

Research Article

Context-Based Topology Control for Wireless Mesh Networks

Pragasen Mudali and Matthew Olusegun Adigun

University of Zululand, Richards Bay, South Africa

Correspondence should be addressed to Pragasen Mudali; mudalip@unizulu.ac.za

Received 26 May 2016; Revised 9 September 2016; Accepted 21 September 2016

Academic Editor: Juan A. Gomez-Pulido

Copyright © 2016 P. Mudali and M. O. Adigun. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Topology Control has been shown to provide several benefits to wireless ad hoc and mesh networks. However these benefits have largely been demonstrated using simulation-based evaluations. In this paper, we demonstrate the negative impact that the PlainTC Topology Control prototype has on topology stability. This instability is found to be caused by the large number of transceiver power adjustments undertaken by the prototype. A context-based solution is offered to reduce the number of transceiver power adjustments undertaken without sacrificing the cumulative transceiver power savings and spatial reuse advantages gained from employing Topology Control in an infrastructure wireless mesh network. We propose the context-based PlainTC+ prototype and show that incorporating context information in the transceiver power adjustment process significantly reduces topology instability. In addition, improvements to network performance arising from the improved topology stability are also observed. Future plans to add real-time context-awareness to PlainTC+ will have the scheme being prototyped in a software-defined wireless mesh network test-bed being planned.

1. Introduction

Wireless mesh networks (WMNs) are increasingly used both as an inexpensive alternative to broadband provisioning in urban areas and as a primary method for broadband provisioning in rural areas. The most common form of WMN deployment consists of a two-tier architecture comprising an access and a backbone network. This type of WMN is commonly referred to as an Infrastructure WMN (I-WMN). Client devices connect to the I-WMN backbone which is typically self-organising and self-configuring. These backbone nodes, comprising Mesh Points (MPs), Mesh Access Points (MAPs), and Mesh Portals (MPPs), collaborate to maintain network connectivity and deliver traffic to the intended destinations (see Figure 1).

Despite the stationary nature of the I-WMN backbone, maintaining network connectivity is made difficult by the transient nature of wireless links, making them unreliable when deployed [1–3]. Network connectivity is traditionally assured by ensuring that each device in the I-WMN backbone utilises its maximum transceiver power. The use of maximum transceiver power is disadvantageous, resulting in high levels of interference, increased contention for the shared

transmission medium, a reduction in network capacity, and unnecessary transceiver power consumption.

As a result of the inefficiencies associated with maximum power consumption in ad hoc networks, several Topology Control (TC) schemes have been developed that can be applied to the WMN backbone in order to maintain network connectivity whilst reducing interference, enhancing the network capacity, and reducing transceiver power consumption. Within the context of TC, power consumption usually refers to the power consumed by a node's wireless transceiver. Power consumed by the wireless transceiver is reported to account for between 15% and 35% of the total energy consumed by the device [4]. TC aims to enhance the QoS capabilities of the WMN backbone by optimizing the transceiver powers of all backbone devices whilst maintaining network connectivity. Distributed TC, however, does have the potential to add to the inherent variations experienced by wireless links by dynamically changing nodes' transceiver power levels. This may cause link quality to exhibit higher-than-normal variation or links may disappear altogether.

Several simulation studies [5–8] have demonstrated the efficacy of TC in ad hoc networks whilst the study of TC prototypes for I-WMNs remains rare. It is important to fill

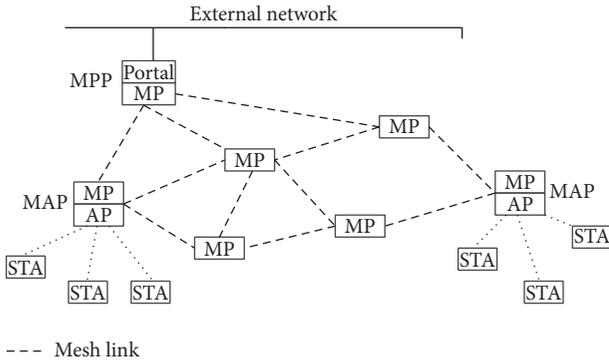


FIGURE 1: Infrastructure WMN architecture [13].

this gap in the literature since there is often a significant discrepancy between what models and simulations predict and the actual performance experienced in a deployed mesh network. TC implementation for the laptop [9] and sensor [10] platforms is available but these devices are not typical I-WMN backbone nodes. A study reported in [11] used a commercially available wireless router platform, but these routers were arranged in a string topology which is an unrealistic topology for the mesh networking use cases described in [12]. The study also reported that the transceiver powers of the participant nodes were manually adjusted.

A TC prototype based upon a popular hardware platform for I-WMN backbone nodes was reported in [14]. The prototype platform was also used as the basis for experimentation with various instances of the Critical Number of Neighbours connectivity strategy in [15]. The prototype, dubbed PlainTC, was found to be capable of maintaining network connectivity whilst achieving cumulative transceiver power savings.

Wireless network topologies experience inherent variations in link quality and the sudden breaking and reestablishment of links when deployed. This leads to temporary fluctuations in network performance as routing protocols and other QoS mechanisms adjust to the new network topology. These natural variations in link quality are difficult to eliminate and must be separated from the topology instability caused by varying the wireless transceiver power of a node as directed by a TC scheme. In the context of this paper, topology instability refers to the forced changes in the network topology caused by a TC scheme.

In this paper, we pay further attention to the PlainTC prototype. We first demonstrate the negative impact that the PlainTC prototype has on topology stability. To the best of the authors' knowledge this is the first instance of such an observation for a TC scheme. This observation is made possible by the use of a test-bed prototype and would have been prohibitively difficult to make with a simulation tool due to the tool's high levels of abstraction. This paper then proposes a context-based solution to reduce the topology instability caused by PlainTC. This solution allows for changes in a node's context to be identified and the quantum of context change to be computed. The quantum of context change is then used to regulate, using a context-change threshold, the adjustment of a node's transceiver power output.

The evaluation of PlainTC+ on our indoor I-WMN test-bed indicates that the incorporation of context information produces a 45% reduction in the number of transceiver power changes when compared to the original PlainTC scheme. In addition, the reductions in the oscillations of observed neighbourhood size and link quality are indicators of reduced topology instability. The reduction in topology instability is also shown to improve the PDRs and throughputs being measured. These improvements are achieved without significant increases in the computational resources being required.

The remainder of this paper is organised as follows. Section 2 reviews earlier TC prototypes that can be considered applicable for an I-WMN. This section also reviews earlier attempts to address topology instability in I-WMNs. Section 3 demonstrates the negative impact of TC on topology instability. A context-based TC prototype is presented in Section 4 and our indoor I-WMN test-bed environment is described in Section 5. The test-bed evaluation of the PlainTC+ prototype is contained in Section 6 and the paper is concluded in Section 7. Section 7 also describes future work that would have PlainTC+ being prototyped in a software-defined I-WMN test-bed.

2. Literature Review

This section reviews existing TC prototypes that are applicable to I-WMNs. These prototypes are discussed in terms of their usage of context information and their control of topology instability. In addition, this section also reviews other attempts to reduce topology instability in I-WMNs.

2.1. Topology Control Prototypes Applicable to Infrastructure Wireless Mesh Networks. The overwhelming majority of TC schemes in the literature have only been subjected to simulation-based evaluations and the reliance on this evaluation type has provided an idealised version of the efficacy of TC in wireless ad hoc networks. These TC schemes have not easily resulted in prototypes, forcing researchers to adopt simpler, more practical approaches in order to study the efficacy of TC in test-beds or deployed networks.

Numerous sleep-based TC prototypes can be found in the literature but these prototypes are better-suited for wireless sensor networks. Typical I-WMN usage scenarios discussed in [16] require that all backbone nodes remain available, thus rendering sleep-based TC schemes inappropriate. Therefore, only those TC prototypes that do not employ sleep states are considered in this review.

To the best of our knowledge, only five TC prototypes that can be considered for use in an I-WMN appear in the literature. The *COMPOW*, *CLUSTERPOW*, and *MINPOW* schemes were reported in [9]. These schemes create routing tables for each specific power level supported by the wireless cards employed. Each scheme requires the modification of existing packet headers as well as the routing protocol messages, thus resulting in tight coupling between the schemes and the protocol stack. The evaluation of these schemes was limited only to the correctness of the routing tables created since the authors reported that the hardware used crashed

repeatedly. These schemes are not context-aware and exacerbate topology instability due to the near-constant changing of transceiver power levels in order to maintain updated routing tables for each supported power level.

The scheme reported in [11] required the predetermination and subsequent manual tuning of a common transceiver power level, on all network nodes, to ensure network connectivity. This results in a completely static scheme that lacks scalability. The scheme was found to have a negative effect on a Network Layer routing protocol since the protocol performed best when the nodes operated at maximum transceiver power. The evaluation of the scheme was based on a string topology, which is an uncommon network topology for most I-WMN deployment scenarios. This scheme is not considered to be context-aware due to the manual tuning involved and topology instability is not a concern as the nodes are not involved in distributed decision-making.

