IOP Conf. Series: Materials Science and Engineering **433** (2018) 012085 doi:10.1088/1757-899X/433/1/012085

Network Topology Detection System

***G. B. Abdulhussein[1], T. Kanakis[2], A. A. R. Altahir[3]**

[1] Computer Network Engineering Dept. - Faculty of Arts, Science & Technology – University of Northampton, Northampton, UK
[2] Faculty of Arts, Science & Technology – University of Northampton, Northampton, UK
[3] Electrical Engineering Department, Faculty of Engineering, University of Kerbala

Email: moonlight950@yahoo.com

Abstract. Detecting network topology is essential to cope with any existing or future network changes such as the loss of network devices (routers, switches, or hubs), interfaces going down, or cable issues, any if which might change packets' paths, interfere with the network structure, interface status, or neighbour status, or disconnect and reconnect devices. This paper develops a low-cost network detection system that is responsible for discovering network topology by identifying each network device's neighbours to allowed the network administrators to know where each device and its neighbours are located, maintaining these devices' information in a local database to be used for visualisation purposes to demonstrate relevant changes inside the devices such as IP address, interfaces, and neighbours by using a configured hardware device. This system can be implemented in any large to medium size of network, measured depending on the number of users that join such a network. In addition, it can be used to remotely fix any minor issues that are detected which may later affect the entire network. The experimental results substantiate the fact that the system resolves the high cost of hardware experienced by previous work in this field.

1. Introduction

Discovering network topology is very significant, as increasing companies' network structure makes them become bigger and more complex, and thus inserting network data into a database manually has become difficult. As well as dealing with any existing and future adjustments in network structure, detecting and resolving network failures takes time to achieve [1] and [2]. There are a number of hardware devices that can perform this function. However, the hardware device price is generally high compared with the network topology size, and the administrators are required to insert the network topology manually into these systems. The current system thus focusses on allowing the network managers to display more information on their networks by discovering connected devices using configured hardware; experimental work suggests that this depends on the available devices on the network, along with their features [3].

This paper is organised as follows: Section 2 examines previous work that had been produced in this field, while section 3 deals with the experimental work of this paper, including choosing the hardware and the programming language before implementing the proposed system on the chosen hardware device. Section 4 discusses the experimental results of the system, and includes some screenshots. Finally, section 5 draws some conclusions about the implementation of the system and the results, as well as offering future recommendations.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

2. Literature review

Some researchers have acknowledged previous art in the tools listed in their patent that embody the observation and control of network performance. Alongside the strengths of their tool, there are a number of weaknesses about which they do not offer sufficient information, though these are readily distinguishable as the source of the problem in the network [4].

A current system that applies the same approach [5] illustrates the progress of the Customer Operation Service Management Interface Control system (COSMIC) in order to observe the private huge telecommunication used by the AT&T corporation network control centre. This showcased network health status monitoring using individual computers. Its major aptitude is displaying simultaneous network management outcomes. However, the architecture of the hardware is relatively expensive.

LAN status monitoring was investigated in [6]; however, this work merely considered token ring, FDDI, and Ethernet networks. In [7], an effective and automatic network monitoring system that continuously monitors network switches to inform the network managers by SMS or Email when a switch is lost by using smart communication Request Tracker (RT) and Nagios software available in a Linux environment was investigated; more negatively, this system generates tickets for each failure and only takes the issue's ticket into consideration when resolving it.

The authors in [8] aimed to monitor network architecture by installing a proxy system in all the devices of the network which acts as a requisite for interconnecting with the central management unit remotely. In each device, the data are gathered and sent to the central management unit over the internet as a message. The strengths of the patent are the discovery of failed devices in order to report these to the administrators. However, this system must be installed on each device and each device must be able to connect remotely with the central management unit, which means that whenever a new device joins the network, a new agent system must be installed.

