97-0835: Proposed Appendix B of Testing Baseline Text on Scalable Configurations.

Arjan Durresi, Raj Jain, Justin Dolske, Gojko Babic

Raj Jain is now at
Washington University in Saint Louis
Jain@cse.wustl.edu
http://www.cse.wustl.edu/~jain/

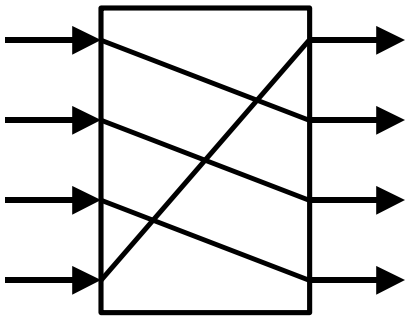
- Need for a standardized way to implement scalable connection configurations
- A simple algorithm to obtain standard scalable connection configurations
- Examples of the algorithm application

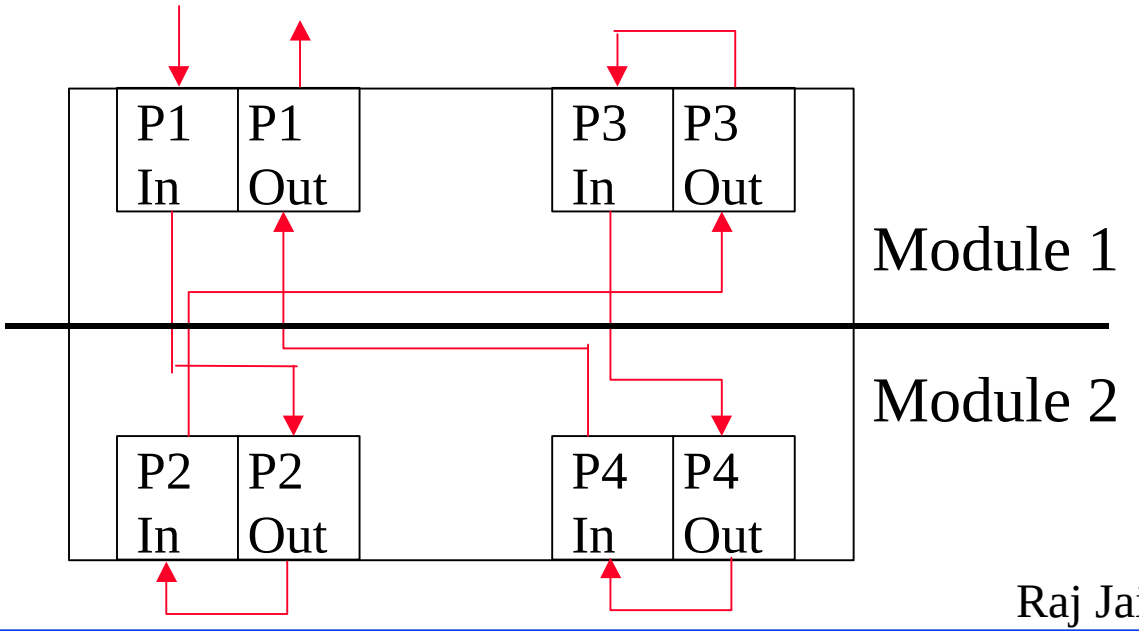
Scalable Configurations

- ATM testing equipment are expensive.
- Scalable Configurations permit to simulate the desired basic configuration using a limited number of generators.
- But there are many ways to set up the scalable connections configurations and the results could vary with the set up.

A Sample 4-to-4 Configuration

- Different implementations could provide different results.
- P1-P2-P3-P4-P1 Four module crossings
- P1-P3-P4-P2-P1 Two module crossings





VCC Chain

- Performance testing requires virtual channel connections (VCCs) to be established through the switch.
- The VCCs are formed by setting up connections between ports of the switch
- The connection order of the ports is referred to here as a **VCC Chain**
- The proposed algorithm permits to create standard VCC Chains for any number of generators and any number of ports $\Rightarrow$ Scalable and basic (both)

Step 1. Numbering the Ports

Module 1	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
	P1	P4	P7	P10	P13	P16	P19	P21
Module 2	[1]	[2]	[3]	[4]	[5]	[6]	[7]	
	P2	P5	P8	P11	P14	P17	P20	
Module 3	[1]	[2]	[3]	[4]	[5]	[6]		
	P3	P6	P9	P12	P15	P18		

[x] indicates Port number in the module. Px indicates Port number used in our algorithm

Numbering the Ports (Cont)

- Generate a schematic of modules placed one below the other, arranged in a decreasing order of number of ports per module
- Then the switch ports are numbered sequentially, along the columns, starting from the top left corner of the schematic
- This port numbering helps creating VCC chains that cross modules using a simple algorithm.

