# An Efficient Network Monitoring and Management System

Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Muhammad Inayatullah Babar

*Abstract.* **Large organizations always require fast and efficient network monitoring system which reports to the network administrator as soon as a network problem arises. This paper presents an effective and automatic network monitoring system that continuously monitor all the network switches and inform the administrator by email or sms when any of the network switch goes down. This system also point out problem location in the network topology and its effect on the rest of the network. Such network monitoring system uses smart interaction of Request Tracker (RT) and Nagios softwares in linux environment. The network topology is built in Nagios which continuously monitor all of the network nodes based on the services defined for them. Nagios generates a notification as soon as a network node goes down and sends it to the RT software. This notification will generate a ticket in RT database with problematic node information and its effect on the rest of the network. The RT software is configured to send the ticket by email and sms to the network administrator as soon as it is created. If the administrator is busy at the moment and does not resolve the ticket within an hour, the same ticket is automatically sent to the second network responsible person depending upon the priority defined. Thus, all persons in the priority list are informed one by one until the ticket is resolved.**

*Index Terms*—**Network Monitoring, Ticketing System, Reporting Back System, Nagios, RT.**

## I. Introduction

An efficient and automatic network monitoring is always required for large organizations like universities, companies and other business sectors where the manual network monitoring is very difficult [1],[2]. Since large organizations have a big network topology, the manual network monitoring causes waste of time to point out problem location [3]. The Multi Router Traffic Grapher (MRTG) has been extensively used for network traffic load monitoring. MRTG generates graph for all the nodes of the network topology from which the traffic load information can be accessed [4]. It consists of perl script which uses simple network management protocol (SNMP). Manual monitoring of all nodes of a huge network with MRTG is inefficient and time consuming.

The network monitoring scheme presented in this paper, make use of the smart interaction of Request Tracker (RT) and Nagios software to obtain an intelligent and automatic network monitoring system. This system is intelligent in a sense that it can specify the problem location in the network topology as well as its effect on the other nodes. If a parent node stops functioning then all the child nodes also become unreachable but problem notification of only parent node is sent to the administrator. Thus, this efficient network monitoring and reporting back system quickly inform the administrator about the network problem location [5].

The role of network monitoring is performed by the nagios software [6]. The whole network topology is constructed in nagios [7]. The administrator apply different services on the network nodes that are to be monitored by nagios software. Nagios continuously monitors all network nodes and generate notification when a node goes down after making a pre-defined number of attempts [8],[9]. The key role in network management is performed by the RT software which manage the tickets generated by the nagios software. RT is heavily used worldwide and can be customized and configured according to the organization needs. RT perform several important functionalities like it can provide multiuser interface, authentication and authorization to the organizations [10].

The rest of the paper is organized as follows. Section 2 describes characteristics of ideal network monitoring system. Section 3 briefly describe the automatic network monitoring system. Network monitoring by nagios and its configuration are explained in section 4. Section 5 describes the importance of RT ticketing system and its configuration for network management. Section 6 describes the interfacing of nagios with RT. Section 7 explain important results obtained. Finally, section 8 concludes the paper.

## II. Ideal Network Monitoring System

The term ‘network monitoring’ describes a system that continuously monitor the whole network topology for jamming, slowing down or failing components and notifies the network responsible person via email, sms or other alarms in case of any problem [1]. The network monitoring is usually associated with the functions involved in network management [11]. Network management is required to ensure that the network is up and running [12]. The ideal network monitoring system should have the following properties:

- It should be automatic and continuously monitor the network.
- It should quickly inform the administrator about the problem as soon as it arises.
- It should be intelligent enough to point out the problem and its exact location in the network topology. It should also be able to identify the

Manuscript received July 25, 2012; revised September 10, 2012.

