PERFORMANCE EVALUATION FOR WEB APPLICATIONS WITH WEB CACHING IN A DISTRIBUTED WIRELESS SYSTEM USING OPNET™

M. GANJER ALI and SAJJAD ZAHIR
University of Lethbridge
Lethbridge, Alberta T1K 3M4, Canada

ABSTRACT

We analyze and evaluate the performance of a distributed wireless system with an added Web cache server. We use OPNET™ simulation tools to perform two experiments. In the first one, keeping the number of wireless clients constant, data traffic at the remote server running Web applications is analyzed as the cache hit rate of the caching device is varied. We noticed that the load at the Web server is improved by having increased caching capabilities at the cache server. Interestingly, it is observed that the traffic improvement is the best at a certain range of caching. The second experiment investigates the pattern of data dropped, and the delay at the remote Web server as the number of wireless clients is varied at a fixed cache hit rate. The results from this study are expected to help us understand the importance of cache servers while planning and designing a distributed wireless system with many clients.

Keywords: Wireless Local Area Network (WLAN), Web Caching, Cache Server, Simulation, OPNET™, Networking, Web Traffic

INTRODUCTION

Simulation is a very convenient tool in the field of network modelling that defines feasible network topologies, functioning devices and a context of particular scenarios. Results from simulations can be used in what-if analysis and performance evaluation of the network. Simulations can be used for LAN configuration decisions (11) and network traffic analysis (9). OPNET™ (Optimized Network Engineering Tool) provides a comprehensive development environment for the specification, simulation and performance analysis of communication networks (2, 5, 18). Recently, OPNET™ has become a very popular tool in network modelling and simulation research (4, 6, 12, 13, 17, 22, 23) and also in teaching (5). It employs a discrete event simulation approach that allows large numbers of closely spaced events in a sizable network. Specific OPNET™ tools that include wireless local area network (WLAN) capabilities are considered in this research.

Web caching and prefetching (3, 6, 8, 16, 19) are the most popular and widely used solutions to remedy Internet performance problems. Web cache reduces the bandwidth consumption and network latency by serving the clients’ requests from the cache (instead of the original source), and prefetching is a technique to preload and cache the Web object that is not currently requested by the user, but can be requested in the near future (19). The technology of caching effectively transfers copies of popular Web documents from Web servers to cache servers usually located near the Web clients. Although client machines also manage cache to provide immediate links to recently accessed Web sites, this paper concerns Web caching in a server, not with the clients.

A Web cache is an application residing in servers and clients and it watches requests for information objects identified as HTML pages, images, documents and files. Cache devices reply to the clients by sending the requested object and by saving a copy in the cache itself for future use, instead of requesting the original server again for the same object. The cache satisfies a request and a cached object is retrieved from the server only once, thus reducing the bandwidth use of the network. Therefore, the Web cache is used to reduce the latency and the traffic. The caching ability of a Web cache server provides similar functionality of an original server.

However, in this work we are less concerned about the prefetching and caching methodologies. As the World Wide Web becomes a dominant medium for information distribution, mechanisms are needed for efficient and reliable delivery of Web traffic. We will model a distributed wireless network with a cache server and study the performance of a Web server with variable cache hit rates and variable wireless client numbers. The results from this study are expected to help us understand the importance of cache servers while planning and designing a distributed wireless system. In addition, at the educational level such an exercise can be of immense pedagogical interest in both senior undergraduate and graduate level courses on networking, especially through the use of hands-on exercises (21).

Because of the reducing cost, technological advances and convenience of mobility, wireless networking supported by laptop computing is gaining momentum recently (e.g., Intel Centrino chip and platform-based technology). This will lead to higher levels of using the Internet and the Web. However, user perception about such benefits will eventually be affected by the performance of the network traffic and delays while browsing the Web. Investigating the performance evaluation of Web applications with Web caching in a wireless system is important. This is a first step toward a more elaborate study that may include a comparison of wired line versus wireless networks in a similar context. However, such a comparison requires studies that need controlling many other network parameters tied to specific vendor provided technologies and thus is beyond the scope of this paper.