The *ConTPC* power control scheme was presented in [17]. *ConTPC* attempts to minimise packet loss caused by transceiver power reductions. A transceiver power reduction is only allowed if the reduction would not affect the delivery rate of the wireless link involved. This method ensures that only high-quality links are considered for transceiver power reductions. The mechanism for computing the delivery rate requires that probe packets be periodically sent at every permissible power level. The disadvantages of such an approach are that it adds to the existing communications overhead and may result in unstable network topologies as node power levels are continuously changing in order to send these probe messages at the various power levels. In addition, the sending and receipt of these probe messages are subjected to the prevailing scheduling mechanism which may result in a delayed view of network conditions.

The TC prototypes discussed above may actually exacerbate topology instability by continuously switching between the various supported transceiver power levels. Each change in transceiver power level causes a change in the network topology.

The *PlainTC* prototype presented in [14] attempted to maintain network connectivity whilst producing cumulative transceiver power savings. The prototype employed the Critical Number of Neighbours connectivity strategy [16] to achieve its goal. Nodes iteratively adjusted their transceiver power levels to maintain the required number of one-hop neighbours. This prototype differed from the previous ones by not requiring the broadcast of probe messages at every supported power level and being evaluated on Linksys WRT54GL devices which are commonly used in wireless mesh network deployments. However, despite these differences, *PlainTC* may also exacerbate topology instability due to the transceiver power changes required to maintain the required neighbourhood size.

2.2. Efforts to Control Instability in Infrastructure Wireless Mesh Networks. Instability has been recognised as being detrimental to the QoS offered by an I-WMN. Thus, there are existing efforts to control instability. Instability in the I-WMN is usually discussed from a network routing perspective where link quality fluctuations and frequent route flapping

are observed in deployed networks [18]. This observation led to the proposal of a stability-aware routing protocol in [18] and a stability-aware routing metric can be found in [19].

The work in [18] was motivated by observations of a deployed I-WMN. The network deployment made it easier to observe that the link quality varied often and route flapping was excessive. These observations point to the instability of the underlying network topology. Additional factors leading to network instability are revealed in [19]. These factors include interference, topology and traffic patterns, congestion, and the presence of distributed decision-making agents whose local actions have wider-reaching consequences. These observations in [19] led to the development of a stability-aware routing metric.

Both examples react to the instability of the underlying network topology whereas the TC scheme being proposed in this paper proactively seeks to minimise the topology instability caused by TC in I-WMNs. A demonstration of the topology instability caused by TC in I-WMNs is presented in the next section.

3. Demonstrating the Impact of Topology Control on Topology Instability

The distributed QoS mechanisms employed in wireless ad hoc networks work best in stable network topologies [19]. However, topology instability is a feature of deployed networks. This section practically demonstrates the additional instability introduced by employing a distributed TC scheme in an I-WMN. The basis for this demonstration is the PlainTC prototype described in [14].

3.1. The Side-Effects of Topology Control Exposed by the OLSR Routing Protocol. The OLSR routing protocol is a useful tool to observe the side-effects of a TC prototype's iterative transceiver power adjustment process. These side-effects are exposed by the metrics measured by the OLSR routing protocol and can be thought of as depicting a microlevel view of PlainTC's iterative transceiver power adjustment process. Figure 2 depicts the effect of iterative transceiver power adjustment on the number of neighbours maintained by a node in the network. There is an almost-constant oscillation in the number of neighbours which causes topology instability. Figure 2 only depicts this behaviour for a single node in the test-bed but every node experiences a similar phenomenon.

I-WMN deployments possess some inherent instability due to variations in link quality but this does not account for the almost-constant variations in the number of neighbours being observed via OLSR. The changes in the number of neighbours can only be caused by changes in a node's transceiver power levels and these power levels are under the direct control of PlainTC's transceiver power adjustment process. The effect of the almost-constant variations in the number of neighbours is topology instability.

The topology instability negatively affects the operation of OLSR's ETX routing metric as paths to destinations are affected by the changing ETX values experienced by a node as depicted in Figure 3. Excessive route flapping is the result.

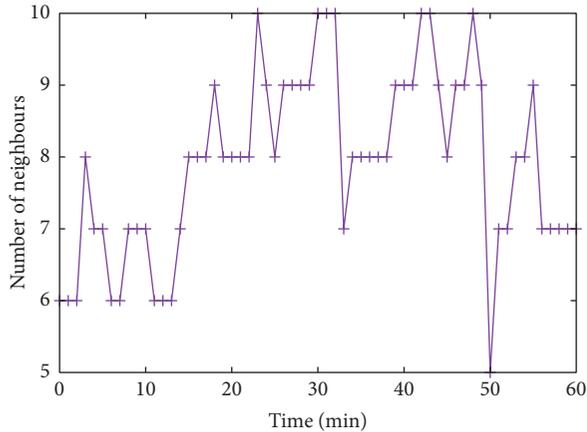


FIGURE 2: Varying neighbourhood size as a result of transceiver power adjustments.

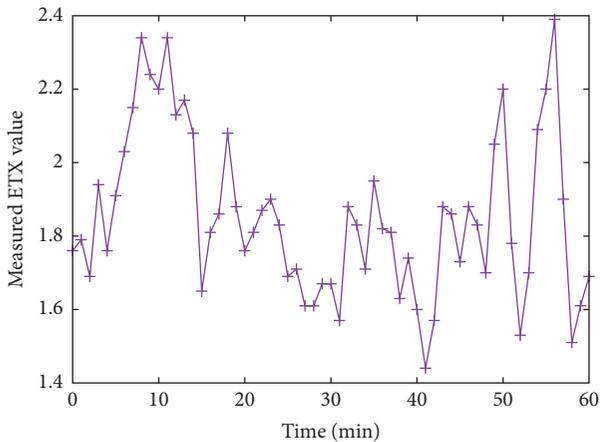


FIGURE 3: Varying link quality as a result of transceiver power adjustments.

Link quality is known to be subjected to natural variations but these variations are exacerbated by PlainTC's iterative transceiver power adjustment process.

3.2. Topology Instability. According to [19], wireless ad hoc networks are inherently unstable due to the high bandwidth demands and dynamic traffic variations. Therefore, this particular experiment was carried out with PlainTC activated but without data traffic being transmitted so as to expose the amount of instability being measured.

The previous experiment has shown that the use of PlainTC, which is based upon iterative transceiver power adjustment, injects instability into the network. The instability caused by PlainTC is explored further and differs from the existing literature in that the instability being observed is caused by a TC scheme rather than being reacted to by a routing protocol as in [18] or a routing metric as found in [19].

This experiment was conducted over a 24-hour period to track the changes that occur in an I-WMN test-bed that employed PlainTC. Readings were collected every second and the number of recorded changes for each context variable was

reported for every hour period. The test-bed is described in Section 5. The experiment logged the number of changes to important topology variables such as the transceiver power output level, neighbourhood size, network size, and link quality. These variables are chosen for their relevance to WMNs and TC but other variables such as speed, altitude, orientation, location coordinates, and available sensors may be more applicable for mobile ad hoc or sensor networks.

The four chosen variables collectively represent the network topology and changes to these variables signify changes to the topology. These variables were collected during the normal operation of a test-bed node.

The number of recorded changes for each context variable was reported across all the network nodes and these changes are presented for twenty-four one-hour intervals. As stated before, the test-bed carried no data traffic as the objective was to explore the global effect of PlainTC's distributed transceiver power adjustment process on topology instability.

The number of changes recorded for the four context variables is recorded in Table 1. The first hour recorded the greatest cumulative number of changes for the four context variables. This is due to the nodes adjusting their individual transceiver power levels (from the initial maximum power level) to satisfy the required connectivity requirements. It would be expected for the subsequent one-hour intervals to show little or no further changes in transceiver power output levels, neighbourhood size, and network size as the network topology should have stabilised within the first hour. This is clearly not the case.

Nodes are found to be constantly adjusting their transceiver power output levels throughout the 24-hour period. This is caused by nodes reacting to the transceiver power changes effected by their neighbours. Changes in transceiver power output levels made by neighbouring nodes caused a change in the neighbourhood size recorded by the affected node. The affected node responded by adjusting its own transceiver power output level. This action, in turn, caused the neighbours of the affected node to adjust their own transceiver power output levels in response. Thus a cycle of transceiver power output level adjustments and counteradjustments took place amongst the backbone nodes.

The cascading effect of changes in the transceiver power output caused changes in the other observed variables. The data contained in Table 1 shows that changes in transceiver power output levels have a multiplier effect on the other observed variables. Consider a node, node *A*, and its immediate neighbours. *A* adjusts its transceiver power output level. This change causes a change in its neighbourhood size (in order to maintain the required CNN) and link quality. The neighbours of *A* may record changes in their own neighbourhood sizes, network sizes, and link quality until the local topology stabilises. Note that *A*'s neighbours have not had to adjust their own transceiver power output levels in order to experience changes to their own observed variables. This example shows that a single change in transceiver power output can cause changes in neighbourhood size, network size, and link quality amongst the originating node and each of its immediate neighbours. Thus, changing the transceiver power

TABLE 1: Number of changes to node variables over a 24-hour period.

