The ETOMIC Active Probing Infrastructure – Demo Proposal

István Csabai, Péter Hága*, Gábor Simon, József Stéger and Gábor Vattay
Dept. of Physics of Complex Systems, Eötvös University, Budapest, Hungary
{csabai,haga,simon,steger,vattay}@complex.elte.hu

Eduardo Magaña, Daniel Morató, Mikel Izal and Javier Aracil
Departamento Automática y Computación
Universidad Pública de Navarra, Pamplona, Spain
{eduardo.magana,daniel.morato,mikel.izal,javier.aracil}@unavarra.es

***Abstract*— ETOMIC (`www.etomic.org`) is a European Union sponsored effort, that aims at providing a Paneuropean traffic measurement infrastructure. This infrastructure contains 15 PC based active probing nodes equipped with high-precision, sending capable DAG cards and GPS receivers to achieve time synchronization. Such cards are specifically designed to transmit packet trains with strict timing, in the range of nanoseconds. Every kind of active probing techniques can be applied on the nodes, from the quite simple ping application to the complex network tomography methods which are based on the synchronized sending capability of the DAG cards. The measurement nodes are centrally managed via a web platform, where the new arbitrary measurement jobs can be uploaded to and handled. The management system schedules the jobs and does the maintenance tasks. Now, the infrastructure is opened to the networking community. This paper describes the node architectures, the management system, and the proposed conference demonstration.**

I. Introduction and System Architecture

Internet traffic is growing at an extraordinary pace and as a consequence, traffic control and forecasting are becoming fundamental issues for network operators. The *ETOMIC* (European Traffic Observatory Measurement InfrastruCture) project is especially focused on realizing a Paneuropean measurement infrastructure, consisting measurement nodes which are deployed at selected European locations.

ETOMIC is targeted to provide the scientific community with a measurement platform that is i) fully open and reconfigurable and ii) extremely accurate (nanoseconds) and GPS-synchronized. ETOMIC has been designed to allow researchers to perform *any kind* of measurement experiments. To do so, researchers are provided with an interface from which software upload to the measurement nodes is possible. A choice of measurement scripts are also provided, that cover the most popular measurement techniques (packet pairs, etc). Researchers may also provide their own code for the experiments, that will be automatically compiled by the ETOMIC management system. The node reservation can be performed through the web-based user interface, while the ETOMIC management kernel takes care of the software upload and experiment execution in a fully automated fashion.

ETOMIC is a high-precision infrastructure, due to the fact that Endace DAG cards are incorporated into the nodes. Such cards are specifically designed to transmit packet trains with strict timing, in the range of nanoseconds. Actually, the DAG card transmit code has been specifically modified to serve the ETOMIC requirements. Furthermore, a GPS is also incorporated into the measurement nodes, so that the whole measurement infrastructure is synchronized to the same reference clock.

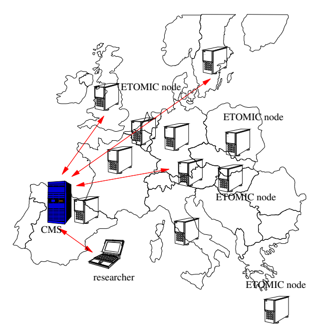

Fig. 1. Schematic of the ETOMIC infrastructure

The system components are depicted in figure 1. A central management system (CMS) is in charge of system control, comprising not only the scheduling and execution of measurements experiments, but also system monitoring (survivability) and configuration. The central management system is composed of a server and a traffic repository to which measurements can be downloaded to for subsequent processing. The software running in the management system is called the *management kernel*. On the other hand, researchers interact with the management system through the *user interface and account database*. ETOMIC provides an account to any registered researcher, so that they can upload software and download measurement results. Finally, the interface also serves the system manager to enter configuration data about the measurement nodes and researcher accounts.

ETOMIC is not the only one measurement infrastructure that is currently available in the state of the art, which includes

Surveyor, Felix, IPMA, AMP, and NIMI that is a large-scale measurement infrastructure with diverse administered sites. ETOMIC differs from the previous measurement infrastructures since it combines in the same platform i) high-precision, ii) GPS synchronization, iii) capabilities to run all kinds of measurement experiments and iv) central management for measurement job scheduling.

A. Hardware architecture

The measurement nodes are based on standard PC hardware. The motherboard is an Intel S875WP1-E with Intel Pentium 4 2.6 GHz processor, 1GB RAM, 200GB hard disk, two ethernet interfaces (one Gbit and another 10/100Mbit). The PC also includes watchdog functionalities. The operating system is based on a Debian GNU/Linux 3.0 (Woody) with enhanced kernel capabilities for low level network access without root privileges.