In the second section, we present a discussion about wireless network scenarios; third section reviews OPNET™ simulation environment; in the fourth section, the network model is presented; simulation scenarios and results are presented in the fifth and sixth sections respectively and conclusions are in the seventh section.
WIRELESS NETWORK SCENARIOS

Wireless Local Area Networks (WLANs) are now common on many academic and corporate campuses. In practice, the network is growing rapidly and expanding everywhere; hence, the extension of the network is essential. The office grows as traffic grows. Specifically, there may be satellite offices at remote areas where infrastructure is not feasible or cannot be installed immediately. In such a case, the wireless scenario must be considered. Again, should the main server be far away from the satellite office, then a wireless distributed system is essential.

Wireless computer networks, especially the 802.11b Wireless Local Area Network (WLAN) specifications (10), have encountered increasing recognition during the last few years, resulting in a world that is progressively more mobile. The 802.11b WLAN architecture basically consists of one or more Basic Service Sets (BSS) and a Distribution System (DS). The basic part of the network architecture is the BSS. It consists of a group of Stations (STA) that are under direct control of a single coordination function. Network connections and client devices on the Internet are becoming increasingly heterogeneous. New types of client devices continue to emerge. The STAs are computing devices (e.g., desktops, laptops, handheld computers) with wireless network interfaces that communicate through a wireless medium. The geographical area covered by the BSS is known as the Basic Service Area (BSA) (10, 18). In an infrastructure network, all STAs communicate by channeling all traffic through a centralized Access Point (AP) (18). The AP controls the communication in the BSS as well as provides network connectivity between other BSSs. Thus, it has a bridging function. In contrast, in an independent BSS (IBSS), also known as an ad hoc network (10, 15, 18), any STA can communicate with any other STA without channeling the traffic through an AP. When interconnecting a wireless to other networks an AP is required.

A common distribution system (DS) integrates multiple BSSs. The DS does not specify any particular backbone technology and can be wired to a wide range of mediums. The integration of multiple BSSs using a DS is called an Extended Service Set (ESS) (10, 15, 18). The ESS provides not only access for multiple wireless users but also gateway access for wireless users into a wired network such as the Internet.

As the network traffic grows, the request processing also increases. There should be a way to minimize the traffic to the main server. A cache device near the clients is very important to reduce the traffic of the network. A scenario of a distributed wireless system where the application server is far away from the clients has been considered for the experiments. A cache device is connected near the client side by a layer 4 switch that redirects the Web application request from the clients to the cache device. That is, a remote satellite office with WLAN can get service from the remote server as the cache server reduces the traffic.

Layer 4 switches combine Network Address Translation (NAT) with higher-layer address screening and are very powerful. Having a wide functionality, layer 4 switches can reduce server load by balancing traffic across a cluster of servers based upon individual session information and status. These switches make packet-forwarding decisions based not only on the MAC address and IP address, but also on the application to which a packet belongs. They greatly simplify the DNS setup procedures. Switching takes care of multiplexing of incoming TCP/IP connections in place of round-robin DNS resolution (20). In most cases, a layer 4 switch is placed in front of a cluster of servers running a particular application. When a request for that application is made by a client, the switch determines which server should handle the request, often based upon current server loads. Once the forwarding decision is made, the switch sends that session to a particular server (14). The layer 4 devices enable establishing priorities for network traffic based on application and one can assign a high priority to packets belonging to vital in-house applications. The layer 4 switches also provide an effective wire-speed security shield for the network because any company or industry specific protocol can be confined only to the authorized switching ports or users. This security feature is often reinforced with traffic filtering and forwarding features (1). In the context of OPNET™, the particular features of the layer 4 switches are displayed in Figure 1.

We analyze and evaluate the above scenario for Web application with different cache hit rates and variable number of clients using OPNET™ simulation tools.

DISCRETE EVENT SIMULATION WITH OPNET™

OPNET™ is a commercially available tool for modelling and simulation of communications networks, devices and protocol (educational versions and access are available, as well). It is a vast software package with an extensive set of features designed to support general network modelling and to provide specific support for particular types of network simulation projects. It provides a comprehensive development environment supporting the modelling of communication networks and distributed systems. Both behaviour and performance of modelled systems can be analyzed by performing discrete event simulations. The OPNET™ environment incorporates tools for all phases of a study, including model design, simulation, data collection and data analysis.

