Wireless Netw
DOI 10.1007/s11276-009-0221-y

Real-time detection of traffic anomalies in wireless mesh networks

Zainab R. Zaidi · Sara Hakami · Bjorn Landfeldt · Tim Moors

Abstract Anomaly detection is emerging as a necessary component as wireless networks gain popularity. Anomaly detection has been addressed broadly in wired networks and powerful methods have been developed for correct detection of a variety of known attacks and other anomalies. In this paper, we propose a real-time anomaly detection and identification scheme for wireless mesh networks (WMN) using components from previous methods developed for wired networks. Experiments over a WMN testbed show the effectiveness of the proposed scheme in isolating different types of anomalies, such as Denial-of-service attacks, port scan attacks, etc. Our scheme uses Chi-square statistics and it is based on similar ideas as the scheme presented by Lakhina et al. although it has lower computational complexity. The original method by Lakhina et al. was developed for wired networks and used Principal Component Analysis (PCA) for reducing the dimensions of observed data and Hotelling's t^2 statistics to distinguish between normal and abnormal traffic conditions. However, in our studies we found that dimension reduction is the most computationally intensive process of the scheme. In this paper we propose an alternative way of reducing dimensions using flow variances in a Chi-square test. Experimental results show that the Chi-square test performs similarly well to the PCA-based method at merely a fraction of the computations. Moreover, we propose an automatic identification scheme to pin-point the cause of the detected anomaly and its contribution in terms of additional or lack of traffic. Our results and comparison with other statistical tools show that the Chi-square test and the PCA-based method with identification scheme make powerful tools for real-time detection of various anomalies in an interference prone wireless networking environment.

Keywords Anomaly detection · Wireless mesh networks · Principal component analysis · Chi-square statistics · Denial-of-service · Port scan

Z. R. Zaidi (✉)
Networked Systems Group, NICTA, Locked Bag 9013, Alexandria, NSW 1435, Australia
e-mail: Zainab.Zaidi@nicta.com.au

S. Hakami · T. Moors
School of EE&T, UNSW, Sydney, NSW 2052, Australia

B. Landfeldt
School of IT, USyd, Sydney, NSW 2006, Australia

S. Hakami · B. Landfeldt
NICTA, Alexandria, Australia

1 Introduction

With growing popularity of wireless networks, it is becoming critically important to work towards providing similar service characteristics to the users as they are accustomed to in wired infrastructure networks. Wireless Mesh Networks (WMN) is a specific class of wireless networks that have attracted much interest from both academia and industry because of potential back-haul cost-effectiveness. There are however, only pilot and experimental deployments of wireless mesh networks at the moment, e.g., MIT Roofnet [21], Wray village [12], etc. Compared with wired networks, there is therefore limited experience with real incidents and security issues. It is not difficult to imagine the extent of additional vulnerabilities associated with WMN that should be taken care of before any major deployment can occur.

The most significant difference, between a wired and a wireless infrastructure network, lies in the fact that links are

Published online: 27 October 2009

relatively unreliable, dynamic, and resource constrained in nature. Moreover, the unprotected locations of wireless routers expose them to malicious intrusions, such as, jamming, Denial-of-Service (DoS) attacks, and environmental hazards, such as thunder storms, etc. [25]. As a consequence of faults (natural and man-made), wireless mesh networks might perform inefficiently and malfunctions such as node failures, DoS attacks, etc., can have a more severe impact on wireless networks than they would have on wired networks, due to limited and shared resources.

Since WMN is a recent construct, not many schemes have been developed for fault management taking the specific considerations of such networks into account. A brief overview of fault detection/management techniques is provided in Sect. 2 and their respective merits and demerits are pointed out. Typically, existing schemes monitor a single or collection of nodes, e.g., destinations, for unexpected behavior in order to detect malicious intrusions. To the best of our knowledge, [22] presents the only method for WMN where traffic traces from a gateway node can be fed into a processing engine in order to be classified as normal or abnormal. The processing engine in Qiu et al. [22] consists of a simulator which is previously trained using similar traces. On the other hand, a rich collection of mature research is available in the anomaly detection area for wired networks. These techniques generally incorporate some statistical or analytical tool to process measured data, for example, empirical densities [1], wavelets [10], etc. However, as wireless links have higher interference and variability than wired links it is not known if a direct implementation of these methods would be able to detect anomalies in noisy environments while keeping the false alarm rate and additional overhead at a reasonable level.

In this paper, we present a computationally economical algorithm for real-time anomaly detection in WMN. Our algorithm is derived from the technique developed in Lakhina et al. [16] for wired infrastructure networks. The method of Lakhina et al. [16] employs Principal Component Analysis (PCA) [13] for dimension reduction and filtering of observed data and uses Hotelling's t^2 statistics to detect the data points deviating far from the mean traffic conditions. We have evaluated the scheme of Lakhina et al. [16] in Hakami et al. [8] for the scenario of a WMN testbed deployed in Sydney. Our experimental results show that PCA with t^2 statistics is able to detect the anomalies, such as, DoS, port scan, and node failures, in a rich mix of network traffic over wireless mesh networks albeit with a non-trivial number of false alarms. While analyzing the computational complexity of the PCA-based method we realized that dimension reduction is in fact the most computationally intensive component of this method. In order to overcome this severe limitation and provide a computationally tractable and reasonable method, we here propose a cost-effective algorithm that uses Chi-square statistics instead of PCA, still yielding similar performance to the PCA-based method in our experimental study.

Also, the PCA-based method, even though it has potential for real-time detection, is presented and evaluated as an off-line process in Lakhina et al. [16] and Hakami et al. [8]. In this paper, we use the on-line version of the algorithm in comparison analysis with the Chi-square test and other statistical tools used in anomaly detection in wired networks, such as statistical distributions (the Kolmogorov-Simrnov test) [1]. To the best of our knowledge, this is the first attempt to evaluate a real-time anomaly detection tool for the wireless mesh networking scenario.

Moreover, we have developed an algorithm for automatic identification of the source of a detected anomaly and its contribution in terms of number of packets or flows. Our identification method maps the detected anomalous time bin into the specific flow or flows causing the anomaly. The automatic identification is very useful in reducing the number of false alarms and increasing the efficiency of the anomaly detection process. We show through experiments that the identification method points out the traffic flows responsible for real faults.

Our contribution in this paper is summarized as follows:

- According to our knowledge this is the first attempt to propose and evaluate real-time anomaly detection tools for wireless mesh networks. Our evaluation experiments are performed in real outdoor wireless environment although with synthetically generated traffic. Besides being more vulnerable, wireless links have higher interference, variability, and limited resources than wired links which could result in higher false alarm rate and prohibit the use of any heavy-duty detection tool.
- Our computational analysis revealed that dimension reduction as performed by the PCA-based method is computationally complex and not suitable for real-time anomaly detection in resource constraint environment of WMN. An alternative method based on Chi-square statistics is proposed and comparison is done in terms of detection performance and computational complexity.
- We have enhanced the theory of detection methods based on PCA and Chi-square statistics by proposing a novel automatic identification scheme which could be used with either of the methods to identify the cause of the detected anomaly and its contribution in terms of increased or decreased traffic. The identification scheme is also useful in differentiating false alarms from real anomalies and initially developed to reduce the false

alarm rate in WMN. However, this scheme can also be used in wired networks.
- Similarly the PCA-based and Chi-square based methods are also enhanced by proposal of an alternate threshold when normal traffic could be heavy tailed, which is a typical case for much of the current Internet traffic. Use of standard thresholds of both methods yields higher false alarms in our study when traffic is dominated by self-similar flows.
- Experimental study also included Comparison of both schemes with other statistical tools used in anomaly detection in terms of detection accuracy, low computational complexity, and ability to identify the culprits.

Both methods, discussed in this paper, are effective in detecting anomalies which create spikes or sudden changes in network traffic, such as, DoS, port scan, node failures, etc. Other anomalies which do not result in rapid changes in network traffic, such as, MAC-spoofing, man-in-the-middle attack, mis-forwarding attack, etc., cannot be detected using these methods unless all nodes are individually monitored. For anomalies causing spikes or sudden changes, variances or 2nd moment analysis of data traffic is sufficient for detecting data points far away from the mean trend as done by Hotelling's t^2 and Chi-square statistics.

