

Proyecto 2

Verificación funcional de un router bus

Jeremy Córdoba, Jesús Rojas y Mario Montero

1. Introducción

En este informe, se presentan los resultados clave derivados de la implementación de un entorno de verificación funcional para un Router Bus, utilizando la técnica de aleatorización controlada en capas en System Verilog. Este modelo de verificación contiene elementos esenciales, incluyendo un test, un generador, un scoreboard, un agente y un checker, además de 16 drivers y monitores destinados a gestionar todas las señales del Dispositivo bajo Prueba (DUT).

Los escenarios del test se desarrollaron siguiendo un plan de pruebas previamente establecido, lo que permitió abordar exhaustivamente las capacidades del diseño. Además, la creación de una interfaz de programación de aplicaciones (API) resultó fundamental para facilitar la comunicación entre los distintos componentes que conforman este ambiente de verificación funcional.

2. Plan de pruebas

2.1. Diagrama del ambiente de verificación

En la Fig. 1 se presenta el diagrama en capas correspondiente al testbench para el bus parametrizable, en este se muestran todos los componentes así como las conexiones entre estos para llevar a cabo la verificación.

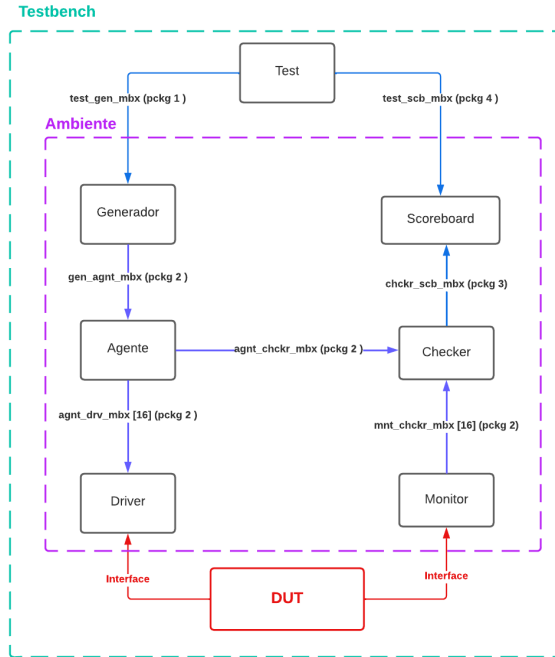


Fig. 1: Estructura del Ambiente de Verificación Funcional Aleatoria en Capas

2.2. Paquetes e interfaces de comunicación

Como se observa en la 1 cada unidad tiene su conexión correspondiente, las cuales se indican como *unidad _ unidad _ mbx (pkg #)*. A continuación se presenta el formato de cada uno de los paquetes de comunicación entre todas las unidades de prueba.

1. **test_agnt_mbx (pkg 1)**: Se trata de un mailbox que comunica al test con el agente, el tipo de mailbox es de *instrucciones_generador* el cual es un tipo de estructura que contiene las pruebas a ejecutar, las características de este mailbox son:
 - Tipo de secuencia:
 - trans_aleatoria (Transacción aleatoria).
 - trans_especifica (Transacción específica).
 - sec_trans_aleatorias (Secuencia de transacciones aleatorias).
 - trans_aleat_x_terminal (Transacciones aleatorias por terminal).
 - una_a_todas (Una terminal específica envía a todas las demás).
 - todas_a_todas (Todas las terminales envían a todas las terminales).
 - llenar_fifos (Llenar las fifos de entrada).
2. **gen_agnt_mbx (pkg 2)**: Se trata de un mailbox que comunica al generador con el agente, es de tipo *trans_router* , los atributos de la clase son:
 - Parámetros:
 - ancho del paquete por defecto en 40.

