A Distributed Challenge Detection System for Resilient Networks

Yue Yu

School of Information Technologies
University of Sydney
NSW 2006, Australia
tinayu@it.usyd.edu.au

Abstract. The network has become essential to our daily life. With the increase in dependence, challenges to the normal operation of the network bear ever more severe consequences. Challenges include malicious attacks, misconfigurations, faults, and operational overloads. Understanding challenges is needed to build resilience mechanism. A crucial part of resilience strategy involves real-time detection of challenges, followed by identification to initiate appropriate remediation. We observe that the state-of-art to challenge detection is insufficient. Our goal is to advocate a new autonomic, distributed challenge detection approach. In this paper, we present a resilient distributed system to identify the challenges that have severe impact on the wired and wireless mesh network (WMN). Our design shows how a challenge (malicious attack) is handled initially by lightweight network monitoring, then progressively applying more heavyweight analysis in order to identify the challenge. Non-malicious challenges could also be simulated by our network failure module. Furthermore, WMNs are an interesting domain to consider network resilience. Automatic detection and mitigation is a desirable property of a resilient WMN. We present guidelines to address the challenge of channel interferences in the WMN. The feasibility of our framework is demonstrated through experiment. We conclude that our proof-of-concept case study has provided valuable insight into resilient networks, which will be useful for further research.

1 Introduction

With the growth of networks and the integration of services, increasingly severe consequences come from the disruption of networked services. Quality of life, the economic viability of businesses, and the security of nations are directly linked to the resilience, survivability, and dependability of the global network. However, the network becomes vulnerable with the increased dependence and sophistication of services. The scale of growth and deregulation bringing numerous service providers has resulted in a network that is difficult to manage. There is a pressing need for better resilience, manageability, and security for the future network [1]. Therefore, our research in distributed challenge detection is aimed to make networks more resilient to various challenges. Resilience means the ability of the network to provide an acceptable level of service in the face of challenges to normal operation. This

X. Zhang and D. Qiao (Eds.): QShine 2010, LNICST 74, pp. 584–597, 2011.

service includes the ability for users to access information, the maintenance of end-to-end communication, the operation of distributed processing and networked storage. The challenges that may impact normal operation include unintentional misconfigurations; malicious attacks; environmental challenges; unusual but legitimate traffic; provider failure. Therefore, the definition of resilience is a superset of commonly used definitions for survivability, dependability, and fault tolerance [1].

The main objective of the EU ResumeNet (Resilience and Survivability for Future Networking: Framework, Mechanisms and Experimental Evaluation) is to propose a multilevel, systemic, and systematic approach to network resilience. To achieve this, our approach addresses the challenge diagnosis problem as followings: it first monitors traffic for anomalies in real-time, which is online, when traffic traverses the network, rather than processing trace files offline. It is too costly to undertake detection operate all the time to perform the root cause analysis of ongoing challenges. They should only be enabled after the detection of the basic symptoms that may lead to the anomalies arise. Therefore, for example, once a link monitor detects a possible anomaly, the alert will be generated in the core router of the infected sub-network, which effectively shrinks the network range for detection. The detection is followed by instantaneously locating the victim. In addition, the network failure scenario is considered. Such a challenge based on the node and link failures can impact single or multiple network components [2], area-based challenges could affect multiple network elements.

In wireless mesh networks (WMNs), challenges are different from those in other networks, and can also have more severe impact than wired network. This is due to the inherently less reliable (compared to Ethernet) wireless technology and high reliance between the mesh elements. Furthermore, it is more vulnerable than wired networks due to the broadcast nature. WMN is an appealing technology for flexibly interconnecting computers. In contrast to wired network, it is a cost effective and simpler solution for rural areas. The WMN could build a resilient infrastructure via a combination of wireless network and ad-hoc routing protocols. With such deployment, problems on the physical and link layer are more likely to occur and have bigger impact. In this paper, we focus on wireless interference, as the big impact challenge needs to be met since it could affect the QoS of individual applications and bring severe trouble to the network [3]. Our work investigates a more systematic approach to meeting challenges to the network than has occurred before.

The paper is organized as follows. In Section 2, we review the shortcomings with the current state-of-art to challenge detection, and highlight aspects requiring additional work. The framework of the distributed challenge detection system is introduced in Section 3. In Section 4, we compare different experimental platforms, and explain why OMNeT is the most suitable network simulator. We demonstrate the simulation of our distributed challenge detection system in Section 5, and explain how we are populating our resilience strategy with the new network monitor, network failure and anomaly detection mechanisms. In Section 6, our approach is validated on the OMNeT and the interference challenge in the WMN will be discussed. The current work is summarized and future directions are concluded in section 7.