The rest of the paper is organized as follows: Sect. 2 contains a literature survey summary, Sect. 3 presents the real-time anomaly detection schemes including a brief summary of the PCA method proposed in Lakhina et al. [16] and details about the Chi-square test, Sect. 4 develops the method for automatic identification of causes, Sect. 5 presents details about our wireless mesh testbed, experiments and results, and finally Sect. 6 concludes the paper.

2 Related work

As mentioned above, since WMN is a recent development, not many schemes have been developed for fault management taking the particular nature of these networks into account. A comprehensive survey of fault management techniques in wireless *multihop* networks, i.e., including mobile ad hoc and sensor networks, can be found in Zaidi et al. [28]. In wireless multihop networks, largely for ad hoc and sensor networking scenarios, research is limited to either threshold-based techniques where loss of a certain number of ACKs or periodic updates trigger route recovery [28], or intrusion detection techniques, where incoming and outgoing links of each node are monitored locally for abnormal behavior, such as, artificial immune system (AIS) [25] and watchdog [20]. These techniques either require all nodes to monitor their neighbors, causing trust issues, or they require a high density of robust monitors which makes them very expensive.

Recently, work has been done to establish the requirements of a fault management system for WMN [19, 2] and to study the effects of monitoring on network performance [7]. Moreover, specific security issues and challenges of the WMN scenario are outlined in Refs. [26, 24]. To the best of our knowledge, Qiu et al. developed the first fault detection scheme for WMN [22]. They trained a simulator using traffic traces from WMN and used it for distinguishing normal and abnormal traffic patterns. Even though the scheme was able to detect a variety of faults, its performance heavily depended on the accuracy and flexibility of the simulator, selected to implement the scheme, and the quality of traffic traces used for training. Moreover, in order to adapt to a dynamic system, the simulator needed frequent training periods, increasing the cost of the scheme. Furthermore, even though the results were encouraging, a simulator driven approach is not suitable for real-time fault detection.

On the other hand, a huge volume of mature research is present in anomaly/fault detection in wired networks. According to Ye and Chen [27], anomaly detection methods complement the capability of a rule-based or signature matching intrusion detection technique. Signature matching techniques store the signature of known intrusion detection scenarios, but detection of a zero day attack, i.e., a new threat with no available security fix, is only possible through anomaly detection techniques. In wired network research, statistical methods are widely used for anomaly detection. These methods use statistical distributions such as Kolmogorov-Simrnov test [1], wavelets [10], and frequency distributions [14]. The Kolmogorov-Smirnov (KS) test [1] compares the empirical distributions of observed data with that of normal data and deviations exceeding a pre-determined threshold are classified as anomalous. The KS test is able to detect anomalous traffic but it is not able to identify the cause of an anomaly as shown from our experimental results. Packet content frequency distribution is used in Karamcheti et al. [14] to detect worm attacks. Wavelets can detect congestion by comparing the energy distributions over various wavelet components in normal and anomalous situations but the method cannot be used in real-time as it is computationally intensive and generally requires to process data sets collected over multiple days. The PCA-based method is, however, considered computationally feasible to be used to run online. Chi-square statistics based intrusion detection method is also proposed in Ye and Chen [27], where long-term profile of normal traffic is used to train the parameters of a classifier for different event types. Events of recent past are then compared against the normal profile and significant departures are termed as anomalies.

There are other techniques proposed in Dickinson et al. [3] and Feather et al. [4] that are very similar to the PCA-based method and where the difference in normal and anomalous traffic is quantified and compared against certain thresholds. In Feather et al. [4], values of performance parameters, such as, throughput, latency, etc., are saved from a training period as signatures of normal or abnormal trends. Observations are compared with the signatures for closest match. Time series of graphs are developed from the traffic observations in Dickinson et al. [3] which are used to construct a median graph representing typical behavior over a longer period. Any subsequent graph deviating from the median graph by a distance more than a threshold is deemed to be abnormal. Both methods require substantial traffic history to develop the signature library in Feather et al. [4] and median graph in Dickinson et al. [3]. Experiments done in Dickinson et al. [3] used over 100 days of data. Moreover, an efficient method to update the signatures or median graph with according to network dynamics is critical, especially for wireless networks.

Preliminary evaluation results of the PCA-based method [16] over a WMN testbed are presented in Hakami et al. [8], where the method is shown to detect anomalies though yielding some false alarms. In the present paper, we use a real-time version of the PCA-based method to compare against our algorithm based on Chi-square statistics for different parameter settings and load.

3 Real-time anomaly detection

In this section, we present a brief summary of the PCA-based method and highlight the issue of computational complexity. We then present details of our novel method, the Chi-Square test, which is significantly less complex than the PCA-based method. Functional blocks of anomaly detection schemes such as the PCA-based and the Chi-square based schemes are shown in Fig. 1, where X is the normalized traffic data collected from specific nodes in the network (cf. Sect. 3.1). In order to detect anomalies affecting different aspects of network traffic, there could be different ways to populate X such as:

1. Number of packets transmitted during each interval of the observation window (time bins), sorted according to OD (Origin-Destination) flow.
2. Number of transmitted bytes during each time bin for each OD-flow.

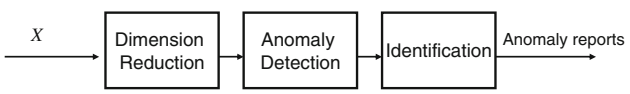


Fig. 1 Functional block of anomaly detection scheme

3. Number of OD-flows for all time bins.
4. Number of flows sorted according to the origins for all time bins.
5. Number of OD-flows sorted according to the port numbers for all time bins.

Once an observation window is specified, the nodes collect data within the window, passing it to the anomaly detection system. In typical networks, X might contain hundreds or thousands of flows. It is desirable to reduce the dimension of X before any further mathematical processing, here shown by the *Dimension reduction* block. The major difference between the PCA-based method and the Chi-square test lies in the way they reduce data dimensionality. After passing through 'Dimension reduction', the reduced data enters the *Anomaly Detection* block where data is compared with pre-selected thresholds according to specific confidence level and values exceeding the thresholds are deemed to be anomalous. Although, the mathematical formulations are different for both tests considered in this study, the 'Anomaly Detection' block is equivalent in both cases if the same confidence levels are selected based on the same underlying assumption about the distribution of traffic data.

The last block is *Identification* which maps back the anomalous time bins into traffic flows and identifies which nodes trigger a detection and then quantifies their contribution towards the anomaly in terms of excess or loss of packets/flows. This method is also useful in order to differentiate false alarms from real anomalies. Finally, anomaly reports are generated as output of the anomaly detection system. The anomaly reports consist of the following:

- Time bin where the anomaly is detected
- Responsible source
- Amount of excess/reduction of traffic (packets/bytes/flows)
- Type of traffic where anomaly is found, i.e., packets/bytes/flows

A sample of the anomaly report is shown in Table 3 in Sect. 5.

3.1 PCA-based method

As summarized in Lakhina et al. [17], PCA is a coordinate transformation method that maps the measured data onto a new set of axes called the principal axes or components. Each principal component has the property that it points in the direction of the maximum variance remaining in the data, given the variance already accounted for by the preceding components. This way, the first principal component is directed towards the maximum variance of the original

data. The second principal component is orthogonal to the first and represents the maximum residual variance among the remaining directions.

Let $\mathcal{X}$ of size $p \times l$ contains l columns of observed data which could be number of bytes/packets/flows for p time bins. Each principal component v_i is the ith eigenvector computed from the spectral decomposition of X^TX,where $X = \mathcal{X} - \bar{\mathcal{X}}$ and $\bar{\mathcal{X}}$ is the time average of $\mathcal{X}$. This normalization ensures that PCs capture the common temporal trends in traffic and are not skewed by the differences in mean traffic rates. Moreover the normalization along the mean traffic allows an easier way to differentiate flood attacks (data with positive sign) and outages (data with negative sign).

$$X^TXv_i = \lambda_i v_i, \quad i = 1, \cdots, l \tag{1}$$

where λ_i is the eigenvalue corresponding to v_i. Furthermore, because X^TX is a symmetric positive definite, its eigenvectors are orthogonal and the corresponding eigenvalues are nonnegative real. By convention, the eigenvectors have unit norm and the eigenvalues are arranged from large to small, so that $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_l$.