- Paquete.
 - `nxt_jump`.
 - `mode`.
 - `row`.
 - `colum`.
 - `payload`.
 - `src`.
 - `id`.
 - `retardo_max`
 - `retardo`
 - `term_envio`
 - `term_recibido`
 - `tiempo_envio`
 - `tiempo_recibido`
 - `cola_rutas`
3. **`agnt_drv_mbx [filas*2+columnas*2] (pkg 2)`**: Se trata de un arreglo de mailboxes que comunica al agente con el driver correspondiente, en este caso será un arreglo de 16 mailboxes, uno para cada uno de los dispositivos, es de tipo *trans_router*, por tanto, es el mismo caso del *pkg 2*
 4. **`agnt_chckr_mbx (pkg 2)`**: Se trata de un mailbox que comunica al agente con el checker, es de tipo *trans_router*, por tanto, es el mismo caso del *pkg 2*
 5. **`mon_chckr_mbx [filas*2+columnas*2] (pkg 2)`**: Se trata de un arreglo de mailboxes que comunica a cada uno de los monitores con el checker, en este caso será un arreglo de 16 mailboxes, uno para cada uno de los dispositivos, es de tipo *trans_router*, por tanto, es el mismo caso del *pkg 2*
 6. **`chckr_scb_mbx (pkg 3)`**: Se trata de un mailbox que comunica al checker con el scoreboard, es de tipo *trans_sb* , los atributos de la clase son:
 - Parámetros:
 - ancho del paquete por defecto en 40.
 - `paquete_enviado`.
 - `paquete_recibido`.
 - `term_tx`.
 - `term_rx`.
 - `tiempo_envio`.
 - `tiempo_recibido`.
 - `latencia`.

- completado.

7. **test_scb_mbx (pkg 4)**: Se trata de un mailbox que comunica al test con el scoreboard, tipo de mailbox es de *reporte_scb* el cual es un tipo de estructura que contiene los reporte a mostrar, las características de este mailbox son:

- Tipo de reporte:
 - *reporte_transacciones* (Reporte de todas las transacciones ejecutadas).
 - *retardo_general* (Retardo general entre todas las transacciones ejecutadas).
 - *reporte_ancho_banda* (Reporte el ancho de banda promedio, maximo y minimo).
 - *reporte_trans_inc* (Reporte de las transacciones que no se completaron correctamente).

3. Escenarios

En esta sección se definirán los diferentes escenarios de prueba para el DUT, tanto casos de uso común y casos de esquina.

En la Tabla 1 se muestra un listado de los casos generales de pruebas para el dispositivo bajo prueba (DUT).

En la Tabla 2 se muestra las pruebas para los casos de esquina para el dispositivo bajo prueba (DUT)

Tabla 1: Casos generales para el DUT

Prueba	Objetivo	Recursos
Transacción específica desde el terminal 0 al terminal específico id_row = 5 y id_column = 4 con modo 1, retardo de 5 ciclos de reloj y payload = 8'haa	Constatar que un paquete enviado por un terminal específico, llegue correctamente a un terminal específico, cumpliendo con el modo de enrutamiento asignado.	El ambiente debe ser capaz de poder generar el paquete específico (que incluye target row, target column, modo y payload) y también el terminal desde el cual se realizará la transacción con su respectivo tiempo de envío.
Transacción aleatoria desde un terminal aleatorio entre 0 y 15 a un terminal aleatorio entre 0 y 15, evitando envíos a sí mismos, con paquete aleatorio (target row aleatorio entre 0 y 5, target column entre 0 y 5, modo entre 0 y 1 y payload aleatorio) con un retardo aleatorio entre 0 y 40.	Verificar que un paquete enviado por un terminal asignado aleatoriamente, llegue correctamente al terminal asignado aleatoriamente, cumpliendo con el modo y el retardo asignado aleatoriamente.	El ambiente debe ser capaz de generar paquetes aleatorios (incluyendo target row, target column, modo y payload), tiempos de envíos aleatorios y elegir de manera aleatoria desde cuál terminal disponible se realizará la transmisión. Asimismo, debe ser capaz de limitar el retardo máximo y los posibles valores del ID del paquete (row y column) para que solo se generen transacciones válidas.
Secuencia de transacciones aleatorias con numero de transacciones igual a 100 y retardo maximo de 40 con paquete aleatorio (target row aleatorio entre 0 y 5, target column entre 0 y 5, modo entre 0 y 1 y payload aleatorio)	Verificar que cada una de las transacciones generadas de manera aleatorio lleguen al destino cumpliendo con el modo y el retardo asignado aleatoriamente.	El ambiente debe ser capaz de generar el número de transacciones que se indique y aleatorizar el paquete y el terminal de envío para cada una de las transacciones de la secuencia.
Transacciones aleatorias por terminal con numero de transacciones por cada terminal igual a 10 con retardo máximo de 20 y paquete aleatorio (target row aleatorio entre 0 y 5, target column entre 0 y 5, modo entre 0 y 1 y payload aleatorio)	Verificar que los paquetes aleatorios en cada uno de las transacciones generadas para cada una de las terminales llegue a su destino cumpliendo con el modo y el retardo asignado aleatoriamente.	El ambiente debe ser capaz de generar el número de transacciones que se indique para cada una de las terminales existentes y aleatorizar el paquete para cada una de las transacciones de cada terminal.

