

# Methodology for Implementing Scalable Test Configurations in ATM Switches

Arjan Durrezi, Raj Jain, Gojko Babic  
Department of Computer and Information Science,  
The Ohio State University,  
2015 Neil Ave, Columbus, OH 43210-1277 USA  
Phone: 614-688-5610, Fax: 614-292-2911,  
Email: {durrezi, jain, babic}@cis.ohio-state.edu

Bruce Northcote  
Fujitsu Network Communications  
289 Great Rd, Acton, MA 01720 USA  
Phone: 978-266-4594,  
Fax: 978-266-2300,  
Email: bsn@nexen.com

**Abstract**—In performance testing of ATM switches and network of switches a variety of connection configurations is needed. In most of the cases, these configurations require one traffic generator and/or analyzer for each switch port. Since this equipment is rather expensive, it is desirable to define scalable configurations that can be used with a limited number of generators. In this paper we present a methodology for the implementation of scalable connection configurations. The methodology is simple and offers a general solution to generate scalable connection configurations. Several examples of scalable configurations illustrate the methodology. The application of this methodology helps users to repeat easily performance tests under the same traffic load conditions.

## I. INTRODUCTION

In order to compare different ATM equipment, it is important to use a set of standard configurations and workloads [1, 2, 3, 4, 7]. One problem that we run into when trying to compare ATM switches is that these switches can have a large number of ports. To test these switches at high loads, a traffic generator and analyzer may be required for each port. These generators and analyzers are expensive. It is therefore, desirable to find ways to minimize the number of such generators and analyzers. In this paper, we discuss such ways and issues related to them [2, 5, 6].

The key problem with scalable configurations is that there are many ways to set up the connections and measurement results could vary with the setup. For example, in the case of unicast, it may not be possible to overload a port with only one generator. Using two generators in scalable configurations may exhibit different behavior, such as overloading, that may not show up with one generator. The application of this methodology helps users to create easily the same traffic load for independent performance tests and make the latest more comparable.

Performance testing requires two kinds of virtual channel connections (VCCs): foreground VCCs (traffic that is measured) and background VCCs (traffic that simply interferes with the foreground traffic) [3, 4]. The methodology for generating configurations of both types of VCCs is covered by this paper.

The VCCs are formed by setting up connections between ports of the switch. The connections are internal through the switch fabric and external through some transmission medium or wires (which could be cables, fibers, or even wireless links), depending on the port technology. In this paper, internal connections are shown by thin lines and external connections by thick lines. An arrow indicates the direction of the connection. If a connection is bi-directional, which is often the case, arrows are not used. It should be

noted that whenever external connections are used in the test configurations, only permanent VCCs can be established.

Throughout the paper it is assumed, for improved readability, that the implementation under test (IUT) consists of a single ATM switch, although the methodologies presented here apply equally to test cases in which the IUT is a network of ATM switches or, alternatively, a subset of modules of a single switch. The notation  $P_{ij}$  is used to refer to the  $j^{\text{th}}$  port of the  $i^{\text{th}}$  module of the IUT, and  $(P_{ab}, P_{cd})$  indicates that a connection (either internal or external to the IUT) exists between  $P_{ab}$  and  $P_{cd}$ .

Two generic categories of scalable configuration are presented in this paper, namely:

1) *Parallel Traffic Replication* configurations, discussed in Section II, which employ the point-to-multipoint capability of a switching system (other than the IUT) to artificially generate more traffic than is possible with a limited number of traffic generators, and

2) *Serial Traffic Replication* configurations discussed in Section III, which employ external connections to serially relay traffic egressing from the IUT back in to the IUT, thereby emulating additional traffic generators.

## II. PARALLEL TRAFFIC REPLICATION

The point-to-multipoint capability of a switching system is intended primarily for broadcast communication services, but which also lends itself well to the task of generating the traffic inputs to the IUT that are required for the test configurations. Identical ATM cells are broadcast in parallel from multiple output ports – hence ‘parallel traffic replication’. Given a single traffic generator, and a switching system (other than the IUT) with a point-to-multipoint capability (a multicast switch), an IUT may receive traffic on as many input ports as the multicast switch has available output ports. This form of scalable configuration is depicted in Fig. 1, where G is the single traffic generator. Internal to the IUT, any of the configurations specified in [1] may be used.

