MIRA: A Distributed and Scalable WAN/LAN Real-time Measurement Platform

Ricardo Romeral[1], Alberto García-Martínez[1], Ana B. García[2], Arturo Azcorra[1], and Manuel Álvarez-Campana[2]

[1] Department of Telematic Engineering
Carlos III University of Madrid (UC3M)
{rromeral, alberto, azcorra}@it.uc3m.es
[2] Department of Telematic Systems Engineering
Technical University of Madrid (UPM)
{abgarcia, mac}@dit.upm.es

Abstract. In this paper we describe MIRA, a distributed and scalable architecture for flow-based monitoring and traffic analysis. MIRA relies on data inspection to provide advanced monitoring services such as Acceptable User Policy auditing. It is based on a low cost hardware platform that can be deployed in both LAN and WAN environments. The distributed architecture of the measurement platform is designed to provide real-time global results from a complex network. Processing rate can be seamlessly increased by the addition of new hardware elements. The MIRA architecture has been thoroughly tested in a field trial on RedIRIS, the Spanish National Research Network.

1 Introduction[3]

Network services are playing a capital role in our society, and this trend is steadily gaining momentum. As a consequence, demand for network resource monitoring is increasing to asses a given Acceptable Usage Policy (AUP) in the network, to prevent, for example, the transport of inappropriate contents or the abuse of network bandwidth by certain users.

In this article we present the architecture of MIRA, a distributed and scalable measurement platform based on a novel approach for IP traffic analysis. While most analysis tools rely on address and port inspection, MIRA incorporates real-time pattern search to inspect the transferred data applying user definable heuristics to obtain more elaborated information, for example, to classify the traffic into acceptable or non-acceptable.

The modular and distributed architecture allows traffic inspection to be performed over complex networks comprised of physically remote links. Although the system was initially designed for ATM STM-1 WAN environments, MIRA

[3] This research was supported by the MIRA (Methods for IP traffic Analysis) project, funded by the Spanish National R&D Programme under contract CICYT 2fd-97-2234-c03-01.

has extended its functionality to Ethernet LAN inspection, enabling combined backbone and campus network analysis.

Data processing is performed in a low cost hardware platform, based on conventional PC boxes equipped with a free Unix operating system. Scalability, in terms of analysed traffic, is achieved by replication of the capture and processing elements.

The platform has been adapted to monitor IPv6 packets, since this protocol is expected to spread in the near future. The MIRA platform allows simultaneous monitoring of networks carrying a mix of IPv4, IPv6, and tunnelled traffic such as IPv4 over IPv4 and IPv6 over IPv4 (expected in the initial transition stages to IPv6).

The development of this platform has continued the work initiated by previous Spanish R&D projects CASTBA [1] and MEHARI [2]. The presented architecture is complemented by the analysis tools and GUIs support developed by the CCABA group at Universitat Politècnica de Catalunya.

The remainder of the paper is structured as follows: in section 2 we summarize related work. In section 3 we present the functional architecture of the MIRA platform, describing in detail each functional module. Section 4 is devoted to the study of distribution and scalability, with some examples of possible configurations. Finally, conclusions and future work are presented in section 5.

2 Related Work

Several tools have been developed for traffic classification, starting with the well-known tcpdump. However, few of them are based on content pattern inspection, and less allow measurement distribution and scaling. We will review some of the related measurement approaches.

Snort [3] is a traffic analysis and packet-logging tool that performs protocol analysis as well as content search to perform security hazard detection. It uses a flexible rule language to describe the traffic that it should collect. However, it cannot be neither distributed nor easily scaled, so it is limited to LAN environments. Bro [4] and the Network Flight Recorder [5] are similar tools presenting the same drawbacks.

CoralReef is a software suite developed by CAIDA that provides a programming library to collect traffic using PC boxes. It supports IP traffic capture over several link layers, including ATM (with the monitor components OC3MON and OC12MON [6], that have been used for monitoring the vBNS network [7]). CoralReef includes software for header-based analysis and web report generation, but no content inspection is performed and no special support is provided for measurement distribution.

