ADVANCED NETWORK MONITORING SYSTEM USING A HYBRID OF AGENT AND AGENTLESS CONCEPTS

Rahul S Sreedhar, Prithvi Vihari R, Dr. N Guruprasad
[1,2]UG student, [3]Professor,[1,2]CMR Institute of Technology,[3]New Horizon College of Engineering, Bangalore
rahuls.1234sreedhar@gmail.com , prithvivihari11@gmail.com, nguruprasad18@gmail.com

Abstract-Network monitoring system on agent based protocol provides benefits such as better security, increased bandwidth efficiency and increased monitoring capabilities. However in today‘s world, an agentless network monitoring system would seem to be more apt as it would reduce the cost of installation & maintenance, reduced administrative overhead, fast detection of network outages and protocol failures and is easy to deploy. For large organizations, factors such as cost of installation, reduced deployment time and maintenance are as critical as security. Hence, this paper would compare the concepts of both agent and agentless monitoring protocols and discusses the best ways to combine and reap their benefits.

Keywords— Agent-Based, Agent less, NagiOS, RT Ticket Management System, Simple Network Management Protocol

I INTRODUCTION

Data communications and networking influences the way business are done and the way we live today. Quicker business decisions are required to be made which requires immediate access to accurate information for decisions makers. Business today relies more often on computer networks and inter-networks. Larger organizations reply more on big network topology to avoid the manual networking options that causes waste of resources and that could lead to efficiency improvement and has become best industry practices to date.

Conventional Network management systems involves monitoring, testing, configuring and troubleshooting network components to meet smooth and efficient operation of network that provides adequate quality of service for the users or meet the organizational requirements. This task is accomplished generally through a most common management system, the Simple Network Management Protocol (SNMP) concept that leverages on the use of hardware, software and humans. SNMP uses the concept of manager and agent, with the former being usually a host, who controls and monitors a set of agents, usually routers. The application level protocol is so designed so that it can monitor devices made by different manufacturers and installed at different physical networks [7].

Over the years, various updates of SNMP have been released. As of date, the SNMPv3, have progressively increased security features such as Authentication, Remote configuration and administration capabilities Privacy, Authorization and access control, that were lacking in the previous versions [7]. Nevertheless, the notable deficiency in the difficulty in monitoring networks as opposed to nodes on networks, improved ease of operational efficiency, performance and reduced infrastructure cost for deployment continues to be challenging while adopting the agent based concept.

The concept of agent and agent-less network monitoring, as shown schematically in Figure 1, of various network devices without the need to install software agents across all the monitoring devices, appears to be an attractive option to overcome few of these challenges. However, it is noted that the agent less options does have major deficiencies in respect of security aspects.

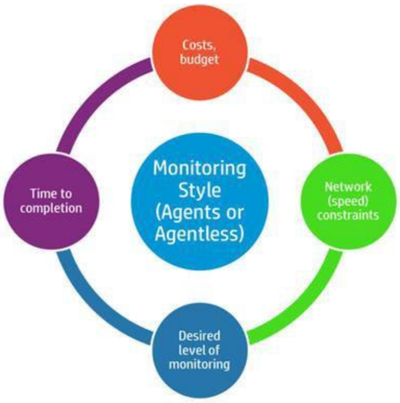

Costs, budget
Time to completion
Monitoring Style (Agents or Agentless)
Network (speed) constraints
Desired level of monitoring

Figure 1: Factors affecting network monitoring

This paper explores a hybrid approach which proposes to exploit advantages of both the Network Monitoring Concepts towards four focal areas such as ease of deployment, adaptability and less node on network and security aspects.

a) AGENT BASED SYSTEM

Monitoring with agents has the cost of installation, configuration (proportionate to number of managed elements), platform support needs and dependencies. It involves installing the agent software in all the systems connected to the network.

In general, agent-based monitoring is regarded more secure since it does not necessitate storage of passwords centrally. Organizations that have high-level of security compliance needs would prefer agent-based monitoring as a result.

In each node system of each sub-network, the Agent Processor, which processes management information based on a management information model, processes the information of each of the equipment by each management service and stores the situation information of recent equipment and network as an object-oriented model in the Management Information Base. The center operating system analyzes the relations among the sub-network units.