The weakness of this solution is that the measurement may be influenced by the performance of the multicast switch. Also if the multicast switch does not support multipoint-to-point connections, then this form of parallel traffic replication cannot support bi-directional connection configurations. In such cases and when the influence of the multicast switch performance in the measurement is not desirable, it may be necessary to use serial-traffic replication, as described in the next section.

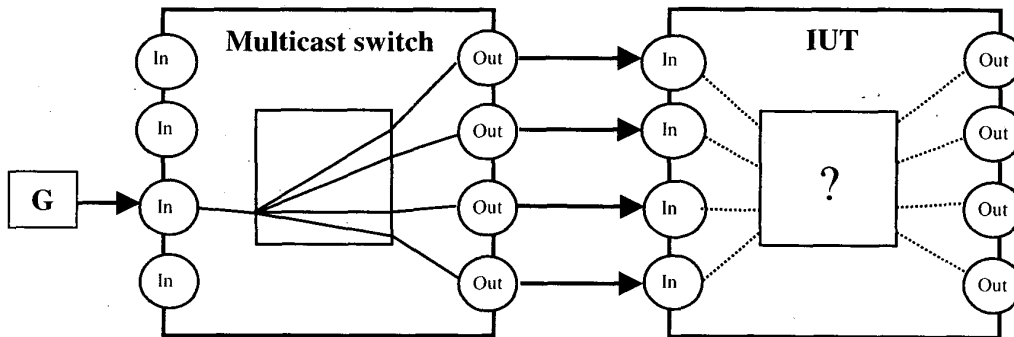


Fig. 1. A parallel traffic replication scalable configuration

### III. SERIAL TRAFFIC REPLICATION

An example test configuration employing serial traffic replication is provided in Fig. 2a, which shows a 4-port switch with ports labeled as  $P_{11}$ ,  $P_{12}$ ,  $P_{21}$  and  $P_{22}$ . Of these, ports  $P_{21}$  and  $P_{12}$  are connected by a wire  $W_1$ , while port  $P_{22}$  has a "loopback" wire LB that connects the output of the port to its input. Internally, a PVC has been set up to connect ports  $P_{11}$  with  $P_{21}$  and  $P_{12}$  with  $P_{22}$ . Note that all external connections (wires) and internal connections (PVCs) in this case are bi-directional, except the loopback. During testing with this configuration, cells first enter the switch at  $P_{11}$  and are passed through every port of the switch in series, before looping back at  $P_{22}$  and following the reverse path back to exit the switch for the last time at  $P_{11}$ .

The methodology presented here has two phases. During the first phase the switch ports are connected externally by numbered wires, as in Section III.A. The second phase consists of setting up PVCs, i.e. internal connections between ports, as explained in Section III.B.

The sequence of concatenated connections (internal and external) is called a VCC Chain. For example, the VCC shown in Fig. 2b. is formed by setting up a VCC chain starting from  $P_{11}$ -In. ATM cells flow internally from  $P_{11}$ -In to  $P_{21}$ -Out, externally via wire  $W_1$  to  $P_{12}$ -In, internally to  $P_{22}$ -Out, externally via wire LB to  $P_{22}$ -In, internally to  $P_{12}$ -Out, externally through wire  $W_1$  to  $P_{21}$ -In, finally exiting at  $P_{11}$ -Out. This VCC chain can be indicated as:

Generator- $P_{11}$ - $P_{21}$ - $P_{12}$ - $P_{22}$ - $P_{22}$ - $P_{12}$ - $P_{21}$ - $P_{11}$ -Analyzer.

Of these connections,  $P_{22}$ - $P_{22}$  is a unidirectional external connection (loopback, denoted as LB) and  $P_{12}$ - $P_{21}$  is a bi-directional external connection (wire, denoted as W). All of the internal connections (VCCs) are bi-directional. The sequence of external connections used in this VCC chain is: Generator- $W_1$ -LB- $W_1$ -Analyzer.