Tabla 2: Casos de esquina para el DUT

Enviar desde cada uno de los terminales existentes a todos los demás, con modo específico igual a 0, retardo de envío aleatorio entre 0 y 40 y payload aleatorio.	Verificar que todos los terminales en el sistema pueden realizar transacciones a todos los demás terminales, menos a sí mismos. Cumpliendo con el modo enrutamiento de 0, es decir primero filas, y el retardo asignado aleatoriamente.	El ambiente debe ser capaz generar paquetes de transacción con modo específico, retardo de envío aleatorio y payload aleatorio y no debe permitir envíos a sí mismo.
Enviar desde cada uno de los terminales existentes a todos los demás, con modo específico igual a 1, retardo de envío aleatorio entre 0 y 40 y payload aleatorio.	Verificar que todos los terminales en el sistema pueden realizar transacciones a todos los demás terminales, menos a sí mismos. Cumpliendo con el modo enrutamiento de 1, es decir primero columnas, y el retardo asignado aleatoriamente.	El ambiente debe ser capaz generar paquetes de transacción con modo específico, retardo de envío aleatorio y payload aleatorio y no debe permitir envíos a sí
Llenar las fifos de entrada de cada una de las terminales, desde la terminal 0 a hasta la 15 con un numero de transacciones igual al parámetro <i>profundidad</i> definido al inicio de la prueba. Las transacción tendrán un paquete aleatorio (target row aleatorio entre 0 y 5, target column entre 0 y 5, modo entre 0 y 1 y payload aleatorio).	Asegurar que todas las transacciones cargadas en las FIFOs de entrada de cada una de las terminales sean transmitidas correctamente llegando al destino indicado y cumpliendo con el modo y retardo asignado. Además con esta prueba de busca calcular el ancho de banda de transmisión, a si como los retardos en cada una de las terminales.	El ambiente debe ser capaz de llenar las FIFOs de todas las terminales según el valor de profundidad que se defina en la simulación. Además debe aleatorizar el paquete para cada una de las transacciones de cada terminal.

4. Indicaciones para correr la prueba

Se dispone de un shell script llamado *cmd.sh* para realizar la corrida de la prueba de aleatorización controlada en capas. Para correr el script se puede utilizar el siguiente comando en el directorio en el que se encuentren los archivos fuente de la prueba:

```
#!/bin/bash
source cmd.sh
```

Dicho script le pedirá que ingrese en la línea de comandos la profundidad de las FIFOs del DUT y de los drivers. Para efectos de esta prueba, fueron utilizadas profundidades de entre 2 y 30, por

lo que si se excede de esos límites, cabe la posibilidad de que no tenga suficiente tiempo para ejecutar todas las instrucciones que fueron escritas en el test.