The patent authors' in [9] indicated that their network management station used a tool and technique with the purpose of discovering and monitoring all the devices on the network simultaneously to display the outcomes in a desktop-based application with all associated information.

Based on the patent in [10], those researchers concentrated on a method to detect the topology of the network. The authors built the network from their own design, which was provided before applying their technique in order to create a comparison. Using the invented method will lead to detection of an actually constructed network topology, and matching the designed topology with the detected topology is the aim. This requires the designed topology to be identified for the matching step, and as such, this is restricted to applications on small networks.

Based on the patent in [11], these inventors focused on a technique to detect the topology of the network to produce a drawing that could be useful for both network managers and designers. The topology producer collects information that from both the ARP (Address Resolution Protocol) cache and MAC (Media Access Control) caches of the network devices. It depends on the gathered data to perform the topology detection stage and uncovers links or unknown domains in the network devices. The association data may be used to define the cables in the topology along with the type of network devices found.

In this paper, the focus is on the selection of a suitable hardware device to discover the construction of a network in order to access all the network devices, in order to create a main copy of the network topology which will be modified as network devices change. Moreover, a low-cost constraint will be applied for purchasing and implementing when compared with current hardware. This system will be developed to be harmonious with any medium to large network where the size of the network is measured in terms of the number of users that connect to the same network structure, including universities, companies, and other business sector networks.

3. Experimental Work

3.1. Hardware selection

A number of hardware devices could be used to implement the proposed system. Recently, the most popular hardware devices that have been used are Raspberry Pi and Arduino, due to their distinctive

features. Arduino does not have an operating system which can be run under Arduino IDE as a platform, however, so only one program could be run at a time, and so on. Table 1 shows the comparative technical features of Raspberry Pi and Arduino.

Table 1: Raspberry Pi and Arduino features

Feature	On Raspberry Pi	On Arduino
Backwards pin compatibility	40 pin GPIO connector	Programmed with FTDI breakout boards
Processor Choice	a Broadcom BCM2837 SoC with 1.2 GHz 64-bit quad-core ARM Cortex A53 processor	ATmega238P
Memory	1 GB RAM	32 KB
Operating System	Customised Debian Linux: Raspbian	None
price	Between $5 and $35	22.57$
Based on	Single board computers	Microcontroller based development boards

Based on these features, Raspberry Pi was selected for the experimental work. The price of a Raspberry Pi currently starts at around $5 (raspberry pi zero), rising to $35 for a board with a power supply without a case, HDMI cable, or RAM. Additional components will cause the price to rise. The cost of the basic Arduino board is higher, and as the Raspberry Pi is a micro-computer that runs under the Raspbian operating system, it can also be used as a separate system if provided with a mouse, keyboard, and monitor without requiring an additional operating system as in the case of Arduino. A number of ports are available to connect such items, and a number of programs can be run at the same time on Raspberry Pi, while only one program can be executed at a time with Arduino. Further, Raspberry Pi supports 802.11n Wireless LAN (Local Area Network) and Bluetooth and offer good open source documentation. [12-15, 3]

3.2. Programming language selection

Raspberry Pi supports a number of programming languages that could be used to detect network topology such as Python and Java, which are preinstalled with the operating system [14]. As Python is the most commonly used programming language, it can be used as an interpreter for another program, while Java requires a compiler that runs from the processor. It is also object-oriented, open source, and fast as compared with Java, which is slow in processing; Python was thus used to implement the proposed system [13, 16].