2 Related Work

The vulnerabilities of the current network and the need for resilience are widely acknowledged. There has been considerable research into network monitoring, anomaly detection, fault tolerance, attacks, anomaly modeling engine separately. We reviewed the latest technologies in these areas. With the current network monitoring techniques, threshold based random walks for fast portscan detection is unscalable [4]. The proposed (threshold crossing alerts) TCAs requires the cooperation of the manufacturers to run on the network devices, which will be difficult. Or they can run on separate hardware, which will be complex [5]. Jackson et al. [6] cope with the distributed monitor problem in internetworks, but the capability to monitor every link cannot be assumed. Today most detection use signature-based IDS that detect known attacks only. In contrast, anomaly detection is effective in foiling known and unknown attacks. Real time volume based anomaly detection is resource challenge [7]. The causes for DoS attacks and mechanisms for defending is surveyed [8], however it is not yet practical to identify attack paths and we require global cooperation to combat (Distributed denial of service)DDoS attacks. To evaluate the impact of faults, fault injection is considered as the first stage. It offers a cost and time effective way to test system. The method to inject faults into the real network is proposed in [9]. Other approaches are presented to study network survivability. Random events affect node and link availability, so cause the failures. There is a complete survey of fault localization [10]. Open research problems still remain with multi-layer fault localization, temporal correlation, and distributed diagnosis.

In the wireless domain, the completely decentralized networks WMN depend on every node to provide packet forwarding services for normal operation. The need for collaboration is highlighted in such network due to the lack of a central entity to supervise the activity of the nodes. The challenges that are inherent in the wireless domain include weakly connected channels, mobility of nodes in an ad-hoc network, and unpredictably long delays, radio interference and error-prone links [2]. Furthermore, WMNs are particularly susceptible to node selfishness. The key challenge discussed in this paper is interference and two general approaches could be applied to simulate interference and radio propagation, by using a complex and computational expensive model [11] or a simple one with the risk of gaining misleading conclusions [12]. Therefore, we need a method to recreate interference traffic from real measurements and seamlessly build into the network simulator. To conclude, so far yet little has been done to systematically embed resilience into the future network or develop novel, distributed mechanisms for monitoring resilience to detect challenges as they occur. Therefore, our research contributes towards the development of new experimental systems to perform challenge detection.

3 Distributed Challenge Detection Framework

In the ResumeNet project, we are investigating a framework for resilient networking. Resilience is required to be a key property of future networks because of our unrelenting demand for network services, the challenging environments, and the continued existence of intelligent adversaries. Two corresponding approaches are adopted: network and service resilience. Our work focuses on the network resilience, which is adding resilience to the services the system provides [13]. From the project's outset, we should understand the complexity of systems, the various challenges and its root cause. Assessing which challenge affect the system in which way is essential to deciding the corresponding mitigation strategy. For example, a web server could be overloaded in the short term with many requests, greater than what is provisioned for, and this could be a DDoS attack or a 'flash crowd' event, which is the unusual but legitimate demand for service. We need to distinguish between challenges that have similar symptoms, since they require different remediation. To do this, the first task is to construct a network that could tolerate foreseen adversarial events. The optimal topology needs to be designed. It utilizes the best possible way to interconnect the access nodes with the corresponding transmission technology. The fine tuned routing mechanism allow the good conditioned connection be built so that they could react to the failure quickly. Then we need a distributed monitoring and assessment platform that could detect network anomalies. This is because no perfect protection is provided by the defensive measures [13]. The unforeseen challenge will degrade the service.

Our proposed resilient network system could be applied in the heterogeneous network, which covers the wired and wireless network. In rural areas, WMNs are often deployed as the affordable and simple way to access internet, challenges could lead to a service such as internet connectivity being impaired or unacceptable depends on the severity. The in-depth understanding with the nature of the system and its challenges is significant to find the matching identification method and improve the resilient network. In WMN, various challenges could arise to the normal operation. They are particular vulnerable to infrastructure-based attack as a result of the relative simplicity of physical access to the mesh. Misconfiguration of devices could be a major issue when non-expert users manage the network. In addition, WMNs are particular vulnerable to the elements. Figure 1 listed the potential challenges that could affect WMN [14]. The hardware resources that can be used to detect the challenges may have wildly varying computational capabilities [3]. Nevertheless, it is not easy to prevent these challenges from leading to significant network outages.