Algorithm

f=1;

for (k = 1 to r, step 1){

if(k>1){f=0; for($q = mod * (1 + \sum_{d=1}^{k-1} NP(d), N)$ to q<=1, step -1){

f=f+1; while P(f) is source or destination {f =f+1;}}}

for (j = 1 to m, step 1){

if (r is equal to 1 and j > 1) { f = mod*(f+1, N);}

if (r>1 and j>1) { f=C(2,j-1,k);}

for (i =1 to NP(k), step 1){

while (P(f) is source or destination or is full

{ f = mod*(f+1, N);}

C(i, j, k) = P(f); f = mod*(f+1, N);

} end for i

} end for j

} end for k.

Algorithm Rules

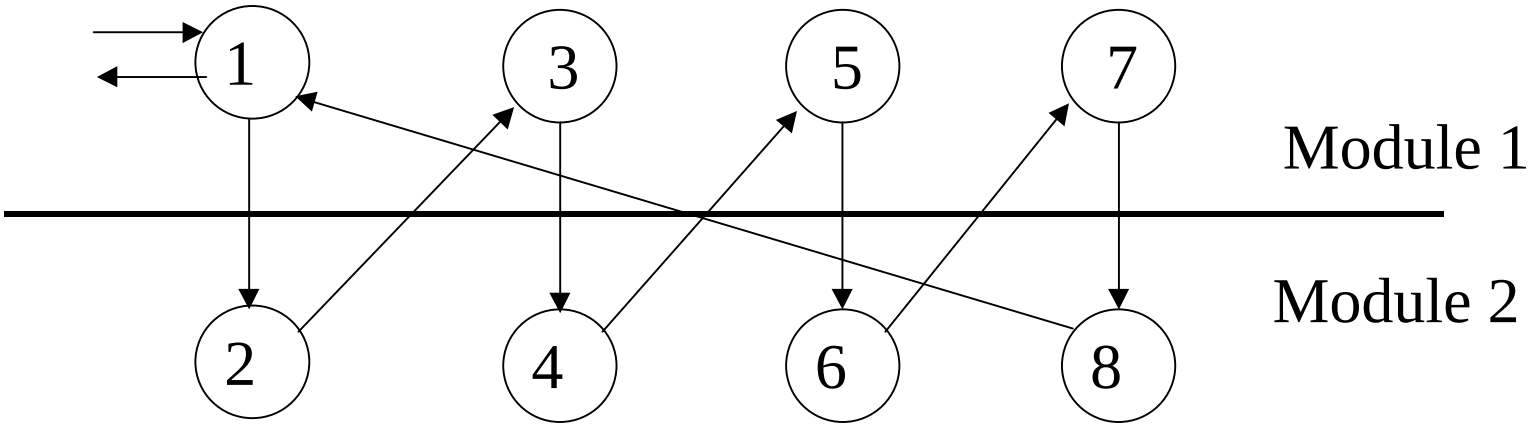
- Each chain generally goes sequentially from port i to port i+1 unless the port has already been fully used by other chains. Use modulo N arithmetic.
- Multiple Chains/Generator: Each new VCC chain is obtained from the previous one shifting by one its port number
- Multiple Generators: Divide the switch ports between the generators. Each generator will start its traffic from its ports.