3.3. The experimental work of proposed system

A network detection system is used to access network devices such as routers and switches remotely via the connected network device, which is the router that connects to the entire network on one side and to the outside world on the other; to do this, the Raspberry Pi is connected using an Ethernet cable [3]. Figure 1 illustrates the connection between the case study network and the Raspberry Pi. Telnet [17] is the protocol used on the internet or Local Area Network (LAN) in order to offer interactive command line communication capability for remotely accessed computers [18] or network equipment [19]. Thus, this is used to obtain access to the connected network device via the Raspberry Pi, and through this connection, the whole network topology can be accessed, along with data such as IP addresses, interfaces, neighbours, and hostnames. This is done by reading the first accessed router's data in order to find its neighbour(s), and then generating a loop that goes through these one by one remotely, using Telnet commands to find other neighbours and data by sending different network commands to the accessed network device. Each time a new neighbour is found that has not been accessed previously, it is accessed in turn until the whole network's devices are stored in a database (DB). The program will execute repeatedly to find any network changes.fi

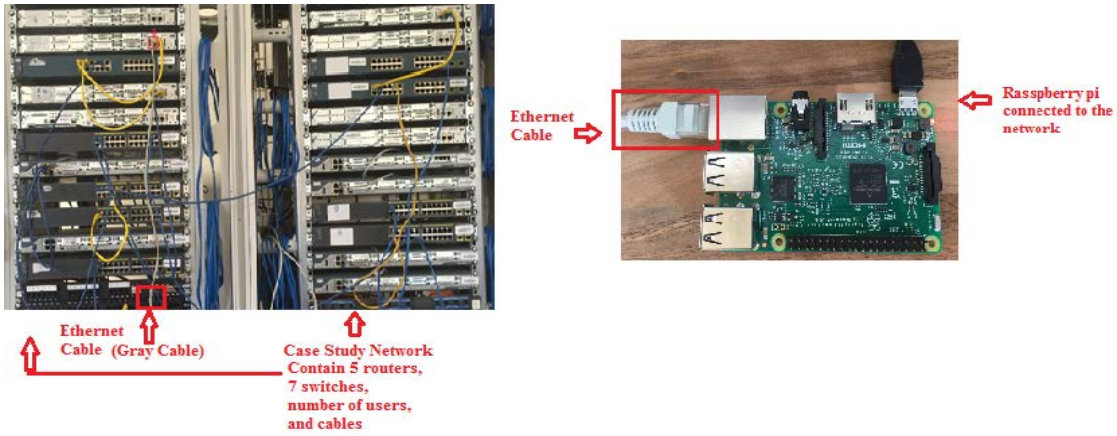


Figure 1: Case study network connected to Raspberry Pi

The initial connected network device's (first router) IP address must be known, and the network device credentials that allow remote connection through Telnet must be provided to the system in order to assess all devices. An overview of the proposed system is shown in figure 2.

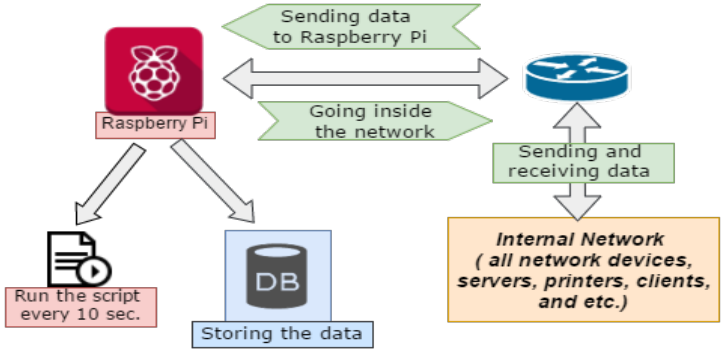


Figure 2: Network Detection System Overview

The main function of this system is to find the devices on the network through the main router connected to the Raspberry Pi; depending on the information from this device and its neighbours, any new device can be accessed remotely, which leads to the detection of other network devices such as routers, and switches, until the complete network topology is known. After that, the collected data are stored in a DB [20] which continues to close the connection with the current device to access another device until the whole network discovered; this process is then repeated every ten seconds after the whole network is determined [3]. The time for re-executing the program depends on the size of the network, which means that whenever the network increases, the time required increases as well. This experimental work was part of a master's dissertation [3], which used the DB in its later stages to create a website for visualisation, network health status tests, and real-time updates.