Considering the data mapped onto the principal components, it is clear that the contribution of principal axis i as a function of time is given by Xv_i [17]. This vector can be normalized to unit length through division by $\sqrt{\lambda_i}$. Thus, for each principal axis i,

$$u_i = \frac{Xv_i}{\sqrt{\lambda_i}}, \quad i = 1, \cdots, l. \tag{2}$$

The u_i's, called eigenflows, are vectors of size p that are orthogonal by construction [17]. Since the principal axes are in order of contribution to the overall variance, u_1 captures the strongest temporal trend common to all OD flows, u_2 captures the second strongest, and so on. The eigenvalues are useful for gauging the potential for reduced dimensionality in the data. For example, if the first two eigenvalues are 99.9% of the aggregated eigenvalues, the first two PCs might explain up to 99.9% of the total variability of the specific data set. Specifically, finding that only r_t singular values are non-negligible, implies that X effectively resides on an r_t-dimensional subspace of $\mathbb{R}^{<}$. In this case, the original X can be approximated as $X' \approx \sum_{i=1}^{r_t} \sqrt{\lambda_i} u_i v_i^T$. The rest of the PCs define the residual subspace $\tilde{X}$, i.e., $X = X' + \tilde{X}$.

In Lakhina et al. [16], the residual subspace is also referred to as an abnormal subspace and the calculation of squared prediction error (SPE) of eigenflows in an abnormal subspace is suggested as a method to detect anomalies. Although, in our experiments, we realized that since most of the network traffic is effectively characterized by X', the anomalies are also contained in X' rather than $\tilde{X}$. As a result, the performance of SPE is very poor as compared to t^2 statistics and results in significant number of missed detections and false alarms.

Hotelling's t^2 is a statistical measure of the multivariate distance of each observation from the center of the data set (for eigenflows, the center is zero by construction [15]). This represents an analytical way to identify the most extreme points in the data by calculating the sum of squares at each interval j of the eigenflows in the normal subspace, as follows:

$$t_j^2 = \sum_{i=1}^{r_t} u_{ij}^2, \quad j = 1, \cdots, p. \tag{3}$$

A peak in the t^2 graph exceeding the threshold δ_t as defined in Lakhina et al. [15] and shown in (4) is considered an anomaly.

$$\delta_t = \frac{r_t(p-1)}{p - r_t} F_{r_t, p-r_t, \alpha}, \tag{4}$$

where $F_{r_t, p-r_t, \alpha}$ is the value of F distribution with r_t and $p - r_t$ degrees of freedom at the $1 - \alpha$ confidence level.

The PCA-based method reduces dimensions of data by picking up the dominant trends and filtering the insignificant components in the 'Dimension reduction' block of Fig. 1. PCA calculation, or eigenvector decomposition, requires $O(l^3)$ computations or multiplication operations (cf. Cholesky factorization [5]), where l is the number of columns in X. The 'Anomaly Detection' block requires $O(r_t p)$ computations and 'Identification' is of order $O(l)$ per detected anomaly, as discussed later in Sect. 4. As traffic dimension or number of flows, i.e., l, grows, the 'Anomaly Detection' block largely remains static but the 'Dimension reduction' becomes increasingly costly. No computationally intensive post-reduction processes are used in Lakhina et al. [16] and even if dimensions are not reduced, the penalty would be to use all l vectors in 'Anomaly Detection' rather than r_t, where typically $r_t \ll l$. Although, a more severe penalty is the additional number of false alarms as normalized eigenflows [cf. (2)] can inflate the small perturbations along insignificant PCs and make them comparable to real anomalies if all PCs are used in calculation of t^2 statistics. The overall computational complexity of the PCA-based method is $O(l^3)$.

3.2 The Chi-square test

An alternative way of dimension reduction would be to choose the significant columns of the traffic matrix X on the basis of their respective variances instead of selection of dominant eigenvalues. This method of reducing dimensions is computationally inexpensive but is not optimal as the case of PCA-based method [6]. Also, it treats all flows to be independent of each other, although, there may exist

mathematical or physical dependencies between flows (columns of X) specially in the case of coordinated and distributed attacks. We expect that PCA-based method will be more suitable for distributed attacks. In this experimental study, we are only concerned with detecting flows which are deviating from their own mean trend and orthogonality can be assumed between the flows without sacrificing accuracy of detection. This interesting property makes it possible to avoid eigenvector decomposition and to save $O(l^3)$ computations.

Let σ_i^2 denote the variance of the ith flow or column of traffic matrix X, i.e.,

$$\sigma_i^2 = \frac{1}{p-1}\sum_{k=1}^{p} X_{ki}^2, \quad i = 1, \cdots, l. \tag{5}$$

Similar to the PCA method, variances are arranged in descending order, i.e., $\sigma_1^2 \geq \sigma_2^2 \geq \ldots \sigma_l^2$, and r_q significant components are selected, such that the sum $\sigma_1^2 + \cdots + \sigma_{r_q}^2$ comprises of significant portion of total aggregated variances, e.g., 95%. This alternative way of dimension reduction requires $O(lp)$ computations instead of the $O(l^3)$ required for PCA.

We now define test statistics q_j as

$$q_j = \sum_{i=1}^{r_q} \frac{X_{ji}^2}{\sigma_i^2}, \quad j = 1, \cdots, p. \tag{6}$$

where X_{ji} is the element of matrix X at column i and row j. Note that the matrix X is re-arranged to match the associated variances in descending order. If X follows a multivariate Gaussian distribution, then q_j is a Chi-square random variable. This assumption about distribution of data is also an underlying postulation in the PCA-based method when Hotelling's t^2 statistics is used to represent summation of eigenflows. Both methods' 'Anomaly Detection' modules (cf. Fig. 1) differ only in mathematical forms but they are in fact equivalent. A peak in the q graph exceeding the threshold δ_q is considered an anomaly, where δ_q is the value of the Chi-square distribution with r_q degree of freedom at the $1 - \alpha$ confidence level.

In the Chi-square test, we have replaced the 'Dimension reduction' block of Fig. 1 by a selection of flows with higher variances. Computations required for this method are of the order of $O(lp)$. The 'Anomaly Detection' block requires $O(r_q p)$ computations. The 'Identification' block still requires $O(l)$ per detected anomaly, details are discussed in Sect. 4. The overall complexity of the Chi-square test is therefore $O(lp)$.

3.3 Alternative threshold

The threshold for t^2 in (4) and δ_q for q are accurate if the underlying data, i.e., X is normally distributed. In our case, network traffic is typically self-similar which would result in a heavier tailed distribution. Thresholds defined for q and t^2 may result in a higher rate of false alarms, something that was also shown in Hakami et al. [8] for t^2 statistics.

We propose a new threshold according to the Chebyshev inequality which states that

$$\mathrm{P}(|Y - \mu_y| \geq k\sigma_y) \leq \frac{1}{k^2}, \tag{7}$$

where Y could be any data set with mean μ_y and standard deviation σ_y and k is a constant. The standard deviation σ_y could be σ_{t^2} or σ_q for t^2 vector calculated from (3) or q vector calculated from (6) respectively. A threshold of $\delta_{yc} = 4\sigma_y$ will ensure that 94% of the data will reside under the threshold. A threshold of $\delta_{yc} = 7\sigma_y$ could be used for 98% confidence bounds. The alternative threshold δ_{yc} could be either δ_{tc} or δ_{qc} depending on the statistics used for detection, i.e., t^2 or q respectively. Since the Chebyshev inequality is a weaker upper-bound, the chances of missed detection will increase with a higher threshold. A positive aspect of this threshold is its distribution independence. However, if the data distribution is closer to normal distribution, δ_t from (4) and δ_q for q constitute better thresholds.

4 Automatic identification of anomalies

Once an anomaly has been detected, it is vital to know: who caused it and what is the impact of the anomaly on the network traffic. The identification scheme can identify the OD flow or flows, or more generally sources, causing peaks in q or t^2 vectors and their contribution in terms of number of packets or flows. Our scheme helps eliminate false alarms due to traffic perturbation as well as providing an efficient way of identifying the anomalous OD flows and can even help classify the type of specific anomaly in some cases.

The principle behind our identification scheme is the reverse mapping of q or t^2 vectors into the measurement space. Once q or t^2 detect an anomaly in a particular time bin j, all significant contributing flows are identified as follows:

$$q_j \text{ or } t_j^2 = \sum_k c_k, \tag{8}$$

where c_k is the contribution of OD flow k. For q statistics,

$$c_k = \frac{X_{jk}^2}{\sigma_k^2}, \tag{9}$$

where σ_k^2 is given in (5) and for t^2 statistics,

$$c_k = X_{jk} \sum_{n=1}^{l} X_{jn} \sum_{i=1}^{r_t} \frac{(v_i)_k (v_i)_n}{\lambda_i}, \tag{10}$$

where $(v_i)_k$ is the kth element of eigenvector v_i [cf. (1)]. The OD flow k with the largest c_k calculated for q or t^2 is the major contributor in the peak of the respective statistics.