Commercial software is available for traffic measurement, mainly based on address/port flow analysis. An example is NetFlow [8], from Cisco Systems, aimed to collect resource utilisation on a per flow basis. However, NetFlow does not perform content inspection. Further processing can be performed by applications

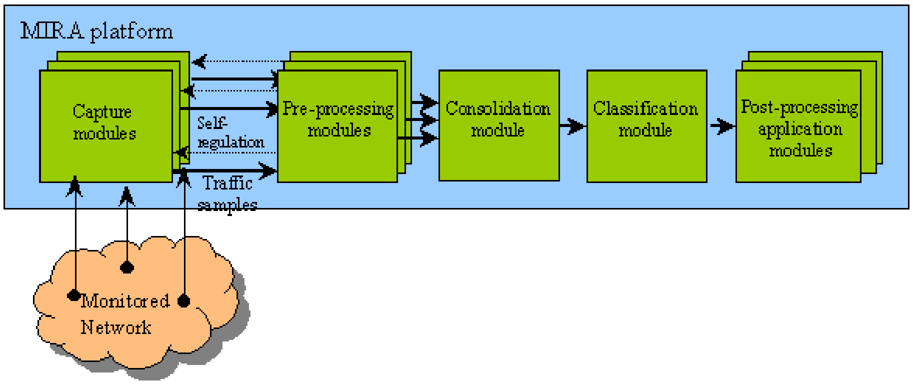

Fig. 1. Basic MIRA architecture

running on PCs or Workstations, either with Cisco proprietary software or with free-software tools like cflowd, but data capture scalability is not considered.

RTFM's NeTraMet [9] proposes a distributed measurement architecture comprised of *meters*, *meter readers*, *applications* and *managers*, to allow several network traffic processing applications to access to the same data. However, meters only provide traffic information based on addresses and ports, and data capturing scalability is not tackled.

RMON [10] defines an SNMP MIB interface for accessing the data measured on a network probe, including address and port inspection, and for programming trace recording. However, the architecture is not intended for intensive data inspection, since access to captured data is performed using SNMP.

The NIMI (National Internet Measurement Infrastructure) project [11] has defined a distributed performance measurement architecture. Internet performance is characterised by means of tests that are performed among the monitoring platform nodes, to obtain end-to-end figures. This approach is completely different to ours, since its aim is network performance characterisation.

3 Functional Architecture of the MIRA Platform

The basic architecture of the MIRA system is shown in figure 1. The capture and analysis functions are split into several modules, which are arranged in cascade. Each module is implemented by one or several processes, which can be hosted in different physical devices. Communication between processes is performed by means of files that are periodically written and read. When the modules are to be hosted in different physical hardware platforms, NFS is used to allow file sharing.

In the rest of the section we will describe the functionality of each module.

3.1 Capture Module

This functional block is responsible for capturing the traffic (IPv4 and IPv6 packets) that will be analysed by the rest of the modules and for dumping this data into capture files. Modularity provides several benefits. First, it eases including additional link layer technologies. Second, several capture modules can simultaneously collaborate even if different link layer technologies are used.Finally, the overall system throughput can be increased by placing several capture probes over a single link or subnet.

The available capture modules are:

An ATM WAN module The ATM capture module relies on PC Fore network interface cards connected to passive splitters that probe each direction of a send/receive fibre pair. Splitters are used to asure that the measurement process does not affect negatively network performance. A FreeBSD driver has been adapted for the capture of complete IP packets. The capture can be performed either in promiscuous mode or applying VPI/VCI filtering. Cells are reassembled into AAL5 frames, that carry the whole IP packet. The capture driver adds some information, such as the VPI/VCI pair used for the transmission and the capture timestamp.

A LAN module This module for Ethernet and Fast-Ethernet networks is based on the services provided by the libpcap library.