Project modelling, node modelling and process modelling are three stages of basic network modelling (2, 15). Project modelling is where the general network topology is defined in terms of the scale and size of the network (e.g., world, enterprise, campus, office and x and y span in degrees, metres, kilometres), technologies to be used (e.g., ATM, Ethernet, wireless), and nodes and links. Node Model typically depicts the interrelation of processes, protocols and subsystems. It defines the behaviour of each network object defined in the project model. Behaviour is defined using different modules, each of which models some internal aspects of node behaviour such as data creation and data storage. Modules are interconnected by either packet streams or statistic wires (15). Packets are transferred between modules using packet streams. Statistic wires could be used to convey numeric signals (i.e., to support transmission of numerical state information). A network object is typically made up of multiple modules that define its behaviour. Process modelling is represented by the graphical power of a state-transition diagram with the flexibility of a standard programming language and a broad library of predefined modelling functions. It is created with icons that represent states, and lines that represent transitions between states. Also, some reasonable input values are defined before the network model is finally validated.

The WLAN models of OPNET™ support wireless-LAN backbones that consist of routers with WLAN interfaces belonging to the same BSS. These backbones can serve also the WLAN ESSs that are connected to the wireless backbone via their access points as they would be connected to a wired backbone. The scenario being discussed in this paper is built to provide an example on how such networks can be configured.

When a client sends a http request to the Web and if the requested page is stored (cache hit) in the cache server, it will
return the Web object to the client from its cache; if not (i.e., miss hit) then the cache server will send requests to the original server (8, 19). In some instance there can be an additional proxy server in between the client and the cache server. In that case, the http response will be returning the Web object from the cache server to the proxy server which relays the object to the requester. During this session the proxy server saves a copy of this object in its cache, as well.

**FIGURE 1**
The OPNET™ PROVIDED PARAMETERS FOR THE LAYER 4 SWITCH

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>switch</td>
</tr>
<tr>
<td>model</td>
<td>ethernet4_layer4_switch</td>
</tr>
<tr>
<td>Address</td>
<td>Auto Assigned</td>
</tr>
<tr>
<td>Priority</td>
<td>32768</td>
</tr>
<tr>
<td>Spanning Tree Protocol</td>
<td>RSTP (002.1w)</td>
</tr>
<tr>
<td>Timers</td>
<td>Default</td>
</tr>
<tr>
<td>BPDU Service Rate</td>
<td>1000000</td>
</tr>
<tr>
<td>Packet Service Rate</td>
<td>5000000</td>
</tr>
<tr>
<td>Layer 4 Redirection Info</td>
<td>(...)</td>
</tr>
<tr>
<td>rows</td>
<td>1</td>
</tr>
<tr>
<td>Application</td>
<td>HTTP</td>
</tr>
<tr>
<td>Destination Device</td>
<td>cache_server</td>
</tr>
<tr>
<td>Switch Port Configuration</td>
<td>(...)</td>
</tr>
<tr>
<td>rows</td>
<td>4</td>
</tr>
<tr>
<td>row 0</td>
<td>P0:Enabled,Auto,Auto,Link Speed Base...</td>
</tr>
<tr>
<td>row 1</td>
<td>P1:Enabled,Auto,Auto,Link Speed Base...</td>
</tr>
<tr>
<td>row 2</td>
<td>P2:Enabled,Auto,Auto,Link Speed Base...</td>
</tr>
<tr>
<td>row 3</td>
<td>P3:Enabled,Auto,Auto,Link Speed Base...</td>
</tr>
<tr>
<td>VLAN Parameters</td>
<td>None</td>
</tr>
</tbody>
</table>

Functions on the application processor of *gna clsvr_mgr* (15) provide cache applications. *gna clsvr_mgr* is a built-in process model of OPNET™ and its standard library and is part of the application processor. One of its functions is to contact the specified server for the requested data contained in the request packet, and another function is to determine if there was a hit or miss in the cache. If the outcome from a uniform distribution from 0 to 100 is less than or equal to the threshold, there has been a hit. This effectively models a random chance of a hit on the user inputs. Functions in *gna clsvr_mgr* determine if there was a cache hit or miss based on the Cache Hit Rate (CHR) input. Function for cache gets the application type from the gna packet. Since the cache server has the Ethernet working in the promiscuous listening mode, it will get packets from all other currently enabled applications. However, caching is only relevant for http applications. Summarized pseudo code for cache functions follows.