Time period	Number of changes in transceiver power	Number of changes in neighbourhood size	Number of changes in network size	Number of changes in link quality
08:00–08:59	107	239	303	364
09:00–09:59	73	102	178	319
10:00–10:59	70	106	231	304
11:00–11:59	71	96	220	295
12:00–12:59	70	104	221	323
13:00–13:59	69	102	225	302
14:00–14:59	67	104	220	303
15:00–15:59	60	87	207	275
16:00–16:59	70	107	230	319
17:00–17:59	68	99	221	317
18:00–18:59	71	97	219	304
19:00–19:59	69	102	223	314
20:00–20:59	70	100	227	303
21:00–21:59	67	97	218	321
22:00–22:59	69	102	229	309
23:00–23:59	76	114	235	375
00:00–00:59	67	93	213	299
01:00–01:59	67	100	223	312
02:00–02:59	63	88	205	281
03:00–03:59	66	91	211	296
04:00–04:59	68	93	212	296
05:00–05:59	73	107	220	327
06:00–06:59	67	94	202	286
07:00–07:59	65	91	207	289
<i>Total</i>	<i>1683</i>	<i>2515</i>	<i>5300</i>	<i>7433</i>

level caused multiple changes to be recorded for the other observed variables.

The observed changes contribute to instability in the network topology. This topology instability affects the QoS offered by the backbone network. This is because the frequent adjustment of transceiver power output levels causes changes in neighbourhood sizes. The changes in neighbourhood sizes affect the routes being maintained by the OLSR routing protocol as these neighbouring nodes are the potential next hop to an intended destination node. Constant changes to the neighbourhood size thus result in changes to the routing table maintained by OLSR. This, in turn, means that routes are constantly changing which causes an increase in OLSR's route creation and maintenance traffic. This additional routing protocol overhead consumes bandwidth at the expense of data traffic and thus the data throughput rates would be reduced in a network that is forwarding traffic.

It must be remembered that the instability measured in this experiment only captured the instabilities caused by the TC scheme as the network was not carrying data traffic. Other works in the literature have shown the instability caused by traffic forwarding in the I-WMN. Thus, PlainTC's use

in a deployed I-WMN is likely to exacerbate the existing network instability. Solutions to the instability caused by traffic forwarding have been proposed in [18, 19] in the form of a stability-aware routing protocol and routing metric, respectively, but these existing solutions are not able to reduce topology instability. A solution to reduce the topology instability caused by a TC scheme is required. Section 4 presents a context-based solution to reduce the topology instability introduced by a distributed TC scheme.

4. A Context-Based Topology Control Prototype: PlainTC+

PlainTC+ is an improvement upon the PlainTC scheme in [14]. PlainTC+ uses context information to reduce the topology instability caused by TC in the I-WMN. This instability was demonstrated in Section 3.

4.1. Context Sources. Topology instability cannot be measured directly but it can be inferred from the number of recorded changes to the transceiver power output level,

neighbourhood size, network size, and link quality variables, respectively, for the entire network. Thus, these four variables are employed as sources of contextual information for the context-aware PlainTC+ scheme.

The four variables represent a node's view of its local neighbourhood and the global network and are taken to represent the context of a node. A change to any of these four variables represents a change to the node's observed context.

4.2. Quantifying Context Changes. Changes in the observed context variables are referred to as events. The weights associated with each of the observed variables are meant to identify the high-impact events that PlainTC+ should respond to. This differs from PlainTC's current operation where events taking place outside of a node's immediate vicinity cause the node to adjust its transceiver power output. The envisaged result is a reduction in the number of changes made to node's transceiver power levels and a reduction in the multiplier effect that such changes have on the other observed variables. This multiplier effect was shown in Section 3.

The weights associated with each variable are meant to be used to quantify the change in context experienced by each node. The process that is established here can be used on any deployed I-WMN to determine the appropriate weightings for that particular deployment. The use of this process will be demonstrated using data obtained from an indoor I-WMN test-bed. The data found in Table 1 is specific to our test-bed but the methodology employed to record the changes to the context variables and the process described here can be applied to any WMN.

The data contained in Table 1 is subjected to a statistical method described in [20] called Principal Component Analysis (PCA). PCA is commonly used to reduce the dimensionality of large data sets but it can also be used to determine the contribution of each variable to the total variability contained in multivariate data. PCA is being employed, in this instance, to determine the contribution (weight) of each variable to the overall context change experienced by a node.

The *FactoMineR* package introduced in [21] is used within *R* statistical environment to perform the PCA. Apart from being able to perform classical PCA, *FactoMineR* also has a mechanism for determining the contribution of a variable to total variability of multivariate data. This mechanism is documented in [22]. This makes *FactoMineR* and its PCA methodology a reliable choice for determining the contribution of each of the four variables to topology instability.

The process defined by *FactoMineR* for calculating the variable weights is as follows.

Step 1. The first step is to load the contents of Table 1 into *R*.

Step 2. The second step is to perform the PCA on the input data.

Step 3. The third step determines the proportion of variation retained by the Principal Components. The amount of variability retained by each Principal Component is calculated from the *eigenvalues*. The variability for each component is depicted below:

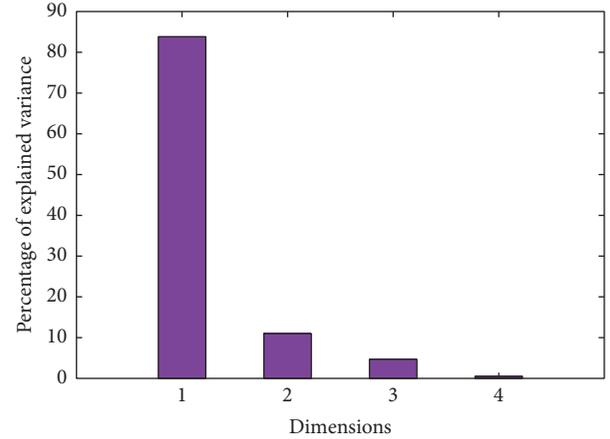


FIGURE 4: Contribution of dimensions towards the total variability in the observed data.

	eigenvalue	percentage of variance	cumulative
comp 1	3.35336277	83.8340693	83.83407
comp 2	0.43964847	10.9912116	94.82528
comp 3	0.18722414	4.6806035	99.50588
comp 4	0.01976462	0.4941155	100.00000

Eigenvalues less than 1 are commonly used to eliminate their associated dimensions as these dimensions do not greatly account for the total variance in the data. Thus, only the first component is selected for further use as it has an eigenvalue greater than 1 and accounts for approximately 84% of the total variance contained in the data. Contributions towards the total variance contained in the data are depicted in Figure 4.

Step 4. The fourth step determines the percentage contribution of each variable on the total variance explained by the identified components:

	Dim.1
Transceiver.Power	27.95265
Network.Size	24.82228
Neighbourhood.size	27.66850
Link.Quality	19.55657

These contributions correspond to the weights that can be associated with each variable and represent the relative impact of a change in that variable on the context change experienced by a node. This step represents the end of the PCA on the observed data.

Step 5. The fifth step uses the normalised weights (obtained in the previous step) to determine the appropriate context-change threshold for the network. The goal of the threshold value is to reduce the number of changes made to the transceiver power output level of a network node. The multiplier effect of the number of changes to a node's transceiver power output on the other context variables was shown in Section 3. Thus, reducing the number of transceiver power

changes performed by a node will have a positive impact on reducing the number of changes observed for the other context variables. The role of the threshold is to act as a sentinel value that determines whether an observed change is sufficiently significant to warrant a change to a node's transceiver power output level.

4.3. Deriving a Context-Change Threshold. The context-change threshold is proposed to limit the number of transceiver power changes produced by the TC scheme. This threshold value quantifies the minimum amount of change required to trigger a transceiver power adjustment.

The weightings derived for each of the four observed variables and knowledge of the multiplier effect of the transceiver power variable on the other three variables provide valuable input into the formulation of the context-change threshold value. This value has to be sufficiently high to reduce the number of changes to the transceiver power output variable but sufficiently low to cause PlainTC to react to significant events.

A significant event is defined as one that takes place within the immediate neighbourhood of an affected node and affects a variable with network-wide impact such as the network size. Less significant events originate from outside of the immediate neighbourhood. These definitions allow for a node to react to significant events originating in its immediate neighbourhood by adjusting its transceiver power output whilst maintaining the current transceiver power output for events originating elsewhere. A balance between reducing the number of transceiver power changes and reacting to significant events can thus be achieved.