X_{jk} is either the flow contribution from the kth origin or the packet/byte contribution of the kth OD flow if c_k is found to be exceeding a percentage contribution threshold of T_c. In our experiments, an arbitrary value, 10, is selected for T_c. Since X is the normalized traffic vector, X_{jk} could be positive or negative. A negative value for X_{jk} shows the absence of packets with respect to the mean flow rate, which could happen in a node outage scenario, and a positive value indicates excess packets, for example found during a DoS attack. The type of traffic where an anomaly is found, is helpful in characterizing the anomaly. For example, DoS and flooding should be visible in packet and byte count data where as port scan should be seen in the flow count. According to our experiments, this method is very effective in identifying different anomalies, such as DoS, port scan, and node outages. Although, 'Identification' block in both methods calculates c_k by different mathematical formulations they are essentially the same and yield very similar results. The only difference between both methods lies in their respective strength in detecting the time bins with anomalies. After detection, identification yields similar performance for q and t^2 statistics.

5 Experiments and results

We used NICTA's outdoor mesh network testbed for our experiments. The testbed has 7 nodes deployed at traffic intersections and one gateway mesh node inside the School of IT at The University of Sydney. The layout of the testbed with all wireless links, is shown in Fig. 2. The testbed operates as a WMN with no direct access to the outside world apart from a fixed link at the gateway node. Each node is equipped with three wireless interfaces: 2 WiFi (unlicensed bands) and 1 Unwired^{TM} (licensed band). Unwired^{TM} is a wireless broadband provider in Sydney operating a proprietary radio at 3.5 GHz. Unwired radios are used for control purposes in our testbed and are shown as curly arrows in Fig. 2. links exist between every two adjacent nodes and operate at channel 9 of the 2.4 GHz band (802.11 g), shown as thick lines in Fig. 2. Some nodes have extra WiFi links which operate at either 2.4 GHz channel 1 (between mesh02 and mesh03), or 900 MHz channel 4 (between mesh01–mesh05 and mesh04–mesh07). WiFi links use Omni-directional antennas. More details about the testbed can be found in Lan et al. [18].

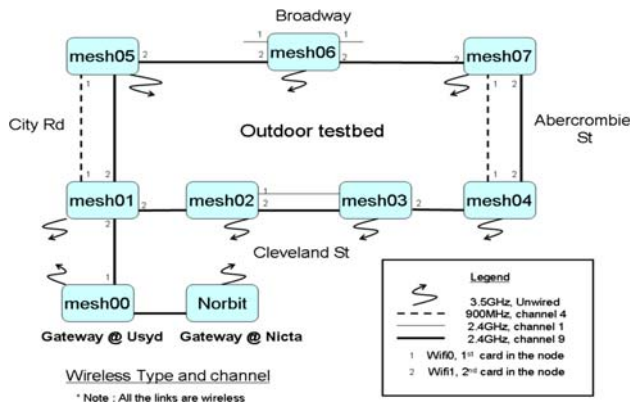


Fig. 2 Layout of NICTA's outdoor testbed in Sydney

The major purpose of NICTA's wireless mesh testbed is to explore the technical feasibility and issues for a city-wide network used for control and monitoring applications such as, traffic signal control. Such a network requires high reliability although the typical traffic consists of packets with only a few bytes and bandwidth requirements are not as stringent as in a public access network. Based on this, our initial experiments used low volume data with small packets sizes. We used ICMP ping packets to simulate the situation of a control network. In subsequent experiments we used a traffic generator to provide a rich mix of flows and more diverse scenarios.

5.1 Experiments with low traffic volume

Table 1 summarizes the initial ping based experiments. Experiments with diverse traffic flows are discussed in the next section. The column for normal traffic in Table 1 shows the flows in the form of "OD, (interval between successive pings in seconds, start time bin)", where OD denotes origin and destination and 1 start bin refers to the start of the experiments and each time bin is 1 min long. Note that in OD description, 1 refers to mesh01, 2 refers to mesh02, and so forth. All experiments ran over 2 h and traffic data was collected from mesh01. The observation window is also set at 2 h for these experiments. Experiment 2 used Unwired^{TM} links where the rest of the experiments used 802.11 b/g links. The confidence level, i.e., $1 - \alpha$ was set to 95% for all experiments. The threshold based on the Chebyshev

Table 1 Experiments with ping packets only

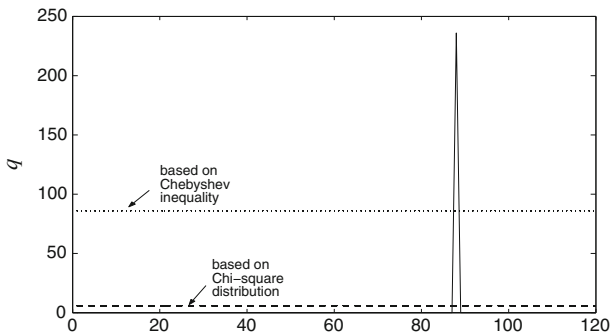
	Normal traffic OD (ping interval(s), start bin)	Anomaly Nodes (type, start bin)	Wireless type
1	01 (4, 1), 21 (10, 1), 51 (30, 30)	4 to 1 (ping flood, 88)	WiFi
2	01 (4, 1), 21 (10, 1), 51 (30, 30)	3 to 1 (ping flood, 83–84)	Unwired^{TM}
3	10 (4, 1), 12 (10, 1), 15 (30, 1)	2 (node failure, 106–109)	WiFi

inequality was taken as 4σ, where σ denotes the standard deviation of q or t^2 statistics. In order to select significant flows, the r_q components with largest variance, i.e., σ_i^2, $i = 1, ..., r_q$, contributing not less than 95% of the aggregated variances were selected for calculating q statistics. Similarly, r_t components with the largest eigenvalues, i.e., λ_i, $i = 1, ..., r_t$, adding up to at least 95% of the aggregate eigenvalues were selected to calculate t^2 statistics.

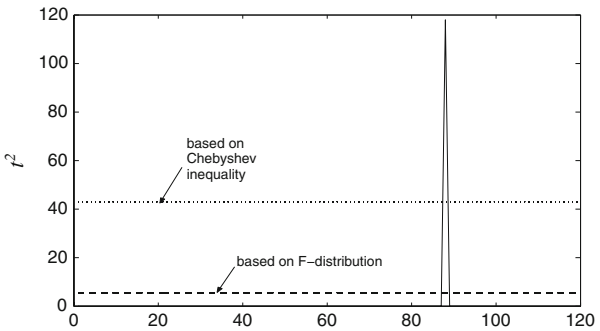
Figure 3a and b show q and t^2 values for experiment 1 respectively. The horizontal dotted line in both figures shows the alternative threshold, given in (7), based on Chebyshev's inequality. The dashed line is δ_q for q statistics and δ_t for t^2 statistics calculated for a 95% confidence level. The peak at 88th time bin in both figures refers to the ping flood. The anomaly identification scheme indicated that approximately 639 additional packets, with respect to the mean flow rate, were being transmitted between nodes mesh01 and mesh04 each way. The identification scheme calculated the relative contribution in different ways for q and t^2 statistics. Even though the calculations were different, both methods represent deterministic mappings and yield similar performance once a time bin is detected to contain an anomaly. Both methods, with both thresholds, yielded similar detection results for this experiment without false alarms.

A similar experiment was repeated over Unwired^{TM} links and we observed symmetrical results. In experiment 2, our identification scheme counted an aggregated number of 1594 excess packets for the ping flood detected at time bins of 83 and 84 between mesh01 and mesh03. We identified approximately 919 excess packets transmitted from mesh03 to mesh01. Unlike experiment 1 where both directions contributed equal numbers of packets, this shows that mesh03 is the most likely cause of the anomaly rather than mesh01. In experiment 1, an attacker could be either of the two nodes.