Foreground vs Background

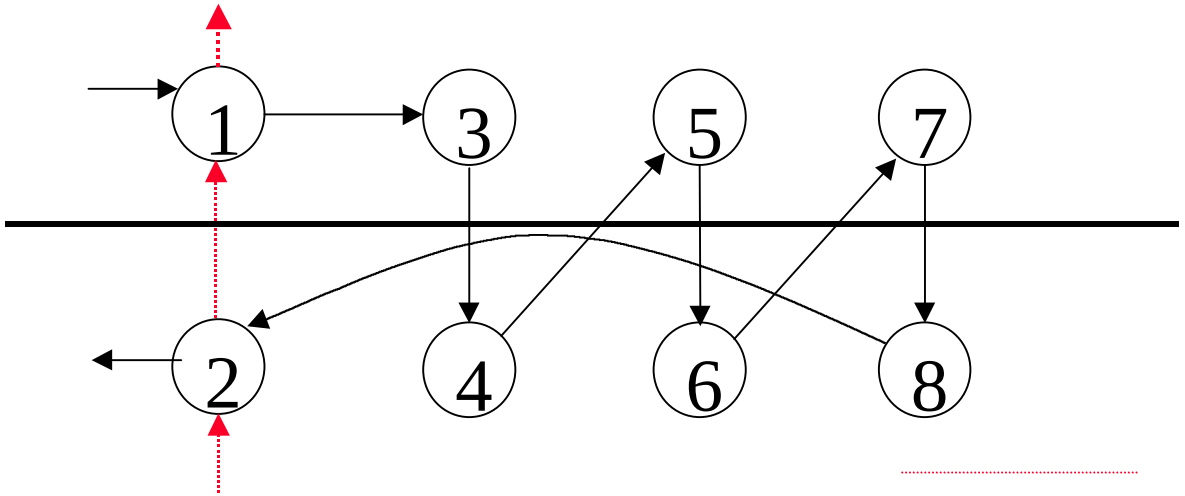
- Two kinds of virtual channel connections (VCCs):
  - Foreground VCCs (traffic that is measured) and
  - Background VCCs (traffic that simply interferes with the foreground traffic).
- Throughput measurements require only foreground traffic
- Latency measurements require both foreground and background traffic
- Foreground and background traffic should not use the same generator/analyzer $\Rightarrow$ OK to share ports in opposite directions

Example: Throughput Measurement

- 8-to-8 Straight with one generator
- Two modules with 4 ports each.
- One VCC chain: P1-P2-P3-P4-P5-P6-P7-P8-P1