For the network monitoring interface an Endace DAG 3.6GE is used, which is specifically designed for active measurements. Such cards do not use interrupts to signal packet arrivals to the kernel, and, thus, packets can be captured at gigabit speeds. Shared memory is used as a mean to relay packets to the analysis program running in user space, in such a way that interrupts are avoided. Furthermore, the GPS reference signal is used to timestamp the incoming packets with high resolution. On the other hand, instead of having the kernel timestamp the arriving packets, timestamping is performed and as soon as the packet arrives by the card itself. As a result, no kernel induced jitter is present in the packet timestamp. DAG cards provide advanced capabilities for transmission: a burst composed of several packets can be transmitted with precise inter-packet timing (nanoseconds). To this end, the DAG card has an internal high resolution clock and also uses the GPS reference.

The GPS reference is based on a Garmin GPS 35 HVS. It is a low-cost water-resistant GPS receiver that is used to synchronize the measurement nodes. Specifically, the GPS provides a PPS signal (pulse per second) directly to the DAG card. The consequent accuracy is 100 nanoseconds in the packet timestamps and inter frame generation intervals.

B. Measurement node software and API

The measurement node software is implemented over GNU Linux operating system. Measurement nodes are always waiting for commands from the management kernel, meanwhile the nodes are stateless and their functionality is reduced to a minimum in order to improve the robustness of the whole platform in case of a single node failure. Accordingly, the node software only provides transfer resume functionality, and some other basic management tasks.

An API (Application Program Interface) has been developed for the DAG card, providing a high level interface that mimics the BSD socket interface programming style. In the future, we plan to extend the API with advanced capabilities such as traffic filtering inside the DAG card or even on-line calculation of basic traffic statistics.

II. Management kernel

The management kernel is the core of the central management system. It is in charge of scheduling the experiments and keeping the corresponding results in the traffic repository.

The researcher is expected to use the (web-based) interface in order to book several nodes during a certain time interval and in order to upload the measurement client and server. Such bookings are stored in the central database that will be described in the next section. Note that the database contains all the relevant information and software about the measurement experiment. Thus, the management kernel is continuously checking the database for new measurement requests. Then, Secure Sockets Layer is used by the kernel to securely command the measurement nodes. Such nodes are totally stateless for failsafe reasons, and communication is always started by the server. Once a new experiment has been defined and the deadline for execution approaches, the management kernel performs the following tasks: i) software upload and measurement node configuration, ii) experiment execution and iii) results download. Such tasks are internal to the system and transparent to the researchers, that will only use the web interface to upload software to the central management system and retrieve measurements.

The *software upload and measurement node configuration* task is performed before the experiment starting time. At this stage, the measurement programs are uploaded to the measurement nodes. Accompanying data files may also be uploaded that contain traces or measurement parameters. This allows to upload many different parameters for one single experiment. The data transfer has been enhanced with transfer resume functionalities in order to be able to manage very large data files. Once the files are uploaded, the measurement nodes are programmed with the starting and ending time for each of the executables that are going to be run for the experiment. Several subtasks are performed for this task, such as quota reservation for the experiment, file upload and time programming for each of the measurement nodes.

During the *experiment execution* task, the management kernel is on standby until the end of the experiment. It only has to ensure that no other experiment is using the same measurement nodes in the same time interval. The aim is to completely isolate the measurement nodes so that the high-precision measurement hardware can be exclusively devoted to a single experiment.

Once the experiment is finished, the management kernel will *download* the resulting data files from each measurement node. Note that during the download phase the measurement node cannot be assigned to another measurement experiment, since the measurement and the download would interfere with each other. An estimation of the download time for an experiment is performed by the management kernel and stored in the database as part of the reserved time interval for such experiment. However, since download time is highly dependent on the network connectivity, transfer resume capabilities are necessary. Thus, the download phase may be interrupted by the management kernel and resumed in the future, depending on the number of pending measurement requests.

Besides the experiment management tasks described above, maintenance tasks are also performed by the management kernel. Such tasks have low priority and they take place when no other task is scheduled for a given measurement node.

In order to do the scheduling of experiments and maintenance tasks a calendar will be used in the management kernel, as shown in figure 2. There is a calendar per measurement node that enforces resource isolation by not allowing concurrent experiments to run over the same node. The gaps between experiments are used for maintenance tasks. Figure 2 shows an example of scheduling for experiments that are running on non-overlapping sets of measurement nodes.

