PROCEEDINGS

UNIVERSITI TENAGA NASIONAL

ICIM μ 2001

INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY AND MULTIMEDIA AT UNIVERSITI TENAGA NASIONAL

13th - 15th August 2001

Recent Advances and Future Trends in IT and Multimedia

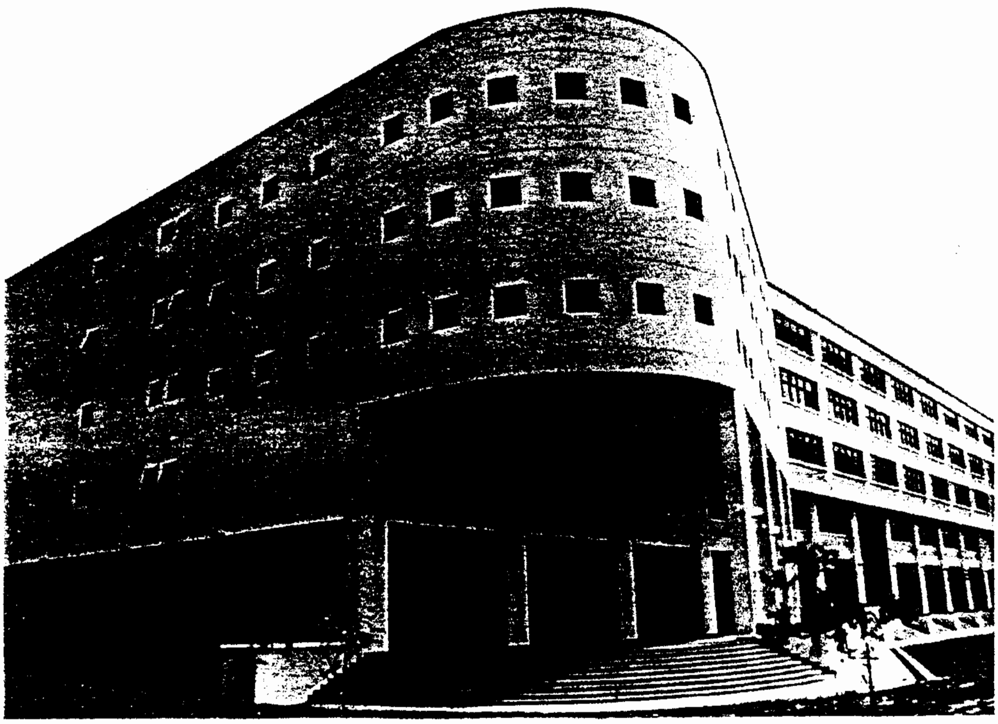

TABLE OF CONTENT

TRACK 2 : IFORMATION SYSTEM

PAGE

TRACK 3 : MULTIMEDIA, COMPUTER BASED TRAINNING AND COURSEWARE — PAGE

TRACK 6: SOFTWARE ENGINEERING AND WEB-BASED APPLICATIONS	PAGE

Proceedings of the International Conference on Information Technology and Multimedia at UNITEN(ICIMU 2001)
Recent Advances and Future Trends in Information Technology and Multimedia
13th - 15th August 2001, Malaysia

Improper Website Blocking Protection Using Active Networks

Akbar, Buhari, Sahalu, Saleem
Information and Computer science Department,
King Fahd University of Petroleum and Minerals,
31261 Dhahran, Saudi Arabia.
Phone: 966 3 860 4374
Fax: 966 3 860 2174
Email: {akbar,mibuhari,sahalu,msaleem}@ccse.kfupm.edu.sa

Abstract

The use of the Internet has become part of the daily routines of many people in this age. The Internet has no specific domain control and one can find on the Internet almost anything imaginable including obscene material. Identification of such obscene websites is often done manually on a trial-and-error basis. Not only is this approach time consuming, it also leads to incorrectly blocking decent websites like www.middlesex.ac.uk.

We propose the use of active networks to support a more systematic and accurate way of identifying and blocking improper websites. This is achieved by carrying out customized user computations inside network routers that are also capable of sharing websites blocking information.

Key words: Active Networks, Routers.