5. Resultados de la prueba

Para la prueba se utilizó el script insertando la profundidad de las FIFOs en 10, con un identificador de broadcast de 0xff (valor por defecto) y un ancho de palabra de 40 bits. Las instrucciones generadas en el test se detallan a continuación:

5.1. Transacción Aleatoria

En esta prueba se envía un paquete con ID, payload, modo, retardo aleatorio desde una terminal especifica, es decir, todo el paquete y sus valores son aleatorios, a excepción de la terminal de envío, los resultados se muestran a continuación.

```
[5] Generador: transacción creada pkt=002526c3f2 Retardo=10 trgt_row=2 trgt_col=5 mode=0 pyld=26c3f2 Terminal_env=3 Tiempo_env=0 Terminal_rcb=13 Tiempo_rcb=0
[15] Agente: Transacción recibida pkt=002526c3f2 Retardo=10 trgt_row=2 trgt_col=5 mode=0 pyld=26c3f2 Terminal_env=3 Tiempo_env=0 Terminal_rcb=13 Tiempo_rcb=0
[155] Driver: transacción completada pkt=002526c3f2 Retardo=10 trgt_row=2 trgt_col=5 mode=0 pyld=26c3f2 Terminal_env=3 Tiempo_env=0 Terminal_rcb=13 Tiempo_rcb=0
[210] Checker: Se recibe transacción desde el Agente pkt=002526c3f2 Retardo=10 trgt_row=2 trgt_col=5 mode=0 pyld=26c3f2 Terminal_env=3 Tiempo_env=115 Terminal_rcb=13 Tiempo_rcb=0
[210] Checker: Paquete: 002526c3f2, ID_Recibido: 14, ID_Esperado:14 -> Ruta Correcta
[300] Checker: Paquete: 022526c3f2, ID_Recibido: 24, ID_Esperado:24 -> Ruta Correcta
[300] Checker: El Paquete: 022526c3f2 ha llegado a su destino por la ruta correcta
[353] Monitor: Transacción Recibida pkt=012526c3f2 Retardo=0 trgt_row=2 trgt_col=5 mode=0 pyld=26c3f2 Terminal_env=0 Tiempo_env=0 Terminal_rcb=13 Tiempo_rcb=345
[360] Checker: Se recibe transacción desde el Monitor pkt=012526c3f2 Retardo=0 trgt_row=2 trgt_col=5 mode=0 pyld=26c3f2 Terminal_env=0 Tiempo_env=0 Terminal_rcb=13 Tiempo_rcb=345
Numero de terminal de recibido: 13
El dato calza con el esperado
[360] Checker: Transacción Completada dato_tx=0x002526c3f2, dato_rx=0x012526c3f2, t_tx=115, t_rx=345, term_tx=3, term_rx=13, ltncy=230, state=1
```

Fig. 2: Verificación de la transacción aleatoria

En la figura 2 se aprecia como el dato fue enviado desde la terminal 3 y se recibió en la terminal 13 de ID 25, se aprecia como se verifica cada camino que toma para llegar a su destino.