Figure 4a and b show the q and t^2 statistics for experiment 3. Also in this case, both methods perform in a similar manner. However, as shown in Fig. 4a and b, the alternative threshold based on Chebyshev's inequality is not able to detect the anomaly as it constitutes a loose upper-bound. The node failure is detected at time bins 106–109 as shown in Fig. 4a and b. For the detected anomaly at 106–109, our identification scheme shows the deficiency of approximately 6 packets in flows between mesh01 and mesh02, in both directions, when compared against the mean flow rates. The negative contribution values from the identification scheme serve as indicators for link or node outages. The second peak in Fig. 4a and b is identified as a false alarm. This was actually an attempt to establish a ssh

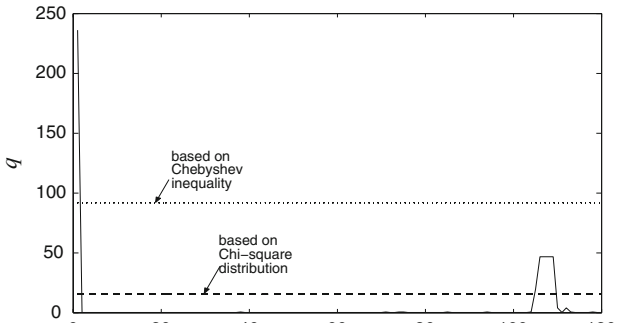


(a) q statistics.

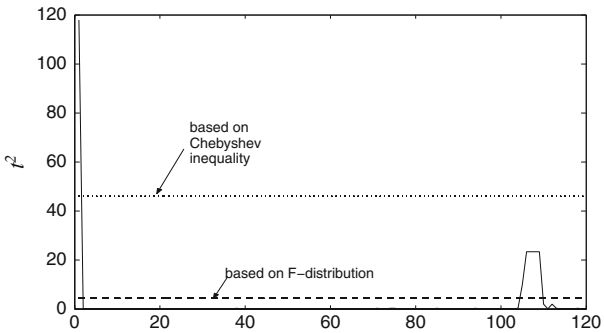


(b) t^2 statistics.

Fig. 3 Anomaly detection for experiment 1



(a) q statistics.



(b) t^2 statistics.

Fig. 4 Anomaly detection for experiment 3

connection from a remote computer to mesh01 through the gateway, i.e., mesh00. All together 75 excess packets are counted for the false alarm peak due to the ssh attempt.

5.2 Experiments with traffic generator

In order to create more interesting scenarios for experimentation, we used a traffic generator [11] which generates self-similar traffic flows according to an on-off model. Based on studies in Ridoux et al. [23], this traffic generator provided us with realistic IP traffic typical for a wireless LAN. The following parameters are tunable for the traffic generator:

1. a_1 = session arrival rate (no. of sessions/sec)
2. a_2 = in-session packet arrival rate (no. of packets/sec for each session)
3. a_3 = session duration parameter (sec)
4. a_4 = Hurst parameter
5. a_5 = service time distribution (no. of packets/sec)

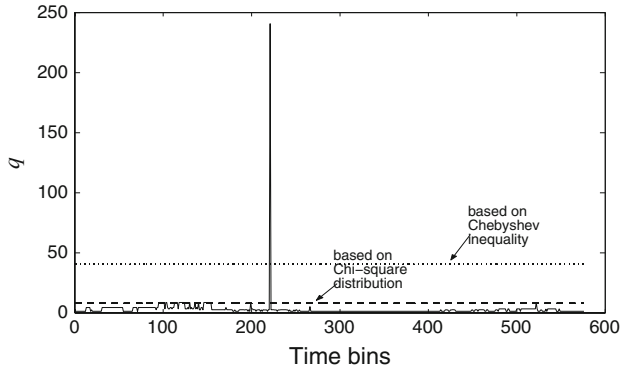
Table 2 summarizes three 24 h long experiments. In experiments 1 and 2, two classes of traffic are generated from mesh06 for mesh05 and mesh04. The parenthesis in the second column contains the parameter settings for the traffic generator for each class. Data was collected at mesh05. Experiments 1 and 2 used 5 min time bins. As before, the thresholds for q and t^2 statistics were calculated for a 95% confidence level for all experiments. The threshold based on Chebyshev inequality is set to 4σ, where σ is the standard deviation of the q or t^2 statistics. In order to select significant flows, the r_q components with largest variances, i.e., σ_i^2, $i = 1, ..., r_q$, contributing not less than 95% of the aggregated variances were selected for calculating q statistics. Similarly, the r_t components with largest eigenvalues, i.e., λ_i, $i = 1, ..., r_t$, adding up to at least 95% of the aggregate eigenvalues were selected to calculate the t^2 statistics. We introduced a single anomaly of port scan in experiment 1, and experiment 2 contained a natural link outage at node 5 as shown in Table 2. Table 2 shows the detection result when the observation window is set to the experiment length, i.e., 24 h. In the next section, we show the effects of changing the length of the observation window.

Figure 5a and b show the q and t^2 statistics for experiment 1 when data matrix $\mathcal{X}$ contains the flow count for each time bin sorted according to the sources. The port scan clearly results in a sharp peak in Fig. 5a and b at the 221st time bin. Flow count data without sorting also resulted in positive detection of the port scan anomaly but it was unable to pinpoint the node causing it. The identification analysis with flow counts sorted according to the sources was able to identify mesh06 with approximately 8 additional flows when compared against the mean trend. This anomaly is not visible in packet and byte count analysis. As shown in Table 2 and Fig. 5a, q statistics also results in couple of false alarms. The identification scheme shows a deficiency of 1 or 2 flow(s) from each source causing false alarms in the q statistics. Interestingly, PCA breaks flows into eigenflows which sometimes helps in filtering small perturbations as shown in Fig. 5b.

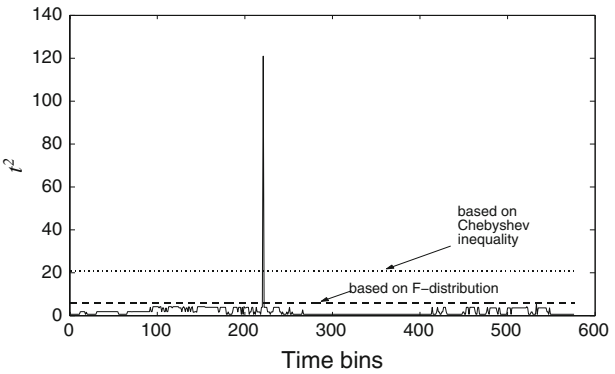
Figure 6 shows the q statistics for experiment 2 when the traffic matrix contains flow count for each time bin. In this case q and t^2 statistics are essentially the same as $\mathcal{X}$ only containing one column comprised of the total flow count in the network. Figure 6 shows the detection of link outage at the 234th time bin as the identification scheme yields a deficiency of approximately 4 flows with respect to the mean flow count. When flows are sorted according to the sources, we are able to identify the failed node as being mesh05. In this experiment, packet and byte count analysis do not yield detection of link outage. The threshold δ_q with 95% confidence level also yields a couple of false alarms which are

Table 2 Experiments with traffic generator observation window = 24 h, threshold = δ_q or δ_t

	Normal traffic	Anomaly	Detection		False alarms	
	OD (a_1, a_2, a_3, a_4, a_5)	Nodes (type, start bin)	q	t^2	q	t^2
1	65 (5, 50, 1, 0.9, 2500.0), 64 (50, 5, 1, 0.9, 2500.0)	6 to 5 (port scan, 221)	✔	✔	8	0
2	65 (5, 50, 1, 0.9, 2500.0), 64 (50, 5, 1, 0.9, 2500.0)	5 (link outage, 234)	✔	✔	5	5
3	15 (1, 1, 1, 0.9, 2500.0), 13 (1, 2, 1, 0.9, 2500.0)	0 to 2 (ping flood, 594),	✔	✔	6	5
		0 to all (node scan, 605–606),	×	×		
		0 to 1 (UDP DoS, 920–922),	✔	✔		
		0 to 1 and 2 (port scan, 940–941)	✔	✔		
		2 to 3 (port scan, 1211–1228),	✔	✔		
		0 to 5 (port scan, 1235),	✔	✔		



(a) q statistics.



(b) t^2 statistics.