1. Introduction

The emergence of Active Network technology has attracted many academics and researchers to become involved in the development of this technology. Active networks has potentially much to offer the Information Technology world. First, exploiting this technology can reduce the time required for the standardization process of new network services. Second, active network shifts the conventional network paradigm: from a passive node that only transfers bits to a more general processing engine like an end station which supports customized computation on user's data. Furthermore, active networks can also be used for enabling on-the-fly modification of network functionality, for example to adapt to changes based upon link conditions.

The use of active networks potentially exposes a high amount of computations taking place inside network routers. Parallel processing can be used to speedup those potential computations. We envisage that parallel processing will be a powerful tool when used in conjunction with active networks especially in a LAN environment: when the load distribution on the nodes in a LAN is skewed with some machines heavily loaded and others lightly loaded these two technologies can be used to good effect taking advance of low-communication latency of a typical LAN.

In this article, we set out to use a synergy of these two techniques in a LAN environment to investigate the prospects of adding user-customized computations inside networks. Another goal of this paper to use parallel processing to improve the performance and resource utilization of the underlying active networks system in a LAN setting.

The remainder of this paper is organized as follows. We outline the system description of our active networks system in Section 2. Our system model, processing of the system and steps are described in Section 3, 4 and 5. Our process model is presented in Section 6. Analysis of experimental results is discussed in Section 7. The advantages and the limitations of our system are outlined in Section 8. Section 9 concludes.

2. System description

2.1. Types of active network system

There are two possible approaches to build active networks. A discrete or out-of-band approach and an integrated or in-band approach [1].

2.1.1. The Discrete Approach:

This may also be called a Programmable Node (Switch/Router) Approach. Here programs are injected into the programmable active node separately from the actual data packets that traverses through the network. 'User' would send the program to the network node (switch or router), where it would be stored and later executed when the data arrives at the node, processing that data. The data can have some information that would let the node decide how to handle it or what program to execute.

2.1.2. The Integrated Approach:

In an Integrated Approach, also termed as the Encapsulation Approach, the program is integrated into every packet of data send over the network. Each message or capsule as the literature calls it contains a program fragment that may or may not have some embedded data. When these capsules arrive at the active node, it interprets the programs and sends the embedded data depending on its interpretation of these programs. This concept is similar to Postscript code, where actual data is embedded in program fragments that the printer understands. In this approach, each active node would have built-in a mechanism to load the encapsulated code, an execution environment to execute the code and a relatively permanent storage where capsules would retrieve or store information.

3. Our system model

We prefer for our active network platform that the packet carries the information about the websites that ought to be blocked due to the following reasons:

1. Each information about a website is very small, because it is just an URL.
2. We got to process the same thing on all the routers, the main idea is just to block the entry of packet from that respective URL or block access to that URL (normally this is done using ipchains in UNIX platform).

We have setup an active network platform with three computers. All the three computers act as routers as well as clients for HTTP protocol. The configuration is shown in Figure 1.

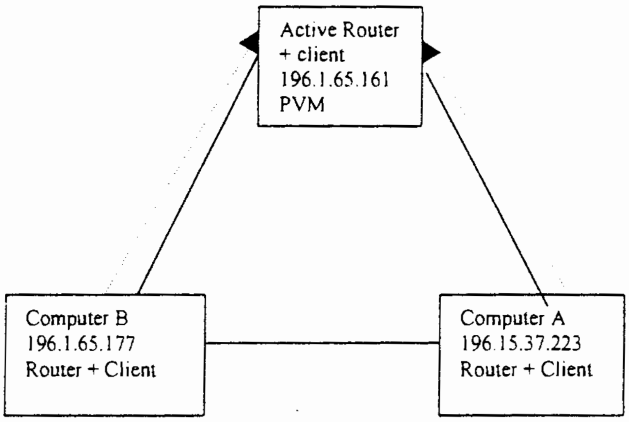

Figure 1. The Configuration Scenario

All the three machines are HTTP clients and so they can access the Internet and gain access to any website. As the industry grows bigger, there might be more than one gateway which connects to the Internet and so, if one gateway blocks one website, the other might not. This reduces the manual update on all the gateways or routers available. In order to check for a certain blocked website by one router, will consume more system processing time. The filter consumes more time than the actual processing time. To reduce this traffic, we distribute the processing as processing on the fly at various routers.