The normalised versions of the threshold values obtained in Step 4 of the PCA process, of 0.1956, 0.2482, 0.2767, and 0.2795, respectively, are too low to ensure that the TC scheme only reacts to local events with network-wide impact. A threshold value of 1.0000 is too high and the hurdle would only be met in extreme circumstances such as network partitioning or network merging immediately after the affected node has adjusted its transceiver power output. The TC scheme would thus be rendered unresponsive to all events, even those in its immediate neighbourhood, in this scenario.

The only means that a TC scheme possesses in order to react to network events is to adjust the transceiver power output of the affected node. Calculating the threshold using the transceiver power variable with any combination of the other variables would enable continuous adjustment of node transceiver power output levels. This is a situation to be avoided as the aim of the threshold value is to reduce the number of transceiver power changes performed by the network nodes.

PlainTC currently adjusts its transceiver power output in order to maintain the desired connectivity at each network node. The number of neighbouring nodes to be maintained is derived from the network size variable and this variable has been shown to change quite frequently. Thus, the connectivity value changes frequently in response to the network size variable and the required transceiver power adjustment is attempted. The transceiver power change should only occur

if the change in the observed network size is caused by a change in the observed neighbourhood size. This would force the TC scheme to react only to events in the affected node's immediate neighbourhood which have network-wide impact. Thus, the network size and neighbourhood size variables must contribute to the threshold value. The link quality variable changes most frequently because it encapsulates natural variations in link quality as well as changes caused by TC. Thus, the link quality variable must also contribute to the threshold value. Therefore, the most appropriate threshold value to force the TC scheme to react only to local events with a network-wide impact is one that ensures that PlainTC+ will only adjust a node's transceiver power output in situations where the affected node simultaneously experiences changes in neighbourhood size and network size and in link quality. The selected threshold value is derived from the weightings determined in Step 4 and is computed as

$$\begin{aligned} contextChangeThreshold &= 0.2767 + 0.2482 + 0.1956 \\ &= 0.7205. \end{aligned} \quad (1)$$

4.4. PlainTC+ Design. The execution logic consists of four phases, each one corresponding to the *Autonomic Control Loop* described in [23] and shown in Figure 5. Autonomous systems must collect information to determine the current situation or context in which they operate. The collected information is analysed to inform the adaptation decisions to be taken. These decisions are subsequently implemented to complete the adaptation response.

PlainTC+ is designed to employ the *Autonomic Control Loop* as follows. In the first phase, PlainTC+ collects the current node, neighbourhood, and network states. These states are comprised of low-level variables such as the current transceiver power level, number of neighbouring nodes, and the total backbone network size. Thus, the current context of both the node and the network can be ascertained. This contextual information is then analysed in the second phase to quantify the change in context experienced by the node and to determine the current neighbourhood size. The neighbourhood size is used to maintain network connectivity as this approach has been shown in [15] to be effective. PlainTC+ now decides whether to change the node's transceiver power output in the third phase. If the decision is made for the node to adjust its transceiver power output, then this decision is acted upon in the final phase. The node's transceiver power will be adjusted upwards or downwards depending upon the decision taken in the previous phase. PlainTC+ is designed to iteratively adjust its transceiver power to reach the required level as this approach does not add to the computational complexity of the scheme. Thus, a faster convergence time is sacrificed for simplicity. Figure 6 depicts a single execution cycle.

The use of the *Autonomic Control Loop* in PlainTC+ does not impact on the performance of the scheme as all the data being processed is always locally available at the node. In addition, there are no additional messaging overheads and latency introduced since all the required data is collected in the normal operations of the routing protocol and the nodes

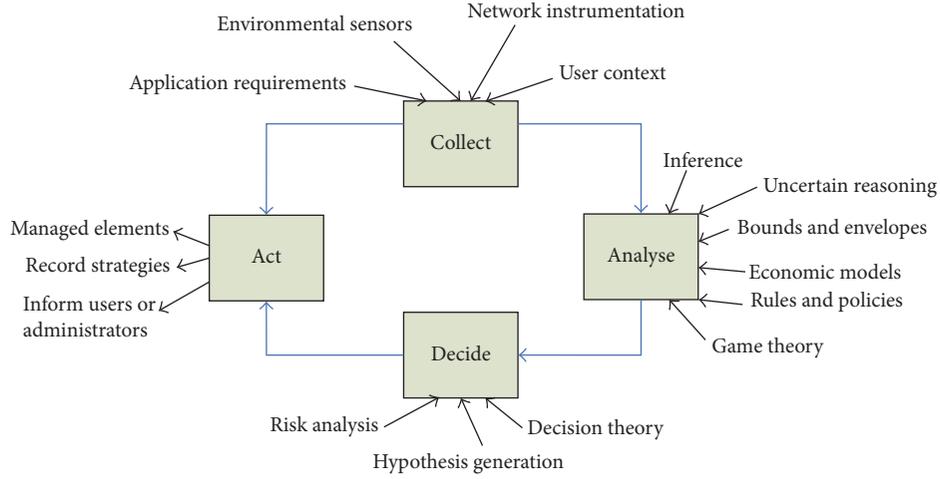


FIGURE 5: Phases of the Autonomic Control Loop for network scenarios [23].

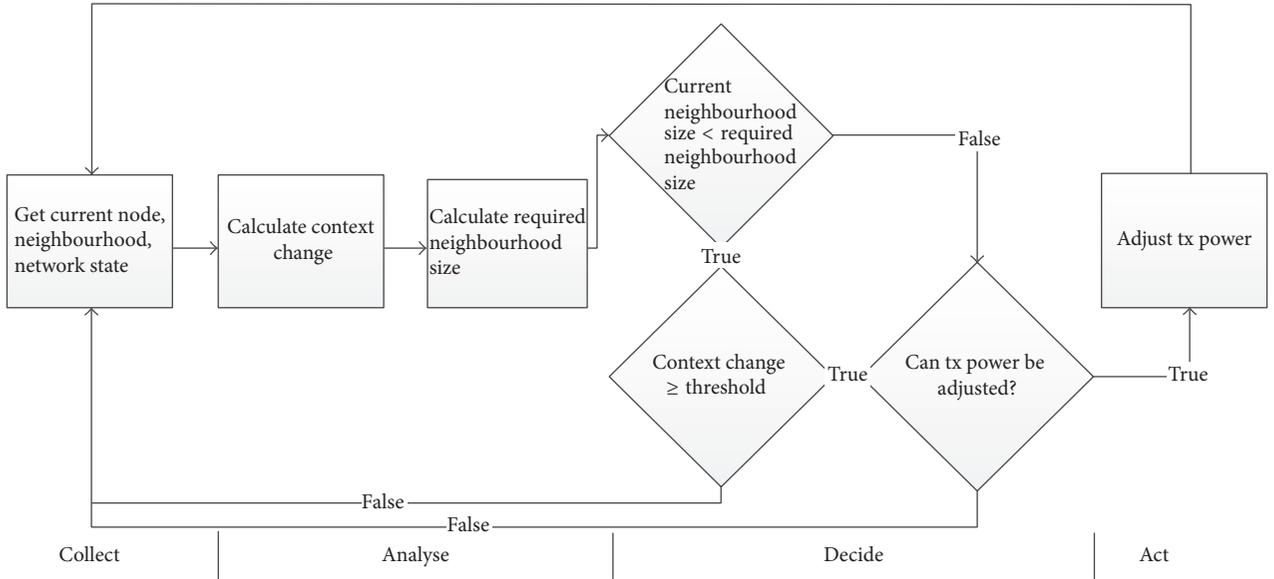


FIGURE 6: Flowchart depicting PlainTC+'s execution logic.

do not collaborate with, or communicate their decisions to, each other.

PlainTC+ is an amalgamation of the original PlainTC algorithm and the results of the process to quantify topology instability. The process (contained in Section 4.2) produces two outcomes: the first outcome is the determination of the weights associated with each observed context variable and the second outcome is the appropriate context-change threshold value for the observed network.

The context change for the observed test-bed network can be computed as

$$\begin{aligned} contextChange = & (0.2795 \times A) + (0.2767 \times B) \\ & + (0.2482 \times C) + (0.1956 \times D), \end{aligned} \quad (2)$$

where

$$\begin{aligned} A &= \begin{cases} 1, & \text{if } |\Delta transceiverPower| > 0, \\ 0, & \text{otherwise,} \end{cases} \\ B &= \begin{cases} 1, & \text{if } |\Delta neighbourhoodSize| > 0, \\ 0, & \text{otherwise,} \end{cases} \\ C &= \begin{cases} 1, & \text{if } |\Delta networkSize| > 0, \\ 0, & \text{otherwise,} \end{cases} \\ D &= \begin{cases} 1, & \text{if } |\Delta linkQuality| > 0, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (3)$$

```

(1) no_nodes ← getRoutingTableSize()
(2) current_no_neighbours ← getNeighbourhoodSize()
(3) CNN ←  $\lceil 5.1774 \log(\textit{no\_nodes}) \rceil$ 
(4) max_tx_power ← 78 qdBm
(5) min_tx_power ← 1 qdBm
(6) current_tx_power ← getCurrentTxPower()
(7) tx_power_change_level ← 3 qdBm
(8) context_change_threshold ← 0.7205
(9) if current_tx_power ≤ max_tx_power then
(10)   if current_no_neighbours > CNN then
(11)     if (current_tx_power − tx_power_change_level) ≥ min_tx_power then
(12)       current_tx_power ← current_tx_power − tx_power_change_level
(13)     end if
(14)   end if
(15) if current_no_neighbours < CNN then
(16)   if getContextChange() ≥ context_change_threshold then
(17)     if (current_tx_power + tx_power_change_level) ≤ max_tx_power then
(18)       current_tx_power ← current_tx_power + tx_power_change_level
(19)     end if
(20)   end if
(21) end if
(22) end if

```

ALGORITHM 1: Execution Logic of PlainTC+.