4. Experimental Results

The network topology was discovered using a Raspberry Pi in a format that connects to any medium to large size network using an Ethernet cable, which is authorised by the system's user to find all the attached devices. The data is collected via communication from the source to the destination. Whenever the hardware device accesses a network device, a new file is created or overwritten on the same file if it already exists in order to store and extract the network device data obtained through sending different network commands, as shown in figure 3; this includes the name of the device, IP addresses, interface status, and neighbours.

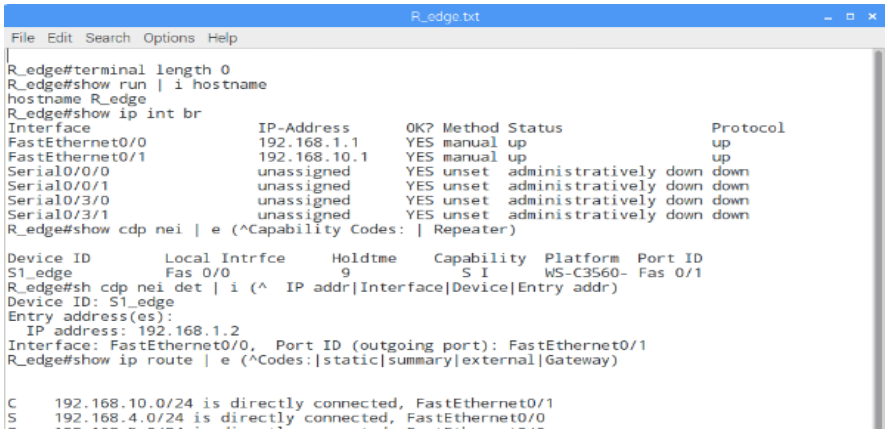


Figure 3: Network Device File Content (Router)

All the network topology data is stored in a local database that can be managed by the network administrator; part of this is listed in Table 2. This step has not been provided in previously published work, where databases were inserted manually into the system or only notifications that a device has a problem are provided [6]. High-cost hardware and software [5] solutions also cannot compare with this information gathering approach.

Table 2: Sample Database Results

Device_name	IP	Interface	int_status	int_portocol	network_add	connect_to	connect_to_int
R_edge	192.168.1.1	FastEthernet0/0	up	up	192.168.1.0/24	Switch	192.168.1.2
R_edge	192.168.10.1	FastEthernet0/1	up	up	192.168.10.0/24	NULL	NULL
R_edge	unassigned	Serial0/0/0	administrativelydown	down	None	NULL	NULL
Active_core	192.168.1.3	FastEthernet0/0	up	up	192.168.1.0/24	Switch	192.168.1.2
Active_core	192.168.2.3	FastEthernet0/1	up	up	192.168.2.0/24	Switch	192.168.2.2
S1_edge	unassigned	FastEthernet0/1	up	up	None	Router	192.168.1.1

Network detection time plus ten seconds is the time used for repeating the same process, though this could be changed depending on the network administrators' detection of network changes, This includes the storage of the network topology information in a database, which reads the contents of the file in detail. The repeating process time thus relies on the number of network devices. The experimental results demonstrate that the system can solve the high-cost issue of buying and implementing systems to allow any network to detect its network topology. Thus, the achieved outcomes of checking the system promote the significance of obtaining true network topology with a low-cost network discovery system when compared with the AT&T company hardware used to detect and monitor that company's network [5, 12]. However, the time of detection process remains an issue; this could be minimised by accessing the network devices simultaneously using a number of Telnet sessions.