There are crucial systems that support SLA‘s (Service Level Agreements) for 99.999% (5-nines) availability. It is important to monitor during each minute and second, how the system is performing – the peaks and trough of usage, not just average utilization levels. This is crucial for capacity analysis and prediction heuristics. While it is theoretically possible to poll at shorter intervals than a minute, it is not efficient to go to really low intervals using remote polling approach (as minor delays due to network latency can cause havoc with the data collections).

This system enables in-depth monitoring and management. The agents to application / OS (Operating System) communications are handled internal to the server. Hence, no additional firewall rules need to be configured since it is much more secure.

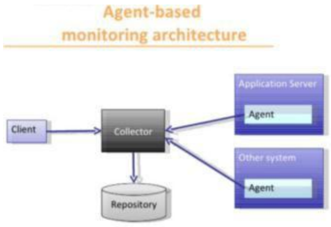


Figure 2: Agent Based Monitoring Architecture

Agents need to be deployed on each server. The data is collected locally and only the processed final results are transported to the console and therefore the bandwidth is utilized efficiently. Any software vendor will tell you that their agents work the best with their platform. While this may be true, it may also be because their management platform is built only to work with their proprietary agents. The result is vendor lock-in, and changing vendors can mean expensive, large-scale, long-term deployments of replacement technology. Because of this, when IT requirements change, it can be extremely costly to meet them. Generally, open standards and flexibility work far better in the long run.

b) AGENTLESS SYSTEM

Agentless monitoring is easy to deploy, as the monitoring and configuration happens at a central place with a good UI. Costs of installation (and the ongoing maintenance) are really low.

Considering one of the case studies of HP global financial service incorporation, when the company began reconsidering its monitoring technology, it identified a number of objectives. One was cost. ―We wanted to reduce the amount of money we were spending on agent maintenance and support‖, notes William Gillen, Director of Systems Engineering, global financial services corporation. ―At the time, I was helping to support an environment with about 12,000 servers, and we recognized that managing our monitoring agents was increasing our support costs by 40%.‖ [6].

In reality, all computer tasks require related programs to run and these programs could be considered agents. However, operations can generally be performed on a controlling machine that may use an agent, while its target is agentless in that it need not install or run new software related to the task itself. This capacity can save time required for managing agents on numerous target machines, especially in large enterprises. That said, even some software called agentless may use built-in services in an operating system, and they may require configuring. So the term agentless can be a bit misleading.

Agentless software generally requires the creation of a user account on the target machine or machines to facilitate access to the data on the account related to the desired operation. Agentless programs directly access the files, often remotely, via this user account. So the created profile must have the necessary access to these files and the software must store the login credentials to access the profile. This access may be facilitated through a number of different Internet communication standards (e.g., FTP (File Transfer Protocol), TELNET, SSH (Secure Shell)).

For agentless monitoring, implementation ranges from built-in SNMP agents to remote shell access, such as SSH. ―Agentless‖ is a bit of a misnomer. All management requires an agent, whether the agent is embedded in the management platform, the managed device, or a separately installed piece of software. The industry has accepted the definition of agentless as a management agent that is embedded in the software of the device or as a capability of the manager, requiring no separate installation or licensing.

Agentless monitoring really means the use of existing, embedded capabilities.

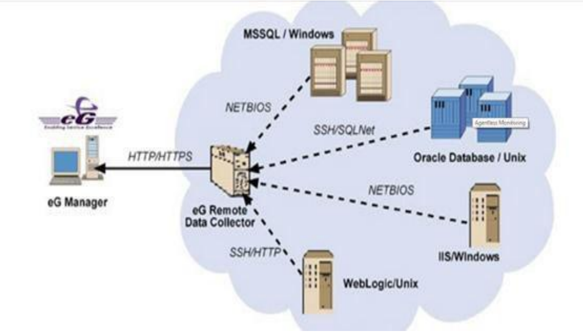


Figure 3: Agentless Network monitoring Architecture

II. NETWORK MONITORING BY NAGIOS

Nagios is open source and web based software used for Network monitoring [1]. It monitors network nodes and services applied on them and inform the network administrator when any change happens in the network [2]. 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 a good web interface, automatic alerts if condition changes and various notification options [3]. When any node or service in the network faces a 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 (Hyper Text Transfer Protocol), NNTP (Network News Transfer Protocol), Ping, SMTP (Simple Mail Transfer Protocol), 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 Figure 4.

