

A Framework for ATM Testing

J.B. Micheel, C. Tittel, J. Tiemann

ATM Test Center and BATES Projects

GMD FOKUS

Hardenbergplatz 2, 10623 Berlin, Germany

Phone: +49-30-25499-268, Fax: +49-30-25499-202

Email: {micheel,tittel,tiemann}@fokus.gmd.de

Abstract

ATM technology is of very complex nature. Different views exist on how to get implementations of this network technology robust and interoperable. This paper describes a unified approach to ATM testing (i.e. interoperability, conformance, performance, debugging, monitoring, etc.), which reflects in a modular set of building blocks that can be combined for a particular test.

Keywords

ATM testing, monitoring, troubleshooting

1 HISTORY AND MOTIVATION

GMD FOKUS' first ATM testbed dates back to June 1992 when 2 SPARCstation 2 were connected with SBA-100's from FORE systems. Some tests were made to evaluate the state of ATM development in practice. The first switch, a FORE ASX-100, arrived in December 1992 and was one of the first workgroup switches being sold in Europe.

Since then the network has grown permanently (Micheel, 1997) and now consists of more than a dozen switches from various vendors, a couple of interworking units and around 50 workstations directly connected to ATM.

The FOKUS inhouse ATM network immediately joined BALI (Berlin ATM LAN Initiative) , one of the first metropolitan area ATM networks. It is based on dark fibers of the former BERKOM high-speed testbed with major sites at DeTeBerkom (Deutsche Telekom Group), GMD FOKUS and Technical University of Berlin.

A network of such size (with DeTeBerkom there are more than 100 networked entities) demands for equipment to test and monitor individual devices and the network in its entirety, the more, since a number of extremely challenging demonstrations are run across this net. As an example, BALI now has transcontinental links into north america (MAY testbed, 155 MBit/sec trunk speed) and south-east asia (Singapore, 2 MBit/sec). GMD FOKUS is equipped with an HP 75000 BSTS, a Tekelec ChameleonOpen and a number of smaller testers. An ATM tester project was set out - resulting in the BATES (Berkom ATM Technology Evaluation System) platform. The ATC (ATM Test Center) project at DeTeBerkom and FOKUS evaluated a large number of ATM testers currently being available commercially (Kühn, 1996). The result of these efforts remains unsatisfying:

- ATM testers are usually grouped into protocol testers / decoders and “high-end” testers dedicated to performance measurements at the ATM layer. There is no single tester providing both in a satisfactory way.
- ATM testers come with a fixed set of software. Test software for a certain ATM feature usually becomes available with a delay of one year after first implementations showing up.
- ATM testers provide a poor environment to describe test scenarios. Some provide a C programming library interface, which is clumsy and inherently tester-dependend. There also is a very limited number of users who would go into programming a tester just to customize it for a certain test case.
- ATM testers are generally expensive devices. While organizations like FOKUS are able to spend money for one or two dedicated systems, these testers are far from being the workhorse for any kind of testing for a large number of researchers doing their job in parallel.
- ATM testers do not allow for testing or monitoring at different layers (e.g. ATM, AAL, TCP/IP, ...) in parallel. So when debugging a TCP session by looking at the data packets you’ll not be able to see cell lossage due to traffic contract violation causing troughput degradation, which you are, in fact, hunting for.
- High-end testers are usually non-portable, at least to carry them with you when being at a demonstration or outhouse testbed, where a tester is of much help “in the case something breaks”.

In the fire of these constraints the idea of workstation-based tools was born. A simple ATM adaptor provides for access to the net or device under test. All else is done in software. An example is ATMCell - a tool based on FORE's SBA-100 hardware, which allows for cell-level protocol en- and decoding and microsecond-granularity cell delay measurements. With this tool switching delays of local ATM switches could be measured for less than US\$1000. Higher layer decoding and debugging in multi-vendor environments highlighted bugs in protocol implementations (EC LACE project). However, the approach was rejected for both technical and esthetic reasons. Implementation, user interface and test model would not run transparent on other hardware providing similiar services (i.e. HP BSTS). Then, the power of the user interface would not scale with our demands for configurability. The idea for CellFile - a stream interface to access ATM-level test hardware and to combine ATM-level tools - was born.

