Distributed Intranet Health Monitor

Gordon Hugh Martin, Kyle Bremner, Graham McDonald, Paul Mullen
Distributed Algorithms and Systems
Group D
School of Computing Science
University of Glasgow
{0800902m, 0801738b, 0909169m, 0705786m}@student.gla.ac.uk

***Abstract*—The abstract goes here.**

I. Introduction

Computer connections via wired and wireless communications channels are ubiquitous in todays digital landscape. The interconnected network of networks known as the internet has enabled communications on a world wide scale to be conducted in milliseconds. Within a large variety of settings, both commercial and in the home, the most appropriate network setup is an intranet set up on a local area network. Intranets can grow to a considerable size and can contain numerous important nodes which peers can depend on heavily, such as mail servers and database servers. As the size of the intranet increases complexity is introduced to the system which can affect system performance and lead to failures.

In order to maintain a working system a network administrator must be able to perform network analysis. L. Bosack and C. Hedrick [1] identified Ethernet meltdown and broadcast storm as two sources of degrading performance which can manifest as inability to establish a connection or unintentional severing of connections between nodes. Our system aims to present the user detailed network statistics to aid in the diagnosis of possible failures and highlight poorly performing communication channels within an intranet. The system uses one of the most fundamental network analysis tests available, pinging a node and measuring the response. This can provide a metric to measure system performance and identify unreachable nodes. Statistics collected by monitor nodes within the system are aggregated and presented as a snapshot of the system to registered viewers. Snapshots present the viewer an overview of factors relevant to network performance such as ***.

There are numerous available solutions to perform network analysis. Most, however are implemented in low level programming languages such as C and require the user to have more than a passing familiarity with network protocols. Propriety solutions also exist which can provide network administrators a statistical overview of network performance, such as Cisco Network Analysis modules. We aim to provide a solution which is not tied to any specific vendors' system and which can provide a service to users less familiar with nuances of lower layer protocols. Platform independent Java RMI enables our system to abstract the complexity inherent in heterogeneous systems and solve this problem.

In the following report we first present summary of previous related work (II). We then outline the system design (III) before going on to describe the system implementation IV. Finaley we present an evaluation of the system (V) and our conclusions (VI).

II. Related Work

There are a range of popular proposed techniques that attempt to tackle the problem of fault detection and diagnosis in distributed systems. This section will discuss contributions in the field of fault detection and contrast them with our approach to establish its merits and weaknesses.

Verissimo et al take a centralised approach, where a Monitor System monitors all protocol traffic between nodes in a distributed system. The system develops a model of the send and receive ordering of a protocol (hence modeling the system state, rather than a nodes internal state) in addition to Quality of Service rules. When a failure occurs, the system traces the error propagation back to the origin of the failure using the developed model as a diagnosis of the problem.[2] Our focus is more general, with more focus on Quality of Service issues than service failures. Furthermore, as our approach is peer based, where peers gather snapshot data from their perspective on different branches of the network we are able to identify Quality of Service issues specific to certain regions of the network. However, some failures can also be identified if a host becomes unreachable. Although our method can identify nodes that are failing to provide expected QoS levels through rudimentary tests, it is less strong at identifying the specific source of the issue which is left to the network administrators judgement using other tools.

Some other systems, like our own, take the approach of 'testing' key nodes the network and passing the result for analysis to a ḿonitorńode, rather than monitoring all packets sent for an instance of a failure from a centralised point.[**?**] Holt and Smith discourage using centralised monitoring nodes as it runs counter to fundamental principles of distributed systems where there is no fault-free all-knowing supervisor.[4] There are advantages to this distributed model of fault detection. It is possible, for example, through collaboration of the testing nodes and the aggregation of their data by a monitoring node to identify which nodes are getting the least efficient service on a network.

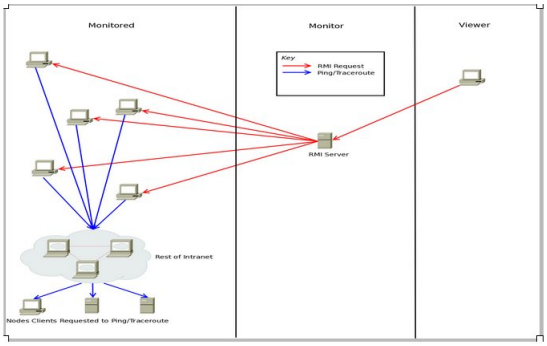
III. System Design

This section will outline the structure of our system, describing the role of components and the interactions between components in the system to collaborate to perform the overall function of providing a network snapshot. The motivations behind our choices will also be discussed.

A. System Overview

Our system creates a snapshot overview of the system through the collaboration of multiple nodes presenting their view of the network, by performing some basic diagnostics on key nodes, to a monitoring system which aggregates the data. The monitor and collaborating clients act as a middleware layer for the network, by exposing an API which can be used by a node to retrieve a snapshot of the network at any given moment. A node accessing the middlewares API can then perform some analysis on the retrieved snapshot to infer some useful information to their function about the state of the network.

B. High Level Diagram



C. Component Descriptions

Our system has three main components: the monitor, the clients and the viewers.

The monitor provides a list of node addresses of interest to the registered clients to probe with network diagnostics. The monitor periodically commands the registered clients to run their diagnostics, and provide the results to the server.

The monitored clients register with the server, then await a command (method invocation) from the monitor to gather a view of the network from its perspective by issuing some basic tests; a ping, and a TraceRoute. The result is returned to the Monitor (server) which aggregates the other client results into a snapshot.

The viewer adds and removes nodes of interests for the clients to probe. The viewer subscribes to the server to receive the periodical network snapshot.

D. Motivations

We have elected to take a decentralised approach to network analysis. Networks can be highly heterogeneous and large in size; some nodes in our system may feature on highly loaded sections of the network for example. As such, we believe it is useful to have information about the network from the perspective of different nodes rather than employing tests from any specific node.

We have attempted to make our system highly flexible, for use as a middleware layer within a network. This motivated the choice to have any clients on the network be registered to perform diagnosis. In different setups, different nodes may be selected as nodes typical to certain characteristics of the specific area of the network and registered with the server. As these nodes become less important, they can be deregistered from the server. Furthermore, we have attempted to make our API to the monitor flexible so that the viewer may use it for different ends. The API can be used by different viewers with different goals to generate data relevant to their needs.

IV. Implementation

A. Functionality

V. Evaluation

A. Testing Strategy

B. Proof of System

VI. Conclusion

A. Future Work

VII. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] Bosack, L. and Hedrick, C. *Problems in Large LANs*, Network, IEEE, vol 2, no.1, pp.49-56, jan 1988

[2] http://www.gsd.inesc-id.pt/ mpc/pubs/khanna-monitor-tdsc.pdf pages 266-267

[3] FAULT-DIAGNOSIS IN FULLY DISTRIBUTED SYSTEMS J.G. Kuhl and S.M. Reddy, page 306

[4] http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1676512