SONAR: A Statistic Repository of Mobility Platforms

Koshiro Mitsuya[1], Jean Lorchat[2], Thomas Noel[2], and Keisuke Uehara[1]

[1] Keio University, Japan. {mitsuya,kei}@sfc.wide.ad.jp
[2] University Louis Pasteur, France. {lorchat,noel}@dpt-info.u-strasbg.fr

Abstract. It is important for both network researchers and operators to know the behavior of mobile nodes (mobile hosts and mobile routers), and to find anomalies in their behavior. This paper describes an on-going effort within the Nautilus Project to provide a set of free tools to build a statistics repository, called SONAR, containing detailed information of mobile nodes. SONAR is especially designed for mobile routers inside cars because they are the only running testbed in a real environment so far. Users can send statistics information about their mobility system to a statistics repository by using these tools. The repository shows history and analysis results of collected data. The analysis includes classification of mobility, MIPv6/NEMO protocol evaluation, network access technologies benchmark and correlation between L2 and L3 technologies.

1 Introduction

Many new wireless access technologies are emerging on the market, and some like Wireless LAN or Bluetooth are widely deployed in our daily life. Nowadays, people can connect to the Internet where they want, when they want. Nodes are also used during movement. For this reason, the need for IP Mobility is significantly increasing and IP Mobility protocols such as Mobile IP [1] and NEMO [2] were standardized at the Internet Engineering Task Force (IETF). It is expected that e.g. the fourth-generation (4G) cellular networks or the Intelligent Transport Systems (ITS) will be built upon all-IP based technologies, where different access networks are seamlessly integrated and host or network mobility needs to be supported [3, 4]. Especially, it is said that a communications system based on the NEMO protocol exactly meets the requirements from the ITS industry [5].

While IP Mobility is being deployed, we are lacking benchmark tools to monitor mobility protocols. Most of the researchers have collected data necessary to design a mobility protocol through computer simulation. It is therefore unclear, at this time, whether the mobility protocols work as expected in the actual environment, and whether protocol configuration parameters (e.g. lifetimes of signaling messages, or re-transmit timers) are suitable for the real-life user. Thus, it becomes important to develop a monitoring system for mobility platforms. The monitoring system is used not only to evaluate mobility protocols, but also

to know the behavior of network access technologies, physical movement of mobile nodes, or whatever may have impact on designing mobility protocols. For example, in the case of the NEMO Basic Support protocol [2], there is no route optimization mechanism standardized yet. Thus, large-scale early deployments are likely to face issues when the load increases. This is one of the information that can be obtained using this system. Additionally, it is also worth improving the accuracy of simulation models by using parameters obtained from this system, which enables us to simulate actual behaviors of network access technologies and mobility protocols.

This paper describes an on-going effort within the Nautilus Project to provide a set of free tools to build a statistic repository, called SONAR, containing detailed information of mobile nodes (mobile hosts and mobile routers). These tools support several operating systems, protocol stacks and network access technologies. Users can send statistic information to the data repository, and the repository shows the history and some analysis results of collected data. At the moment, SONAR is designed for a mobile router inside a car because it is the only running testbed of IP Mobility in an actual environment. And the analysis results are aimed at being used for classification of mobility, IPv6 layer benchmark (MIPv6/NEMO protocol evaluation), L2 benchmark and correlation between L2 and IP layer.

2 Related Works

Network activity monitoring has become a very easy task nowadays thanks to many tools that are able to query the status of networking equipments, especially using the SNMP protocol. However, the rise of Wireless LANs and mobility protocols usage did not draw the attention of these tools yet, maybe because of their complexity.

Most large scale WLAN deployments resort to the use of SNMP to query the state of the Access Points (AP). This allows to know the L2 address of attached devices and the amount of traffic in bytes units. In a such deployment [6], authors went a step further by using remote syslog facility to log L2 management messages from each AP. These statistics are used to study the behavior of users, but have no focus on protocols.

Another very good approach to system evaluation was made in [7]. This paper describes a very powerful measurement system which is completely focused on L2 statistics. It uses broadcast messages sent from each station one by one, and recorded on all listening stations. This study allowed to classify L2 variables that have impact on the link itself, so that they can be accounted for in upper layers.