With ATM devices gaining a certain degree of useability, more and more functionality became available at AAL and higher layers. Standard ATM adaptor cards implementing AAL3/4 and 5 in firm- or hardware provided a basis to design a number of important tools: sigdump (Tittel, 1995) (Tittel, Micheel, 1996), lanedump (Mochalski, Barth, 1996) (Mochalski 1996) and mpoadump. Again, to keep tools free from handling ATM adaptor and API specifics, separation of functionality became a necessity. PacketFile - a stream interface for variable-length messages - would serve our needs.

Running an ATM network at this time is not just challanging due to implementations still getting adult, but because there are no tools to identify the source of trouble, thus forcing a network administrator to "fix" problems heuristically by guessing and rebooting switches and workstations. A troubleshooting system would be very beneficial. FOKUS has started to instrument it's own network in order to overcome such problems and within the ATM test framework to develop the necessary set of tools for private network troubleshooting. A central role play decoding tools (sigdump, lanedump, etc.), but these tools require as input the link state (channel activity plus higher layer protocol information). We consider to develop a number of state agents holding this information for later use by decoders. These agents provide their information via SNMP. An umbrella agent (called CIA) gives a unified view onto these standalone agents. sigagent (Sträuber, 1996) is a first implementation that holds link state based on information of the UNI signalling flow. (Micheel, Tittel, 1996) provides more information about the approach.

The current framework thus stems its power from a few paradigms:

- Hardware is used to convert a CPU-processable representation of test data (read: CellFile, PacketFile) from and to the wire.

- Any semantics describing a certain test scenario are assembled out of atomic modules providing basic features. Examples are filtering or statistics modules. These modules are currently implemented as CPU-instances (processes or streams modules), but the abstraction allows for later implementation in firm- or hardware. In any case (for now), the CPU should be kept in the data stream to provide a maximum of flexibility when designing and assembling a test scenario.
- Writing new modules shall be lots of fun. The programming environment must be simple and easy to understand, forcing human cycles to be spend on solving the problem instead of fighting against the framework.
- Keeping the CPU in the data stream implies destroying unnecessary bandwidth as early as possible and generating useful information instead. An example would be an ATM layer statistics module taking any cell on the wire and producing once a second a summary about activities.
- At the very end of the chain of modules there is always a human sitting in front of the tester who is most often interested to get a very simple answer. The framework shall give this answer by allowing to formulate questions.

2 CONCEPTS

The core idea of the framework is to identify a minimal set of building blocks that satisfy complex testing. We define ATM testing as the superset of all kinds of activities ensuring proper operation of an ATM-based network or system.

Assembling a test case out of modular building blocks requires defining an interface between those modules. Keeping the number of different interfaces low ensures a maximum of flexibility when combining tools. We have, at current, decided to stick to 3 simple protocols: **CellFile**¹, **PacketFile**² and **SNMP**.

The CellFile (CF) interface is based on the definition of an ATM cell (ATM Forum UNI) (ATM Forum B-ICI), extended with a timestamp and layed out (padded) in a way that allows for easy processing by 32bit CPU's (see also Cooper et.al., 1991). The Cell File is the core protocol of the framework, since anything in an ATM network happens as ATM cells being transmitted over bidirectional links. Higher layer information may be recovered from or encapsulated in these cells.

¹. The term "File" is a misnomer, it really is a stream interface format. The name remains for historical reasons.

². see 1.)

Timestamping captured cells is a way of data compression (elimination of idle or unassigned cells) and a representation of the traffic profile - essential feature for a network technology with complex descriptions of application behaviour and Quality-of-Service requirements.