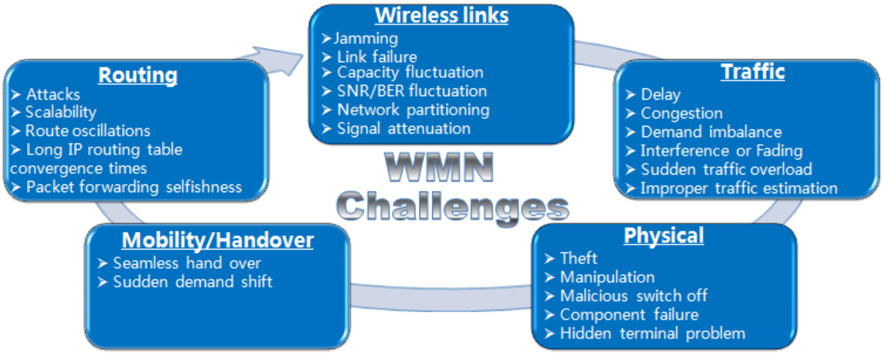

Fig. 1. Wireless Mesh Network Challenges

4 Comparison of Experimental Platforms

Building a distributed challenge detection system in the real world is a challenge. Firstly, a large topology network is required to get meaningful results. Secondly, such large topology will increase the hardware cost and administration effort. Thirdly, it is not convenient to experiment with attacks on the current available real world platform, e.g. AARNET, PlanetLab, since it's hard to ensure the experiment will not disrupt the normal operational network, which will cause more severe effect. Fourthly, we need to have full control with all the system nodes so that they could easily be configured to suit our case [15]. However, this is also difficult to realize.

Another possibility is to use off-line IDS datasets such as DARPA [16], KDD Cup, which could help to relieve from the real world difficulties but with the realistic dataset. DARPA intrusion detection is collected between 1998 and 2000 from Lincoln Lab. The 1998 version covers 38 attack types. The data slightly improved in 1999, so that 201 instances of 56 attack types distributed. Whilst the 2000 scenario specific datasets include two attack situations. KDD Cup dataset gathered in 1999 with 41 features. After investigation we decided not to use them. Because first, the datasets are not up to date, so that the most recent and the unforeseen attacks couldn't be measured. Secondly, the DARPA dataset was fundamentally broken due to numerous irregularities [17]. Thirdly, the performance measure applied in DARPA'98 evaluation, ROC curves, has been widely criticized [18]. Lastly, it was still useful to evaluate the true positive however any false positive results were meaningless [19].

Network simulators could overcome all the mentioned hurdles and meet the needs by integrating real world applications. However, it still requires us to compare different simulators to recognize the most appropriate environment. So we surveyed the widely applied simulators, NS-2, NS-3, OMNeT, SSFNet, JiST/SWANS and J-Sim. NS-2 and OMNeT are continuously supported today. NS-3 is the latest updated platform developed from 2008. While SSFNet, JiST/SWANS and J-Sim are nearly inactive since 2004, 2005, 2006 respectively. Our project needs the updated simulator, so NS-2, NS-3, OMNeT could better suit. Moreover, our work needs large topology, but NS-2 has the scalability issue with memory usage and simulation run-time [20]. However, NS-3 and OMNeT are scalable. Considering the run time, JiST/SWANS are the fastest, whilst J-Sim is the slowest. OMNeT is slower than NS-3 but faster than NS-2 [20]. NS-3 has lowest computational and less memory demands whereas JiST/SWANS exhaust memory [21]. OMNeT consume more memory than NS-3 but less than NS-2. In terms of GUI, NS-3 and NS-2 are relying on source code but OMNeT has a rich GUI with online visualization.

NS-2, NS-3 and OMNeT are widely used in wireless network simulation. Dedicated Short Range Communications (DSRC) is a good medium for inter-vehicle communications. The features of the simulators applied in DSRC are listed in [22]. Many researchers did DSRC related simulation based on NS-2 [23], NS-3[24] or OMNeT [25]. Among them, OMNeT and NS-2 are the most mature ones. We also studied the survey on the performance of wireless network simulators [26]. OMNeT reduce the complexity, and become an excellent tool for wireless network simulation as a result of its scalability, efficiency and the simplicity of modifying the network properties. OMNeT simulation API is more powerful than NS-2's. NS-2 is only lightly maintained now. NS-3 will eventually replace NS-2, but it is not backward compatible. It is trying to avoid some problems with NS-2. The NS-3 goals include some features (e.g. real-life protocol, parallel simulation) that have already proven to be useful with OMNeT. Moreover, the new animators, configuration tools and etc. are still in work. In addition, OMNeT offer basic modules, which is extensible using C++, whilst NS-2 is not easily modifiable. OMNeT provides well online documentation and active discussion forum. Overall, our preferred platform should be updated and reliable, so the flexible and generic simulator OMNeT best suits us.