**THE NETWORK MODEL**

A key performance measure for the World Wide Web is the speed with which content is delivered to users. As traffic on the Web increases, users face increasing delays and failures in data delivery. Web caching is one of the key strategies that improve performance.

The proposed network model (see Figure 2) contains wireless email, ftp and http applications at the clients. The wireless server supports these applications, acting as the email, ftp and http servers simultaneously. The clients and server belong to different wireless-leans, BSS 0 and BSS 1 respectively. These two LANs are connected to each other with two routers forming a backbone of the distributed system. The cache server contains only the http application and thus mimics Web caching alone.

`router0` and `router1` are `wlan_ethernet_router` nodes and `router2` is a `wlan2_router` node (15). `wlan2_router` is equipped with two wireless receivers/transmitters, and hence it has two wireless interfaces (IF0 and IF1), whereas `wlan_ethernet_router` has only one wireless interface (IF). `router0` and `router1` are connected with a layer 4 switch (denoted by the node "switch" in Figure 1) using Ethernet interfaces. The switch only redirects the http traffic to the cache server. WLAN interface of `router0` is used as AP for the BSS 0. One WLAN interface of `router2`
serves as the access point for BSS 1. The remaining WLAN interface of router2 and WLAN interface of router1 compose the WLAN-backbone, which is BSS 2. To achieve this, the WLAN interface of wireless router0 was configured as an access point and its BSS ID was set to 0. The access point functionality of the WLAN interface of router1 was disabled and its BSS ID was set to 2. IF0 on the router2 became the access points, and IF1 was connected to the backbone with the access point disabled and BSS ID being set as 2. Hence, the backbone-LAN, BSS 2, does not have an access point and does not need to have one, though it is possible to configure one of the backbone interfaces as an access point. Additionally, the physical layer technology used by WLAN backbone-interfaces on the routers is set to "OFDM (802.11a)" to enable 802.11a data rates and their data rates are set to 54 Mbps. In other words, BSS 2 deploys the 802.11a PHY, while BSS 0 and BSS 1 use 802.11/11b PHY.

if (application type == Http application)
    if (uniform distribution <= hit_rate_threshold)
        
        1. There was a cache hit, process this request normally, i.e., Data is present in cache, respond to client.
        2. Update the cache hit counters: cache_hits++;
    
else

        1. There was a cache miss, i.e., data is not present in cache, contact the origin server for the data.
        2. Spawn a new client session to contact the origin server for the requested data.
        3. Update the cache miss counters: cache_misses++;

FIGURE 2
OPNET™ Modeler’s device creator is used to design the Web cache server node (Figure 3). Initially, the Web cache server is created as the default (i.e., as provided by OPNET™) and the various components at different levels are denoted by the rectangular boxes. Some of them supported by computer code (C++) and can be to change or add functionalities. For example in (7), the authors made significant changes to the given OPNET™ WLAN workstation’s media access control (mac) and the physical layers to make a better representation of an actual 802.11 b wireless LAN. They even added new code to allow modeling MANET routing, data-rate and power adaptation, more accurate path loss. In our case, we also redefine its application process (modelled by gna_clsrv_mgr in OPNET™) by removing the cache component to the node as a separate attribute called the “cache hit rate” or CHR.

**FIGURE 3**
Internal Design of the Cache Server

**FIGURE 4**
Setting Parameters for Designing the Cache Server Using Device Creator of OPNET™ Modeler

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache</td>
<td>None</td>
</tr>
<tr>
<td>L. Cache Hit Rate</td>
<td>None</td>
</tr>
<tr>
<td>SIP</td>
<td>None</td>
</tr>
<tr>
<td>[Applications] ACE Tier Configuration</td>
<td>None</td>
</tr>
<tr>
<td>[Applications] Destination Preferences</td>
<td>20%</td>
</tr>
<tr>
<td>[Applications] LAN Supported Profiles</td>
<td>30%</td>
</tr>
<tr>
<td>[Applications] Multicasting Specification</td>
<td>50%</td>
</tr>
<tr>
<td>[Applications] RSVP Application Parameter</td>
<td>60%</td>
</tr>
<tr>
<td>[Applications] RSVP Application Parameter</td>
<td>70%</td>
</tr>
<tr>
<td>[Applications] RSVP Application Parameter</td>
<td>90%</td>
</tr>
<tr>
<td>[Applications] RSVP Application Parameter</td>
<td>100%</td>
</tr>
<tr>
<td>Extended Attr.</td>
<td>E1K</td>
</tr>
<tr>
<td>Apply changes to selected objects</td>
<td></td>
</tr>
</tbody>
</table>