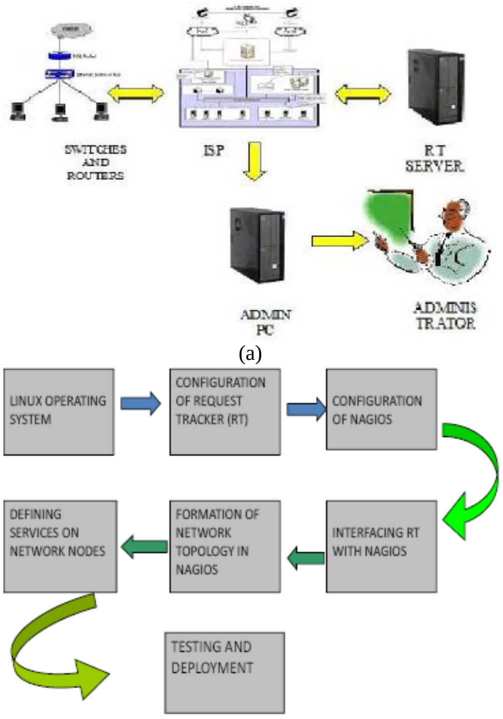
Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer are with Politecnico Di Torino, 10129 Turin, Italy (e-mail: rafiullah.khan@studenti.polito.it,sarmad.khan@polito.it,{rifaqatzaheer,babar}@nwfpuet.edu.pk).

Muhammad Inayatullah Babar is with University of Engineering and Technology, Peshawar Pakistan

DOI: 10.7763/IJIEE.2013.V3.280

problem effects on the rest of the network and the services that will become unavailable.

- It should keep a record of the changes in the network which makes easier to find the cause of the problem due to configuration change.
- It should provide remote authentication and authorization for the administrator to get access to the monitoring system from everywhere.


SWITCHES AND ROUTERS
ISP
RT SERVER
ADMIN PC
ADMINIS TRATOR
LINUX OPERATING SYSTEM
CONFIGURATION OF REQUEST TRACKER (RT)
CONFIGURATION OF NAGIOS
DEFINING SERVICES ON NETWORK NODES
FORMATION OF NETWORK TOPOLOGY IN NAGIOS
INTERFACING RT WITH NAGIOS
TESTING AND DEPLOYMENT

(a)

(b)

Fig. 1. (a) Proposed Network Monitoring System; (b) Work Flow.

## III. Automatic Network Monitoring

An overview of the automatic network monitoring and management system defined in this paper is shown in Fig. 1(a). The CISCO switches were used in the network which support SNMP. These network switches are configurable and different services e.g. ping, ssh etc are applied on them. All the attributes of the switches and routers are also defined in nagios software. After every 10 sec (pre-defined time), nagios monitors all the services that are applied to the switches. Nagios is configured to make 5 attempts if a service appears to be down. After 5 failed attempts, a notification is generated in nagios which generate a ticket in RT. The RT server has a continuous interaction with internet service provider (ISP) [8]. As soon as a ticket is generated in RT, it is forwarded via email to the network administrator with the information of the problematic switch. The administrator can remotely access the RT server and resolve the ticket [10]. If the administrator does not resolve the ticket within an hour then it is forwarded to the next responsible person in the priority list and so on. This network monitoring system is implemented in linux environment and its basic work flow is shown in Fig. 1 (b).

## IV. Network Monitoring by Nagios

Nagios is open source and web based software used for network monitoring [7]. It monitors network nodes and services applied on them and inform the network administrator when any change happens in the network [6]. Nagios is well suited application for linux environment but it can also run on other platforms as well. Nagios is a secure and easy manageable application which provides nice web interface, automatic alerts if condition changes and various notification options [13].

When any node or service in the network gets problem, nagios generates notification to the network administrator in the form of email or sms. Nagios is developed under GNU general public license and supports different services like HTTP, NNTP, Ping, SMTP, etc. Nagios allow administrator to build complete network topology and define child-parent relationship among nodes. This child-parent relationship among nodes enable nagios to send only one notification if a parent node goes down with the information that child nodes become unavailable. A generic network topology created in nagios is shown in Fig. 2 (a).