A capture module can generate one or more sequences of constant-size files, each one containing a specific subset of the captured data. The captured file sequence determines the unit of data that can be separately handled in the following steps of the process, so it is an important parameter for scalability. Different criteria can be used for the generation of these sequences: in ATM, VPI/VCI filtering, outgoing and incoming traffic (depending on the fibre of the pair used); for Ethernet, grouping can be based on IP/MAC origin/destination addresses.

A MIRA *flow* is defined as a group of IP packets that have the same pair of IP address and application port both at source and destination. Inside the file, the data captured for a given flow is marked with a label to distinguish among different aggregated flows. The aggregated flow is in most cases the unit of data for which final results are going to be generated. Different flows can be assigned to the same aggregate. Typically, it will be defined taking into account geographical or organisational criteria. For ATM, VPI/VCI pairs can be used for defining aggregated flows. For Ethernet, IP/MAC origin/destination address filters can be used, allowing for example the definition of incoming, outgoing, transit or internal traffic with regard to a given network.

3.2 Pre-processing Module

The capture module generates large amounts of sampled information. It is necessary to process this information as soon as possible in order to reduce the storage demand discarding unnecessary data. Data privacy also justifies fast data removal. Fortunately, early data deletion is possible if the final goal is traffic classification or accounting, although selective packet logging can still be programmed if required. For each captured file, the pre-processing module extracts the relevant parameters from the samples and deletes the files resulting from the capture process when they are no longer necessary, generating a much smaller file. This module carries out several tasks:

Fast classification For some purposes, IP addresses and TCP/UDP ports can provide the required information. For example, if one of the IP address corresponds to a known server of inappropriate content, no additional information is required for classification, so pattern scanning is not performed.

Symptom detection This is the core activity of the MIRA system, and also the one that consumes most resources. Its aim is to identify *symptoms*. A *symptom* is a group of patterns, represented by a label, that refer to a similar usage profile.
A pattern can be a character string or a binary sequence. An example of a possible symptom could be the "MAIL" symptom, defined to mark all the flows that are expected to belong to SMTP email transference. Some patterns associated to the symptom "MAIL" could be the presence of the string "RCPT To:", or the presence of the four letters "EHLO" at the beginning of a line. The association among patterns and symptoms is defined in a configurable database. As we can see, in general, several patterns can be associated to the same symptom. Whenever a pattern is found on a packet, the corresponding symptom is added to the characterisation of the corresponding flow, and a record of the number of occurrences of each symptom is maintained (note that the patterns themselves or its occurrences are not stored). For the specification of the patterns that define a symptom, heuristics, including natural language strings, application protocol commands or binary data format (e.g. delimiters of an MP3 audio file), can be used.

Flow aggregation Aggregation, i.e. accumulation of statistics and symptoms related with a flow stored in different captured files, is performed periodically, with a period that can be adjusted by the administrator. For tunnelled packets, flow aggregation is not based on the outer packet header data, but on the tunnelled addresses and ports. For each flow, the pre-processing module computes the number of captured packets, the number of captured octets, and a list of the symptoms found and the number of occurrences.

Pre-processing is intensive in computing resources. To prevent the fast producer capture module from flooding the slow consumer pre-processing module, a self-regulation mechanism is introduced that stops the capture when the number of files that have not been pre-processed exceeds a given threshold.

3.3 Consolidation Module

Although MIRA is a real-time distributed traffic inspection platform that allows several monitoring systems to be placed in different network locations, the results obtained on each meter element, analysing different flows, are gathered together into a single report. This process will be performed in one consolidation module per MIRA platform. The fact that each meter can work at a different capture rate has to be taken into account by this process to guarantee that the consolidated results are correct.

Before generating a single report, bi-directional flow correlation is performed (obtaining the so called *biflows*), to account for IP communication involving both IP directions. This process increases classification accuracy, since in many cases relevant symptoms only arise on one direction. Flow correlation allows, for example, linking the symptoms derived from natural language text patterns found in HTTP responses with HTTP requests in which no relevant content for the considered symptoms is transmitted.