Both the above notations are symmetric in the sense that the second half of the chain is a mirror image of the first half. For example,  $W_1$ -LB is the mirror image of LB- $W_1$ .

Another possible configuration for this "n-to-n single generator configuration" is  $P_{11}$ - $P_{12}$ - $P_{21}$ - $P_{22}$ - $P_{22}$ - $P_{21}$ - $P_{12}$ - $P_{11}$ . The various VCC chains may be distinguished by

the order of, and the direction through which, each wire is initially traversed by the generated traffic.

The four-port switch shown in Fig. 2a consists of two modules with two ports each. The measured performance for a given test configuration may depend upon whether the internal connections of the VCC chain are inter-module, intra-module, or a mixture of both. The methodology presented in this paper ensures that it is possible for exclusively inter-module, or intra-module traffic to be carried.

#### A. Implementation of External Connections

The methodology for implementing the external connections consists of the following three steps:

- 1) Identify the modules to be included in the IUT and label the ports (using  $P_{ij}$  format).
- 2) Connect the generators and analyzers to the appropriate ports.
- 3) Establish and number external connections (wires) to use all the remaining ports of the IUT.

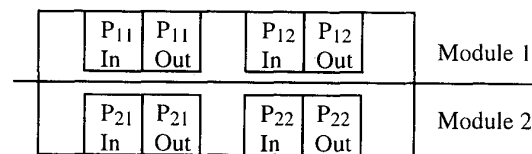


Fig. 2a Representation of physical switch

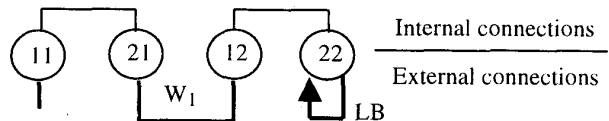


Fig. 2b Logical representation of the switch and its port connections

These steps are now explained.

#### Step 1. Identifying the modules to be included in the IUT

In order to ensure that it is possible for the configuration to support exclusively inter-module and/or intra-module internal connections, the IUT should consist of pairs of similar

modules. If this constraint is not satisfied, the VCCs that are established may be a mixture of inter-module and intra-module connections. It is not necessary that the modules/ports be labeled, although we use the  $P_{ij}$  format here to assist in the description of the methodology.

Consider a switch with several modules of different port types. The ports could be different in speed and/or connector type. Each module may have a different number of ports. For example, a switch may have two modules of eight and six 155-Mbps single-mode fiber ports, respectively, another module with eight 155-Mbps UTP ports and a fourth module with six 25-Mbps UTP ports. Fig. 3 shows an example IUT where the modules are grouped by type. The first group consists of two 25-Mbps UTP modules, the second group consists of two 155-Mbps single fiber modules. External connections may only be established between ports that are co-located within the same group (hence the constraint that modules come in pairs for inter-module connectivity).

*Step 2. Connect the generators and analyzers to the appropriate ports.*

A port must be reserved for each generator/analyzer that is to be used in the test. These reserved ports cannot be used in the next step that establishes external connections. The methodology presented here allows any given number,  $r < n$ , of generators. Some additional constraints on the number of ports in the IUT are explained in the next step.

*Step 3. Establish and number external connections.*

After selecting the ports that are reserved for connection to generators/analyzers, the remaining ports have to be externally connected with numbered "wires".

The following guidelines should be followed when establishing the wires:

1) Partition the remaining (non-reserved) ports into subgroups, whilst ensuring that there is an odd number of ports in each subgroup.

2) Assign each generator/analyzer to a subgroup. To establish a "straight" connection configuration it is necessary that there exist a 1-to-1 correspondence between subgroups and generators with an even partitioning of ports into subgroups. For other test configurations, such as "full cross" or "partial cross", more than one generator may be assigned to a subgroup, and the ports need not be evenly distributed between subgroups.