Nagios decide about the condition of nodes and services with two factors: ‘status’ and ‘type of state’. The status can be either up, down, critical or unreachable while the type of state can be either soft state or hard state. The type of state has great importance for alerting process. It decides about the final status before a notification is sent out. In order to avoid false notifications, nagios check the nodes and services for pre-defined number of times before declaring them to have real problem [13]. The number of attempts can be controlled by ‘max_check_attempts’ option in the node and service definitions. Node or service is declared in soft state if status check results in a non-OK state but the number of attempts is less then ‘max_check_attempts’. This is also called ‘soft error state’. In the ‘soft recovery state’, the node or service recovers from ‘soft error state’. Node or service is declared in hard state if status check results in a non-OK state for the number of attempts specified in ‘max_check_attempts’. This is also called ‘hard error state’ when the node is either unreachable or down. In the ‘hard recovery state’, the node or service recovers from ‘hard error state’. The hard state of node or service will change if the status check changes from hard OK state to hard non-OK state or vice versa. If during the hard state change, the node or service is declared in non-OK state then the hard node or service problem is logged and administrator is notified about the problem. But if during the hard state change, the node or service is declared in OK state then the hard node or service recovery is logged and administrator is notified about the recovery. Furthermore, if the hard state change occurs from one non-OK state to another non-OK state then the administrator is re-notified about the problem [7]. A screen shot of host problem in nagios is shown in Fig. 2 (b).

### A. *Nagios Installation*

Nagios installation in Ubuntu is quite simple. Just follow the steps below as root user:

```
# apt-get install nagios3
```

After installation, assign the web user password with the following command:

```
# htpasswd -c /etc/nagios3/htpasswd.users username
```

Now nagios can be accessed from browser using Fully Qualified Domain Name (FQDN) by visiting the web page at http://FQDN/nagios3. Use the login information specified above:

username: *username*
password: *password*

### B. Nagios Configuration

Make the network topology by defining every node in /etc/nagios3/conf.d/ directory. File name should be the same as host_name. A generic node1 can be defined as follows:

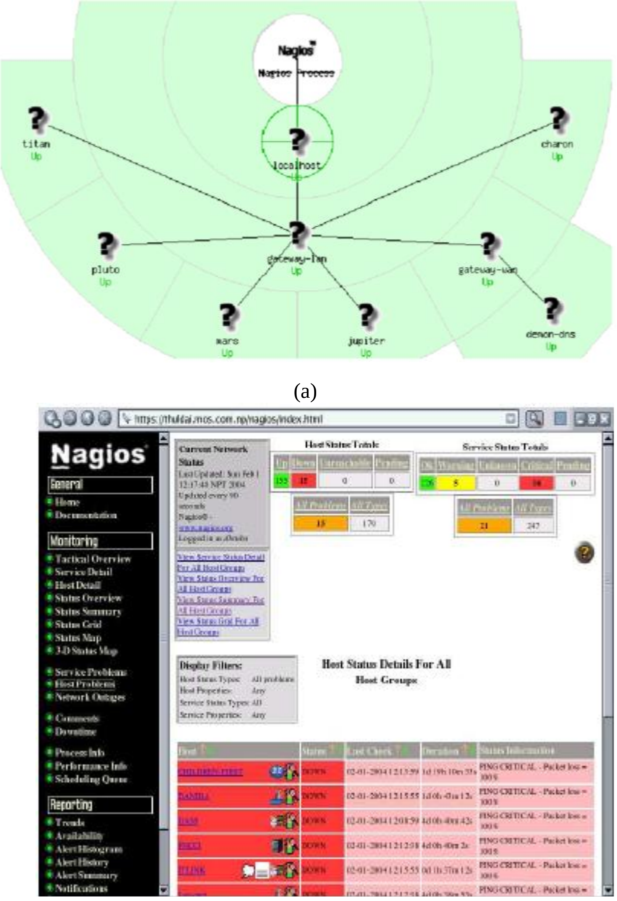
```
define host {
    use          generic-host
    host_name    node1
    alias        node1 in network
    address      [node1 IP address]
    parents      node1's parent if any
}
```