A change in every variable is mapped to the contribution of that variable to the total variability in the collected data. Thus, the formulation quantifies the change that a node detects in its environment. The result is a normalised value in $[0, 1]$ that is independently computed by each node and the result depends upon the variable changes being reported by the node. This quantified context-change value can now be subjected to the appropriate threshold value within PlainTC+ in order to reduce the number of transceiver power output changes performed by the network nodes.

The resultant algorithm employed by PlainTC+ is shown in Algorithm 1. The use of the CNN value is augmented with the context-change threshold value when deciding to adjust a node's transceiver power output. The potential reduction of a node's transceiver power output is a key motivator for the use of a TC scheme in the I-WMN. Thus, PlainTC+ is designed to allow for the easy reduction of a node's transceiver power output by basing the adjustment decision only upon the CNN to be maintained.

Raising a node's transceiver power output requires that the quantified change being observed meets or exceeds the context-change threshold value. The threshold value is defined in line (8) and the additional context-change condition can be found in line (16) of Algorithm 1. These additions to the algorithm are meant to reduce the number of transceiver power adjustments made by a node whilst allowing for savings in a node's transceiver power output to be achieved.

It should be noted that the threshold value derived above is based upon data obtained from our test-bed and this value is therefore unique to this particular test-bed. However, the procedure for deriving the threshold value is still applicable to

any other WMN test-bed or real-world deployment and this procedure will yield a value that is unique to that deployment.

In the event that other context variables are considered, then the process for quantifying context changes in Section 4.2 can be used to determine the various contributions to the context change experienced by a node. These variables can be divided into variables that are affected by neighbourhood changes and those variables affected by network changes. The principle of only adjusting a node's transceiver power in response to significant events as defined earlier in this section can still apply to determine a threshold value that will only allow for responses to neighbourhood-originating events that have a network-wide impact. Therefore, the threshold value needs to incorporate all the identified neighbourhood-based context variables but exclude the variable whose changes are being minimised.

4.5. Implementation. Linksys WRT54GL nodes are used as the hardware platform on which to deploy PlainTC+. These devices are popular backbone nodes for I-WMN deployments due to their low cost, rugged reputation, and easily available tutorials and related support documentation.

The Linksys nodes are transformed into I-WMN nodes via the use of the OpenWRT firmware [24]. This firmware is a stripped-down version of the Linux OS that caters for the limitations imposed by embedded devices and wireless routers such as the Linksys WRT54GL. Apart from common embedded Linux tools such as uClibc, Busybox, and a shell interpreter, a package manager is also provided. Due to OpenWRT's modular design, the Linux kernel can be optimized to suit the underlying hardware platform whilst allowing the user-space environment to remain unaffected (requiring only a recompilation).

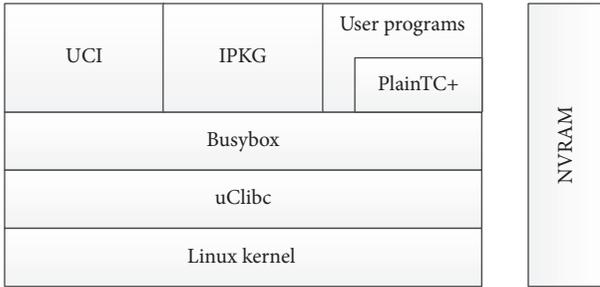


FIGURE 7: Implementation architecture, adapted from [24].

Mesh networking functionality is enabled by the installation of the necessary routing and network management packages via the in-built package manager. The OpenWRT firmware also allows access to the NVRAM (nonvolatile random access memory) partition of the Linksys node. This partition is the equivalent of the secondary storage facilities available on all possible I-WMN nodes. User-space packages can interact with the NVRAM partition. The 64 KB NVRAM partition stores configuration variables that span the entire logical protocol stack and is thus a ready-made source of cross-layer optimization data. It is also possible to alter the values of configuration variables.

As the Linksys nodes were deployed with the OpenWRT firmware to provide the mesh networking capabilities, PlainTC+ has to exist within the OpenWRT ecosystem depicted in Figure 7. PlainTC+ is implemented as a user-space application that is initiated at node start-up and executes at two-minute intervals thereafter. The application is required to interact with the configuration variables (stored within the NVRAM partition) that control the magnitude of the transceiver power output of the Linksys nodes as well as with the OLSR routing protocol that is usually also implemented as a user-space application. This interaction is depicted in Figure 8. PlainTC+ relies upon the topology information collected during OLSR's normal operations. The total number of backbone nodes derived from OLSR's routing table is used to determine the appropriate CNN to be maintained. If the transceiver power output requires modification then the value of the state variable associated with the node's transceiver power level is modified. This modification forces an OpenWRT firmware trigger that sets the updated value of the state variable on the wireless transceiver hardware. The change of transceiver power level is performed without having to perform a node reboot.

No modifications to the OpenWRT firmware were necessary and therefore the advantage of the above-mentioned implementation approach was that the benefit of a cross-layer approach is gained without the need to reengineer the protocol stack by specifying an additional protocol layer, its interlayer interfaces, and the various communication messages that would be required, thus conforming to the logical implementation architecture shown in Figure 9.

5. The Test-Bed Environment

The indoor mesh test-bed operated in a ground-floor laboratory in a 3-storey office building at the University of Zululand's main campus. The test-bed network operated in 802.11g mode on channel 6 in order to mitigate interference caused by a separate WLAN that was operational within the building.

The mesh test-bed consisted of 14 nodes placed in 6 m × 4 m area as shown in Figure 10. The node placement was determined by the availability of plug points which is analogous to the coupling of nodes with existing infrastructure in real-world deployments. Each node in the mesh backbone consisted of a mains-powered, Linksys WRT54GL router with the OpenWRT-based Freifunk firmware (version 1.7.2) used to provide mesh networking functionality. The Freifunk firmware uses the OLSR routing protocol by default as this protocol has been successfully employed in large-scale WMN deployments.

The Linksys WRT54GL routers possessed a 200 MHz processor, 16 Mb of RAM, 4 Mb flash memory, and a Broadcom 802.11b/g radio chipset. The wireless chipset allowed transceiver power output levels to be set from 1 dBm to 19.5 dBm. The latter value is the maximum power output recommended by the manufacturer.

Each node was connected via Ethernet through a switch to a central server. The use of the Ethernet port for data collection was to allow for network traffic to flow over the wireless mesh network and for performance data and other statistical information to flow over the Ethernet network. Thus, the performance data did not interfere with the data flowing over the wireless mesh network. The central server also synchronised the clocks of all the network nodes in order for the collected data to be accurately time-stamped.

Custom data collection scripts were written and installed within each test-bed node. These scripts were designed to amalgamate the time-stamped data from the node and the routing protocol at one-second intervals. This data was "pushed" by each test-bed node to the central data collection server where it was stored whilst awaiting further analysis. The data collected from the individual nodes included their current transceiver power output, network size, neighbourhood size, and the link quality associated with their various neighbouring nodes.

Data traffic was generated by using the *iperf* traffic generator (with its default settings) installed on each test-bed node. The central server remotely controlled the operation of these traffic generators. The test-bed supported two traffic scenarios: the intramesh traffic scenario where the source and destination are within the test-bed backbone and the Portal-oriented traffic scenario where either the source or destination was resident outside of the test-bed backbone. For the Portal-oriented scenario, a PC accessible via the designated gateway node (node 10 in Figure 10) was set up to act as an external source or destination. These two scenarios are depicted in Figure 11.

The *iperf* tool allowed for the generation of both UDP- and TCP-based traffic. The UDP-based traffic was used in the intramesh scenario to emulate activities such as LAN

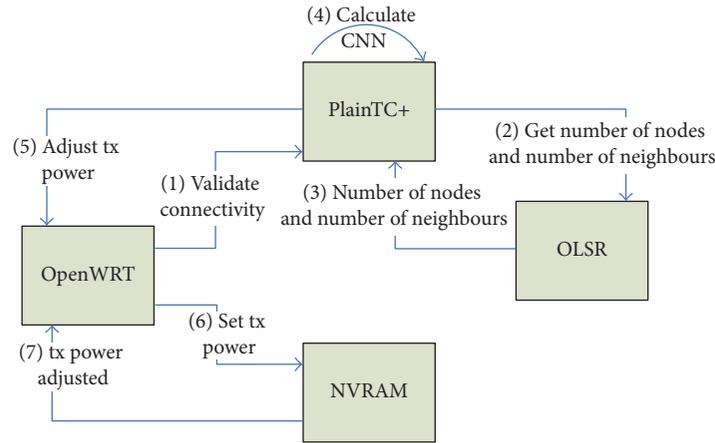


FIGURE 8: PlainTC’s interactions with other system elements.