3) With  $m$  non-reserved ports in a particular subgroup ( $m$  being odd), the first  $m-1$  ports are pair-wise connected by wires, numbered consecutively from 1 across all subgroups i.e. the 1<sup>st</sup> subgroup's wires would be  $W_1$  to  $W_x$ , where  $x = (m-1)/2$ , and the 2<sup>nd</sup> subgroup would have wires numbered from  $W_{x+1}$ . It is preferable that wires be established between ports on different modules to ensure that exclusively inter-module or intra-module traffic may be carried. Wires cannot connect ports of different types.

4) The last ( $m^{\text{th}}$ ) port of each subgroup is occupied by a loopback, labeled as  $LB_g$  for the  $g^{\text{th}}$  subgroup that will redirect all traffic egressing from the port back to the ingress of the port.

The following example illustrates the methodology for establishing the wires:

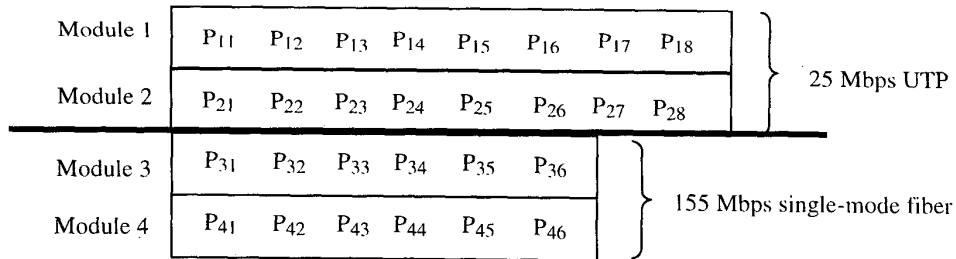


Fig. 3. Example partitioning of modules into groups.

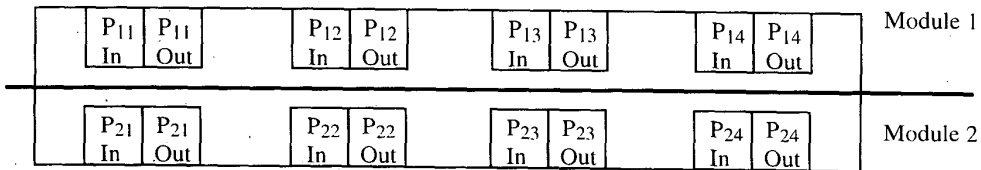


Fig. 4.a. Port labeling of the example switch (2 modules of 4 ports each).

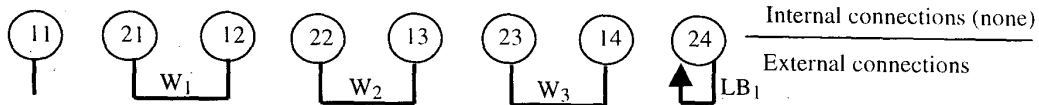


Fig. 4.b. Wiring configuration for the example switch.

Consider the n-to-n straight configuration required for the foreground traffic in throughput measurement. Suppose the switch has two modules with four ports each of the same speed and connection type. The ports are labeled as shown in Figure 4.a.  $P_{11}$  is arbitrarily selected to be connected to the single generator/analyzer that is to be used. There will only be one subgroup, because there is only one generator/analyzer and all the ports are of the same speed and connection type. The non-reserved ports in the subgroup are  $\{P_{12}, P_{13}, P_{14}, P_{21}, P_{22}, P_{23}, P_{24}\}$ . The wires  $W_1=(P_{12}, P_{21})$ ,  $W_2=(P_{13}, P_{22})$  and  $W_3=(P_{14}, P_{23})$  are obtained by alternatively selecting ports of the first and second module from the non-reserved list of ports. Finally, the loopback is placed at the remaining port, namely  $P_{24}$ . Figure 4.b shows the resulting wiring configuration. Note that it will allow for either exclusively inter-module, or exclusively intra-module traffic (this will be shown in the next section).

### B. Implementation of Internal Connections

The methodology presented in this paper can ensure that exclusively inter-module or intra-modules traffic is carried, depending on the implementation of the internal connections. For example, Fig. 4.a and 4.b show configurations for intra-module and inter-module configurations, respectively, although both are based on the wiring indicated in Figure 4.b.