Then in simulation runs we set different values for CHR ranging from 0 to 100 at intervals of 10 as shown in Figure 4 which demonstrates how flexible OPNET™ is in accommodating designer’s choices. Essentially, the cache server is a normal server with caching ability, and the CHR is now configurable because of the minor changes made we made in the default design. If no cache hit rate is set (default setting is “none”), then the server operates as a normal server.

**SIMULATION SCENARIOS**

This scenario of Figure 2 demonstrates the use of Web Caching Server (WCS) and a layer 4 switch in a distributed system. The layer 4 switch will redirect all http traffic to WCS.
The switch implements the Spanning Tree algorithm in order to ensure a loop free network topology (15). Packets are received and processed by the switch based on the current configuration of the layer 4 switch’s redirection information and the spanning tree. The switch is capable of redirecting application traffic based on the application protocol. Normally, we use this switch in conjunction with http traffic to simulate transparent Web caching scenarios. WCS has the CHR attribute that determines how often objects are returned from the WCS directly to the client. If there is a hit, WCS will return the request object immediately; however, if there is a miss, WCS will open an http 1.1 session to the destination server, fetch the object from there and then return it to the client. In the latter case, object response time will be longer than average.

In client server architecture, a workstation connects to the server for traffic exchange. Any workstation can be used to generate traffic. Advanced server node models need to be used when back-end custom application is used. In OPNET™ network models, subnetworks are allowed and can be nested to an unlimited depth to construct complex topologies, and hence, a subnetwork of ten wireless workstations was designed to reduce the complexity of a whole network topology.

Network Model Attributes

In the first part of the simulation experiment, the network model has been designed with the following fundamental characteristics:

1. Wireless LAN Parameters: BSS Identifier was explicitly set on all the WLAN nodes in the network.
2. There was only one access point for a BSS network
   (Wireless LAN Parameters. Access Point Functionality)
3. 802.1 lb uses the following physical attribute values for Wireless LAN Parameters except the backbone WLAN:
   - Data Rate (bps) – 11 Mbps
   - Physical Characteristics – Direct Sequence
4. Client Application:
   - Destination Preferences: the application server.
   - Supported Profiles: Application (email, file transfer, and http as configured in Application and Profile configure nodes)
5. Cache server:
   - Application Supporting Service: http: application
   - Cache Hit Rate: set 0 to 100 for different scenarios.
6. The main server:
   - Application Supporting Service: email, FTP and http application
7. Switch Layer 4 Redirection Information:
   - Application: http
   - Destination: cache server
8. IP auto addressing scheme will assign IP addresses based on the BSS ID, i.e., all the WLAN nodes sharing the same BSS ID will be assigned an IP address from the same IP network.

In the second part, another network model was designed to see the effect of the number of clients. In this model, the cache hit rate was kept constant at 50%, and the client number was increased from 10 to 100, incrementing by 10 clients for each scenario. Results for each scenario indicate the http traffic successfully flowing over the WLAN backbone between the wireless clients and server.

SIMULATION EXPERIMENTS AND RESULTS

In this work, the scenarios were focused on the cache hit rate and client number of a wireless distributed system. Two sets of simulations were performed: the average application response time per 10% cache hit rate intervals and the average application response time per ten clients increased.