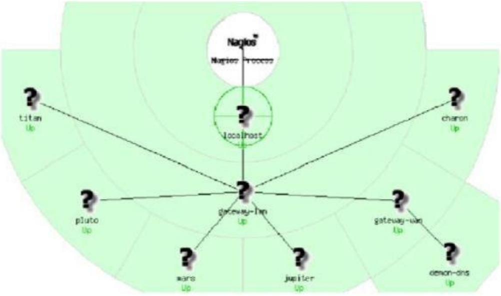
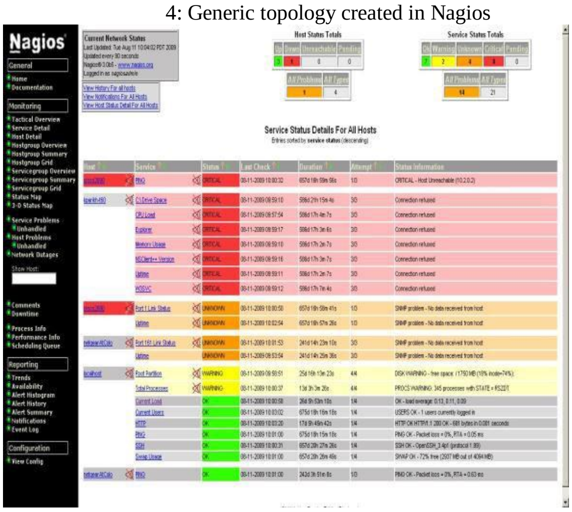


Figure 4: Generic topology created in Nagios



Current Network Status
Last Updated: Tue Aug 11 10:04:02 PDT 2009
Updated every 90 seconds
Nagios® 3.0b6 - www.nagios.org
Logged in as nagiosadmin

Host Status Totals

Up	Down	Unreachable	Pending
3	1	0	0

All Problems	All Types
1	4

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
7	2	4	8	0

All Problems	All Types
14	21

Service Status Details For All Hosts
Entries sorted by service status (descending)

Host	Service	Status	Last Check	Duration	Attempt	Status Information
cisco2600	PING	CRITICAL	08-11-2009 10:00:32	657d 18h 59m 56s	1/3	CRITICAL - Host Unreachable (10.2.0.2)
kparikh-t60	C:\ Drive Space	CRITICAL	08-11-2009 08:59:10	598d 21h 15m 4s	3/3	Connection refused
	CPU Load	CRITICAL	08-11-2009 08:57:54	598d 17h 4m 7s	3/3	Connection refused
	Explorer	CRITICAL	08-11-2009 08:59:17	598d 17h 3m 6s	3/3	Connection refused
	Memory Usage	CRITICAL	08-11-2009 08:59:10	598d 17h 2m 7s	3/3	Connection refused
	NSClient++ Version	CRITICAL	08-11-2009 08:59:16	598d 17h 3m 7s	3/3	Connection refused
	Uptime	CRITICAL	08-11-2009 08:59:11	598d 17h 2m 7s	3/3	Connection refused
	W3SVC	CRITICAL	08-11-2009 08:59:12	598d 17h 7m 4s	3/3	Connection refused
cisco2600	Port 1 Link Status	UNKNOWN	08-11-2009 10:00:58	657d 18h 58m 41s	1/3	SNMP problem - No data received from host
	Uptime	UNKNOWN	08-11-2009 10:02:54	657d 18h 57m 26s	1/3	SNMP problem - No data received from host
netgearAtCalo	Port 161 Link Status	UNKNOWN	08-11-2009 10:01:53	241d 14h 23m 10s	3/3	SNMP problem - No data received from host
	Uptime	UNKNOWN	08-11-2009 08:53:54	241d 14h 25m 36s	3/3	SNMP problem - No data received from host
localhost	Root Partition	WARNING	08-11-2009 08:58:51	25d 16h 13m 23s	4/4	DISK WARNING - free space: / 1750 MB (18% inode=74%):
	Total Processes	WARNING	08-11-2009 10:00:37	13d 3h 3m 26s	4/4	PROCS WARNING: 345 processes with STATE = RSZDT
	Current Load	OK	08-11-2009 10:00:58	26d 9h 53m 10s	1/4	OK - load average: 0.13, 0.11, 0.09
	Current Users	OK	08-11-2009 10:03:02	675d 18h 16m 18s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	08-11-2009 10:03:20	17d 9h 49m 42s	1/4	HTTP OK HTTP/1.1 200 OK - 681 bytes in 0.001 seconds
	PING	OK	08-11-2009 10:01:00	675d 18h 15m 18s	1/4	PING OK - Packet loss = 0%, RTA = 0.05 ms
	SSH	OK	08-11-2009 10:00:31	657d 20h 27m 26s	1/4	SSH OK - OpenSSH_3.4p1 (protocol 1.99)
	Swap Usage	OK	08-11-2009 10:01:00	657d 20h 26m 49s	1/4	SWAP OK - 72% free (2937 MB out of 4094 MB)
netgearAtCalo	PING	OK	08-11-2009 10:01:00	243d 3h 51m 8s	1/3	PING OK - Packet loss = 0%, RTA = 0.63 ms