As indicated earlier, VCC chains may be distinguished by the order of, and the direction through which, each wire is initially traversed by the generated traffic. We only consider VCC chains that use every non-reserved port within a single subgroup. Using the same terminology as in Section III.A, assume that we have established  $x$  wires and a single loopback for a subgroup with  $m$  non-reserved ports, and  $r$  generators. For a given traffic flow (intra-module or inter-module), a VCC chain that passes through each non-reserved port can be expressed as:

Generator – ‘ $x$  wires in series’ – LB – ‘ $x$  wires in reverse series’ – Analyzer.

Stated more succinctly, the VCC chain only depends on the generator/analyzer used and the ordering of the ‘ $x$  wires in series’. The example in Fig. 5 uses the ordering  $W_1W_2W_3$ . For a given traffic generator and wire ordering there exists a unique set of VCCs that provides an intra-module VCC chain. Similarly for an inter-module VCC chain.

Without loss of generality, assume that the wire ordering is  $W_1W_2 \dots W_x$ . Let  $P_g$  be the port connected to the generator for this VCC chain, and  $P_{LB}$  be the port that has the loopback. To complete the construction of the VCC chain, all that remains is to establish the VCCs, as follows:

1. Set  $P_{in} = P_g$  and  $i = 1$ .
2. Set  $P_{out}$  = the port from  $W_i$  that is/isn't located on the same module as  $P_{in}$  for an intra/inter-module VCC chain, respectively.
3. Establish the bi-directional internal connection  $V_i = (P_{in}, P_{out})$ .
4. Set  $P_{in}$  to be the other port from  $W_i$  (not  $P_{out}$ ).
5. If ( $i < x$ ), set  $i = i+1$  and return to step 2, otherwise continue.

6. Establish the bi-directional internal connection  $V_{x+1} = (P_{in}, P_{LB})$ .

The VCC chain is now fully specified as:

Generator- $V_1$ - $W_1$ - $V_2$ - $W_2$ -...- $W_x$ - $V_{x+1}$ -LB- $V_{x+1}$ - $W_x$ -...- $W_2$ - $V_2$ - $W_1$ - $V_1$ -Analyzer.

Clearly, if any other wire ordering or any other generator (out of the  $r$  generators assigned to the subgroup) were used, a different VCC chain would have resulted.

Referring again to the example provided in Fig. 4.a and 4.b, only one VCC chain is established for each traffic flow. In the case of an n-to-m partial cross configuration with a single generator, several VCC chains are required; each based on a different ordering of the same subgroup of wires. The six (3!) possible permutations for the example are:  $W_1W_2W_3$ ,  $W_2W_3W_1$ ,  $W_3W_1W_2$ ,  $W_3W_2W_1$ ,  $W_2W_1W_3$  and  $W_1W_3W_2$ , each of which would result in a distinct VCC chain for the single generator.

The examples given in Section IV illustrate the ways in which multiple VCC chains and/or multiple generators may be used in serial traffic replication scalable configurations.

### C. Background Traffic

To establish a configuration that incorporates background test traffic, the background traffic may:

- 1) ingress and egress on ports distinct from those used by the foreground traffic,
- 2) ingress with, but egress at a port distinct from, the foreground traffic, or
- 3) ingress at a port distinct from, but egress with, the foreground traffic.

To implement a configuration with background traffic, one or more ports need to be reserved for foreground traffic. Therefore, to implement:

- Case 1 (distinct ingress and egress) – reserve 2 ports for each foreground traffic stream.
- Cases 2 & 3 (shared ingress or egress) – reserve 1 port per foreground traffic stream.

## IV. EXAMPLE OF SERIAL TRAFFIC REPLICATION

In this section, several examples of scalable connection configurations are provided. In all of the examples, a switch with two 4-port modules is used (as in Fig. 4.a), where all ports are of the same speed and connector type.

### A. n-to-n Straight (Single Generator)

This configuration may be used for throughput as well as latency measurements. Suitable test configurations that employ serial traffic replication can be obtained as follows:

#### 1) Foreground Traffic (Throughput measurements)

For these tests, only a single chain starting from a single generator is needed. Fig. 4.a and 4.b, provide adequate examples of this type of connection configuration.