There are off the shelf solutions too, using the RADIUS protocol. However, as soon as RADIUS is involved, statistics are L3 specific, and their gathering becomes network centric.

3 Statistics Repository

The SONAR architecture is an hybrid solution that combines local statistics recording, and centralized storage and analysis : the repository is built in a single central location based on measurements results from many Mobile Nodes (MNs).

3.1 Parameters

Parameters listed below are collected and stored into MNs' local storage periodically or based on events. These parameters show a kind of snapshot of MNs, and the behavior of MNs can be known by comparing them. The timing to collect data is discussed in Sec. 4.2.

[p1] IPv6 HoA : the identifier of MN
[p2] Time stamp
[p3] IPv6 Mobile Network Prefix
[p4] CoA : the current location of MNs
[p5] Traffic Input : history of input traffic. number of packets and total bytes.
[p6] Traffic Output : history of output traffic. number of packets and total bytes
[p7] Input Mobility protocol : history of input-related mobility signaling. Number of Mobility Headers, Binding Updates and so on.
[p8] Output Mobility protocol : history of output-related mobility signaling. Number of Mobility Headers, Binding Updates and so on.
[p9] L2 statistics : description of network access technology (MAC address, bandwidth, security mechanism ans so on). Input and output frames count and overall byte amount. SS or SNR.
[p10] Platform, with bit masks for each parameter : description of the mobility system. Operating system, implementation name and protocols used.

3.2 Automated Analysis

Basic analysis such as the consumed bandwidth and the signaling overhead can be obtained simply by statistical processing of collected parameters. This section describes examples of advanced analysis.

Signaling Intensity with Respect to L2 Signal Strength

The first example is a correlation between L2 and L3 technologies, which is signaling intensity with respect to L2 signal strength (SS). This can be obtained by matching the percentage of lost *Binding Update* messages with the corresponding signal strength. We define the X ratio as :

$$X = \frac{N_{BindingUpdate}}{N_{BindingAcknowledgement} + N_{BindingError}}$$

Where $N_{BindingUpdate}$ is the amount of Binding Update messages sent by the MN, $N_{BindingAcknowledgement}$ is the number of Binding Acknowledgement messages received by the MN, and $N_{BindingError}$ is the amount of Binding Error messages received by the MN. X thus represents the percentage of Binding Update messages lost somewhere between MN and HA.

For a given time unit, the percentage of lost Binding Updates is to be compared with the corresponding signal strength. Since the minimum time unit for signal strength monitoring is $1e^{-3}$ second according to current driver implementations in operating systems, we chose to define a t time unit to monitor signal strength. On the other hand, the X percentage is computed for a larger period T. Of course, T should be longer than t and we can check the correlation between SS_T and X where $SS_T = mean(SS_t)$, SS_x being the signal strength over a given period x.

Classification of mobility
The classification of user's mobility is obtained by monitoring the L2 and L3 attachment information. The three kinds of user mobility that we want to monitor are summarized according to the parameters required to decide the user mobility class:

- Nomadic mobility when L3 attachment changes for each session
- Local mobility when L2 attachment changes and L3 attachment stays the same
- Global mobility when L3 attachment changes during the session
- Nested mobility when L3 attachment changes to MR managed subnet (mobile network)

4 Design of Evaluation System

4.1 Overview

Our evaluation system, SONAR, consists of two parts: client programs running on mobile nodes and a data repository server, as described in Fig. 1. One of the client programs, "monitor" in the figure, is running on mobile nodes (MNs) to collect statistic information from "Observed entities" such as mobility protocol stacks and network access technologies. It periodically stores statistic information into its local storage. Another program, "sender", sends the information encoded in an XML format to a data repository server (DS). "receiver" running on the DS stores the information into the local data base. "analyzer" processes and analyzes information collected by the above way on a regular schedule. The analysis results are made available to public via our web site.