Note that information gathering should not be performed at a prior stage: if the data were consolidated before pre-processing, large amounts of data, holding the whole captured data, should be delivered from possible many different places to the central host in which consolidation is performed.

3.4 Classification Module

The aim of this module is to classify each biflow according to usage categories, if it has not been classified before by the pre-processing module (see fast classification, section 3.2). These categories can be later used to audit network usage or to enforce Acceptable Usage Policies. The MIRA administrator can define the categories to process, and the rules to apply to classify a flow into one category, that are based on relations among the number of symptom occurrences (majority of symptoms, comparison of the number of symptoms, etc.). For example, we can define a Leisure class for flows in which there is a majority of symptoms such as MP3, GAMES, VIDEO, SPORTS, etc.

Note that several combined strategies could be required for the detection of the stealthy usage of particular applications, such as MP3 audio distribution in environments where port restriction applies. Different ports and even different applications could be used for data transference, so port identification alone may be useless. In this case, the detection of the MP3 symptom (defined to identify Leisure traffic) could be performed by a search for the string ".mp3" combined with the pattern-based detection of the usage of ftp (that would not necessarily be linked to the Leisure class determination) or other file distribution applications, along with the binary inspection of the transferred data to detect a sequence of eleven consecutive bits with the value of 1 [12].

3.5 Post-processing Application Modules

The MIRA architecture allows the development of application modules that extend the functionality of MIRA. The available data at this stage include not only the classification results, but the addresses, ports and symptoms detected.

An example of a post-processing application module is the Network Usage Statistics Module. This application gathers classification information based on aggregated-flow identifiers (VPI/VCI for ATM networks, or administrator-defined address aggregations for Ethernet), that usually refers to topological locations, for example, a city. The module accumulates data for each aggregated-flow, and extrapolates the results obtained by sampling to allow even comparison among different aggregated-flows.

Other applications organise the resulting data taking into account the source and destination Autonomous Systems, adapt Snort [3] detection rules to be used by MIRA to perform security intrusion detection, or provide billing and charging information [13].

4 MIRA Distributed and Scalable Measurement Architecture

In this section some guidelines for the deployment of MIRA in a distributed network are suggested. Scalability is also addressed, since it is a main concern when high performance networks are inspected.

4.1 Distributed Measurement

One capital problem that arises when distributed measurement is performed is avoiding the measurement of the same traffic in different physical locations. We can illustrate the problem with the example of an ATM WAN star-based network, with a LAN segment attached to it whose internal traffic is required to be inspected (figure 2). If all the traffic were captured on each link, data traversing from one inspected network to another one (e.g., from A to B) would be accounted twice. We would like to assure that each packet can be captured, and that it will not be captured more than once. A way of achieving this would be to capture on each link just the traffic going in one direction, either from the central node to the considered network or vice versa. The same policy should be forced in all the WAN probes. As a consequence, traffic going from network A to network B will only be captured once. Additionally, the whole traffic sent to and received from the central node should be captured, to assure that all the traffic is inspected once.

Note that, in the configuration described above it is useless to try to correlate the flows gathered on the same link, since different directions of the same traffic are probed on different links. However the flow correlation process performed in the consolidation phase will solve the problem.

To prevent the Ethernet MIRA probe placed in the shared Ethernet link from capturing duplicated data, this probe should be configured to capture and analyse only internal traffic. Another option to consider is the removal of the ATM probe placed on link C, delegating the capture of the traffic to the Ethernet probe (provided that traffic destined to or sourced at the C exit router and communicating with networks A or B is not relevant).