A special **CellHeader** (CH) format allows for analysis and testing at higher data rates by stripping payload information, thus reducing the bandwidth resp. storage requirements to 20% compared to the standard CellFile. This format is very convenient for ATM layer performance measurements, where payload contents is irrelevant. It allows to build inexpensive hardware for capturing and transmission of traffic profiles at today's standard link rates (155 MBit/sec) (Micheel et. al., 1996). (TANYA's I-want-it-all design addresses the problems with other ATM testers described in the first section of this paper.) The header fields and the toolset for CH streams are identical to CF, so everything said about CF includes CH in the rest of this paper.

The second major interface format is PacketFile (PF), a sequence of variable-length messages marked with a timestamp. We found the format, in which data packets are stored by tcpdump (McCanne, Jacobsen, 1993), to nicely fit our requirements. Tcpdump's traces may contain header-only information (a predefined number of bytes starting at byte #0), again reducing the number of bytes to be processed by PF tools and keeping the storage size of captured profiles low.

Both cell and packet file formats hold a timestamp per unit (cell or packet) for different purposes. Very important is inter-unit spacing to retain the traffic profile of the original stream in the captured representation or to control the behaviour of a stream to be sent out. For this purpose relative timestamps would be best. A second goal is to allow for a later merge or comparison with other streams: both directions of a communications link might be monitored to look at request-response performance or a 2-point measurement could allow for cell transfer delay and delay variation analysis. For this kind of tests absolute timestamps are a better choice. Instead of deciding for a single method we currently support both and use whatever coding is best for the scenario in question. The downside of this is that modules have to support both formats, which makes implementation and usage harder. The role and usage of timestamps is context-dependent. There are also performance and flexibility tradeoffs when choosing a certain format: the higher the accuracy the more bits you need to describe larger intervals (thin traffic). More overhead bits for timestamping are a performance degradation for the toolset. 32 bits are very easily processed by today's processors, more bits result in a significant reduction. At current, the final choice of the timestamp format is still a research topic.

A SNMP (Case et. al., 1991) MIB is the database of choice to hold information gathered by monitoring an instrumented network. First, SNMP is a fairly simple and

robust protocol. Second, because of ASN.1 BER encoding, it is able to transport almost any kind of data. Agents to be written capture and display data of various kinds: ATM level activity information, measured traffic flow parameters, ATM signalling and routing state, human input and other. Third, data can be conveniently substructured in MIB's. The umbrella SNMP agent (CIA) then allows to combine different agents to give a link-complete view, or to assemble a network view by collecting the information from different monitoring points of the same agent (e.g. signalling). CIA could also implement more sophisticated functionality, such as filtering or scoping, which are not present as SNMP protocol features.

3 EXAMPLES

In this section we'd like to illustrate the broad range of configurability we intend to achieve with modules within our framework. any of the pictures TANYA might be replaced by a similiar hardware for ATM-level access, the combination of TANYA with AAL5 segmentation and reassembly with a standard of-the-shelf workstation ATM adaptor. In fact, since TANYA is non-existing hardware (at the time of this writing), the entire toolset as currently exploited is based on a number of different platforms. Other access points for higher-level analysis (AAL, IP, LANE, ...) might be standard debugging hooks in network driver implementations, such as Sun's NIT, BPF or DLPI.