4.2 Programs Running on Mobile Nodes

The client programs are silently running on MNs as daemons, so that users do not have to take care of any behavior of this system after configuration. And the

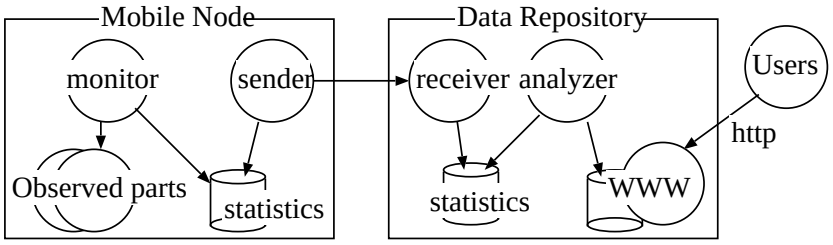

Fig. 1. Overview of SONAR, A Statistic Repository System

programs are provided as packaged software, like PORTS of FreeBSD or APT packages of Debian Linux, in order to be easy to install. These design principles help to involve more users, to collect many data and to provide better statistics information as a result.

The monitoring daemon running on MNs collects parameters needed for the statistics each N seconds and stores them in the local storage. The period N should be different for different parameters because the environment around lower layer changes after shorter periods than upper layer. For example, it is interesting to monitor the Signal Strength every second, but it is almost meaningless to observe L3 information each second. For the details of the parameters, refer to Sec. 3.1.

Additionally, the collection of parameters can happen based on events. These are when Care-of Address, an IP address which identifies the Mobile Node's current point of attachment to the Internet, changed and L2 trigger (the association is newly formed, lost or to be down) happened. This is because it is very interesting to compare the former and new status with respect to the event. These events typically lead to changes in network environment.

Another daemon that sends data to the repository, "sender" in Fig. 1, is defined separately from the monitoring daemon in order to accommodate for a demand of user flexibility. In fact, some users don't want to send data periodically and some want to send when broadband network access such as WiFi and Ethernet is available. The daemon sends an XML flow to DS each P seconds, and users can define the period P length. If users do not want to share the data at all, they set $P = 0$. Or users can start this daemon only when they want to send the data. An XML flow contains one single set of values that corresponds to a given N interval. Even when several results are dumped and stored in the MNs' storage, the result will not be summarized (not be sent at once). Each dump is sent separately.

The monitoring daemon consists of several modules which monitor each function. For example, the following modules are defined at IP Layer: IPv6, IPv4, MIP, FMIP and NEMO. For Data Link Layer: IEEE802.11a,b,g, Ethernet, GPRS, 3G and PHS. This modularity brings ease to manage and extend

for upcoming technologies. For example, what is necessary when a new observed entity appears is as simple as adding a module.

4.3 The Statistics Repository Server

A daemon running on DS receives the XML flow from MNs and stores this information in the data repository database. The XML flow is plain text so that we have a table with some information fields extracted from the flow for faster reference as well as hashing and sorting, and a field containing the whole XML flow.

A daemon analyzes collected data at 6:00AM JST(11:00PM CET) every day. It summarizes analysis results for classification of mobility, MIPv6/NEMO protocol evaluation, network access technologies benchmark and correlation between L2 and L3 technologies, as described in Sec. 3.2.

5 Implementation

We are developing this system on BSD and Linux at the moment. The implementations of mobility protocol used are SHISA and NEPL. SHISA [8] is an implementation of Mobile IPv6 and NEMO on BSD operating system and it has been developed in collaboration by KAME and Nautilus Project. NEPL (NEMO Platform for Linux) is a NEMO implementation for Linux kernel based on MIPL2 [9] and it has been developed by Go-Core project at Helsinki University of Technology in cooperation with Nautilus Project. We use several tools described later in this section to automatically maintain the statistics repository.

5.1 Method

Traffic Statistics: We use *netstat* to obtain network-related statistics data because it is installed on all BSD platforms as part of the default tools. It symbolically displays the contents of various network-related data structures. *netstat*, by default, displays a list active sockets for each protocol at the first form. The second form presents the contents of one of the other network data structures according to the option selected. With "-I interface" it shows information about specified interface including number of packets in and out. It also shows the number of bytes in and out with "-b" option.

On the other hand, we use *ifconfig* to obtain network-related statistics data on Linux. It is the standard network management tool found across many UNIX operating systems. It has many OS-specific extensions but can report on both Linux and BSD : the L2 address, the current L3 address for all existing L3 layers, as well as the traffic count for both packets and bytes, that flew through the interface for the Linux operating system.