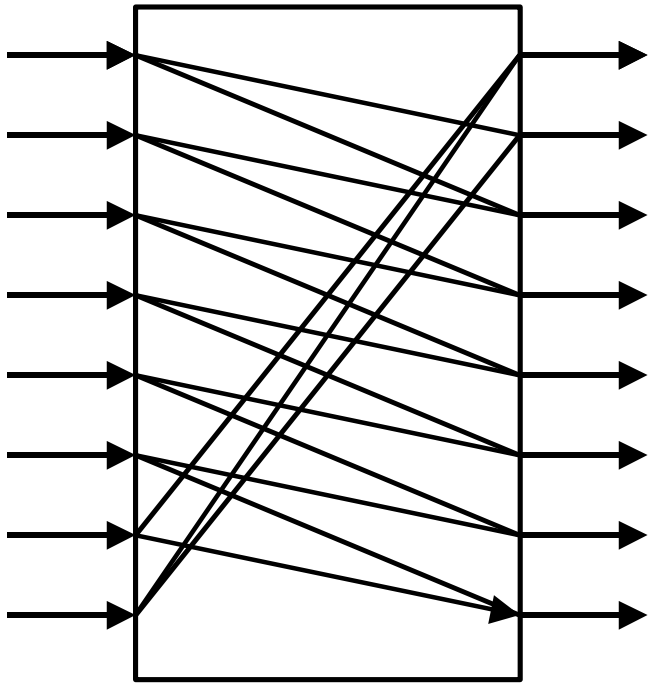


Example: Latency Measurement



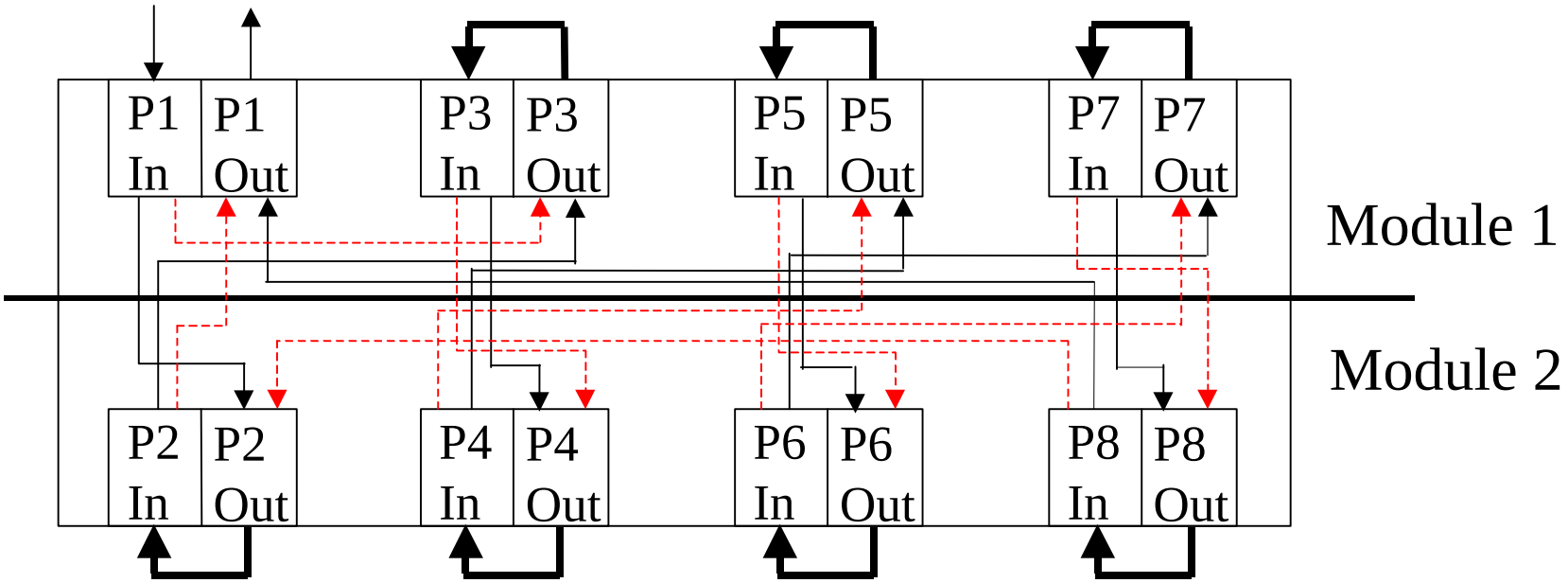
- Foreground traffic: P2-P1
- Background traffic: P1-P3-P4-P5-P6-P7-P8-P2

8-to-2 Partial Cross Throughput Foreground Traffic.