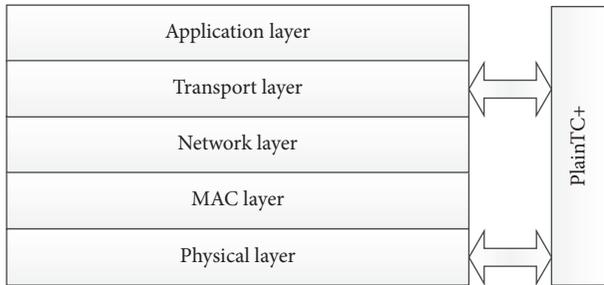


FIGURE 9: Placement of PlainTC with regard to the network protocol stack.

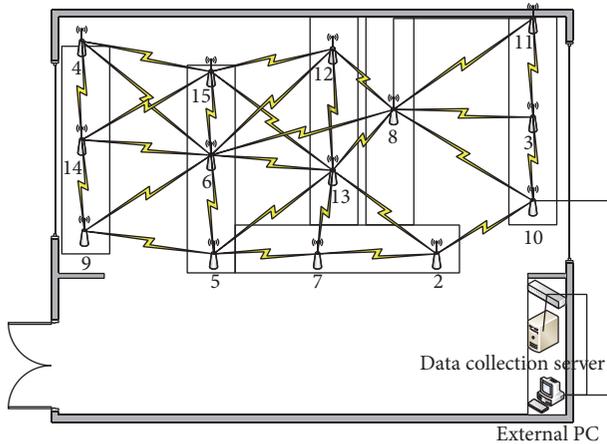


FIGURE 10: Test-bed layout.

gaming. The TCP-based traffic was meant to emulate web activities where at least 80% of all traffic is carried via TCP. The server was set up to initiate traffic flow from 90% of the network nodes to randomly selected, destination nodes for the intramesh scenario. The server initiated traffic flows from 90% of the network nodes to the external PC for the Portal-oriented scenario.

The test-bed allowed for the evaluation of both constant-size and incrementally increasing network backbones. The constant-size networks were initiated with all the network nodes being switched on simultaneously and a ten-minute stabilisation period was allowed to pass before data was collected. The duration of the data collection in the constant-size experiments is 60 minutes. Network sizes ranged from 8 to 14 nodes.

The forementioned experiments were designed to examine the dynamic or run-time behaviour of PlainTC+ when nodes were added to an existing backbone network. These experiments began with an initial network size of eight (8) nodes and edge nodes were incrementally introduced to the backbone network at ten-minute intervals. The initial 8-node network was given a ten-minute stabilisation period before data collection begins. Data collection stopped after a duration of 70 minutes with data being collected whilst each newly introduced node adjusted its transceiver power output. Thus, the collected data showed the response of the newly introduced nodes to the existing network and showed the reaction of the existing nodes to the introduction of a new node. With data being collected at one-second intervals, a total of 600 data points were collected by each node for each observed metric for each evaluation run. Each experiment was repeated five (5) times as the volume of data collected posed a storage challenge.

6. Experimental Results

PlainTC+ is designed to reduce the topology instability caused by transceiver power changes. This is achieved by limiting the number of power changes performed by a node in the backbone network. PlainTC+ is evaluated against PlainTC to determine whether the context-change mechanism can successfully reduce topology instability.

6.1. Topology Instability. The numbers of changes to the observed network variables, when using PlainTC+, are listed in Table 2. This is compared to the outcome for PlainTC found in Table 1. PlainTC+ is found to significantly reduce the

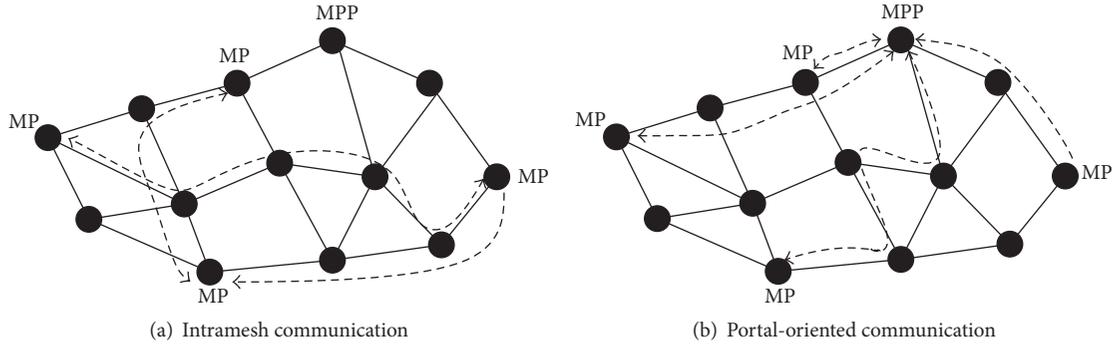


FIGURE 11: Communication Scenarios.

TABLE 2: Number of changes in node variables over a 24-hour period.

Time period	Number of changes in transceiver power	Number of changes in neighbourhood size	Number of changes in network size	Number of changes in link quality
08:00–08:59	59	100	198	277
09:00–09:59	40	68	143	199
10:00–10:59	39	66	139	195
11:00–11:59	39	70	146	202
12:00–12:59	38	65	139	196
13:00–13:59	37	68	136	189
14:00–14:59	36	61	128	184
15:00–15:59	35	60	124	173
16:00–16:59	38	68	151	211
17:00–17:59	37	68	147	201
18:00–18:59	39	66	119	166
19:00–19:59	38	71	135	189
20:00–20:59	38	65	137	192
21:00–21:59	36	60	131	183
22:00–22:59	37	65	127	178
23:00–23:59	42	70	152	207
00:00–00:59	36	59	125	171
01:00–01:59	37	67	136	190
02:00–02:59	35	64	135	186
03:00–03:59	36	66	140	196
04:00–04:59	37	67	129	179
05:00–05:59	40	69	148	199
06:00–06:59	37	62	128	184
07:00–07:59	36	65	139	196
<i>Total</i>	<i>922</i>	<i>1610</i>	<i>3332</i>	<i>4643</i>

number of changes for each observed variable. The context-change threshold mechanism produced a 45% reduction (1683 changes for PlainTC versus 922 changes for PlainTC+) in the number of transceiver power changes recorded over a 24-hour period when compared to the original PlainTC scheme. This result is produced by using the threshold value to suppress unnecessary transceiver power increases and compares favourably with a similar threshold mechanism

described in [18] that reduced route flapping by 60%. It must be remembered that PlainTC+ only applied its context-change threshold mechanism to transceiver power increases. It is expected that PlainTC+ would have achieved a larger reduction in topology instability if the threshold was applied to transceiver power decreases as well. However, in the context of TC, such a move would have been at the expense of cumulative transceiver power savings.

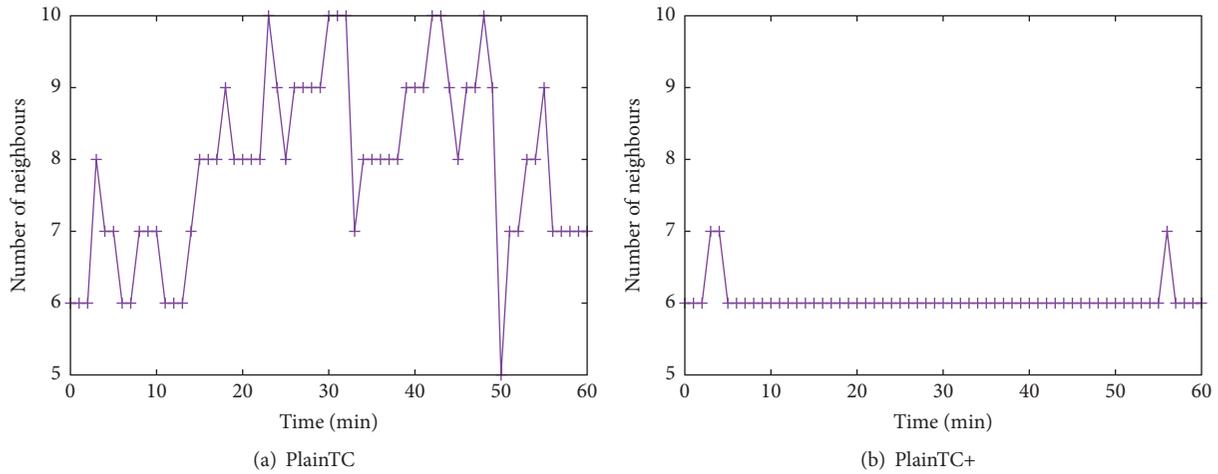


FIGURE 12: Observed neighbourhood size.