Mobility Protocols Statistic: Both SHISA and NEPL have a virtual terminal accessible with the TELNET protocol. With this interface, we can retrieve information related to mobility protocols including the Home Address, the current Care of Address, and Mobility Signaling Statistics (These statistics are available only for SHISA at the moment).

Data Link Statistic: L2 specific information is extracted using specific tools depending on the underlying technology. For IEEE 802.11 WLAN cards, the data comes from *wiconfig* output on BSD, or from the Linux counterpart *iwconfig*. This includes L2 attachment point (AP MAC address and ESSID), security technology being used, bit rate, among others.

5.2 Triggers for local storing:

There are two classes of events that can trigger data collection, depending on the stack level monitored. The L3 events are triggered by the mobility protocol itself, and are usually notified by some specific socket (routing socket in BSD, Netlink socket in Linux). The lower layer (L2) events are recorded either by hooks in the L2 layer (wireless extensions for Linux) or in an easier way by polling the driver using an ioctl interface (both Linux and BSD).

5.3 Sending Data to the Repository:

Each set of parameters values that were measured during a N interval is sent by the *sender* module to the repository in a separate XML flow. The XML flow uses attributes to classify parameter types (IPv4 address, IPv6 address, byte count, packet count, timestamps, etc...) and the extensible nature of XML allows to add new parameters and types in an incremental fashion.

5.4 Implementation Status:

[*Comment*: KM: let's put the status and some analysis result here]

Currently, the implementation is under development. However, the mandatory set of features that allows to monitor an in-car mobile router are present. The *monitor* daemon for BSD system is ready and able to collect all statistics described in section 3.1. Despite the lack of a *sender* daemon for the user-side client, we could inject the first data sets by remotely copying the MR local storage to the DS and using a custom script. On the server-side, the *receiver* daemon is completed and able to inject data to the database.

This implementation in progress allowed us to collect our first results from an in-car mobile router. The detailed setup of the measured system is as follows :

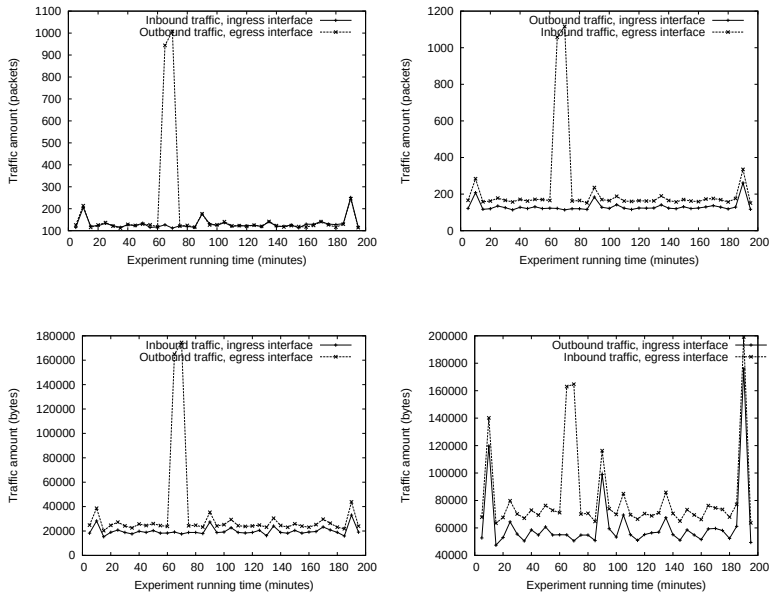

Fig. 2. First results from in-car mobile router

- FreeBSD 5.4 Mobile Router using KAME/SHISA implementation of NEMO Basic Support
- Ingress interface, Fast Ethernet interface
- Egress interface, 3G CDMA interface
- MAC OS X Mobile network node, using IPv6 applications

The results for both packets and bytes that flew through the interface can be seen in Fig. 2. In the bytes representation of outbound trafic in the case of the egress interface, the overhead caused by the mobility protocol is the obvious gap between both curves. The lower curve shows the traffic received on the ingress interface where as the higher curve shows the amount of actual data sent on the access network. This includes mobility protocol overhead caused by additional headers as well as signaling. In fact, it also includes the IPv6 over IPv4 tunneling overhead caused by the tunnel that we use to get IPv6 access using the 3G interface.