5.2. Transacción Específica

En esta prueba se establece todo de forma específica en el paquete, su fuente, ID, payload, modo y retardo, los resultados se muestran a continuación.

```
[10] Test: Enviada la segunda instruccion al generador -> transacción específica
[10] Generador: ha recibido una instrucción
[10] Generador Instrucción: trans_especifica
[10] Generador: transacción creada pkt=00548000aa Retardo=4 trgt_row=5 trgt_col=4 mode=1 pyld=0000aa Terminal_env=0 Tiempo_env=0 Terminal_rcb=11 Tiempo_rcb=0
[55] Agente: Transacción recibida pkt=00548000aa Retardo=4 trgt_row=5 trgt_col=4 mode=1 pyld=0000aa Terminal_env=0 Tiempo_env=0 Terminal_rcb=11 Tiempo_rcb=0
[55] Driver: transacción completada pkt=00548000aa Retardo=4 trgt_row=5 trgt_col=4 mode=1 pyld=0000aa Terminal_env=0 Tiempo_env=55 Terminal_rcb=11 Tiempo_rcb=0
[60] Checker: Se recibe transacción desde el Agente pkt=00548000aa Retardo=4 trgt_row=5 trgt_col=4 mode=1 pyld=0000aa Terminal_env=0 Tiempo_env=55 Terminal_rcb=11 Tiempo_rcb=0
[70] Checker: Paquete: 00548000aa, ID_Recibido: 11, ID_Esperado:11 -> Ruta Correcta
[160] Checker: Paquete: 02548000aa, ID_Recibido: 21, ID_Esperado:21 -> Ruta Correcta
[250] Checker: Paquete: 02548000aa, ID_Recibido: 31, ID_Esperado:31 -> Ruta Correcta
[340] Checker: Paquete: 02548000aa, ID_Recibido: 41, ID_Esperado:41 -> Ruta Correcta
[410] Checker: Paquete: 01548000aa, ID_Recibido: 42, ID_Esperado:42 -> Ruta Correcta
[480] Checker: Paquete: 01548000aa, ID_Recibido: 43, ID_Esperado:43 -> Ruta Correcta
[550] Checker: Paquete: 01548000aa, ID_Recibido: 44, ID_Esperado:44 -> Ruta Correcta
[550] Checker: El Paquete: 01548000aa ha llegado a su destino por la ruta correcta
[595] Monitor: Transacción Recibida pkt=02548000aa Retardo=0 trgt_row=5 trgt_col=4 mode=1 pyld=0000aa Terminal_env=0 Tiempo_env=0 Terminal_rcb=11 Tiempo_rcb=585
[600] Checker: Se recibe transacción desde el Monitor pkt=02548000aa Retardo=0 trgt_row=5 trgt_col=4 mode=1 pyld=0000aa Terminal_env=0 Tiempo_env=0 Terminal_rcb=11 Tiempo_rcb=585
Numero de terminal de recibido: 11
El dato calza con el esperado
[600] Checker: Transacción Completada dato_tx=0x00548000aa, dato_rx=0x02548000aa, t_tx=55, t_rx=585, term_tx=0, term_rx=11, ltncy=530, state=1
```

Fig. 3: Verificación de la transacción específica

Como se puede apreciar en la figura 3 el paquete específico enviado desde la terminal 0 del DUT y pasa por cada router según su modo establecido hasta llegar a la terminal de salida 11 de ID

5.3. Sección de transacciones aleatorias

En este escenario se ejecutan una serie de 100 transacciones aleatorias en total, las cuales pueden salir desde cualquiera de los 16 terminales y hacia cualquiera de los 16 dispositivos, exceptuando el caso en los que se generen envíos hacia el mismo terminal, ya que esto no tendría mucho sentido. En la Fig. 4 se observa la recepción de la instrucción en el generador y su posterior creación en el mismo bloque funcional. Por su parte, la Fig. 5 muestra la verificación de la ruta de cada transacción por parte del checker.

```
[5] Generador: ha recibido una instrucción
[5] Generador Instrucción: sec_trans_aleatorias
[5] Generador: transacción creada pkt=000421ee59 Retardo=19 trgt_row=0 trgt_col=4 mode=0 pyld=21ee59 Terminal_env=15 Tiempo_env=0 Terminal_rcb=3 Tiempo_rcb=0
[5] Generador: transacción creada pkt=002526c3f2 Retardo=10 trgt_row=2 trgt_col=5 mode=0 pyld=26c3f2 Terminal_env=0 Tiempo_env=0 Terminal_rcb=13 Tiempo_rcb=0
[5] Generador: transacción creada pkt=0054a74bab Retardo=6 trgt_row=5 trgt_col=4 mode=1 pyld=274bab Terminal_env=6 Tiempo_env=0 Terminal_rcb=11 Tiempo_rcb=0
[5] Generador: transacción creada pkt=00031def07 Retardo=18 trgt_row=0 trgt_col=3 mode=0 pyld=1def07 Terminal_env=6 Tiempo_env=0 Terminal_rcb=2 Tiempo_rcb=0
[5] Generador: transacción creada pkt=00514be970 Retardo=18 trgt_row=5 trgt_col=1 mode=0 pyld=4be970 Terminal_env=6 Tiempo_env=0 Terminal_rcb=8 Tiempo_rcb=0
[5] Generador: transacción creada pkt=0051510051 Retardo=20 trgt_row=5 trgt_col=1 mode=0 pyld=510051 Terminal_env=12 Tiempo_env=0 Terminal_rcb=8 Tiempo_rcb=0
[5] Generador: transacción creada pkt=00250d4d49 Retardo=10 trgt_row=2 trgt_col=5 mode=0 pyld=0d4d49 Terminal_env=10 Tiempo_env=0 Terminal_rcb=13 Tiempo_rcb=0
```