#### 2) Background Traffic (Latency measurements)

In this example, the background traffic does not use the input/output ports of the foreground traffic. As shown in Fig. 6, generator/analyzer  $G_1$  is used for background traffic while generator/analyzer  $G_2$  is used for foreground traffic.

The background traffic passes through each port not assigned to the foreground traffic. Inter-module external connections are established by following the guidelines described in Section III.A, resulting in wires  $W_1$ ,  $W_2$ , and  $LB_1$ . The VCC chain of the background traffic corresponds to the wire ordering  $W_1W_2$ . So the VCC chain is given by:

$$G_1-V_1-W_1-V_2-W_2-V_3-LB_1-V_3-W_2-V_2-W_1-V_1-G_1,$$

where, by establishing exclusively inter-module VCCs:  $V_1 = (P_{12}, P_{22})$ ,  $V_2 = (P_{13}, P_{23})$ ,  $V_3 = (P_{14}, P_{24})$ .

### B. n-to-n Straight (r Generator) Foreground Traffic

To realize an n-to-n straight configuration with r generators, we need to design r VCC chains. Of the n ports, r ports are used as the source/destination of these chains. The remaining ports are connected among themselves and their wires are divided in subgroups among the generators ensuring an odd number of ports in each subgroup.

As an example, consider the 8-port switch again, with  $r = 3$  generators. In dividing the available ports in three subgroups the goal is to have an odd number of ports in each subgroup. The partitioning of ports into subgroups results in one subgroup with three ports and two with one port each. The three port subgroup is used by the first generator. Note that the wiring configurations of the other two subgroups consist only of a single loopback. Fig. 7 illustrates the implementation of the VCC chains for this case. First we select the source and destination ports. Let,  $P_{11}$  be the input/output port for the first chain,  $P_{21}$  be the input/output port for the second chain, and  $P_{12}$  be the input/output port for the third chain. Using only intra-module VCCs the following VCC chains are obtained:

$$G_1-V_1-W_1-LB_1-W_1-V_1-G_1, \\ G_2-V_2-LB_2-V_2-G_2, \text{ and} \\ G_3-V_3-LB_3-V_3-G_3, \text{ where} \\ V_1 = (P_{11}, P_{13}), V_2 = (P_{21}, P_{24}) \text{ and } V_3 = (P_{12}, P_{14}).$$

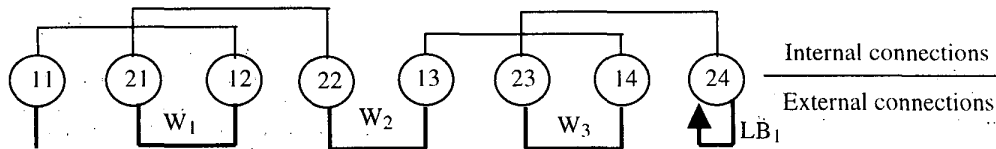


Fig. 5.a. An 8-to-8 straight configuration for the example switch using intra-module VCCs.

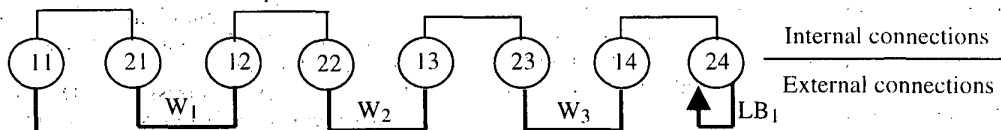


Fig. 5.b. An 8-to-8 straight configuration for the example switch using inter-module VCCs.

### C. n-to-m Partial Cross (r Generator)

#### 1) Foreground Traffic 8-to-2 with two generators ( $m=r=2$ )

This configuration has  $m \cdot r$  VCC chains starting from r generators, where each generator originates m VCC chains. Each chain has a load of  $1/m^{\text{th}}$  of the traffic egressing from the generator. Each intermediate wire has exactly m of these streams flowing through it. There are r subgroups and the switch ports, other than those reserved to be connected to the r generators, are divided among them.