Fig. 5 Anomaly detection (flow count) for experiment 1 using traffic generator

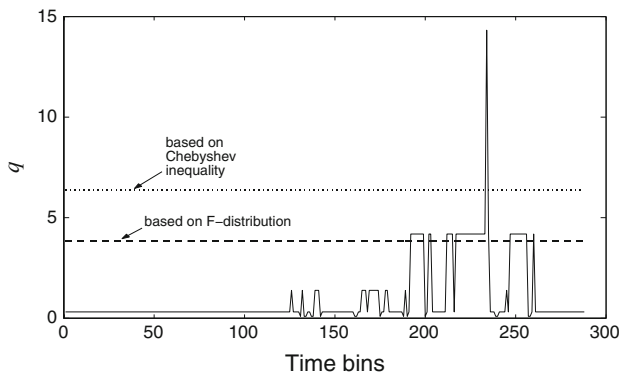
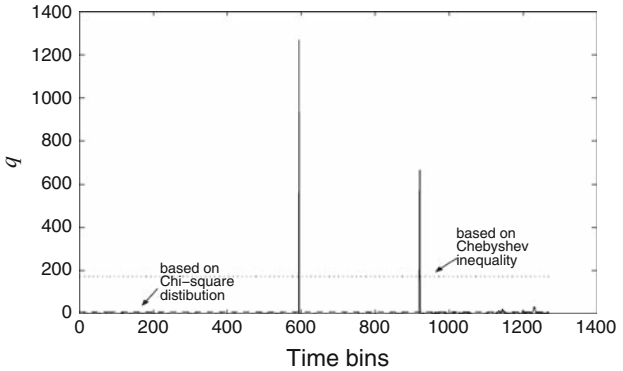


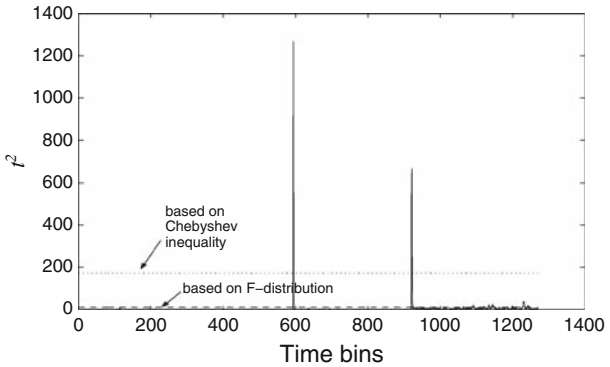
Fig. 6 q statistics (flow count) for experiment 2 using traffic generator

caused by the deficiency of 2 flows as identified by our identification scheme. Thresholds for 99% confidence level and an alternative threshold based on Chebyshev inequality yielded 0 false alarms for experiments 1 and 2.

Experiment 3 used mesh01 as source and mesh03 and mesh05 as destinations for two classes of traffic as shown in Table 2. Each time bin was 1 min long and data was collected at mesh01. We introduced a range of anomalies



(a) q statistics.



(b) t^2 statistics.

Fig. 7 Anomaly detection (packet count) for experiment 3 using traffic generator

in this experiment as shown in Table 2. Figure 7a and b show the q and t^2 statistics when the traffic matrix contained packet count for each time bin and the observation window length was 24 h. The analysis with byte count yielded similar result as for packet count. It is clear from both figures and Table 2 that q and t^2 statistics with 95% confidence threshold yields similar results in terms of detections and false alarms. Table 3 elaborates the detection and identification results of experiment 3 using packet and flow counts. Anomalies associated with port scan may not show up in the packet count as they induce a small number of packets in the networks. Although, the q and t^2 analyses using flow count, where flows are distinguished according to the IP address and port numbers, are successful in detecting these anomalies as shown in Table 3. However, the node scan at 605–606 could not be detected by either method. In a small testbed network of seven nodes, node scan does not produce a significant number of packets and flows, and could therefore not be detected by either method. Sorting flows according to sources makes it possible to identify the attacker in most cases.

Ping flood and DoS attacks were detected by q and t^2 analyses using packet count and the attackers were correctly identified. Packet count analysis yielded 5–6 false

Table 3 Identification results of experiment 3 (with traffic generator)

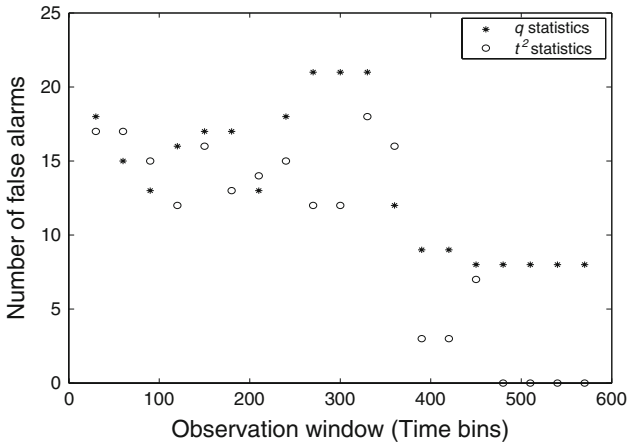
Time bin	Contribution	Responsible nodes	Anomaly	Traffic matrix
594	2 to 0 (2758 packets), 2 to 0 to 2 (3287 packets)	0	Ping flood	Packet count
605–606	Not detected		Node scan	
920–922	0 to 1 (19641 total packets)	0	DoS	Packet/flow count
941	209 flows	0	Port scan	Flow count
1211–1228	13 flows	2	Port scan	Flow count
1235	103 flows	5	Port scan	Flow count
5–6 False alarms	1 to 3 (UDP flow), over 1000 packets			Packet count

alarms, using t^2 statistics, due to changes in normal flows, i.e., from mesh01 to mesh03. Some of these changes happened due to the bursty nature of self-similar traffic. In practise, once a flow is found to be legitimate, any subsequent detections could be readily classified as false alarms.

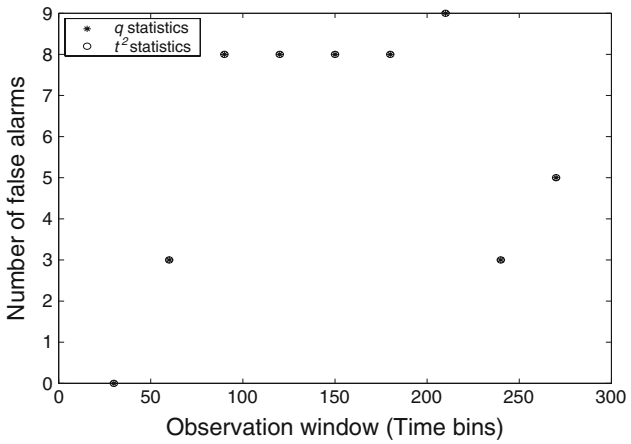
5.3 Observation window

To study the effects of the observation window, we analyzed the 24 h experiments with variable window lengths. After a specified interval (30 time bins in our analysis) q and t^2 statistics were calculated for the last L observations, where L is the observation window length in terms of time bins. In our study, L = 30, 60, 90, ... Packet and flow count data collected for 24 h, from experiment 1–3, was processed as if the detection modules were running in real-time. Figure 8a and b show the false alarms for various observation windows for experiment 1 and 2 respectively. Both experiments contain a single anomaly which was detected for all observation window lengths using both methods. Experiment 2 used flow count data and X contained a single column, therefore q and t^2 statistics were equivalent and yielded the same result as shown in Fig. 8b. In general, a larger observation window results in fewer false alarms as shown in Fig. 8a and b. Similar to Fig. 5, q statistics performs in a similar manner to t^2 statistics, although for larger windows, t^2 yields fewer false alarms. Larger observation windows are desirable to capture normal trends more accurately albeit at the cost of storage space. Since the observation window uses the last L observations after a specific interval, a larger L does not mean later detection. As long as the detection algorithm is processed at a reasonable frequency, network anomalies can be detected quickly.

Figure 9a and b show the detection performance and number of false alarms using q and t^2 analysis for packet count data from experiment 3. We have labeled anomalies DoS or scan anomalies, i.e., port scan and node scan. Experiment 3 contains 2 DoS or flood anomalies and 4 scan



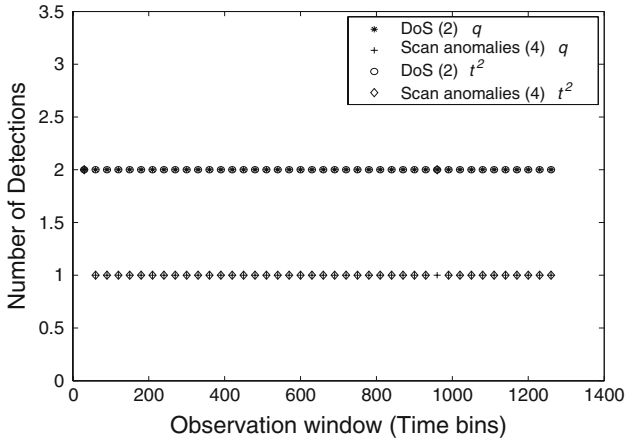
(a) Experiment 1 (traffic generator).



