good quality and stability. Thus, the random selfish

technique may suffer from a channel switching delay between 0 and 0.45 s owing to 0–9 channel switchings, respectively, in the current scenario, as evidenced from Figure 10a,b.

## 7. Conclusions

Conventional high data rate multichannel applications may perform per-packet handshaking, which may result in channel switching overheads and data losses. The solution is to perform stream-based communication that may require channel handshaking before transmitting the whole data stream and, consequently, handle channel switching overheads while ensuring network reliability. Efficient stream-based communication is possible when the behavior of channels is known before transmitting the whole data stream. For effective realization of channel behavior, it is important to consider both channel quality and stability because it results in making better decisions for overall channel quality assessment in stream-based multichannel wireless sensor networks and may increase network efficiency and reliability by avoiding unnecessary channel switchings. However, selecting a channel of good quality and stability is a challenging task. In this work, an initiative is taken for devising, comparing and analyzing different techniques in order to figure out the best among them that can perform both channel quality and stability assessment and assist sensor nodes in selecting the best among all available channels for performing stream-based communication and thereby achieving high performance in high data rate multichannel wireless sensor networks. In this way, channel switching overheads, such as switching delays, energy consumption and data losses, would be minimized, and network efficiency is ensured accordingly.

**Acknowledgments:** This research work was sponsored by Deutscher Akademischer Austauschdienst (DAAD) under Reference No. 91541022.

**Author Contributions:** Waqas Rehan conceived the idea and designed simulations. Waqas Rehan, Stefan Fischer and Maaz Rehan performed simulation work and wrote the paper. Stefan Fischer supervised the overall research work, critically evaluated and revised the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Cevik, T.; Zaim, A.H.; Yiltas, D. Localized power aware routing with an energy efficient pipelined wakeup schedule for wireless sensor networks. *Turk. J. Electr. Eng. Comput. Sci.* **2012**, *20*, 964–978.
2. Cevik, T.; Zaim, A.H. EETBR: Energy efficient token based routing for wireless sensor networks. *Turk. J. Electr. Eng. Comput. Sci.* **2013**, *21*, doi:10.3906/elk-1112-18.
3. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. A survey on sensor networks. *IEEE Commun. Mag.* **2002**, *40*, 102–114.
4. Demirkol, I.; Ersoy, C.; Alagoz, F. MAC protocols for wireless sensor networks: A survey. *IEEE Commun. Mag.* **2006**, *44*, 115–121.
5. Yigitel, M.A.; Incel, O.D.; Ersoy, C. QoS-aware MAC protocols for wireless sensor networks: A survey. *Comput. Netw.* **2011**, *55*, 1982–2004.
6. Akyildiz, I.F.; Melodia, T.; Chowdhury, K.R. A survey on wireless multimedia sensor networks. *Comput. Netw.* **2007**, *51*, 921–960.
7. Pietraski, P.J. Method for Channel Quality Prediction for Wireless Communication Systems. U.S. Patent 7,912,490 B2, 22 March 2011.
8. Hamid, M.A.; Alam, M.M.; Hong, C.S. Design of a QoS-aware routing mechanism for wireless multimedia sensor networks. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), New Orleans, LA, USA, 30 November–4 December 2008; doi:10.1109/GLOCOM.2008.ECP.159.
9. Cabezas, A.C.; Pena, T.; Labrador, M.A. An adaptive multi-channel approach for real-time multimedia wireless sensor networks. In Proceedings of the IEEE Latin-American Conference on Communications (LATINCOM), Medellin, Colombia, 10–11 September 2009; doi:10.1109/LATINCOM.2009.5305150.
10. Chen, Y.; Krishnamachari, B. MCC: A High-Throughput Multichannel Data Collection Protocol for Wireless Sensor Networks; Technical Report; University of Southern California (USC): Los Angeles, CA, USA, 2011.