5 Distributed Challenge Detection Simulation

To simulate various challenges, complex simulation scripts are needed to model the network protocols, topology, and the challenges. The challenges are separated into malicious and nonmalicious challenges. The malicious challenge (e.g. DDoS attack) could be monitored by network monitoring models and detected by anomaly detection models. The challenges like operational mistakes, unintentional misconfiguration, accidental fiber cuts, and node failures could be grouped as nonmalicious challenges. This type of challenge represents most adverse events observed in practice and could be simulated as random node and link failures [27], which cause the network failure.

5.1 Network Monitoring

Distributed network monitoring is needed to detect coordinated attacks. The module we developed to perform the traffic monitor on the link is called linkmonitor module. The monitor could continuously collect traffic information so that values display on the link in real time. Our development based on the OMNeT cDatarateChannel. After programming and debugging with the ThruputMeteringChannel, we further extend its function to monitor threshold. To gain a comprehensive overview of the link we monitor, the display could be customized with different attributes. Properties such as link color, propagation delay, data rate, can be assigned to connections. The characters could be monitored include packets number, current packet/sec, average packet/sec, current bandwidth, average bandwidth, channel utilization, average utilization, traffic volume, threshold. Channel utilization is the ratio of current traffic to the maximum traffic, which assists to understand the network performance and troubleshoot failures. This module implemented as the channel so that offers the flexibility to collect information from any link within any network to gain the clear detailed view of its typical behavior. The threshold monitor could notify that a certain parameter has exceeded a certain threshold and direct attention to those areas, so we could be immediately alert. It could effectively evaluate the network traffic to pinpoint the sub-network where victim located, and meanwhile prevent superfluous and incorrect alerting. The traffic values not only display on the link and shown in the module output in real time, but also recorded into the output vector file in OMNeT, which could be traced back to analyze why and when the anomalies behavior happens. The output vector captures traffic over time. The collected historic data demonstrate the network behavior in terms of performance and reliability. In addition, real-time statistics are important for detailed in-depth analysis. To ensure no false alarm created by the flash crowd also the alert could be raised immediately after the attack, we embed the timer function into the link monitor. The alert won't be generated unless the traffic above threshold for the continuous reasonable interval time. This method effectively avoids the events caused by the flash crowd.

5.2 Network Failure

Part of the network resilience strategy is enhancing the resilience to the network failures also modern networks should design to be fault-tolerant. The fault of the systems component could be another source of event pose the threat to the normal operation. It could lead to the network failure. There is a potential large set of faults. We simulate the faults that could be triggered by the nonmalicious challenges. So the flexible network could offer quick and efficient fault management techniques to provide network survivability. The network failure will result in the packet loss. With a broken link, the packets will be discarded until a new connection is rebuilt. Then the fault restoration will be used, once the network failure is identified, the backup path will be built immediately with the dynamically allocated spare capacity. The overall process of failure recovery shouldn't cause long delay so that could ensure the network robustness. With the above understanding, we simulate the network failure situation. The connection failure could appear in any place within the network structure. We designed connection failure channel based on the cDatarateChannel, which has the flexibility to be placed as a channel between any network objects. We could schedule connection failure event happen at certain simulation time, after it recovers, we could schedule another failure at another time as occasional failure could occur more than once on the same connection in the real network. We could include multiple concurrent connection failure channels in the network. When the connection failure happens, no packet could send through. The monitoring system should quickly raise the alarm once the broken link discovered. It means the failure detection time should be short. Then a real-time solution will be triggered.

5.3 Anomaly Detection

The anomaly detection module has been developed for the evaluation of attack detection and traffic analysis. As the linkmonitor offers the chance to get close to the victim by raising the alarm on the sub network where victim positioned, hence could perform efficient filtering. So the anomaly detection module use the simple algorithm to identify the victim, this effectively reduce the computation complexity and cost. The anomaly detection module implemented as the cSimpleModule and built into the INET compound module Router. In OMNeT, modules communicate by messages, which contain usual attributes as timestamp and arbitrary data. The cPacket class extends cMessage with fields to represent network packets (frames, datagrams, transport packets etc.) [28]. Simple module sends message through output gate. The output gate and input gate linked by a channel. Therefore, the message travels through the channel and arrives at the input gate of another simple module. The Compound module consists of several simple modules and transparently relaying messages between their inside and the outside world. The Router includes the modules NetworkLayer, Routing Table etc. The anomaly detection module interconnected with NetworkLayer, TCP, and UDP by incoming and outgoing gates through channel.

Every packet transfered into the router will pass the anomaly detection module for processing. The anomaly detection module will transparently process different network packets in a unified way. In addition, we use a hashing table to store the objects into the IDS table, and the table could iterate through. The IDS table could be monitored in real time as the module output, and the event be created immediately after identifying the destination IP address of the victim.

6 Experimental Results and Analysis

In this section, the system will be validated. The attack is injected across time to evaluate the performance. The interference challenge in the WMN will be discussed. The results demonstrate the accuracy, flexibility, scalability and efficiency of our method.