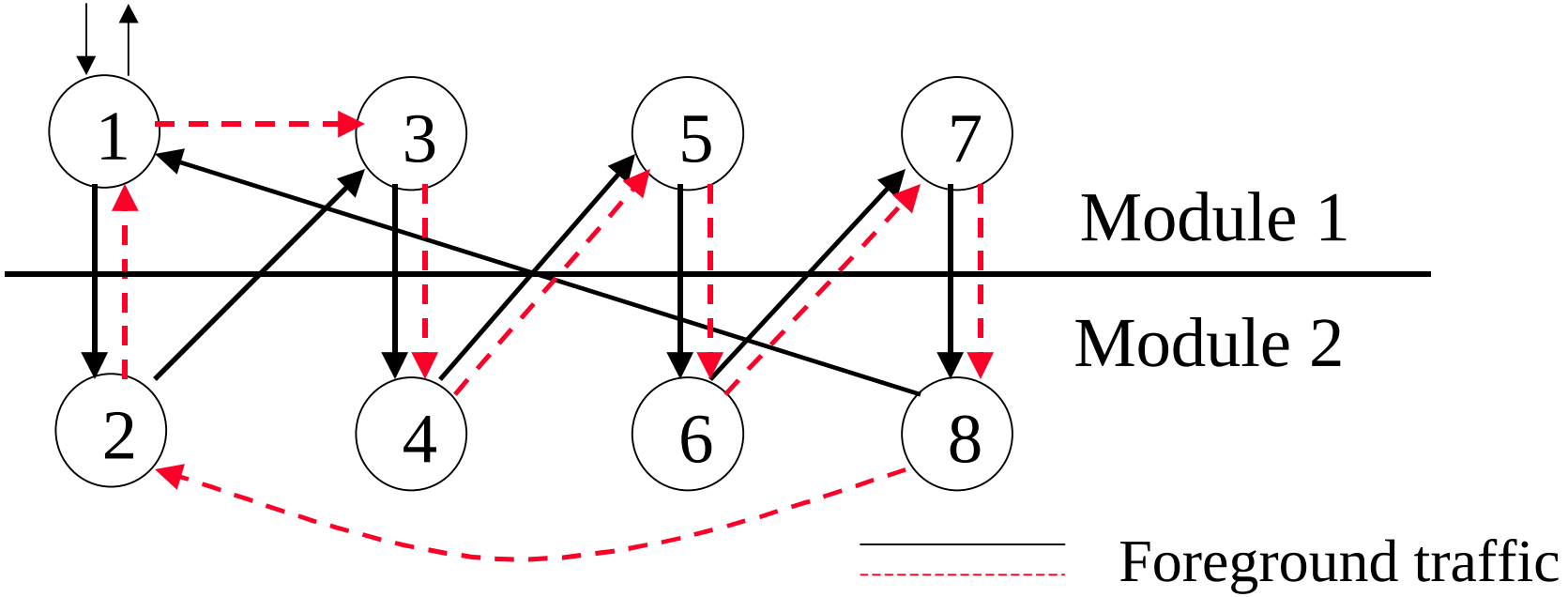


- Basic 8-to-2 Partial Cross configuration

Scaleable 8-to-2 Partial Cross with One Generators

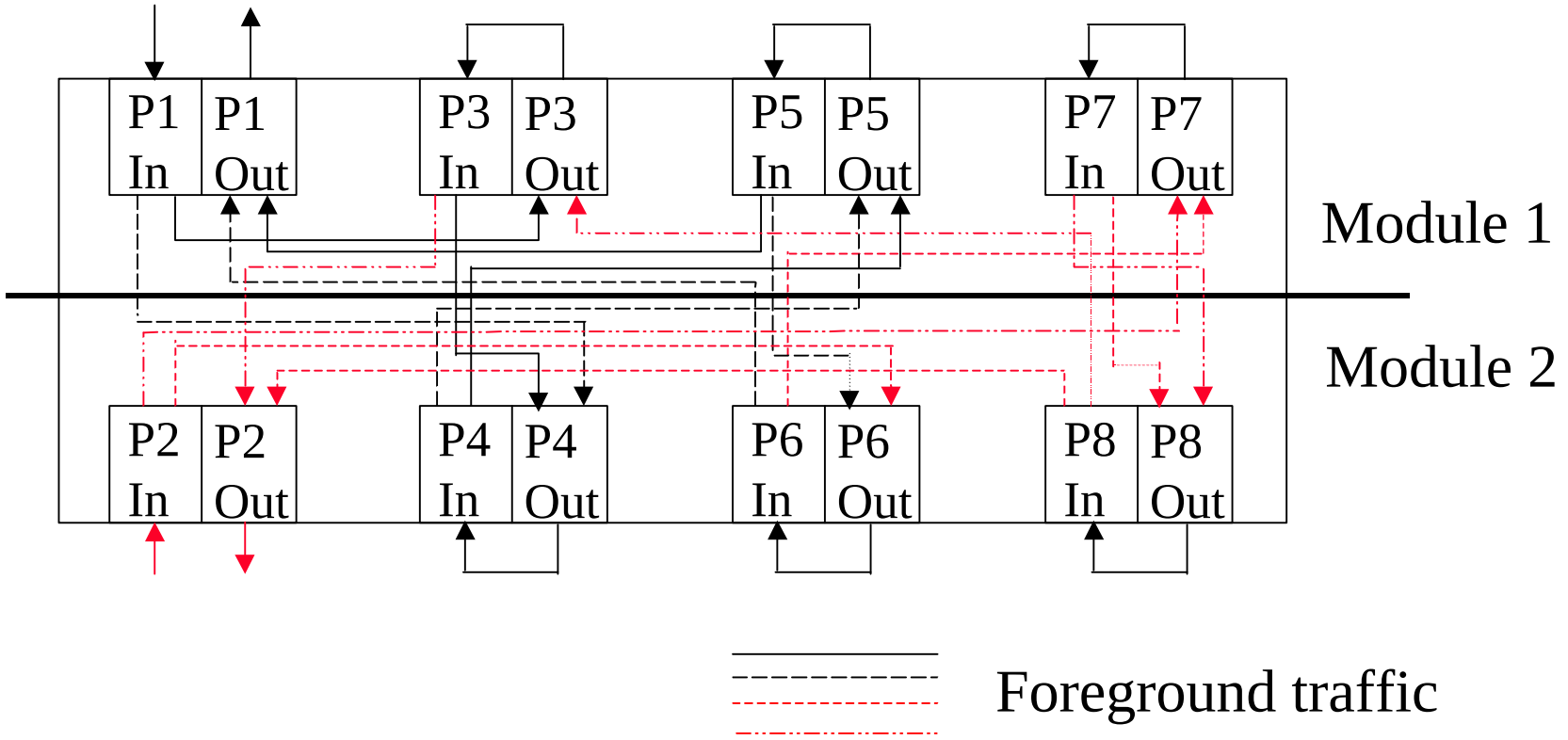


Scaleable 8-to-2 Partial Cross with One Generators (Cont.)