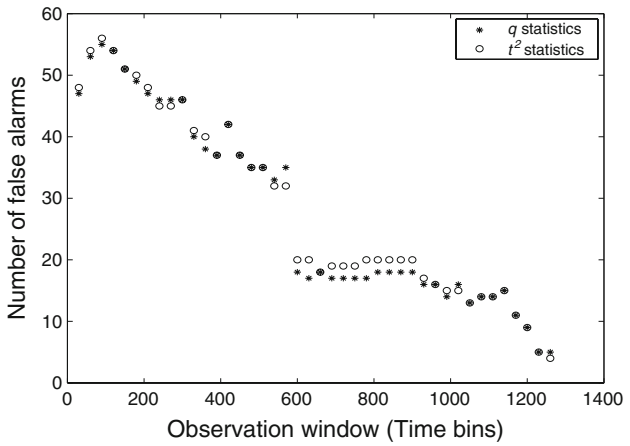
(b) Experiment 2 (traffic generator).

Fig. 8 False alarms vs. observation window for experiment 1 and 2 (flow count). Anomalies in both experiments were detected for every observation window

anomalies. As discussed above, flood anomalies should be detectable in packet count, where as scan anomalies may or may not appear in packet count analysis depending on the number of packets generated in these anomalies. As shown in Fig. 9a, q and t^2 statistics are able to detect all flood

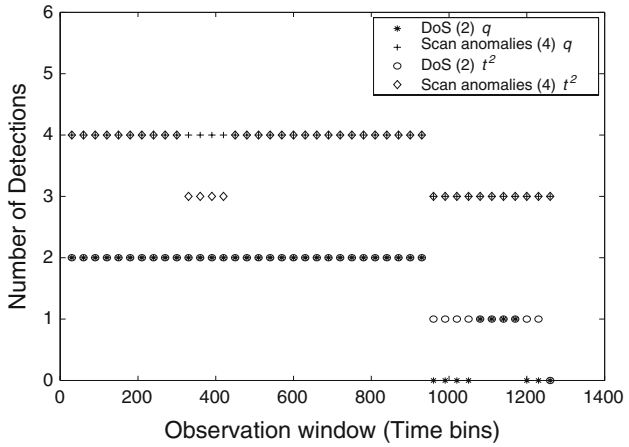


(a) Detections.

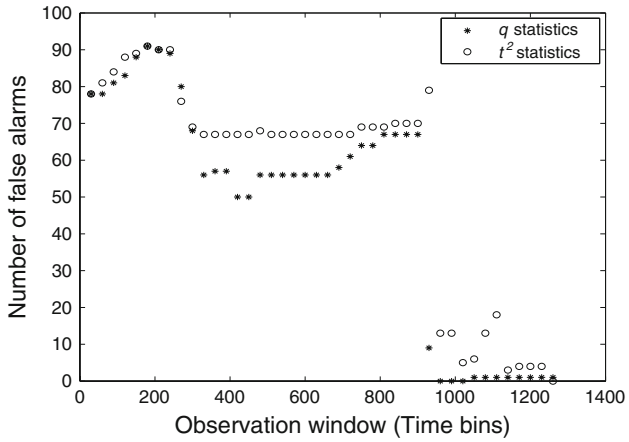


(b) False alarms.

Fig. 9 Anomaly detection vs. observation window for experiment 3 (packet count) using traffic generator



(a) Detections.



(b) False alarms.

Fig. 10 Anomaly detection vs. observation window for experiment 3 (flow count) using traffic generator

anomalies for all observation windows besides detecting 1 or 2 scan anomalies. Both algorithms exhibit similar performance also in terms of false alarms as shown in Fig. 9b, where either one has slightly more false alarms than another for some observation window size. As discussed before, in general, a larger observation window yields better performance.

However, flow count data for experiment 3 shows missed detection for a larger observation window using both methods as shown in Fig. 10a. The missed anomaly is the node scan at the 605–606 time bins. Interestingly, when the observation window is larger, normal changes of few flows cause the node scan to appear as normal. In a small testbed of 7 nodes, a node scan only produced few additional flows. False alarm performance, however, is better for a larger observation window using both methods as shown in Fig. 10b. A larger observation window requires more memory space which in some instances is a limiting factor. As natural variations in traffic from the mean trend increase in amplitude, the noise floor of the anomaly detection algorithms below which anomalies go undetected is also raised. As shown in numerous studies [9], traffic in public access networks changes considerably in 24 h. A larger observation window would encompass all changes from very light to very heavy traffic conditions and would never be able to detect anomalies in lighter traffic scenarios, though they might be significant. To study the effects of light and heavy traffic, it is important to get data from a real public access network. Moreover, both methods have similar performance although the Chi-square test requires a fraction of the computations compared with the PCA-based method.

5.4 Comparison with other statistical methods

Among the anomaly detection methods discussed in Sect. 2, the Kolmogorov-Smirnov test [10] is the only suitable method to be implemented on our experimental data. Since wavelets are proposed to detect congestion [10] and frequency distributions of packet contents are used to

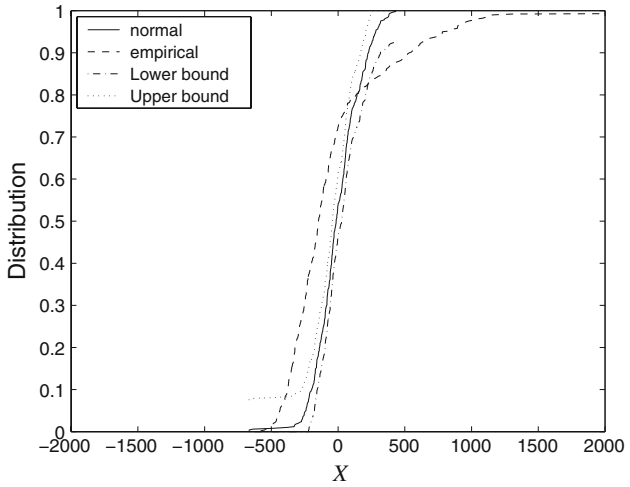


Fig. 11 Kolmogorov-Smirnov Test: Normal and empirical distributions for normalized packet count of experiment 3 (using traffic generator)

detect worm attacks [14], both of the methods were beyond the scope of this paper. Moreover, our experiments were not done for a long enough period to get meaningful training of the signature library of [4] and the median graph of [3]. The Kolmogorov-Smirnov (KS) test [1] compares empirical distribution of observed data with that of normal data and deviations exceeding a threshold are classified as anomalous. We used the first 500 time bins of experiment 3 to generate the normal data distribution as shown by the solid line in Fig. 11. The first 500 min of experiment 3 did not contain any anomaly.

The upper and lower bounds shown by dotted and dot-dash lines respectively in Fig. 11 were drawn for 99% confidence level using KS statistics tables. The dashed line in Fig. 11 shows the empirical distribution which is generated using the remaining data from experiment 3 after the first 500 time bins. According to the KS test, if the empirical distribution is not contained within the upper and lower bounds of the normal distribution, the data contains an anomaly or anomalies as shown in Fig. 11. The KS test is able to detect the anomalous traffic but it is not able to identify the cause of the anomaly and exact time instant when an anomaly happens. The Chi-square test as well as the PCA-based method has much higher potential to detect and identify various anomalies in real-time without significant computational load.

6 Conclusion

In this paper, we presented a computationally economical substitute of the PCA-based method of anomaly detection using a Chi-square test. Our experiments with real-time anomaly detection systems using the Chi-square and the PCA-based methods show that both schemes are very effective in detecting various types of anomalies including DoS and port scan attacks in an interference prone wireless networking environment. The Chi-square test requires a fraction of the computations required by the PCA-based test although their performance is comparable in our experiments. Our experiments used an outdoor testbed with different types of traffic flows.

We also proposed an automatic identification scheme which can be used with both algorithms to identify the cause of a detected anomaly and also to estimate the contribution of each responsible node in terms of excess or loss of traffic. The identification scheme is also useful in isolating false alarms from real anomalies. We have also proposed an alternative threshold to compare test statistics, if data is found to be heavy tailed which is a typical case for self-similar traffic. An alternative threshold based on the Chebyshev inequality constitutes a loose upper bound and may result in missed detections.