6.1 OMNET++/INET

OMNeT is a public source C++ based object oriented discrete event simulator for modeling communication networks, multiprocessors and other distributed or parallel systems [28]. It applies in diverse domains and written in two languages, NED designed for the network topology and C++ programmed for the modules. The compound module assembles from reusable simple modules. OMNeT utilize Tkenv as the GUI and it’s easily debugging and trace. It could animate the flow of messages and present the node state changes in the network charts. Build on OMNeT, INET extends it by package of network protocols and offers objects, which combined with the channels to complete the network. Testing our system contains two steps, create various attacks and detect the anomalies. When consider the background traffic generation, IDS testing is classified into four categories. Compared to no background traffic, real or sanitized background traffic, testing by generating background traffic approach has benefits such as data freely distributed, no unknown attack and repeatable simulated traffic [29]. So ReaSE is chosen as our realistic background traffic and DDoS attack generator. It extends INET by server and client entities.

6.2 Network Topology and Attack Implementation

To build our network, firstly the realistic AS level topologies is generated to connect several separate administrative domains. Each AS is categorized as stub AS or transit AS. One transit AS is built to provide connections through itself to other networks. The stub AS is connected to only one other AS. This ensures each AS is accessible by crossing transit AS only. Two stub ASes and one transit AS are configured, named SAS1, SAS2 and TAS0. SAS1 connect to SAS2 through TAS0. Secondly, the router level topology within each AS is specified. Each AS has core, edge and gateway routers placed. The distinction between different routers is realized by allocating different bandwidth. Within the AS, it has total min 8 routers and max 15 routers. A few meshed core routers with low node degree that forward aggregated traffic of a high number of gateway routers with high node degree [30]. Each edge router connects between 2 and 13 hosts to the network complete the hierarchical topology. Therefore, each AS has different topology sizes and fills with nodes independently.

Thirdly, the network built with different traffic profiles to ensure the reasonable mixture of various protocols. The traffic profiles covers web, Interactive, mail, misc and ping traffic, which are based on transport protocols TCP, TCP, TCP, UDP and ICMP respectively. The router level topology's host systems are classified into clients and servers. Clients correspond to the ReaSE module InetUserHost, whilst servers represents by Web, Mail, and Interactive server. Fourthly, the bandwidth between different types of nodes are assigned from ReaSEGUI, also we configure the server fraction value, which specify the percentage of all router modules of each router-level topology are replaced by special server nodes. Overall, 136 hosts and servers are placed cross the AS in our network. At last, since ReaSE integrate the real attack tool tribe flood network to conduct the DDoS attack, so it is utilized to perform a random distribution by replacing randomly selected clients InetUserHost with DDoS zombies. The compound module DDoSZombie contains simple module TribeFloodNetwork with other INET modules that are essential to achieve the functionality of an attacking system [30]. In our experiment, total 30 DDoS zombies are located across the AS, at simulation time 120s, the zombies conduct the attack based on TCP SYN packets, and 90% of the zombies collectively launch the attack by sending a fix rate TCP SYN packets to the victim Webserver27 which is in SAS1. Figure 2 shows the linkmonitor result within SAS1 right after attack. Two linkmonitor position on the ingress link core router0 to the gateway1 and gateway2 to core router0, as highlighted in red. When the threshold value turns to 1, it generates the alarm, and informs that the attack is detected on the ingress link core router0 to gateway1. The gateway1 connect to edge router7, that is also the router victim connect to.

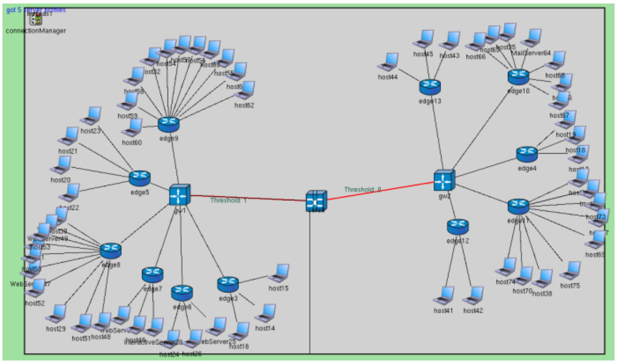

Fig. 2. Link Monitor on SAS1

In service level monitoring, except false alarm, another way to assess the quality of threshold detection is measure the delay between the time a crossing is reported and its actual occurrence [31]. Figure 3 demonstrates the result of the threshold monitor on ingress link core0 to gateway1 router. The alarm is arisen by linkmonitor at 127s.