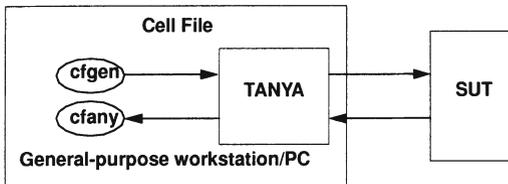


Figure 1 Performance measurement scenario

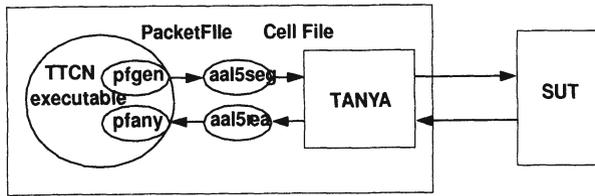


Figure 2 Conformance test scenario

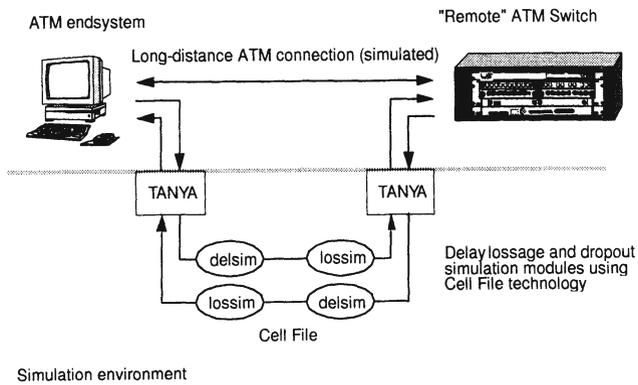


Figure 3 On-the-fly simulation of network behaviour

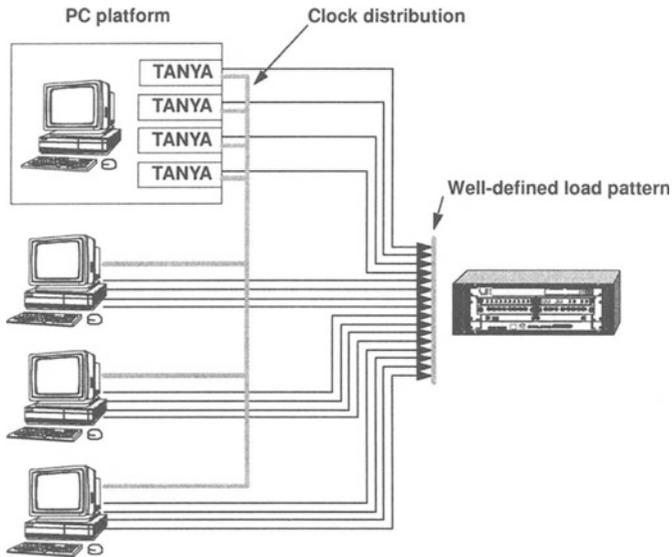


Figure 4 Large-scale distributed testing

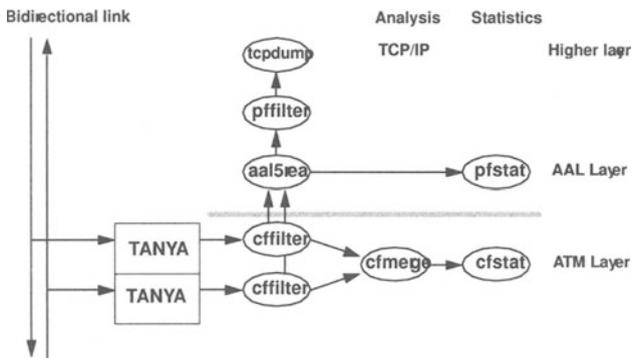


Figure 5 Network monitoring and troubleshooting

4 DEVELOPMENT STATE

At current, a number of protocol decoding tools have been implemented (UNI signalling, ILMI, LAN Emulation). The packet file toolset fitting with these decoders implements ASCII conversion, filtering based on the Berkeley Packet Filter, merging of PF streams. CF tools have been written for a number of experimental purposes (generators, analyzers, etc.), but are still missing the strong split of functionality and cleancut. A first agent is available - sigagent. The instrumentation of the FOKUS and BALI local and MAN ATM networks is in progress, a photonic switch has been obtained to join measurement points independent of the various physical layer speeds to be run on multimode and singlemode fiber. A heterogenous multi-vendor testbed is envisioned for the near future.

The next steps to be taken are:

- to enrich the toolset trough decoders for P-NNI, MPOA, B-ISUP (in this order)
- to develop cheap and powerful access hardware - TANYA
- to evaluate scaleability for the CPU-based approach of ATM testing
- to develop a clean set of CF tools
- to evaluate the feasibility of the framework for permanent ATM monitoring, accounting and billing.