For the configuration example shown in Fig. 8, there are two subgroups (one for each generator) with three ports each.

The following intra-module VCC chains form the required test configuration:

$$G_1-V_1-W_1-V_2-LB_1-V_2-W_1-V_1-G_1, \\ G_2-V_3-W_2-V_4-LB_2-V_4-W_2-V_3-G_2, \\ G_1-V_5-W_1-V_6-LB_1-V_6-W_1-V_5-G_1, \text{ and} \\ G_2-V_7-W_2-V_8-LB_2-V_8-W_2-V_7-G_2, \text{ where} \\ V_1 = (P_{11}, P_{13}), V_2 = (P_{21}, P_{23}), V_3 = (P_{12}, P_{14}), V_4 = (P_{22}, P_{24}), \\ V_5 = (P_{11}, P_{14}), V_6 = (P_{22}, P_{24}), \\ V_7 = (P_{12}, P_{13}), \text{ and } V_8 = (P_{21}, P_{23}).$$

This case is illustrated in Fig. 9. There is only one subgroup composed of wires  $W_1, W_2, W_3$ . The three VCC chains required for this test configuration (each constructed from a distinct wire ordering, with inter-module VCCs) are:

$$\text{generator-V}_1\text{-W}_1\text{-V}_2\text{-W}_2\text{-V}_3\text{-W}_3\text{-V}_4\text{-LB}_1\text{-V}_4\text{-W}_3\text{-V}_3\text{-W}_2\text{-V}_2\text{-} \\ \text{W}_1\text{-V}_1\text{-analyzer,} \\ \text{generator-V}_5\text{-W}_2\text{-V}_6\text{-W}_3\text{-V}_7\text{-W}_1\text{-V}_8\text{-LB}_1\text{-V}_8\text{-W}_1\text{-V}_7\text{-W}_3\text{-V}_6\text{-} \\ \text{W}_2\text{-V}_5\text{-analyzer, and} \\ \text{generator-V}_9\text{-W}_3\text{-V}_{10}\text{-W}_1\text{-V}_{11}\text{-W}_2\text{-V}_{12}\text{-LB}_1\text{-V}_{12}\text{-W}_2\text{-V}_{11}\text{-W}_1\text{-} \\ \text{V}_{10}\text{-W}_3\text{-V}_9\text{-analyzer,}$$

where for tests involving exclusively inter-module traffic, the VCCs are:

$$V_1 = (P_{11}, P_{21}), V_2 = (P_{12}, P_{22}), V_3 = (P_{13}, P_{23}), V_4 = (P_{14}, P_{24}), \\ V_5 = (P_{11}, P_{22}), V_6 = (P_{13}, P_{23}), \\ V_7 = (P_{14}, P_{21}), V_8 = (P_{12}, P_{24}), V_9 = (P_{11}, P_{23}), V_{10} = (P_{14}, \\ P_{21}), V_{11} = (P_{12}, P_{22}), V_{12} = (P_{13}, P_{24}).$$

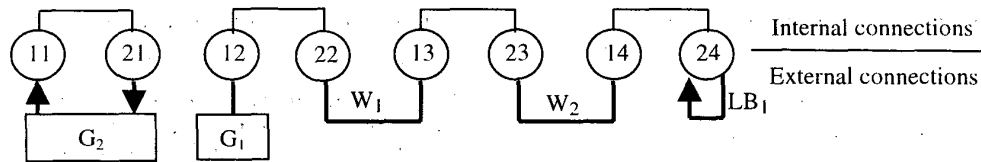


Fig. 6. A 6-to-6 straight configuration with one generator using inter-module VCCs, where the foreground traffic does not share any port with the background traffic.

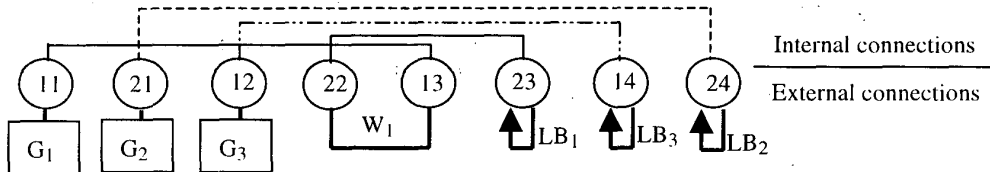


Fig. 7. Implementation of the 8-to-8 straight configuration with 3 generators using intra-module VCCs.