4. Actual processing of the system

The major processing that ought to be done by the active routers can be divided into two.

1. The processing of identification or searching for the presence of some words in the web site accessed. Or use an existing set of websites that ought to be blocked and block those packets. This set of blocking list is divided among various routers and so the total blocking is done at various routers for various URLs.
2. Regular update of the nearby routers with the newly known improper websites. This should be done with the help of cache of the information on the router and updating the nearby routers with the help of some other message. In case of maintenance on one router, the router should update its total list to other routers and make sure the improper websites are blocked. The update of this information should be done faster and in a locked state where the packets are totally blocked and are not allowed to be processed, while an update is going on. This is to prevent illicit usage of packet access during the update time. Also the frequently blockable sites should be taken to the first router so that there are fewer packets to cross it.

Firewalls (hardware or software barrier placed between the network of concern and the rest of the world to prevent unwanted and damaging intrusions on the network) are highly needed to secure the system from an intruder's access. The firewall has to use an IP router to control the passing of any packet from the Internet into the Intranet. The packet filtering parses the headers of the packets to identify the certain websites according to the list specified by a network administrator. The performance of the whole router depends on the procedure on which the rules to check for improper websites are applied to the packet and a decision is made to allow or reject the packet. Also, the performance of the off-the-shelf routers is degraded in proportion to the huge list of the blockable websites. Hence routers, that forward packets more efficiently, are to be explored. The conventional method of packet filtering is sequentially interpreting and checking for the URL, to determine whether a packet should be forwarded or discarded.

Sequential parsing causes the following problems [2,3]:

1. As the number of rows of the rules in the websites that out to be blocked increases, the cost of packet filtering also increases.
2. Because a condition consists of conjunctions of parameters, disjunctive conditions must be specified in several rules. In these rules, each kind of parameter has the same value unless it specifies a disjunctive value. As a result, the same value might be applied to a packet many times.

3. Consider the conditions of the i^{th} rule and the k^{th} rule in the ACL, where $i < k$. If the Condition of the i^{th} rule is always true when the Condition of the k^{th} rule is true, the rule in the k^{th} row of the ACL is redundant because its action, A_k, is never carried out. This is similar to an infeasible path (IFP) in a procedural program and this has to be remedied.

Normally, the necessary information is provided for the packets that are denied entry into our network. The remaining which are not mentioned in the list are mentioned as permitted, by using a last rule which says that the rest are all permitted.

5. Steps for our system processing

From Client side:

1. The client uses a browser (Netscape Navigator or Internet Explorer) to access the Internet.
2. The corresponding HTTP packet is generated and request is just send through any specific gateway.

From Server Side:

1. If access permission is given by the filters of access list rules, then the server responds to the request.

Processing done at router level:

1. After the client gives the send request, the router, actually checks for the list of URLs that ought to be blocked. If the URL requested is in the blocked list, then the message is send back to the client that the concerned page is blocked.
2. If not, the packet is forwarded, the next router also checks for another set of lists. This checking list can also be done parallel when a PVM on the main router or gateway invokes various other PVMD, which checks on the copy of the packet (i.e.) the URL request part, for the list of blocked websites they posses.
3. If one of the router is updated with the set of newly blockable list, the other routers are also updated using some control message. The active packet should have a bit to differentiate between the packet that contains the data and the packet is used for blocked list updates (may be, called as system oriented packets).

6. Our Process model

Processing part of host and hub

The processing portion of the host and the hub are shown, respectively, in Figure 2 and Figure 3. The host's "init" part is used to make the packet destination address fall between 0 and 3 because the network has 4 nodes only. As we have 4 subnets the subnet address in which the node belongs to is also added into the packet. The generated packets are then forcibly sent to the idle part. The transmitter then transmits the packet along any one stream depending upon the destination subnet and the destination address specified.