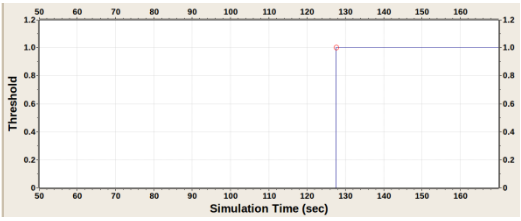

Fig. 3. Threshold Monitor

Our detection technique could identify the victim in a short time. When use linkmonitor to monitor the traffic on the ingress link as shown in Fig 4, the DDoS ramp-up behavior could clearly be observed between 120s and 140s, as each zombie is configured to delay its start for a uniformly distributed time from 0 to 20s. After the threshold raises alarm at 127s, the attack identified the victim IP 0.2.0.28 at 132s.

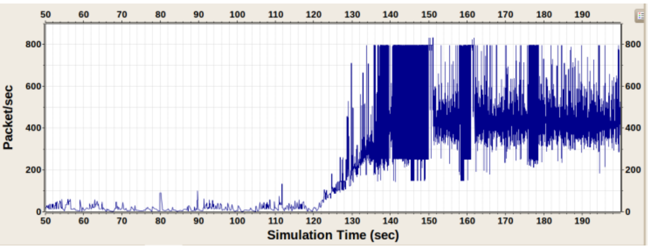

Fig. 4. SAS1 Core Router Traffic Monitor

6.3 Challenges in the Wireless Mesh Networks

There are numerous challenges need to be met in WMNs. It is impossible to look into all the challenges at once, so here we focus on one key challenge, that is interference, and demonstrate the approach to effectively address it in WMN. Because of interference, a challenge causes high traffic between two nodes could affect the available bandwidth between two other nodes. One or more root causes could result in interference which is not easy to identify from the initial symptom. Because of the mesh structure, interference could not only cause by the network itself but also other APs in proximity. Therefore, the detection will request the neighboring nodes to support each other for identifying the challenge then optimize the whole system state as the mitigation strategy. The interferer could easily and correctly detected by investigating how it appears for different nodes and the remediation may request the changing of wireless channel in the whole system [3]. Usually the interference could occur at either the receiver or sender. Interference at receiver caused by the appearance of another transmission that results in a sent frame could not receive directly. Whereas the interference at the sender is the result of the continuing transmissions from other nodes preventing the sender from sending a frame. The interference could not easily be reduced to the physical interference between multiple frequencies, and complex interactions between MAC protocols also involved especially since some of them are very particular to certain manufacturers. There are diverse set of symptoms that indicate the interference, such as SSID mismatch, element unknown, high on-chip retries, bad CRC, channel mismatch [32].

OMNeT relies on external extensions to implement the wireless ad hoc networks. The two major ones are the INET Framework (IF) and the Mobility Framework (MF). The latter is an extension explicitly designed for mobile ad hoc networks [26]. To build the WMN in OMNeT, the simulation is produced by combining the approaches of several existing frameworks into one: the protocol library is obtained from the MAC simulator and the MF; the mobility support, connection management and general structure is taken from the MF module [33]. There are several current available modules could be integrated into the WMN, which include nodes, radio propagation models for multiple signal dimensions, physical layer, receivers and an extensive library of MAC and network protocols. A key element used for the radio channel communication is the channel controller module, which handle the radio propagation, record on-going transmission and offer information for radio devices to use the reception and interference model. To simulate the interference challenge, we need to find a method to integrate the interferences generated from real measurements in a transparent way into the OMNeT INET framework. This could base on the work in [34] where the interference scenarios are represented in two dimensions. On the spatial dimension, the traffic injected is received with a calculated reception power by the wireless nodes. On the temporal dimension, two interference scenarios are produced: 1st, the simulated system respond to the interfering traffic, but has no interaction with the interfering sources; 2nd, there is mutual interactions between the simulated system and the interfering traffic. The typical WMN contains several wireless access points (APs) that route packets from clients to their destination, usually a set of egress points to the internet. Figure 5 shows the WMN simulation, it has four types of nodes, AP1 and AP2 are the clients' access points, and each connects five clients. AP4 is a mesh node. AP3 is the Internet gateway. The simple interference model is depicted in Figure 6. Host1 connects to AP1, Host2 communicates to AP2 separately. AP1 and Host1 combined as an interferer.