Fig. 4: Generación de una parte de las transacciones aleatorias

```
Checker: El Paquete: 000421ee59 ha llegado a su destino por la ruta correcta
Checker: El Paquete: 002526c3f2 ha llegado a su destino por la ruta correcta
Checker: El Paquete: 0154a74bab ha llegado a su destino por la ruta correcta
Checker: El Paquete: 02514be970 ha llegado a su destino por la ruta correcta
Checker: El Paquete: 00031def07 ha llegado a su destino por la ruta correcta
Checker: El Paquete: 00250d4d49 ha llegado a su destino por la ruta correcta
```

Fig. 5: Comprobación de la ruta de las transacciones aleatorias

5.4. Transacciones aleatorias por terminal

También se ejecutó un escenario en el que se realice una cantidad de 10 transacciones aleatorias por terminal con diferentes tiempos de retardo, lo que hace casi imposible el llenado de las FIFOs de entrada, por lo que DUT tiene suficiente tiempo para transmitir dichos paquetes sin generar errores. En las Figs. 6 y 7 se puede observar respectivamente la generación y comprobación de una parte de las transacciones creadas en este escenario.

```
[10] Generador Instrucción: trans_aleat_x_terminal
[10] Generador: transacción creada pkt=0003a7385c Retardo=19 trgt_row=0 trgt_col=3 mode=1 pyld=27385c Terminal_env=0 Tiempo_env=0 Terminal_rcb=2 Tiempo_rcb=0
[10] Generador: transacción creada pkt=004059d926 Retardo=5 trgt_row=4 trgt_col=0 mode=0 pyld=59d926 Terminal_env=0 Tiempo_env=0 Terminal_rcb=7 Tiempo_rcb=0
[10] Generador: transacción creada pkt=00205b68da Retardo=10 trgt_row=2 trgt_col=0 mode=0 pyld=5b68da Terminal_env=0 Tiempo_env=0 Terminal_rcb=5 Tiempo_rcb=0
[10] Generador: transacción creada pkt=0002d38951 Retardo=9 trgt_row=0 trgt_col=2 mode=1 pyld=538951 Terminal_env=0 Tiempo_env=0 Terminal_rcb=1 Tiempo_rcb=0
[10] Generador: transacción creada pkt=0051ab3113 Retardo=17 trgt_row=5 trgt_col=1 mode=1 pyld=2b3113 Terminal_env=0 Tiempo_env=0 Terminal_rcb=8 Tiempo_rcb=0
[10] Generador: transacción creada pkt=0020099e07 Retardo=6 trgt_row=2 trgt_col=0 mode=0 pyld=099e07 Terminal_env=0 Tiempo_env=0 Terminal_rcb=5 Tiempo_rcb=0
[10] Generador: transacción creada pkt=0015d827be Retardo=0 trgt_row=1 trgt_col=5 mode=1 pyld=5827be Terminal_env=0 Tiempo_env=0 Terminal_rcb=12 Tiempo_rcb=0
[10] Generador: transacción creada pkt=0054a66ff4 Retardo=9 trgt_row=5 trgt_col=4 mode=1 pyld=266ff4 Terminal_env=0 Tiempo_env=0 Terminal_rcb=11 Tiempo_rcb=0
```