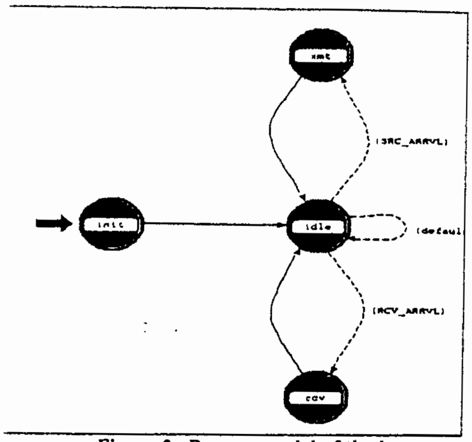

Figure 2. Process model of the host

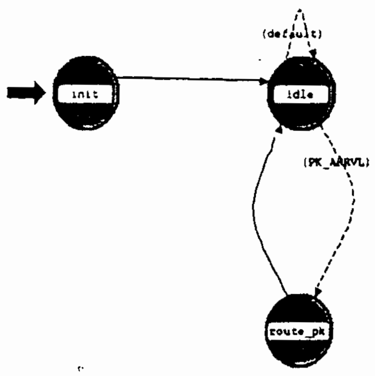

Figure 3. Process model of the hub

The process of the packets is done at the hub. Once the packet reaches the hub, the initial settings for the analysis of the packet is done. Now, the process is handled based on the criteria whether the packet is active or non-active. The processing at the router on the packet is done only when the communication is active during the simulation time. If the simulation run is set as active, the packet is first read for the filename indication and the data is stored down on the packet. If there is data present in the data part, then it means that the filename indicates the name with which the data part needs to be stored in the router. After storing the packet data onto the router, the action is performed on the router based on the command present. This data file that is sent along with the packet contains the information about the various websites that needs to be blocked from being accessed. In some cases, the packet is checked for access control information using the attached file. The forwarding of the packet is mainly based on the destination subnet. If the current subnet is same as the destination subnet, then the packet is just send to the destination node or else, the packet is forwarded to the hub that is present in the destination subnet. If type of the packet is two, that means there is a request for a website blocking information file from the router, in that case, we need to check the whether the file is present with us. If the file is present, then we just send the file from the router itself or else, the requests is send to the next hop to get the required file. In general, the processing includes caching as well as processing on the fly.

7. Analysis of the experimental results

- *Packet format*

The packet structure is made as per the requirement of the application. The packet is made up of destination subnet, destination host address, filename, data and type field (active or non-active). The destination subnet and destination host address are stored as integers as we number the subnet from 1 to 4 and the nodes from 0 to 3. To handle the integer storage in OPNET, the necessary variable type is present. To handle the filename and the data storage as strings, the storage is done in the form of a structure that is made of an array of characters. The filename option is used to indicate the executable file that is used to act upon the packet. The data part is used to store the result of the processing on the packet.

- *Analysis*

The End-to-End (ETE) delay has been calculated both at the router level and the node level. The ETE delay is calculated by finding the difference between the time the packet is created and the time the packet reaches the hub or the destination. Time synchronization problem does not persist here because the whole simulation process runs on a single machine. The ETE delay is calculated for both active as well as non-active system. The ETE delay at the hub and at the node is shown in Figure 4.

The ETE value at the hub and the node vary but they remain the same in both the active and the non-active network models. We can infer from the graph of Figure 4 that the ETE delay for the hub is less than that of the node because the hub is physically closer to the source than the node. Also the packet density is more at the hub because of the following reasons:

- The hub is connected to more than one node,
- A hub receives message from the 4 nodes which are present in the subnet,
- The hub also receives any packet that is destined to any node in its subnet.

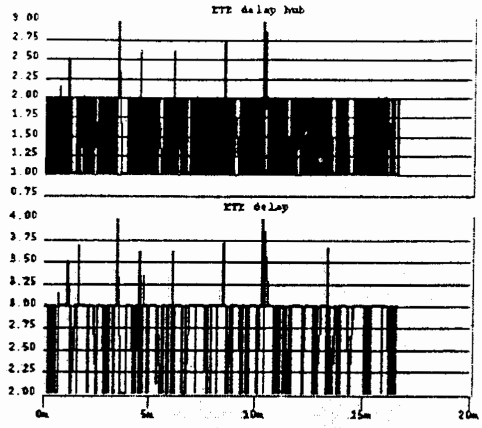

Figure 4. (x-axis: time, y-axis: simulation time) End to End(ETE) delay at the hub and the node

Each node receives its own packet and so less density of ETE entries, see Figure 4. We can also infer that there is an influence at the hub whenever there is a change in ETE at the node. This strongly supports any action to be taken at the hub. That is active processing.