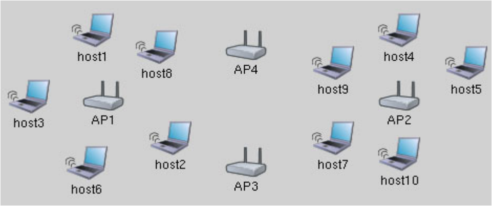

Fig. 5. WMN Simulation

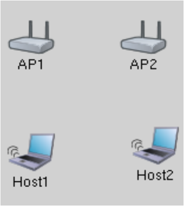

Fig. 6. Interference Model

Considering the WMN resource constraints and to minimize message overhead, we separate detection and remediation into stages. The complex investigation which involves other hosts is triggered to fully recognize the challenge only after local detection of the symptom. The initial detection could use the lightweight machine-learning classification algorithm with low false alarm rate to identify as much interference as possible. We need to find the best solution to minimize the interference by comparing different solutions and accessing its influence. After this stage, further analysis will perform to gain more understanding with the challenges. All the available networks will be detected by network scan locally. In addition, with the distributed cooperation, we could gain accurate estimation of the node utilization in the detected networks. This analysis could effectively avoid the collision caused by different nodes switched to the same channel, so only one neighboring node is responsible for finding a new channel allocation. The two connected nodes are required to have different channels. To select the best channel, we need to evaluate the cost associate with interference and channel changes. We also need to consider the communication on one channel could affect the transmission on the adjacent channel. When minimizing the interference, the channels should be changed in all the nodes simultaneously. Otherwise one channel change could cause the interference at another node, which result in once more channel change and interference, produce a circle. However, changing the channel on an AP need clients to follow the channel that might causes the wireless connection unusable for a short time. Therefore, it is suboptimal to simply switching to the current best channel at every occurrence [33].

7 Conclusions and Future Work

In this paper, we present a new distributed challenge detection system for network resilience. Currently, three activities are carried out by our system: real time network monitoring, detection of the challenge symptoms, and challenge identification. We have surveyed the state-of-art and highlighted their shortcomings. We propose that a systematic approach to resilience is required to consider the complete socio-technical system and the challenges it may face [13].

Our experimentation defined with the aim of addressing various resilience issues in the context of different types of networks and service provision settings. We present a simulation framework to simulate realistic challenges, specifically intelligent attacks and non-malicious challenges that go well beyond the network failure. Currently we validate our system on OMNeT with the wired network and propose an approach to address the significant challenge interference in the WMN. Future work will involve further developing the scenarios we want to evaluate our work through, e.g. we may examine how various types of challenges can influence the wireless networks at local and global level. As WMNs have to cope with a much wider range of challenges than the wired network, the ongoing work for WMN will require further implementation of the current architecture so that corresponding remediation could be carried out. In addition, besides the appropriate identification of a resource starvation attack on an ISP's infrastructure caused by high volumes of traffic from a DDoS attack, other types of attacks will also be considered. In future, fault management is one of the major components of the network management suite. We need to introduce innovative concepts for fault detection, root cause analysis and self-healing architectures. We hope the system could implement root-cause analysis to detect faults once they occur, and also to identify the source for performing automatic fault recovery. Different types of network service faults will be considered: they range from node misbehavior at different network layers, to software misconfigurations.

This paper elaborates an initial proof-of-concept implementation with the understanding of how to ensure resilience for a future network. Our work will persist in the context of strong experimental scenarios that we believe will feature in a future network. Through these scenarios, we will evaluate the validity of our strategies for resilience. We hope this project could have a broader socio-economic impact by contributing to the development for the future internet.

Acknowledgements. This work has been supported by NICTA (National ICT Australia). This research is carried out under the European Union Research Framework Programme 7 via the ResumeNet project with contract no. FP7–224619. The author is grateful for the guidance of Michael Fry. Also the author would like to acknowledge Aruna Seneviratne for his support and thank all the partners of the ResumeNet project for their contribution to this work. In addition, I thank Thomas Gamer and J.C. Maureira for their useful discussions on the simulation.

References

[1] ResumeNet, http://www.resumenet.eu/

[2] Doerr, C., Omic, J., et al.: Resilience and Survivability for future networking: framework, mechanisms, and experimental evaluation, ResumeNet Deliverable D2.1b (2010)

[3] Smith, P., Fry, M., et al.: Resilience and Survivability for future networking: framework, mechanisms, and experimental evaluation, ResumeNet Deliverable D2.2a (2010)

[4] Jung, J., Paxson, V., Berger, A., Balakrishnan, H.: Fast portscan detection using sequential hypothesis testing, pp. 211–225. IEEE, Los Alamitos (2004)

[5] Wuhib, F., Stadler, R.: Decentralised Service-Level Monitoring Using Network Threshold Alerts. IEEE Communications Magazine, 44 (2006)

[6] Jackson, A.W., Milliken, W., Santivanez, C.a., Condell, M., Strayer, W.T.: A Topological Analysis of Monitor Placement, pp. 169–178. IEEE, Los Alamitos (2007)

[7] Fry, M., Fischer, M., Karaliopoulos, M., Smith, P., Hutchison, D.: Challenge identification for network resilience. IEEE, Los Alamitos (2010)