We have also compared the real-time anomaly detection using the Chi-square test and the PCA-based method with other statistical tools proposed in the literature. Both methods are computationally feasible to run in real-time and also have the potential of identifying the responsible nodes. Both methods are very promising to be used in a WMN environment where the Chi-square test is also more economical in terms of computational load.

Acknowledgments The authors want to acknowledge the help of Mr. Rodney Berriman and Dr. Mohsin Iftikhar in setting up experiments.

References

1. Caberera, J. B. D., Ravichandran, B., & Mehra, R. K. (2000). Statistical traffic modeling for network intrusion detection. In *Proceedings of the IEEE MASCOTS, 2000* (pp. 466–473).
2. Chen, T., Kuo, G.-S., Li, Z.-P., & Zhu, G.-M. (2007). Intrusion detection in wireless mesh networks. In *Security in wireless mesh networks*. Boca Raton: CRC Press.
3. Dickinson, P., Bunke, H., Dadej, A., & Kraetzl, M. (2002). Median graphs and anomalous change detection in communication networks. In *Proceedings of the IEEE information, decision and control, 2002* (pp. 59–64).
4. Feather, F., Siewiorek, D., & Maxion, R. (1993). Fault detection in an ethernet network using anomaly signature matching. In *Proceedings of the ACM SIGCOMM 1993* (pp. 279–288).
5. Frenk, H., Roos, K., Terlaky, T., & Zhang, S. (1999). *High performance optimization*. New York: Springer.
6. Fukunaga, K. (1972). *Introduction to statistical pattern recognition*. New York: Academic Press.
7. Gupta, D., Chuah, C.-N., & Mohapatra, P. (2008). Efficient monitoring in wireless mesh networks: Overheads and accuracy trade-offs. In *Proceedings of the IEEE MASS 2008* (pp. 13–23).
8. Hakami, S., Zaidi, Z. R., Landfeldt, B., & Moors, T. (2008). Detection and identification of anomalies in wireless mesh

networks using Principal Component Analysis (PCA). In *Proceedings of the IEEE I-SPAN 2008* (pp. 266–271).
9. Hohn, N. (2004). *Measuring understanding and modelling internet traffic*. Ph.D. thesis in Electrical and Electronic Engineering, The University of Melbourne.
10. Huang, P., Feldmann, A., & Willinger, W. (2001). A non-intrusive, wavelet-based approach to detecting network performance problems. In *Proceedings of internet measurement workshop, 2001* (pp. 213–227).
11. Iftikhar, M., Landfeldt, B., & Caglar, M. (2006). Multiclass G/M/1 Queueing system with self-similar input and non-preemptive priority. In *Proceedings of the IEEE ICI-06*.
12. Ishmael, J., Bury, S., Pezaros, D., & Race, N. (2008). Deploying rural community wireless mesh networks. *IEEE Internet Computing, 12*(4), 22–29.
13. Jackson, J. E. (1991). *A user's guide to principal components*. New York, NY: Wiley.
14. Karamcheti, V., Geiger, D., Kedem, Z., & Muthukrishnan, S. (2005). Detecting malicious network traffic using inverse distributions of packet contents. In *Proceedings of the ACM SIGCOMM workshop MineNet 2005* (pp. 165–170).
15. Lakhina, A., Crovella, M., & Diot, C. (2004). Characterization of network-wide anomalies in traffic flows—Technical report BUCS-2004-020, Boston University.
16. Lakhina, A., Crovella, M., & Diot, C. (2004). Diagnosing network-wide traffic anomalies. In *ACM SIGCOMM 2004* (pp. 219–230).
17. Lakhina, A., Papagiannaki, K., Crovella, M., Diot, C., Kolaczyk, E., & Taft, N. (2004). Structural analysis of network traffic flows. In *Proceedings of the ACM SIGMETRICS 2004* (pp. 61–72).
18. Lan, K., Wang, Z., Berriman, R., Moors, T., Hassan, M., Libman, L., et al. (2007). Implementation of a wireless mesh network testbed for traffic control. In *Proceedings of the IEEE WiMAN 2007* (pp. 1022–1027).
19. Li, N., Chen, G., & Zhao, M. (2008). Autonomic fault management for wireless mesh networks—UMass Lowell technical report 2008–04.
20. Marti, S., Giuli, T. J., Lai, K., & Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of MOBICOM '00* (pp. 255–265).
21. MIT Roofnet. URL- http://www.pdos.csail.mit.edu/roofnet/doku.ph.
22. Qiu, L., Bahl, P., Rao, A., Zhou, L. (2006). Troubleshooting wireless mesh networks. *SIGCOMM Computer Communication Review, 36*(5), 17–28
23. Ridoux, J., Nucci, A., & Veitch, D. (2006). Seeing the difference in IP traffic: Wireless versus wireline. In *Proceedings of IEEE INFOCOM '06* (pp. 1–12).
24. Salem, N. B., & Hubaux, J.-P. (2006) Securing wireless mesh networks. *IEEE Wireless Communications, 13*(2), 50–55.
25. Sarafijanovic, S., & Boudec, J. Y. L. (2005). An artificial immune system approach with secondary response for misbehavior detection in mobile ad hoc networks. *IEEE Trans. on Neural Networks, 16*(5), 1076–1087.
26. Siddiqui, M. S., & Hong, C. S. (2007). Security issues in wireless mesh networks. In *Proceedings of IEEE international conference on multimedia and ubiquitous engineering (MUE) 2007* (pp.717–722).
27. Ye, N., & Chen, Q. (2001). An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Wiley Quality and Reliability Engineering International, 17*, 105–112.
28. Zaidi, Z. R., Landfeldt, B., & Zomaya, A. (2007). Fault management in wireless mesh networks. In *Handbook on ad hoc and mobile computing*. Valencia, CA, USA: American Scientific Publishers.

Author Biographies

Zainab R. Zaidi did Ph.D. and MS in Electrical Engineering (EE) from George Mason University, Virginia, USA, in 2004 and 1999 respectively and BS in EE from NED University of Engineering & Technology, Karachi, Pakistan, in 1997. Currently, she is a researcher in Networked Systems group in NICTA. Before joining NICTA in 2006, she taught in NED University of Engineering & Technology, Karachi, Pakistan, in 2005 and worked as a post doctoral fellow in Network Architecture Lab of George Mason University, Virginia, USA, in 2004. Her research interests include different network layer issues in wireless mesh networks, such as, robust routing, efficiency of link metric, quality-of-service, etc. besides mobility tracking and its applications in mobile wireless networks.

Sara Hakami received her Bachelors degree in Electrical Engineering and Electronics in 2005 from Shahid Beheshti University in Tehran, Iran. She has also completed her Masters degree in Telecommunications in 2007 at the University of New South Wales (UNSW), Sydney, Australia. Since 2008 she has been working towards her Ph.D. in UNSW and in cooperation with the Networked systems group in National ICT of Australia (NICTA). Her research interests include security in wireless mesh networks, content delivery networks and web pref-fetching.

Bjorn Landfeldt started his studies at the Royal Institute of Technology in Sweden. After receiving a B.Sc. equiv, he continued studying at The University of New South Wales where he received his Ph.D. in year 2000. In parallel with his studies in Sweden he was running a mobile computing consultancy company and after his studies he joined Ericsson Research in Stockholm as a Senior Researcher where he worked on mobility management and QoS issues. In November 2001, Dr. Landfeldt took up a position as a CISCO Senior lecturer in Internet Technologies at the University of Sydney with the School of Electrical and Information Engineering and the School of Information Technologies. Dr. Landfeldt has been awarded 8 patents in the US and globally. He has published more than 60 publications in international books, journals and conferences and been awarded many competitive grants.

Dr. Landfeldt is also a research associate of National ICT Australia (NICTA) and the Smart Internet CRC. Currently, he is serving on the editorial boards of international journals and as a program committee member of many international conferences and is supervising 8 Ph.D. students. Dr. Landfeldt's research interests include; wireless systems, systems modeling, mobility management, QoS and service provisioning.

Tim Moors is a Senior Lecturer in the School of Electrical Engineering and Telecommunications at the University of New South Wales, in Sydney, Australia. His research focuses on improving the reliability of communication networks, by ways that include developing tools for troubleshooting network faults, building overlay networks that can route around failed links, and creating multipath routing protocols for wireless mesh networks. Previously, he was with the Center for Advanced Technology in Telecommunications at Polytechnic University in New York, and prior to that, with the Communications Division of the Australian Defence Science and Technology Organisation. He received his Ph.D. and BEng (Hons) degrees from universities in Western Australia (Curtin and UWA).