11. Wang, X.; Wang, X.; Fu, X.; Xing, G. MCRT: Multichannel real-time communications in wireless sensor networks. *ACM Trans. Sens. Netw.* **2011**, *8*, 1–30.
12. Pal, A.; Nasipuri, A. DRCS: A Distributed routing and channel selection scheme for multi-channel wireless sensor networks. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM), San Diego, CA, USA, 18–22 March 2013; pp. 602–608.
13. Cevik, T.; Zaim, A.H. A multichannel cross-layer architecture for multimedia sensor networks. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 457045.
14. Wu, P.J.; Lee, C.N. Connection-oriented multi-channel MAC protocol for ad-hoc networks. *Comput. Commun.* **2009**, *32*, 169–178.
15. Hanninen, M.; Suhonen, J.; Hamalainen, T.D.; Hannikainen, M. Link Quality-Based Channel Selection for Resource Constrained WSNs. In *Advances in Grid and Pervasive Computing*; Riekki, J., Ylianttila, M., Guo, M., Eds.; Springer: Berlin, Germany; Heidelberg, Germany, 2011; pp. 254–263.
16. Srinivasan, K.; Levis, P. RSSI is under appreciated. In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets), Cambridge, MA, USA, 30–31 May 2006.
17. Boano, C.A.; Zuniga, M.A.; Voigt, T.; Willig, A.; Romer, K. The triangle metric: Fast link quality estimation for mobile wireless sensor networks. In Proceedings of the 19th International Conference on Computer Communications and Networks (ICCCN), Zurich, Switzerland, 2–5 August 2010.
18. Zuniga, M.; Irzyska, I.; Hauer, J.H.; Voigt, T.; Boano, C.; Roemer, K. Link quality ranking: Getting the best out of unreliable links. In Proceedings of the International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), Barcelona, Spain, 27–29 June 2011.
19. Gomes, R.D.; Rocha, G.B.; Lima Filho, A.C.; Fonseca, I.E.; Alencar, M.S. Distributed approach for channel quality estimation using dedicated nodes in industrial WSN. In Proceedings of the IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC), Washington, DC, USA, 2–5 September 2014; pp. 1943–1948.
20. Bildea, A. Link Quality in Wireless Sensor Networks. Ph.D. Thesis, Universite de Grenoble, Grenoble, France, 2013.
21. Vlavianos, A.; Law, L.K.; Broustis, I.; Krishnamurthy, S.V.; Faloutsos, M. Assessing link quality in IEEE 802.11 wireless networks: Which is the right metric? In Proceedings of the IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Cannes, France, 31 August–4 September 2008.
22. Upadhyayula, S.; Annamalai, V.; Gupta, S.K. A low-latency and energy-efficient algorithm for convergecast in wireless sensor networks. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), San Francisco, CA, USA, 1–5 December 2003; Volume 6, pp. 3525–3530.
23. Yen, H.H.; Lin, C.L. Integrated channel assignment and data aggregation routing problem in wireless sensor networks. *IET Commun.* **2009**, *3*, 784–793.
24. Yen, H.H. Optimization-based channel constrained data aggregation routing algorithms in multi-radio wireless sensor networks. *Sensors* **2009**, *9*, 4766–4788.
25. Huang, C.; Fu, Y.; Zhong, M.; Yin, C. Multi-channel Wireless Sensor Network MAC protocol based on dynamic route. In Proceedings of the International Conference on Multimedia Technology (ICMT), Hangzhou, China, 26–28 July 2011; pp. 420–424.
26. Tang, L.; Sun, Y.; Gurewitz, O.; Johnson, D.B. EM-MAC: A dynamic multichannel energy-efficient MAC protocol for wireless sensor networks. In Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Paris, France, 16–19 May 2011.
27. Miao, G.; Li, G.Y.; Swami, A. Decentralized optimization for multichannel random access. *IEEE Trans. Commun.* **2009**, *57*, 3012–3023.
28. Wen, J.; Wu, M.; Bo, L.V.; Wang, D. Opportunistic cooperative routing in multi-radio multi-channel wireless sensor networks. In Proceedings of the IEEE International Conference on Communications Workshops (ICC), Budapest, Hungary, 9–13 June 2013; pp. 276–280.
29. Khan, Z.A.; Auguin, M. A multichannel design for QoS aware energy efficient clustering and routing in WMSN. *Int. J. Sens. Netw.* **2013**, *13*, 145–161.
30. Yu, Q.; Chen, J.; Sun, Y.; Fan, Y.; Shen, X. Regret matching based channel assignment for wireless sensor networks. In Proceedings of the IEEE International Conference on Communications (ICC), Cape Town, South Africa, 23–27 May 2010.

31. Halder, S.J.; Kim, W. A fusion approach of RSSI and LQI for indoor localization system using adaptive smoothers. *J. Comput. Netw. Commun.* **2012**, *2012*, 790374.
32. Texas Instruments. 2.4 GHz IEEE 802.15.4/ZigBee-Ready RF Transceiver. Available online: <http://www.ti.com/lit/ds/symlink/cc2420.pdf> (accessed on 9 March 2016).
33. Derivation of the Normal Equations. Available online: [https://en.wikipedia.org/wiki/Linear\\_least\\_squares\\_\(mathematics\)](https://en.wikipedia.org/wiki/Linear_least_squares_(mathematics)) (accessed on 17 December 2015).
34. Chen, H.; Cui, L.; Lu, S. An experimental study of the multiple channels and channel switching in wireless sensor networks. In Proceedings of the 4th International Symposium on Innovations and Real-Time Applications of Distributed Sensor Networks (IRADSN), Hangzhou, China, 18–21 May 2009; pp. 54–61.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).