[8] Peng, T., Leckie, C., Ramamohanarao, K.: Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems. ACM Computing Surveys 1, 39 (2007)

[9] Labovitz, C., Ahuja, A., Bose, A., Jahanian, F.: Delayed internet routing convergence. IEEE/ACM Transactions Networking 9, 293–306 (2001)

[10] Steinder, M., Sethi, A.S.: A survey of fault localization techniques in computer networks. Science of Computer Programming 53, 165–194 (2004)

[11] Qiu, L., Zhang, Y., Wang, F., Han, M.K., Mahajan, R.: A general model of wireless interference, pp. 171–182. ACM, NY (2007)

[12] Kotz, D., Newport, C., Gray, R. S., Liu, J., Yuan, Y., Elliott, C.: Experimental evaluation of wireless simulation assumptions, Technical Report, Dartmouth College (2004)

[13] Fessi A., Plattner, B., et al.: Resilience and Survivability for future networking: framework, mechanisms, and experimental evaluation, ResumeNet Deliverable D1.5 (2009)

[14] Doerr, C., Smith, P., et al.: Resilience and Survivability for future networking: framework, mechanisms, and experimental evaluation, ResumeNet Deliverable D2.3a (2010)

[15] Mayer, C.P., Gamer, T.: Integrating real world applications into OMNeT, Institute of Telematics, University of Karlsruhe, Karlsruhe, Germany (2008)

[16] Lippmann, R., et al.: The 1999 DARPA Off-Line Intrusion Detection Evaluation. Computer Networks 34(4), 579–595 (2000)

[17] Mahoney, M.V., Chan, P.K.: An analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for network anomaly detection. In: Vigna, G., Krügel, C., Jonsson, E. (eds.) RAID 2003. LNCS, vol. 2820, pp. 220–237. Springer, Heidelberg (2003)

[18] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.: A Detailed Analysis of the KDD CUP 99 Data Set. IEEE, Los Alamitos (2009)

[19] Brugger, T.: KDD Cup 1999 dataset considered harmful, White Paper, Department of Computer Science, University of California Davis (2007)

[20] Weingartner, E., vom Lehn, H., Wehrle, K.: A performance comparison of recent network simulators, pp. 1–5. IEEE, Germany (2009)

[21] Kargl, F., Schoch, E.: Simulation of MANETs: A qualitative comparison between JiST/SWANS and NS-2. In: International Workshop on MobiEval (2007)

[22] Young, C.P., Chang, B.R., Chen, S.Y., Wang, L.C.: A Highway Traffic Simulator with Dedicated Short Range Communications Based Cooperative Collision Prediction and Warning Mechanism. IEEE, Los Alamitos (2008)

[23] Schmidt-Eisenlohr, F., et al.: Cumulative Noise and 5.9GHz DSRC Extensions for ns-2.28, University of Karlsruhe, Tech. Rep. (2006)

[24] Johansson B., et al.: Highway Mobility And Vehicular Ad-Hoc Networks In NS-3, CiteSeerX (2010)

[25] Eichler, S.: Strategies for pseudonym changes in vehicular ad hoc networks depending on node mobility. In: Proceedings of the 2007 IEEE Intelligent Vehicles Symposium (2007)

[26] Orfanus, D., Lessmann, J., Janacik, P., Lachev, L.: In Performance of wireless network simulators: a case study, pp. 59–66. ACM, New York (2008)

[27] Cetinkaya, E.K., Jabbar, A., Mahmood, R., Sterbenz, J.P.G.: Modelling Network Attacks and Challenges: A Simulation-based Approach. In: EDCC, Valencia, Spain (2010)

[28] Varga, A.: OMNeT++ User Manual, `http://www.omnetpp.org/doc/manual/usman.html`

[29] Mell, P., Hu, V., Lipmann, R., et al.: An Overview of Issues in Testing Intrusion Detection Systems, Technical Report, National Institute of Standard and Technology (2003)

[30] Gamer, T., Scharf, M.: Realistic Simulation Environments for IP-based Networks. In: ICTS (2008)

[31] Wuhib, F., Stadler, R.: Decentralised Service-Level Monitoring Using Network Threshold Alerts. IEEE Communications Magazine, 44 (2006)

[32] Smith, P., Fry, M., et al.: Resilience and Survivability for future networking: framework, mechanisms, and experimental evaluation, ResumeNet Deliverable D2.2b (2010)

[33] Rasheed, T.: Wireless Mesh Network Simulation Framework for OMNeT++, Create-Net Technical Report (2007)

[34] Maureira, J.C., Dalle, O., Dujovne, D.: Generation of Realistic 802.11 Interferences in the Omnet++ INET Framework Based on Real Traffic Measurements. In: ICST (2009)