Figure-5 shows the Throughput (Packets/sec). Throughput is defined as the average number of packets successfully received or transmitted by the receiver or transmitter channel per second. The throughput is not affected in the active case and it is the same for both the active and the non-active cases.

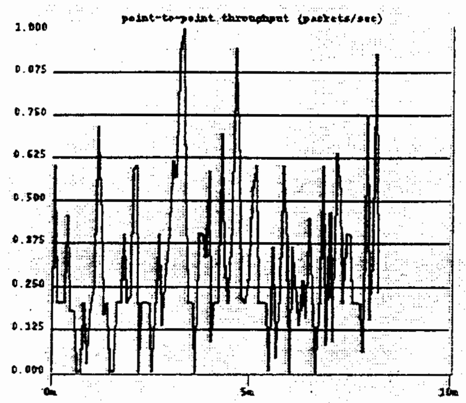

Figure 5. (x-axis: packets/sec, y-axis: simulation time) Throughput of the network for both the active and non-active cases.

The throughput at the node is almost same for active and non-active, as the number of packets destined to or send from the node are the same irrespective of active or non active system.

Packet inter-arrival rate is the time difference between the arrivals of two adjacent packets. If this rate is less then the packet generation is faster. Figure 6 shows the throughput of the node for a packet inter-arrival rate of 40 and 4 simulation time. Here we can infer from the graph that the packets received is high in number for the inter arrival time of 4 than that of 40 which is obvious.

The simulation time is less for packet inter-arrival rate of 4 seconds because of the fast generation of packets.

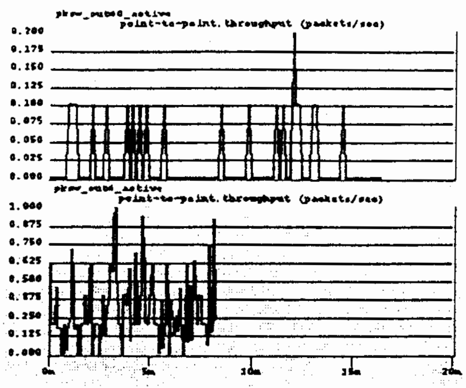

Figure 6. (x-axis: packets/sec, y-axis: simulation time) Throughput for the inter arrival time of 40 and 4 simulation time

8. Advantages and limitations of our system:

8.1. Advantages:

1. The processing can be done on the fly with the help of PVM in parallel or serially with the help of various routers. The router can distribute the job using PVM and PVMD.

2. The whole system is heterogeneous. Currently, it works on Windows NT, Windows 98 and Linux.

8.2. Limitations:

1. The system needs the nearby routers to be active routers so as to facilitate the distribution of the job of checking the packets.
2. When one router wishes to update onto the nearby router, the permission and security in case where on router is possessing corrupted data, it can corrupt the nearby router as well.

9. Conclusion

With the emergence of Active Networks, the network equipments have now become active instead of being passive. Now, by the introduction of parallel and updated checking on the packets for blocking of the improper websites, we help the processing to be done faster and avoid a mesh of regular manual updates on all the routers possible.

10. Acknowledgements

This work is supported by King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

11. References

[1] *Active Networks – A Survey*, Sohail Munir, Ohio State University, 1997
Internet:
http://www.cis.ohio-state.edu/~jain/cis788-97/active_nets/

[2] Takeshi Miei, Mitsuru Maruyama, Tsuyoshi Ogura & Naohisa Takahashi 1997. Parallelization of IP-Packet Filter Rules. Proceedings of Third International Conference on Algorithms and Architectures for Parallel Processing '91, 381-388.

[3] Mansour Esmaili, Rei Safavi-Naini, Bala Balachandran & Josef Pieprzyk 1996. Case-Based Reasoning for Intrusion Detection. Twelfth Annual Computer Security Applications Conference '96, 214-223.