6 Considerations

6.1 Privacy Issue

Data protection and privacy is rapidly becoming one of the most important issues in the networking area today. IP privacy is broadly concerned with protecting user communication from unwittingly revealing information that could

be used to analyze and gather sensitive user data. This problem is particularly applicable to IP Mobility where the Home Address on a visited network can reveal device roaming and, together with a user identifier (such as an NAI), can reveal user roaming [10]. Another work addresses the problem of profiling IP and MAC identifiers [11]. Since the SONAR system tries to collect such sensitive information as IP and MAC identifiers, it is important to understand issues involved in user privacy.

There are two major issues involving user privacy. One is providing anonymity; Any Identifiers such as the Home Address, the Care of Address and the MAC Address must be scrambled by an irreversible hash before sent to the data repository. One is secure transport channel; The communication between mobile nodes and the statistic repository uses TLS/SSL transport.

6.2 Influence on Observations

By using SONAR, an amount of traffic is occurring because the daemon sends data to the statistics repository. This has impact on parameters such as *p5* and *p6*. In order to keep the meaningfulness of statistical data, it is necessary to remove this influence. There are two ways to achieve this. One is sending data after the experiments; this could be implemented by tuning the parameter P mentioned in Sec. 4.2. Another is discarding monitored data when the mobile node is sending data to the repository.

7 Conclusion

This paper described an on-going effort within the Nautilus Project to provide a set of free tools to build a statistic repository, called SONAR, containing detailed information of mobile nodes. Users can send statistics information related to the mobility system to the data repository by using these tools, and the repository shows analysis results. This architecture is currently being used on car-embedded mobile routers and will soon give access to results regarding scalability of the NEMO Basic Support.

This system is a work in progress, and the following topics the future works related to this system. In general, we need more experiments using this SONAR system to get best practice values for system parameters. This statistics repository is targeted at NEMO as it's first experimentation platform, and we expect to get very detailed results for protocol and users behavior. Then it's scope is going to be broaden to account for more mobility protocols from next generation Internet like MIPv6 and FMIPv6. Monitoring at HA would help to analyze scalability of mobility protocol, as HA is famous as being a potential point of failure. Multihoming support is also needed, so that the data structure must be updated to accommodate for this need.

References

1. David B. Johnson, C. Perkins, and Jari Arkko. Mobility Support in IPv6. Request For Comments 3775, IETF, June 2004.
2. Vijay Devarapalli, Ryuji Wakikawa, Alexandru Petrescu, and Pascal Thubert. NEMO Basic Support Protocol. Request For Comments 3963, IETF, January 2005.
3. Narumi Umeda, Toru Otsu, and Tatsuro Masanura. Overview of the fourth-generation mobile communication system. *NTT Techinical Review*, 2(9), September 2004.
4. Thierry Ernst, Koshiro Mitsuya, and Keisuke Uehara. Network Mobility from the InternetCAR Perspective. *JOIN: Journal on Interconnection Networks*, 4(3), September 2003.
5. Koshiro Mitsuya, Keisuke Uehara, and Jun Murai. The In-vehicle Router System to support Network Mobility. *LNCS*, 2662:633–742, Oct. 2003.
6. Kotz David and Essien Kobby. Analysis of a Campus-wide Wireless Network. *Wireless Networks Journal*, 11:115–133, 2005.
7. D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-Level Measurements from an 802.11b Mesh Network. In ACM, editor, *SIGCOMM'04*, Portland, USA, Sep 2004.
8. SHISA, Mobile IPv6/NEMO for BSD, Home Page, As of October 2005. http://www.mobileip.jp/.
9. MIPL, Mobile IPv6 for Linux, Home Page, As of October 2005. http://www.mobile-ipv6.org/.
10. Rajeev Koodli. IP Address Location Privacy and Mobile IPv6: Problem Statement. Internet Drafts draft-ietf-mip6-location-privacy-ps-00, IETF, October 2005. Work in progress.
11. W. Haddad et al. Privacy for Mobile and Multi-homed Nodes: MoMiPriv Problem Statement. Internet Draft draft-haddad-momipriv-problem-statement-02, IETF, October 2004.