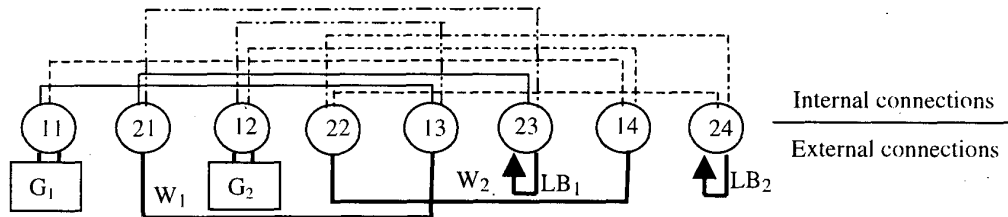


Fig. 8. Implementation of 8-to-2 partial cross configuration with 2 generators for foreground traffic using intra module VCCs.

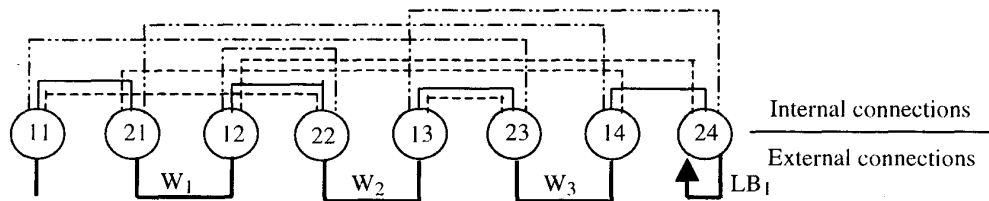


Fig. 9. An 8-to-3 partial cross with one generator using inter-module VCCs

## V. CONCLUSIONS

We have presented a methodology for the implementation of scalable test configurations in ATM switches. The methodology is simple and offers a general solution to the problem of scalable configurations in ATM switches. The application of this methodology helps users to repeat easily performance tests under the same traffic load conditions and therefore independent performance tests will result more comparable.

## REFERENCES

- [1] "ATM Forum/BTD-TEST-TM-PERF.00.11 (Draft)", February 1999.
- [2] A. Durresti, R. Jain, G. Babic, and B. Northcote, "Proposed modified text for methodology for implementing scalable test configurations," *ATM\_Forum/98-0410*, July 1998, <http://www.cis.ohio-state.edu/~jain/atmf/a98-0410.htm>
- [3] A. Durresti, R. Jain, G. Babic, "Experience with ATM Switch Performance Testing," *IEEE International Conference On Networks 1999*, in press, <http://www.cis.ohio-state.edu/~jain/papers/perfexp.htm>
- [4] R. Jain, G. Babic, "Performance testing effort at the ATM Forum: An overview," *IEEE Communication Magazine, Special issue on ATM performance*, August 1997, [http://www.cis.ohio-state.edu/~jain/papers/perf\\_com.htm](http://www.cis.ohio-state.edu/~jain/papers/perf_com.htm)
- [5] A. Durresti, R. Jain, J. Dolske, and G. Babic, "Modifications to appendix B and sections 3.1.7 and 3.2.7 of Performance testing baseline text," *ATM\_Forum/97-1089*, December 1997, <http://www.cis.ohio-state.edu/~jain/atmf/a97-1089.htm>
- [6] A. Durresti, R. Jain, J. Dolske, and G. Babic, "Proposed appendix B of Performance testing baseline text on scalable configurations," *ATM\_Forum/97-835*, September 1997, <http://www.cis.ohio-state.edu/~jain/atmf/a97-835.htm>
- [7] G. Babic, A. Durresti, J. Dolske, and R. Jain "Measurement experiences with the revised MIMO latency definition," *ATM\_Forum/97-859*, September 1997, <http://www.cis.ohio-state.edu/~jain/atmf/a97-859.htm>