Introduce a group in ‘hostgroups_nagios2.cfg’ that will include all above defined nodes.

```
define  hostgroup {
    hostgroup_name       network-group
    alias                network nodes
    members              node1, node2,…
}
```

Associate some services e.g, ssh, ping etc to the defined group in ‘services_nagios2.cfg’ file.

```
define  service {
    hostgroup_name           network-group, other-
groups
    service_description   PING
    check_command
check_ping!100.0,20%!500.0,60%
    use                   generic-service
    notification_interval    0  ;  for  re-notification,
set > 0
  }
```



(a)

(b)

Fig. 2. (a) Generic topology created in Nagios; (b) Nagios host problem screen shot.

Since nagios has to be interfaced with RT for better network management, therefore define RT contact in the file ‘contacts_nagios.cfg’.

```
define contact {
    contact_name                      RT
    alias                             Request-
Tracker
    service_notification_period       24×7
    host_notification_period          24×7
    service_notification_options      c,w,r,u
    host_notification_options         r,d
    service_notification_commands     notify-service-
by-email
    host_notification_commands        notify-host-by-
email
    email                             rt@host.FQDN
  }
```

Now introduce a contact group that will include all defined contacts.

```
define  contactgroup {
    contactgroup_name      Network-admins
    alias                  Network   and   Nagios
admins
    members                RT, other-contacts
  }
```

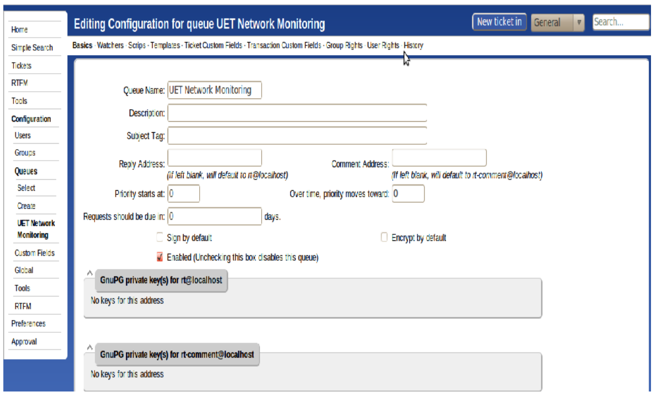
Finally, restart the nagios and check for the applied configurations in the web interface.

```
# /etc/init.d/nagios3  restart
```

## V. RT Ticket Management System

A good management system is usually required for organizations in order to manage their work flow, offering services to clients or manage hardware/software problems [5],[14]. Every ticket has certain attributes and ID number used to identify the ticket. RT is an open source ticket management software developed by Best Practical, Inc. New York University. RT is heavily used worldwide as it provides email friendly interface and keep track of tickets which represent a job to be done. RT provide ease of use, multiuser accessibility, access control, history tracking, remote accessibility, generate notifications and customization according to organization requirements. Different versions of RT software is available to work on windows, Unix and Linux environments. It also requires a database which can be MySQL, POSTGRESQL or ORACLE. Since RT is open source, thus can be customized using Perl script language. RT also requires Apache web server. RT make use of Perl based main engine and a database to store its data and provides web and email interfaces [10].

RT allows to create different users via web interface and assign rights to them. Users can also be arranged in groups and assign rights to them on global basis. RT can also be configured to generate queues of tickets to work on. These queues correspond to a group of different services. The RT queue configuration window is shown in Fig. 3. Ticket is key object in RT which defines a job to be done. RT ticket attributes include status, watcher, time left, time worked, ticket priority, queue and its owner. Main ticket watchers are its owner and requester but additional watchers can also be defined. RT ticket priority can range from 0-99 which determines the importance of ticket with 99 as the highest priority. It is also possible to define initial and final ticket priority which increases or decreases with the time left. RT also allows to define custom scripts which take an automatic action in response to a given condition.