5. Conclusion

A network detection system has been proposed to minimise the system costs, which matches network administrators' requirements. The results of testing the system support the relevance of obtaining accurate and filtered network device data at lower cost. In this paper, the discovery process of the devices connected to a network was carried out using a Raspberry Pi as a configured hardware device. Taking into consideration the fact that the core router might not have wireless signal, an Ethernet cable is essential to form a connection between the Raspberry Pi and the network device that connects the overall network with the outside world. The device data is then stored in a new or existing file using the name of the network device that has been accessed. After the process of discovering the first router, storing the extracted data from the network device is performed subsequently. When the first router's data is stored in the DB, its first neighbour is then accessed remotely and its device data stored, and so on until the whole network topology is discovered. As a result, this system solves the issue of the high cost of

existing systems that only the massive networks can effectively use and offers a system that can be used by any network from medium to large size [5]. The time for discovering a network's topology depends on its size, which means that every time the network grows in size, the time is increased. To decrease this time, a number of Telnet sessions could be run at the same time, as future recommendations to improve the system.
Future recommendations to improve the system include the fact that the database could be stored in the Cloud to offer a recovery method for the network topology when the company replaces hardware, and more information on the network could be used in order to provide network administrators with fuller data on each device.

References

[1] Oliveira O M F 2017 Network topology discovery *University of Aveiro Department of Mathematics, Thesis.*
[2] JENSEN R 2007 Network discovery system *U.S. Patent 20080155386A1*.
[3] Abdulhussein G B A 2017 Network Health Status Monitoring *University of Northampton, MSc Dissertation* (UK: Northampton).
[4] Fletcher R A, Banthia P C and Svensson A 2001 Integrated Computer System and Network Performance Monitoring *U.S. Patent 6,269,401 B1*.
[5] Chang L, Chan W L, Chang J, Ting P and Netrakanti M 1989 A network status monitoring system using personal computer *IEEE Global Telecommunications Conference 1989 and Exhibition Communications Technology for the 1990s and Beyond* vol. 1 pp. 201-206.
[6] Choi T and Tang A 1995 Enterprise network management: LAN status monitoring *In Communications, 1995. ICC'95 Seattle, 'Gateway to Globalization', 1995 IEEE International Conference on* Vol. 3, pp. 1448-1452.
[7] Khan R, Khan S U, Zaheer R, and Babar M I 2013 An Efficient Network Monitoring and Management *System International Journal of Information and Electronics Engineering* Vol. 3(1), pp. 122-126.
[8] Rackus P, Carter C, Fauteux J and Gilbert A 2005 Multi-network monitoring architecture *U.S. Patent 20050235058 A1*.
[9] Chin J, Tran H and Leong L 2002 Method and apparatus for displaying health status of network devices *Patent US 6,456,306 B1*.
[10] Anderson K W, Weichhold P and Simon C S 2005 Network Topology Discovery *US Patent 10/907,929*.
[11] Alexander W M, Shah D and Gibson W S 2013 Network Topology Discovery *US Patent 13/984,976*.
[12] Atwell C 2015 Top 10 Single Board Computers From $9 to $569 *EETimes.com*.
[13] Dennis A K 2016 Raspberry Pi Computer Architecture Essentials *Packt Publishing Ltd* (Birmingham, B3 2PB ,UK).
[14] Gay W 2014 Raspberry Pi Hardware Reference *Apress* (New York, NY 10013).
[15] Shah S 2015 Learning Raspberry Pi *Packt Publishing Ltd* (Birmingham, B3 2PB, UK).
[16] Donat W 2014 Learn Raspberry Pi Programming with Python *Apress* (New York, NY 10013).
[17] Postel J and Reynolds J 1983 Telnet protocol specification *[Online]. Available: https://tools.ietf.org/html/rfc854. [Accessed 22 Nov 2017]*.
[18] MORRIS H C and CALABRESE J S 2002 Detecting an active network node using a login attempt *U.S. Patent 6,347,339*.
[19] LEONG L and LEE F 2001 Method and system for monitoring remote routers in networks for available protocols and providing a graphical representation of information received from the routers *U.S. Patent 6,269,398*.
[20] MATHENY D, TAYLOR D and WINTERTON R 2001 System and method for collecting, aggregating, and coalescing network discovery data *U.S. Patent 09/846,521*.