Figure 5: Service status details for all hosts [8]

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 [3]. 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 ard 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 [1].

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
```

Designed with scalability and flexibility in mind, Nagios gives you the peace of mind that comes from knowing your organization's business processes won't be affected by unknown outages.

Nagios is a powerful tool that provides you with instant awareness of your organization's mission critical IT infrastructure. It allows you to detect and repair problems and mitigate future issues before they affect end-users and customers. It is a popular open source computer system and network monitoring software application. It watches hosts and services, alerting users when things go wrong and again when they get better. It also monitors your entire IT infrastructure to ensure systems, applications, services, and business processes are functioning properly.

In the event of a failure, Nagios can alert technical staff of the problem, allowing them to begin remediation processes before outages affect business processes, end-users, or customers. With Nagios you'll never be left having to explain why a unseen infrastructure outage hurt your organization's bottom line.)

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, othergroups
service_description  PING
check_commands check_ping!100.0,20%!500.0,60%
use   generic-service
notification_interval     0;
for re-notification,
set > 0
}
```

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 RequestTracker
service_notification_period 24×7
host_notification_period 24×7
service_notification_options c,w,r,u
host_notification_optionsr, d
service_notification_commands notify-serviceby-email
host_notification_commands    notify-host-byemail
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
```

III. 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. Every ticket has certain attributes and ID number used to identify the ticket. RT is 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 provides ease of use, multiuser accessibility, access control, history tracking and remote accessibility; generate notifications and customization according to organization requirements.

Different versions of RT software are 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 makes use of Perl based main engine and a database to store its data and provides web and email interfaces.

RT allows creating different users via web interface and assigning 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.

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 defining custom scripts which take an automatic action in response to a given condition.

a) RT INSTALLATION

RT software requires many dependencies for its installation in Linux environment. 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
3.6clients apache2-doc postfix mysql-server lynx libdbd-
pgperllibapache-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".

IV. 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. „rtmailgate" 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-mailgatetesting' 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 evernagios 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.

VI. CONCLUSION

The paper addressed a hybrid approach which deals with a combination of agent and agentless network monitoring system by exploring four typical areas such as ease of deployment, adaptability, less network load and security aspects. Scenario of agent-based and agent-less monitoring has been brought out to clarify and support the hybrid concept. However, it is always a question of complying with quality requirements of network monitoring considering costs, budgets, time and network speeds that the choices are made in a typical industry scenario.

Abbreviations:
EG- Enterprise Group
HTTP- Hyper Text Transfer Protocol
HTML-Hyper Text Markup Language
SSH-Secure Shell
NetBIOS- Network Basic Input / Output System
RT- Real Time
UI- User Interface
HP- Hewlett Packard
TELNET- TELecommunication NETwork
GNU- GNU's Not Unix
SQL-Structured Query Language
URL- Universal Resource Locator
IIS- Internet Information Server

REFERENCES

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

[2] 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.

[3] 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.

[4] Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Muhammad Inayatullah Babar, ―An Efficient Network Monitoring and Management System", International Journal of Information and Electronics Engineering, Vol. 3, No. 1, January 2013

[5]http://www.ijarcsse.com/docs/papers/Volume_3/9_ September2013/V3I9-0238.pdf, International Journal of Advanced Research in Computer Science and Software Engineering

[6] http://www8.hp.com/h20195/V2/GetPDF.aspx/ 4AA5-1319ENW.pdf

[7] Data communication and Networking, 4th Edition, Beherouz A Forouzan

[8]https://www.novell.com/coolsolutions/feature/16723. html