Fig. 6: Generación de las transacciones aleatorias


```

[11490] Checker: El Paquete: 0103a7385c ha llegado a su destino por la ruta correcta
[11540] Checker: El Paquete: 0352c717d4 ha llegado a su destino por la ruta correcta
[11720] Checker: El Paquete: 02205b68da ha llegado a su destino por la ruta correcta
[11780] Checker: El Paquete: 024059d926 ha llegado a su destino por la ruta correcta
[11820] Checker: El Paquete: 0102d38951 ha llegado a su destino por la ruta correcta
[12070] Checker: El Paquete: 0220099e07 ha llegado a su destino por la ruta correcta
[12150] Checker: El Paquete: 0251ab3113 ha llegado a su destino por la ruta correcta
[12310] Checker: El Paquete: 0115d827be ha llegado a su destino por la ruta correcta
[12470] Checker: El Paquete: 0102c51254 ha llegado a su destino por la ruta correcta
[12550] Checker: El Paquete: 011536df57 ha llegado a su destino por la ruta correcta
[12620] Checker: El Paquete: 0154a66ff4 ha llegado a su destino por la ruta correcta

```

Fig. 7: Comprobación de la ruta de algunas de las transacciones aleatorias por terminal

5.5. Envío de todas las terminales a todas las terminales con modo específico

Con esta prueba se verificó que todas las terminales fueran capaces de enviar paquetes a todas las demás terminales del sistema, el ambiente fue capaz de generar paquetes aleatorios para cada transacción que se generaba en cada terminal. En la Fig. 8 se muestra un fragmento del resultado en consola de la simulación hecha con esta prueba, en esta se observa como la terminal 0 envía paquetes aleatorios con el modo especificado en 1, para este caso, a todas las demás terminales, de igual manera la terminal y así con todas las demás. Además, se nota como no se envían datos a sí mismas.

```

[500010] Generador: ha recibido una instrucción
[500010] Generador: Instrucción: todas_a_todas_moda_esp
[500010] Generador: transacción creada pkt=000291bf43 Retardo=1 trgt_row=0 trgt_col=2 mode=1 pyId=41bf43 Terminal_env=0 Tiempo_env=0 Terminal_rcb=1 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0003ba5736 Retardo=8 trgt_row=0 trgt_col=3 mode=1 pyId=3ba5736 Terminal_env=0 Tiempo_env=0 Terminal_rcb=2 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0004af8cee Retardo=6 trgt_row=0 trgt_col=4 mode=1 pyId=2f8cee Terminal_env=0 Tiempo_env=0 Terminal_rcb=3 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0010617847 Retardo=6 trgt_row=1 trgt_col=0 mode=1 pyId=317847 Terminal_env=0 Tiempo_env=0 Terminal_rcb=4 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0020ab2167 Retardo=4 trgt_row=2 trgt_col=0 mode=1 pyId=2b2167 Terminal_env=0 Tiempo_env=0 Terminal_rcb=5 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00308730bf Retardo=5 trgt_row=3 trgt_col=0 mode=1 pyId=0730bf Terminal_env=0 Tiempo_env=0 Terminal_rcb=6 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00408d0d4f Retardo=17 trgt_row=4 trgt_col=0 mode=1 pyId=d0d4f Terminal_env=0 Tiempo_env=0 Terminal_rcb=7 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0051b2044f Retardo=0 trgt_row=5 trgt_col=1 mode=1 pyId=32044f Terminal_env=0 Tiempo_env=0 Terminal_rcb=8 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0052eada5a Retardo=0 trgt_row=5 trgt_col=2 mode=1 pyId=6eada5a Terminal_env=0 Tiempo_env=0 Terminal_rcb=9 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=005381a2b3 Retardo=10 trgt_row=5 trgt_col=3 mode=1 pyId=01a2b3 Terminal_env=0 Tiempo_env=0 Terminal_rcb=10 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00548e84db Retardo=4 trgt_row=5 trgt_col=4 mode=1 pyId=0e84db Terminal_env=0 Tiempo_env=0 Terminal_rcb=11 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0015f1e892 Retardo=4 trgt_row=1 trgt_col=5 mode=1 pyId=71e892 Terminal_env=0 