Editing Configuration for queue UET Network Monitoring

Queue Name: UET Network Monitoring

Description:

Subject Tag:

Reply Address: (If left blank, will default to rt@localhost)

Comment Address: (If left blank, will default to rt-comment@localhost)

Priority starts at: 0

Over time, priority moves toward: 0

Requests should be due in: 0 days.

□ Sign by default

□ Encrypt by default

☑ Enabled (Unchecking this box disables this queue)

GnuPG private key(s) for rt@localhost

No keys for this address

GnuPG private key(s) for rt-comment@localhost

No keys for this address

Fig. 3. RT Queue Configuration.

### A. RT Installation

RT software requires many dependencies for its installation in Linux environment [10]. Ubuntu and Fedora are preferred choices as they allow automatic installation of many required dependencies during the installation process. RT can be installed in Ubuntu environment with the following commands in the terminal:

```
# apt-get install rt3.6-apache2  request-tracker3.6  3.6-clients apache2-doc postfix mysql-server lynx libdbd-pg-perl  libapache-dbi-perl  rt3.6-rtfm
```

During postfix configuration, a pop up window appears to enter the 'system mail name' which is also called Fully Qualified Domain Name (FQDN). FQDN is used to provide global access to the RT software.

### B. RT Configuration

Make the following important changes in the configuration file 'RT_SiteConfig.pm'.

```
Set($rtname, ‘rt-name’);
Set($Organization, ‘organization-name’);
Set($CorrespondAddress , rt@FQDN);
Set($CommentAddress , rt-comment@FQDN);
Set($WebPath , "/rt");
Set($WebBaseURL , "http://FQDN/rt");
Set($DatabaseType, $typemapmysql);
Set($DatabaseUser , ‘user-name’);
Set($DatabasePassword , ‘user-password’ );
```

Now restart the apache to get sure that all the changes have been recorded. Enter the following URL in the browser: “http://FQDN/rt” and finally, log in with the user name "root" and password "password".

## VI. Interfacing Nagios with RT

At the final stage, nagios is interfaced with RT software. The main network monitoring task is performed by nagios but the ticket management task is performed by RT. 'rt-mailgate' plays an important role for creating interface. For this purpose, an alias is created in file called 'aliases' by inserting the following text:

```
rt: "|rt-mailgate --queue `name of RT queue' --action correspond --url http://FQDN/rt"
rt-comment:  "|rt-mailgate --queue `name of RT queue' --action comment --url http://FQDN/rt"
```

The above statements will inform rt-mailgate to send all nagios notifications to the defined queue in RT. Check whether rt-mailgate works properly with the follow statement.

```
echo "checking functionality" | mail -s `rt-mailgate-testing' rt@FQDN
```

The above statement will generate ticket in RT with subject 'rt-mailgate-testing'. Create a queue with the same name in the configuration menu of RT. Also assign required rights to the users as well as groups. When ever nagios will generate a notification, a ticket will be created in RT. The network monitoring system functionality can be tested by making any of the network node unavailable. This will generate a nagios notification which will create a ticket in RT. The ticket will be forwarded to all watchers in the defined queue of RT according to the priority.

## VII. Performance Test

Nagios software is configured to monitor the whole network every 10 sec. Nagios put any of the node or service in 'soft error state' when it becomes unavailable [6]. Nagios is also configured to make 5 re-attempts and finally put node/service in 'hard error state' if the repeated attempts fail [7]. This has been verified by making a generic node 'node1' unavailable. After the hard error state, nagios sent a notification to the RT server. According to the defined configuration, RT has generated and sent a ticket by email to the network administrator. The network administrator checked the ticket but did not change its status to 'resolve' or 'open'. After an hour, the same ticket with same attributes was automatically forwarded to second responsible person by email who has remotely accessed the RT server and changed the ticket status to 'resolve'. The ticket was not forwarded to any other person in the priority queue after its status has changed. Thus, the presented paper has provided automatic and efficient network monitoring and management system.