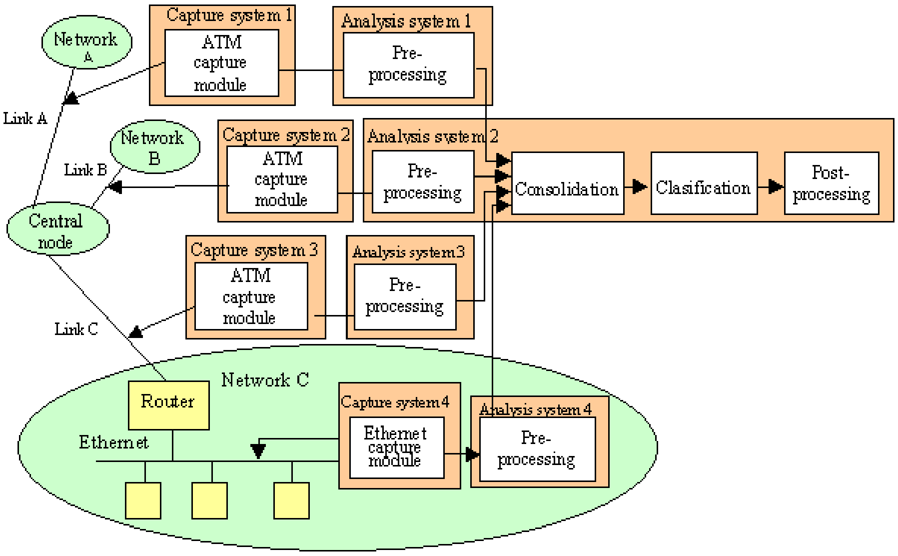

Fig. 2. Distributed Measurement Example

4.2 Guidelines for Deploying a Scalable System

One important concern about traffic inspection is system scalability, as large amounts of data have to be scanned when dealing with high capacity networks.

Although the whole MIRA platform can be hosted in a single device, there are several ways of increasing performance by the addition of new hardware. First of all, we have to stress that different modules can run in different machines, provided that file sharing is allowed. Since real-time consolidation, classification and post-processing are not highly demanding tasks, focus must be set on distributing capture and pre-processing. In the field trial performed in RedIRIS, six PCs (Pentium II, 650 MHz, 256 MB of memory) were arranged into three pairs of capture and analysis devices to inspect three pairs of STM-1 ATM fibres. With this configuration, the percentage of analysed traffic is close to 5%, despite of the CPU-intensive processing required. The computed statistical relevance of the data obtained allows assuring a 5% confidence interval for the mean daily results over a given month with a 95% confidence level. A deeper study on the relevance of the obtained data can be found in [14].

A step further to increase performance would be to split symptom detection into two devices (see figure 3). In this example, outgoing and incoming ATM captured traffic is separated to be pre-processed in two different machines, so two sample files are generated. The capture element can be configured to generate a different number of files that group traffic with the same criteria used for the definition of the aggregated flow. This possibility allows the optimisation of hardware usage; for example, in the test field with the equipment described above, we have measured that pre-processing takes around 2,8 times longer that

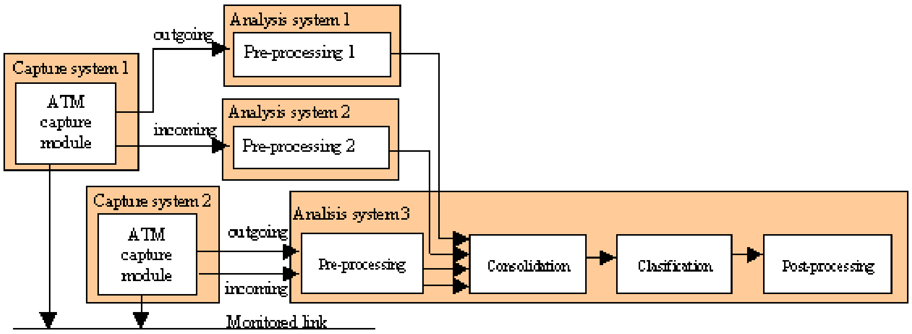

Fig. 3. Example of performance increase by new hardware addition

capturing; therefore, it makes sense to have from 2 to 3 machines performing pre-processing for each capturing element.