There are plans to publicly release parts of the software (binaries and/or sources) as **FAST** - FOKUS ATM ToolSet - but this hasn't been our primary goal in the past, so the release is delayed.

5 ACKNOWLEDGEMENTS

We owe special thanks to our colleague Falk M. Schulz, now with DeTeBerkom, for co-designing an early draft of the framework and implementing first versions of the CF tools as part of his ATMCell test tool. Thanks to all other colleagues of the ATC and BATES projects for enriching the number of miscellaneous CF and PF tools.

6 REFERENCES

- ATM Forum, Technical Committee, Broadband Inter Carrier Interface (B-ICI) Specification Version 1.0
- ATM Forum, Technical Committee, User-Network Interface (UNI) Specification Version 3.1, September 1994.

- Case J.D., Fedor, M.S., Schoffstall, M.L. and Davin J.R. (1990) A Simple Network Management Protocol, *Internet Engineering Task Force (IETF) Request For Comments (RFC) 1157*, May 1990.
- Cooper, E., Menzilcioglu, O., Sansom, R. and Bitz F. (1991) Host Interface Design for ATM LAN's, in *Proc. of the 16th Conference on Local Computer Networks*, IEEE Computer Society, p. 247-258.
- Kühn, K. (1996) Testwerkzeug für ATM Signalisierung, Diplomarbeit an der Technischen Universität Dresden, DeTeBerkom und GMD FOKUS.
- McCanne, S. and Jacobsen, V. (1993) The BSD Packet Filter: A New Architecture for User-Level Packet Capture, in *Proc. of the 1993 Winter USENIX Conference*, January 25-29, 1993, San Diego, CA.
- Micheel, J., Schmidt, R. and Tiemann J. (1996) TANYA - Yet ANother ATM Tester or I-want-it-all, GMD FOKUS research paper, August 5th, 1996.
- Micheel, J. and Tittel, C. (1996) ATMSnoop - Watching The Cells Go By, in *Proc. of the 8th IEEE Workshop on Local and Metropolitan Area Networks*, Potsdam, Germany, 25.-28. August 1996.
- Micheel, J. (1997) The FOKUS ATM and BALI Networks. Informational material, GMD FOKUS.
- Mochalski, K. and Barth, M. (1996) LAN Emulation. ATC Technical Report #2, GMD FOKUS.
- Mochalski, K. (1996) Entwicklung und Implementierung eines LAN Emulation Testtools zur Evaluierung von ATM Komponenten, Diplomarbeit an der Universität Leipzig und GMD FOKUS.
- Sträuber, P. (1996) Entwicklung und Implementierung eines Signalisierungs-SNMP-Agenten, Diplomarbeit an der Universität Leipzig und GMD FOKUS.
- Tittel, C. (1995) Entwicklung und Implementierung eines Signalisierungs-Testtools zur Evaluierung von ATM Komponenten, Diplomarbeit an der Universität Leipzig und GMD FOKUS.
- Tittel, C., and Micheel, J. (1996) UNI Signalling. ATC Technical Report #1, GMD FOKUS.

7 BIOGRAPHY

Jörg Micheel received his Dipl.-Ing. at the Novosibirsk Institute for Electrical Engineering, Russia. In 1992 he started working at GMD FOKUS as a systems programmer for the BERMATE high-speed ATM adaptor. Since 1995 he leads the ATM Test Center joint project of DeTeBerkom and GMD FOKUS. He has been administering the FOKUS ATM and BALI networks since the early beginnings.

Currently, his research focuses on smart solutions for creating a powerful and robust networking service.

Dipl.-Inform. Carsten Tittel joined FOKUS as a student at Leipzig University. Since 1996 he's a full-time researcher involved in the European Community ACTS INSIGNIA project, where he works on functional testing of protocol implementations.

Dipl.-Ing. Jens Tiemann brood over BERMATE as a hard-worker. Since 1995 he led the BATES project and pursued the integration of CellFile technology. In his spare time he spend cycles on making things more efficient and co-designed TANYA, which is now his primary project.