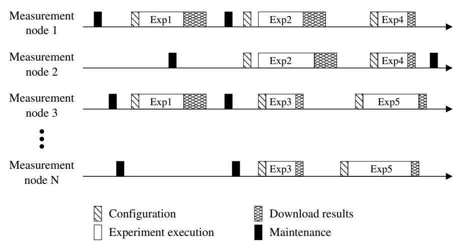

Fig. 2. Management kernel time reservation structure

A. Database and user interface

ETOMIC provides an interface for researchers and administrators which fulfils the different requirements they may have. Researchers require capabilities to define new experiments, to reserve the measurement nodes and to download results from the management system. Administrators require functionalities to manage users, measurement nodes, software and experiments. As now we are focusing on the usability of the system the detailed discussion of the administrative functionalities is not provided here.

Users are provided with a graphical interface for setting up the experiment beforehand. Then, the management kernel is in charge of experiment execution. The interface is based on the ubiquitous Web service. Using HTML4, Javascript, PHP4 and Apache Web server (using a Secure Sockets Layer) we were able to provide interface capabilities which are close to those offered by end-user applications. The aim is to facilitate the definition of the experiments as much as possible. This targeted simplicity and ease of use posed more stringent challenges on the researchers interface.

ETOMIC will mostly be used by researchers willing to define measurement experiments. In the demonstration we will focus on the user functionalities.

III. Proposed Demonstration

During the demonstration we would like to introduce the ETOMIC infrastructure, show the web user interface and its main functionalities. In this process we want to show how to upload the programs to the sender and receiver nodes, create a new measurement bundle, book a time-slot and download the measured data step by step:

- adding a new program: In order to run simple experiments, several scripts for sending and receiving IP packets with the high-precision timestamping DAG cards are provided. However, not only *off-the-shelf* programs are offered but also an API for creating new programs for whatever experiment the researcher may have in mind. Hence, the interface offers capabilities to upload new programs and input data files.
- creating an *experiment-Bundle*: The definition of a new experiment comprises several measurement nodes and several programs running on each node, with different starting and ending times. This is called ”Experiment Bundle”. First, the measurement nodes that make up the experiment must be selected. Next the programs to run on each node are chosen, and finally for each of the programs that make up the experiment bundle a starting and ending time can be provided *relative to the starting time of the experiment* (even asynchronous execution of different programs can be applied).
- booking ETOMIC time: Once the bundle definition is performed, the researcher should define the starting time for the experiment. The interface shows the availability intervals per node, together with a best-fit suggestion, where the user can select the appropriate time slot.
- obtaining results: The management kernel is in charge of downloading the data files created or modified during the experiment. Once the download is over, the researcher will have access to such files through the interface.

We are also going to make a short on-line experiment. We would like to make a one-way-delay measurement between two etomic nodes in both directions. Finally we will discuss the results evolving from the measured data.

During this short introduction (to the usage of the ETOMIC infrastructure) the audience will get the first lesson of how to use this new system to collect their own measurement data. If they interested in, they can immediately apply for an account to the central management system.

During the presentation we are going to use our notebook computer. We need Internet access during the demonstartion because it is an on-line demo. We also require a projector for better visibility if it is possible.

The authors thank the partial support of the National Science Foundation (OTKA T37903), the National Office for Research and Technology (NKFP 02/032/2004 and NAP 2005/ KCKHA005), the EU IST FET Complexity EVERGROW Integrated Project and the Spanish Government (TEC2004-06437-C05-01).

References

[1] D. Morato, E. Magana, M. Izal, J. Aracil, F. Naranjo, F. Astiz, U. Alonso, I. Csabai, P. Haga, G. Simon, J. Steger and G. Vattay, *ETOMIC: A testbed for universal active and passive measurements*, in *Proceeding of TRIDENTCOM 2005*, **Best Testbed Award**, p283-289, Trento, Italy, February 23 – 25 2005. `http://www.etomic.org/`.

[2] E. Magana, D. Morato, M. Izal, J. Aracil, F. Naranjo, F. Astiz, U. Alonso, I. Csabai, P. Haga, G. Simon, J. Steger, G. Vattay, *The European Traffic Observatory Measurement Infrastructure (ETOMIC)*, in *Proccedings of IPOM 2004*, p165-169, 11 – 13 October 2004, Beijing, China. `http://www.etomic.org/`.