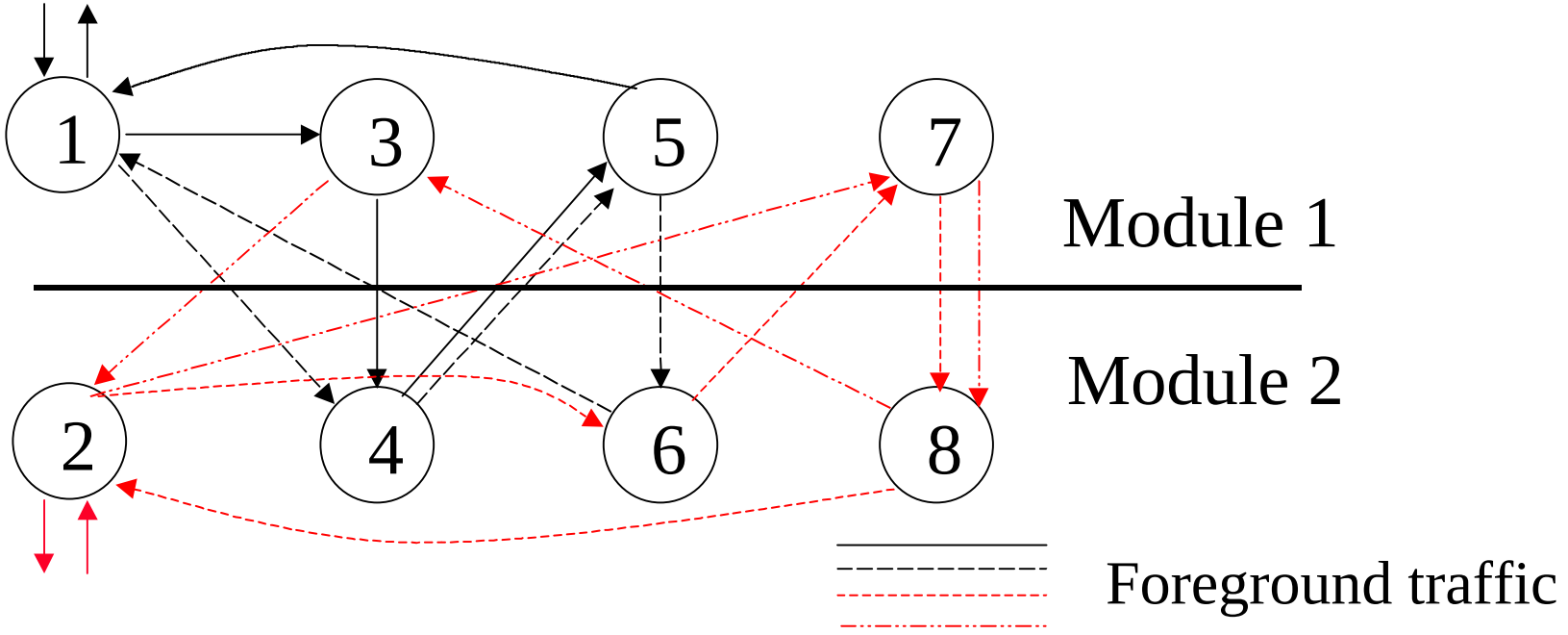


- First VCC Chain: P1-P2-P3-P4-P5-P6-P7-P8-P1
- Second VCC Chain: P1-P3-P4-P5-P6-P7-P8-P2-P1

Scaleable 8-to-2 Partial Cross with 2 Generators

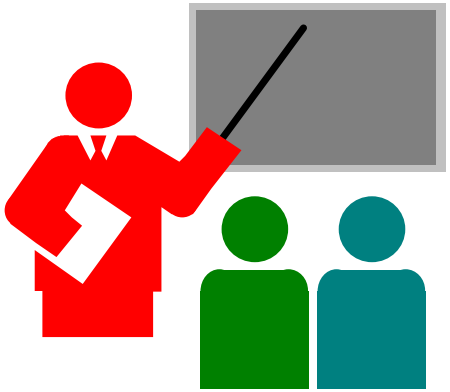


Scaleable 8-to-2 Partial Cross with 2 Generators (Cont.)



- Generator 1: P1-P3-P4-P5-P1 and P1-P4-P5-P6-P1
- Generator 2: P2-P6-P7-P8-P2 and P2-P7-P8-P3-P2

Summary



- The presented methodology permits to implement standard VCC chains.
- This methodology can be used for both scalable and basic configurations.
- The methodology algorithm is simple and can be easily transformed in a computer program.

Motion

- Adopt the text of 97-0835 as Appendix B of Performance Testing Baseline Text.