If performance has to be further increased, new capture elements (and associated pre-processor devices per probe) can be added. In this case, it must be assured that each probe samples different traffic, programming the probe to capture a different set of aggregated-flows.

5 Conclusions and Future Work

Acceptable User Policy monitoring, to allow subsequent enforcement, is a requirement for many networks. MIRA is a measurement platform especially aimed to fulfil this goal, relying on configurable pattern search to perform traffic classification, in addition to classic address and port analysis. The content search strategy implemented in MIRA broadens traffic inspection possibilities, allowing for example the detection of content formats transported over non-expected transport ports or application protocols (like MP3 over new peer-to-peer communication protocols).

To the best of our knowledge, this is the first distributed and scalable pattern-based analysis tool. MIRA's modular architecture allows achieving probe and analysis distribution, and processing scalability. Measurement distribution and scaling have been addressed in the paper, and some configuration guidelines have been issued. The first two stages of the MIRA architecture, capture and pre-processing, can be distributed at administrator convenience. This allows the increment of analysed traffic, since the analysis process can be speeded via hardware replication, and also enables measurement on complex network topologies. WAN and LAN combined analysis, based on the availability of ATM and Ethernet/Fast Ethernet capture modules, has also been discussed. The distributed measurement system has been field tested in RedIRIS, the Spanish National Research Network.

As further work, we should highlight that additional experience in heterogeneous networks is required to fully validate the generality of the MIRA architecture. Additionally, new link technologies could be added to the available capture modules. Interesting choices for WAN analysis would be Packet Over Sonet (POS) and Gigabit Ethernet.

References

1. M. Alvarez-Campana, et al. CASTBA: Medidas de tráfico sobre la Red Académica Española de Banda Ancha. Proceedings of Telecom. I+D, Madrid. October 1998.
2. P. J. Lizcano, A. Azcorra, J. Solé-Pareta, J. Domingo-Pascual, M. Alvarez-Campana. MEHARI: A System for Analysing the Use of the Internet Services. Computer Networks and ISDN Systems (ISSN 0169-7552), Vol. 31, Num. 10, November 1999.
3. M. Roesch. Snort - Lightweight Intrusion Detection for Networks. In proceedings of the USENIX Systems Administration Conference (LISA). November 1999.
4. V. Paxson. Bro: A System for Detecting Intruders in Real-Time. In Proceedings of the Seventh USENIX Security Symposium, pages 31–51, San Antonio, Texas, January 1998.
5. M. J. Ranum et al. Implementing a generalized tool for network monitoring. In Proceedings of the USENIX Systems Administration Conference (LISA), San Diego, CA, October 1997.
6. Claffy et al. OC3MON: Flexible, Affordable, High-Performance Statistics Collection. Proceedings USENIX. September 1996.
7. J. Jamison, R. Wilder. vBNS: The Internet Fast Lane for Research and Education. IEEE Communications Magazine, pp 60-63. January 1997.
8. Cisco Systems. NetFlow Services and Applications. White paper. Jun 2000.
9. N. Brownlee, C. Mills, G. Ruth. Traffic Flow Measurement: Architecture. RFC 2063. January 1997.
10. S. Waldbusser. Remote Network Monitoring Management Information Base. RFC 1757. February 1995.
11. V. Paxson, J. Mahdavi, A. Adams, M. Mathis. An Architecture for Large-Scale Internet Measurement. IEEE Communications, Vol. 36, No.8, pp 48-54. August 1998.
12. ISO. Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 MBit/s. International Standard IS-11172. October 1992.
13. C. Veciana et al. Server Location and Verification Tool for Backbone Access Points. 13th ITC Specialist Seminar IP Traffic Measurement, Modelling and Management. Monterey, September 2000.
14. C. Veciana, A. Cabellos-Aparicio, J. Domingo-Pascual, J. Solé-Pareta. Verifying IP Meters from Sampled Measurements. IFIP 14th International Conference on Testing Communicating Systems. Berlin, March 2002.