## VIII. Conclusions

This paper has presented an efficient and automatic network monitoring and management system which quickly reports to network administrator in case of any problem. Nagios is configured to generate and monitor the whole network topology and send notifications in case of state change anywhere in the network. These notifications will generate tickets in RT. The further network management task is performed by RT. RT is configured to send email or sms to all responsible persons one by one after every pre-defined time interval until the problem is solved. This network monitoring system is fully automatic and the administrator has to check only his emails. The presented network monitoring system is intelligent to quickly identify problem location in the network and also its effect on the rest of the network. Thus, it is highly efficient and provides full control over the network.

## REFERENCES

[1] D. Ten, S. Manickam, S. Ramadass, and H. A. Bazar, "Study on Advanced Visualization Tools In Network Monitoring Platform," in *Third UKSim European Symposium on Computer Modeling and Simulation, EMS '09'*, Minden Penang, Malaysia, December 2009.

[2] L. Chang, W.L. Chan, J. Chang, P. Ting, M. Netrakanti, "A network status monitoring system using personal computer," presented at IEEE Global Telecommunications Conference, August 2002.

[3] R. Talpade, G. Kim, and S. Khurana, "NOMAD: traffic-based network monitoring framework for anomaly detection," *IEEE International Symposium on Computers and Communications* vol. 9, Morristown, NJ, August 2002.

[4] S. Feng, J. Zhang, and B. Zeng, "Design of the Visualized Assistant for the Management of Proxy Server," presented at IEEE Third International Symposium on Electronic Commerce and Security (ISECS), Wuhan, China, August 2010.

[5] X. Wang, L. Wang, B. Yu, and G. Dong, "Studies on Network Management System framework of Campus Network," presented at 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR), 2010, Yantai, China.

[6] D. Oliveira, T. Vasques, F. Vieira, G. de Deus *et al.*, "A management system for PLC networks using SNMP Protocol," presented at IEEE International Symposium on Power Line Communications and Its Applications (ISPLC), '10, Goias, Brazil, June 2010.

[7] M. Schubert, A. Hay, D. Bennett *et al.*, "Nagios3 Enterprise Network Monitoring," *Designing Configurations for Large Organizations*, Chap:2, pp.25-84, 2008.

[8] Y. Cai, "Development of an open source network management & monitoring platform for wireless broadband service provider in rural areas," presented at IEEE International Conference on Electro/Information Technology (EIT), Houghton, MI, USA, October 2010.

[9] B. Stelte and I. Hochstatter, "iNagMon Network Monitoring on the iPhone," presented at Third International Conference on Next Generation Mobile Applications, Services and Technologies, NGMAST '09, Neubiberg, Germany, November 2009.

[10] D. Chamberlain, R. Foley, D. Rolsky, R. Spier, and J. Vincent, "RT Essentials," in Installation, Chap:2, pp. 9-18, August 2005.

[11] H.-C. Lin and C.-H. Wang, "Distributed network management by HTTP based remote invocation," in Global Telecommunications Conference, 1999. GLOBECOM-99.

[12] X. Jiang and F. Peng, "Network Management Capability Model and its Application of Self-Management Capability Analysis," in *International Symposium on Computer Network and Multimedia Technology*, CNMT-09, Wuhan, China, January 2010.

[13] A. Gomez, C. Dafonte, and B. Arcay, "3D Visualization for system and networks monitoring support," presented at 3rd IEEE Conference on Human System Interactions (HSI-10), A Coruna, Spain, July 2010.

[14] L. Jia, W. Zhu, C. Zhai, and Y. Du, "Research on an Integrated Network Management System," presented at 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel Distributed Computing, 2007. SNPD 2007.