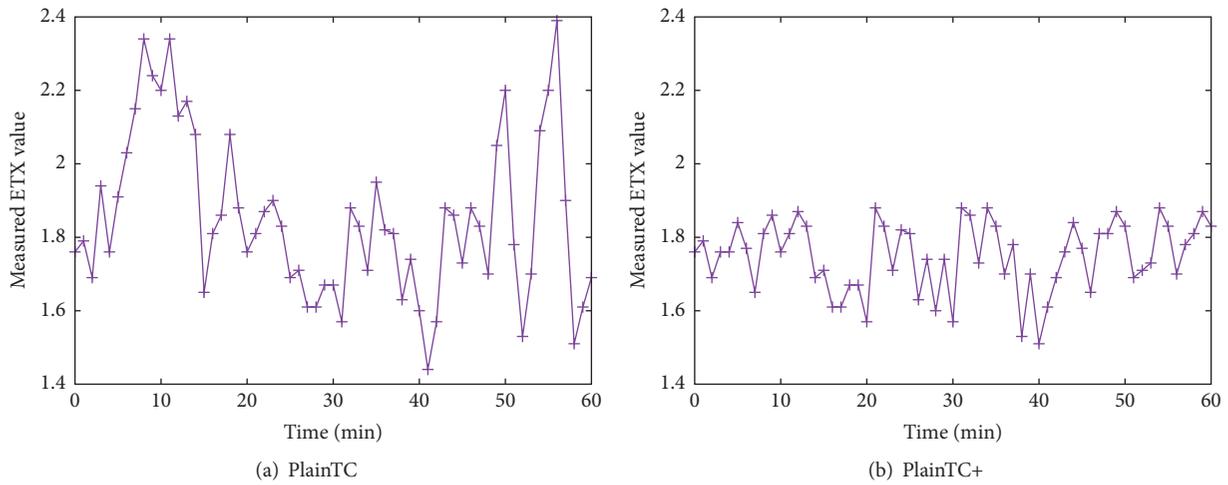


FIGURE 13: Observed link quality.

The reduction in the number of transceiver power output changes also caused reductions in the other variables observed within the network. Reductions of 36%, 37%, and 38% were recorded for the neighbourhood size, network size, and link quality variables, respectively. These values represent a significant improvement to the stability of the network topology.

6.2. Effect on the OLSR Routing Protocol. The OLSR routing protocol reports both the number of neighbouring nodes and the link quality associated with each neighbour. These metrics are affected by transceiver power output changes and the effect of a reduction in the number of transceiver power changes is depicted in Figures 12 and 13.

The restriction on transceiver power changes imposed by the context-change threshold helps to maintain a stable one-hop neighbourhood, as depicted in Figure 12. PlainTC+ causes a node and its neighbours not to deviate from the required CNN. This results in a stable neighbourhood size. Nodes may reduce their transceiver power output level to

maintain the required CNN but the context-change threshold causes the node to avoid increasing its transceiver power output to maintain a new CNN when a new node joins the network and this new node is not an immediate neighbour. Any increases in neighbourhood size are either caused by neighbouring nodes in the process of adjusting their own transceiver power outputs to maintain their CNN or caused by the natural variations in link quality which cause a temporary loss of links to neighbours.

The improved network stability offered by PlainTC+ is also found to reduce the severity of the fluctuations in observed link quality. This is depicted in Figure 13. Nodes do not adjust their transceiver power output levels as frequently as with PlainTC. This improves the performance of the probe packets sent out by OLSR to calculate the ETX metric and leads to reduced route flapping caused by variations in the route cost metric.

6.3. Transceiver Power Output. PlainTC+ results in a lower cumulative transceiver power output level compared to

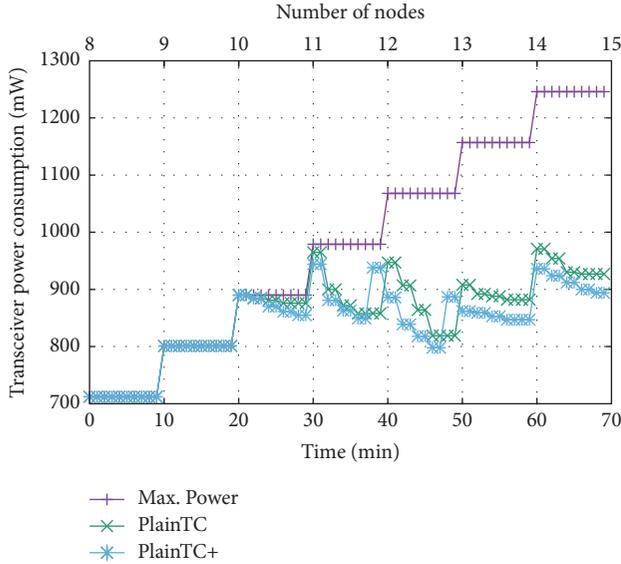


FIGURE 14: Dynamic adjustment of transceiver power by PlainTC+.

PlainTC (see Figure 14). The power savings are a result of PlainTC+ subjecting any increases in transceiver power output to the context-change threshold. This prevents nodes from increasing their transceiver power outputs in response to changes in network size that arise from outside of their immediate neighbourhood. The new CNN to be maintained in such scenarios is overruled by the context-change threshold. Therefore, nodes within an unaffected neighbourhood do not adjust their transceiver power output levels upwards with PlainTC+ whereas these same nodes increase their transceiver power levels when employing PlainTC.

6.4. Network Connectivity. Figure 15 shows that PlainTC+ reduces the time to establish network connectivity when changes to the network size occur. This reduction (when compared to PlainTC) is caused by the positive effect of the context-change threshold on the workings of the OLSR routing protocol.

Network connectivity is determined by the availability of routes to all possible destinations. The reduction in network instability caused by the context-change threshold aids in the propagation of routing information by OLSR. The routing updates propagate faster due to the increased stability of links between neighbouring nodes. Thus, the routing tables at each network node report routes for every other node at a faster rate than with PlainTC.

6.5. Packet Delivery Ratio and Throughput. PlainTC+ outperforms PlainTC in delivering data traffic to the intended destinations as the network size increases. The improvements in topology stability offered by PlainTC+ result in PDR and throughput increases for both the intramesh and Portal-oriented traffic scenarios. These increases cause the data traffic performance of the network created by PlainTC+ to approach the performance achieved when employing Max. Power. Thus, PlainTC's performance gap to Max.

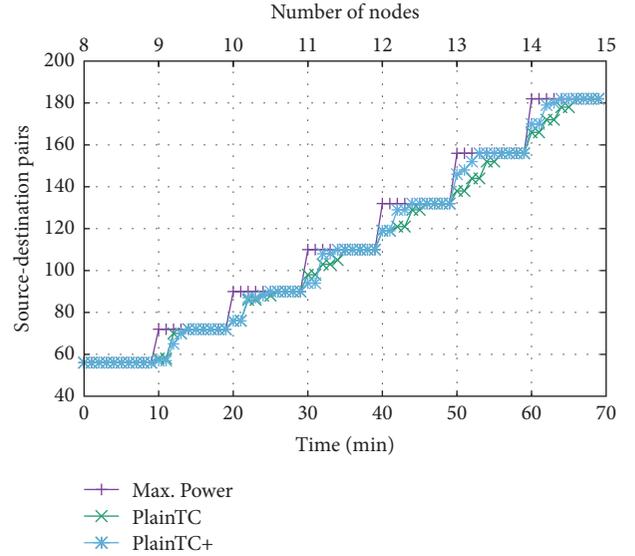


FIGURE 15: Dynamic establishment of network connectivity by PlainTC+.

TABLE 3: Average PDR achieved for intramesh traffic.

Network size	Max. Power	PlainTC	PlainTC+
8	97.13	96.75	96.89
9	97.02	96.29	96.67
10	96.71	95.65	96.04
11	96.67	95.22	95.82
12	96.31	94.76	95.75
13	95.88	94.16	95.38
14	95.4	93.25	94.67

TABLE 4: Average PDR achieved for Portal-oriented traffic.

Network size	Max. Power	PlainTC	PlainTC+
8	99.72	99.46	99.48
9	99.66	99.53	99.56
10	99.53	99.15	99.35
11	99.36	98.95	99.02
12	99.19	98.44	98.62
13	98.98	98.12	98.52
14	98.71	98.02	98.26

Power is decreased despite achieving even greater cumulative transceiver power savings compared to PlainTC.

The PDR performance of PlainTC+ is contained in Tables 3 and 4 and the throughput results are contained in Tables 5 and 6.

6.6. Resource Consumption. PlainTC+ was observed to consume 371 KB of memory (2.3% of total memory) and the CPU utilisation equalled that of PlainTC at 0.3%. Therefore, the addition of the context-change mechanism did not significantly increase the recorded resource consumption.