Since the user’s perception of the communication is the determining factor in terms of network survivability, it seemed reasonable to measure the performance from the user’s perspective in terms of application layer evaluation, specifically the application response times for http Web browsing, file transfer and emailing. However, http application is the main concern of this work. The following statistics are particularly useful for simulation of Web infrastructures:
- Data traffic of the http server
- Data dropped and the delay of performance
  - Data traffic sent and traffic received are recorded at one minute intervals of simulation period for 20% Cache Hit Rate intervals. Clients directed the traffic toward the main server, while the layer 4 switch redirected the http application to the cache server. The result is illustrated in Figure 5. It shows that the traffic is decreasing with the cache hit rate along with the simulation period. When the cache hit rate set 0, the entire http application traffic is directed at the main server. Figure 6 shows the steady state http application traffic of the main server (remember that it is our Web server and application server as well). In both Figures 5 and 6, the results correspond to simulations where the number of clients was 10. The outputs from simulation reached steady states after a ten-minute simulation time. It was seen that the traffic load of the server decreased as the Cache Hit Rate increased. When the CHR is 100, then http server traffic is almost zero. Therefore, it confirms our expectation that a cache device helps reduce the load and traffic congestion of the original server. We make an interesting observation from Figures 5 and 6 that the traffic drops almost by a half when the cache rate changes from 20% to 40% for 10 wireless clients. We repeat the simulation for 40 clients and notice the same large drop in traffic (Figure 7) for the cache rate change from 20% to 40% although the amount of drop is not as dramatic as in the 10 client simulation. However, before we can make a statement about any general rule, we need more experiments to understand the impact of other network parameters as well. In this study we have not manipulated any other network parameters and plan to investigate this phenomenon in more detail in a future study. Figure 8 shows the traffic of the server for http application only with a variable number of clients keeping the CHR fixed at 50%. With the number of clients increasing, the traffic increases and it is seen from Figure 9 that some of the data dropped. Therefore, the number of clients should be such that data dropped is kept at a minimum. Figure 10 shows the delay at the server with the
varying CHR and number of clients. The time delay decreases as the CHR increases; on the other hand, the delay increases with increasing the number of clients. Intuitively, it makes sense.
CONCLUSIONS

This simulation study demonstrates that the cache servers are very important to reduce the load of the application servers. As the cache hit rate increases, the load of the server decreases. Eventually, if the cache hit rate achieves 100% (practically not achievable) then the application server load decreases to zero. That is, caching was extended to Web servers in order to improve client latency, network traffic congestion and server load. Interestingly, we noticed from this simulation study that
the network traffic for Web applications was reduced by a large amount at a certain range of the cache rate (20% to 40%) at the cache server. This behaviour of the Web traffic was observed to be maintained at different numbers of the wireless clients albeit to a variable levels of improvement. Again, if the cache hit rate is fixed at a certain level, it is seen that the data traffic increases gradually as the client load increases. When the client load is substantially higher, some of the requests never reach the application server (i.e., data will be dropped and give errors.).

The bandwidth demand of the World Wide Web continues to grow at a hyper-exponential rate, and wireless networking is growing fast, too. Given this rocketing growth, caching of Web objects as a technical approach to reduce network bandwidth consumption is likely to be a necessity in the near future for WLAN. Again, the number of clients should be of a reasonable size so that data is not dropped. This stresses the need for better network management via segmentation.

Providing some practical knowledge about networking and Web architecture is becoming one of the goals in both undergraduate and graduate level programs in information technology. Use of such exercises built on OPNET™ simulation tools is expected to be a convenient and useful technique in classrooms to convey the importance of Web caching for network traffic and efficiency management. Current trend indicates that wireless technology will play a very significant role on networking and computer usage. In the context of Internet use, delays and congestion in the Web traffic will affect the user perception about the Web which will handle increasingly more multi-media information. Thus, our simulation-based study of the performance evaluation of the Web applications via Web caching in a wireless environment is a timely one. However, further investigation is needed for the comparison of wireless advantage with wireline advantage once various technological implications are taken into consideration.

ACKNOWLEDGMENTS

This research is partly funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Lethbridge. The OPNET Inc. has allowed the authors to use the software for instruction and academic research. We also thank the anonymous referees for their valuable comments that helped us revise the paper for improvements.

REFERENCES

1. Blackbox Website: http://www.blackbox.com/tech_docs/tech_overviews/switching_overview.html
20. Vergenet Website: http://www.vergenet.net/linux/mail_farm/html/node11.html

Spring 2006 Journal of Computer Information Systems 66