TABLE 5: Average throughput achieved for intramesh traffic (KBps).

Network size	Max. Power	PlainTC	PlainTC+
8	328.17	327.54	327.75
9	263.91	263.85	263.89
10	227.43	221.50	223.12
11	213.87	211.09	211.95
12	196.51	180.82	188.52
13	192.63	184.25	185.65
14	185.11	170.92	177.35

TABLE 6: Average throughput achieved for Portal-oriented traffic (KBps).

Network size	Max. Power	PlainTC	PlainTC+
8	313.57	312.27	312.55
9	271.82	270.83	271.02
10	219.65	209.71	212.65
11	192.72	184.18	188.63
12	164.57	159.73	161.88
13	155.79	151.52	151.92
14	147.35	131.61	137.55

7. Conclusion and Future Work

In this article we demonstrated the negative impact that the PlainTC prototype has on topology stability. This instability was diagnosed to be caused by transceiver power adjustments being carried out under the auspices of a distributed TC scheme.

We subsequently proposed the use of context information to reduce the topology instability caused by PlainTC. The PlainTC+ prototype was presented with this prototype incorporating the collection of context information and using this information to filter out unnecessary transceiver power adjustments.

An indoor test-bed evaluation of PlainTC+ showed a 45% reduction in the number of transceiver power adjustments undertaken in the network. This reduction stabilised the neighbourhood sizes and link qualities experienced by network nodes. The improved topology stability enhanced the network performance with improvements to the observed PDR and throughput metrics when compared to PlainTC. The performance gap to the Max. Power scheme was narrowed and PlainTC+ offers a deployed I-WMN the possibility of achieving significant cumulative transceiver power savings with a minimal sacrifice in network performance.

Our future work seeks to provide a mechanism for the dynamic, real-time calculation of the context variables' contributions to the total variability in topology stability. The current process to determine these contributions that is defined in Section 4.2 is backwards-looking and the results may not be an accurate reflection of current network conditions. We are in the process of deploying a software-defined I-WMN test-bed and the centralised software-defined network controller is ideally suited to the task of collating the context information from the network nodes and dynamically

computing each context variable's contribution to the total variability in topology instability. The appropriate context-change threshold value can be subsequently determined in real time at the controller and this threshold value can be pushed to the WMN nodes to regulate the adjustment of the node's transceiver power. PlainTC+ will thus become a real-time, context-aware TC mechanism with the ability to regulate nodes' transceiver powers in response to current network conditions.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this article.

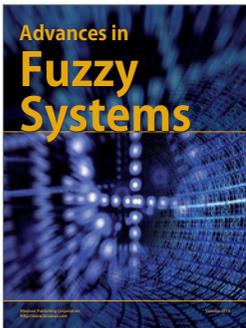
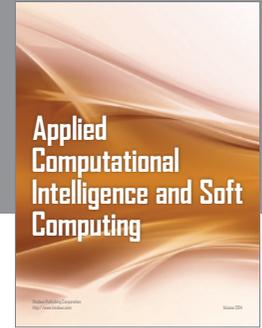
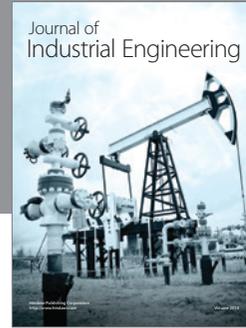
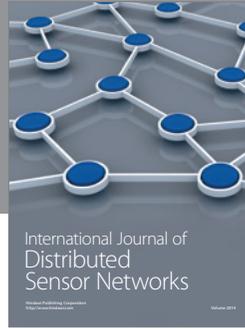
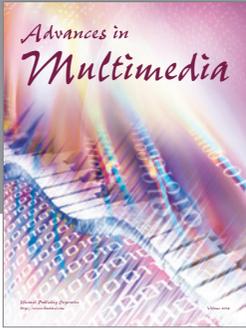
Acknowledgments

The authors would like to acknowledge the financial and technical support provided by both the Centre of Excellence for Mobile e-Services at the University of Zululand, South Africa, and the Meraka Institute at the Council for Scientific and Industrial Research, South Africa.

References

- [1] W. Allen, A. Martin, and A. Rangarajan, "Designing and deploying a rural ad-hoc community mesh network testbed," in *Proceedings of the IEEE Conference on Local Computer Networks*, pp. 740–743, November 2005.
- [2] H. Lundgren, K. Ramachandran, E. Belding-Royer et al., "Experiences from the design, deployment, and usage of the UCSB meshnet testbed," *IEEE Wireless Communications*, vol. 13, no. 2, pp. 18–29, 2006.
- [3] J. Camp, J. Robinson, C. Steger, and E. Knightly, "Measurement driven deployment of a two-tier urban mesh access network," in *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*, pp. 96–109, June 2006.
- [4] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*, John Wiley & Sons, Chichester, UK, 2005.
- [5] F. O. Aron, T. O. Olwal, A. Kurien, and M. O. Odhiambo, "Energy efficient topology control algorithm for wireless mesh networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '08)*, pp. 135–140, August 2008.
- [6] N. Li and J. C. Hou, "Localized fault-tolerant topology control in wireless ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 4, pp. 307–320, 2006.
- [7] K.-D. Wu and W. Liao, "Interference-efficient topology control in wireless ad hoc networks," in *Proceedings of the 3rd IEEE Consumer Communications and Networking Conference (CCNC '06)*, pp. 411–415, January 2006.
- [8] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "A cone-based distributed topology-control algorithm for wireless multi-hop networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 147–159, 2005.
- [9] V. Kawadia and P. R. Kumar, "Principles and protocols for power control in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 76–88, 2005.

- [10] A. Cerpa and D. Estrin, "ASCENT: adaptive self-configuring sensor networks topologies," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 272–285, 2004.
- [11] P. Valera, P. Lee, Y. Wong, W. Seah, H. Tan, and H. Ju, "An experimental study on connectivity and topology control in real multi-hop networks," in *Proceedings of the 4th International Wireless Internet Conference (WICON '08)*, Maui, Hawaii, USA, November 2008.
- [12] I. F. Akyildiz and X. Wang, *Wireless Mesh Networks*, John Wiley & Sons, Chichester, UK, 2009.
- [13] X. Wang and A. O. Lim, "IEEE 802.11s wireless mesh networks: framework and challenges," *Ad Hoc Networks*, vol. 6, no. 6, pp. 970–984, 2008.
- [14] P. Mudali, T. C. Nyandeni, N. Ntlatlapa, and M. O. Adigun, "Design and implementation of a Topology Control scheme for Wireless Mesh Networks," in *Proceedings of the IEEE Africon*, September 2009.
- [15] P. Mudali, T. C. Nyandeni, N. Ntlatlapa, and M. O. Adigun, "A test-bed evaluation of connectivity strategies for infrastructure wireless mesh networks," in *Proceedings of the Southern African Telecommunications and Network Applications Conference (SATNAC '10)*, 2010, <http://www.satnac.org.za/proceedings/2010/engineering.htm>.
- [16] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Communications Magazine*, vol. 43, no. 9, pp. S23–S30, 2005.
- [17] K. Kowalik, M. Bykowski, B. Keegan, and M. Davis, "An evaluation of a conservative transmit power control mechanism on an indoor 802.11 wire-less mesh testbed," in *Proceedings of the International Conference on Wireless Information Networks and Systems (WINSYS '08)*, July 2008.
- [18] K. Ramachandran, I. Sheriff, E. Belding, and K. Almeroth, *Passive and Active Network Measurement: 6th International Workshop, PAM 2005, Boston, MA, USA, March 31–April 1, 2005. Proceedings*, vol. 3431 of *Lecture Notes in Computer Science*, Springer, Berlin, Germany, 2005.
- [19] M. Boushaba, A. Hafid, and M. Gendreau, "Local node stability-based routing for wireless mesh networks," in *Proceedings of the 2013 IEEE Wireless Communications and Networking Conference (WCNC '13)*, pp. 1950–1955, Shanghai, China, April 2013.
- [20] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Springer, New York, NY, USA, 2nd edition, 2002.
- [21] S. Lê, J. Josse, and F. Husson, "FactoMineR: an R package for multivariate analysis," *Journal of Statistical Software*, vol. 25, no. 1, pp. 1–18, 2008.
- [22] Tools for high-throughput data analysis, S. S., Principal component analysis: How to reveal the most important variables in your data? -r software and data mining, World Wide Web, 2015, <http://www.sthda.com/english/wiki/principal-component-analysis-how-to-reveal-the-most-important-variables-in-your-data-r-software-and-data-mining>.
- [23] S. Dobson, S. Denazis, A. Fernández et al., "A survey of autonomous communications," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 1, no. 2, pp. 223–259, 2006.
- [24] F. Fainelli, "The openwrt embedded development framework," Tech. Rep., 2008, <http://downloads.openwrt.org/people/florian/fosdem/fosdem.pdf>.



Hindawi

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