Tiempo_env=0 Terminal_rcb=12 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0025984e28 Retardo=2 trgt_row=2 trgt_col=5 mode=1 pyId=d184e28 Terminal_env=0 Tiempo_env=0 Terminal_rcb=13 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0035d8fd02 Retardo=20 trgt_row=3 trgt_col=5 mode=1 pyId=58fd02 Terminal_env=0 Tiempo_env=0 Terminal_rcb=14 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0045af2f8e Retardo=1 trgt_row=4 trgt_col=5 mode=1 pyId=2f2f8e Terminal_env=0 Tiempo_env=0 Terminal_rcb=15 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0001a04e81 Retardo=3 trgt_row=0 trgt_col=1 mode=1 pyId=204e81 Terminal_env=1 Tiempo_env=0 Terminal_rcb=0 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0003a132c2 Retardo=3 trgt_row=0 trgt_col=3 mode=1 pyId=d132c2 Terminal_env=1 Tiempo_env=0 Terminal_rcb=2 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0004a6fc24 Retardo=11 trgt_row=0 trgt_col=4 mode=1 pyId=66fc24 Terminal_env=1 Tiempo_env=0 Terminal_rcb=3 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00108f4241 Retardo=7 trgt_row=1 trgt_col=0 mode=1 pyId=d0f4241 Terminal_env=1 Tiempo_env=0 Terminal_rcb=4 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0020ecd4a4 Retardo=8 trgt_row=2 trgt_col=0 mode=1 pyId=d6cd4a4 Terminal_env=1 Tiempo_env=0 Terminal_rcb=5 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0030d1ae02 Retardo=6 trgt_row=3 trgt_col=0 mode=1 pyId=51ae02 Terminal_env=1 Tiempo_env=0 Terminal_rcb=6 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0040db37b4 Retardo=13 trgt_row=4 trgt_col=0 mode=1 pyId=5b37b4 Terminal_env=1 Tiempo_env=0 Terminal_rcb=7 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0051c54f23 Retardo=17 trgt_row=5 trgt_col=1 mode=1 pyId=454f23 Terminal_env=1 Tiempo_env=0 Terminal_rcb=8 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0052a4e4af Retardo=1 trgt_row=5 trgt_col=2 mode=1 pyId=24e4af Terminal_env=1 Tiempo_env=0 Terminal_rcb=9 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0053a2153d Retardo=18 trgt_row=5 trgt_col=3 mode=1 pyId=23153d Terminal_env=1 Tiempo_env=0 Terminal_rcb=10 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0054c2c996 Retardo=16 trgt_row=5 trgt_col=4 mode=1 pyId=42c996 Terminal_env=1 Tiempo_env=0 Terminal_rcb=11 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00158de84 Retardo=2 trgt_row=1 trgt_col=5 mode=1 pyId=0de84 Terminal_env=1 Tiempo_env=0 Terminal_rcb=12 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00258266d2 Retardo=3 trgt_row=2 trgt_col=5 mode=1 pyId=0266d2 Terminal_env=1 Tiempo_env=0 Terminal_rcb=13 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0035aff6a4 Retardo=4 trgt_row=3 trgt_col=5 mode=1 pyId=2fff6a4 Terminal_env=1 Tiempo_env=0 Terminal_rcb=14 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0045b6a45f Retardo=10 trgt_row=4 trgt_col=5 mode=1 pyId=3b6a45f Terminal_env=1 Tiempo_env=0 Terminal_rcb=15 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0001aebd7c Retardo=20 trgt_row=0 trgt_col=1 mode=1 pyId=2aebd7c Terminal_env=2 Tiempo_env=0 Terminal_rcb=0 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0002ceb074 Retardo=16 trgt_row=0 trgt_col=2 mode=1 pyId=4eb074 Terminal_env=2 Tiempo_env=0 Terminal_rcb=1 Tiempo_rcb=0

```

Fig. 8: Verificación de la transacción de envío de todas a todas modo 1

También, en la Fig. 9 se observa el resultado de la prueba con modo de enrutamiento en 0.

```
[500010] Generador: Instrucción: todas_a_todas_modo_esp
[500010] Generador: transacción creada pkt=00023d900d Retardo=6 trgt_row=0 trgt_col=2 mode=0 pyld=3d900d Terminal_env=0 Tiempo_env=0 Terminal rcb=1 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=000361bb35 Retardo=7 trgt_row=0 trgt_col=3 mode=0 pyld=61bb35 Terminal_env=0 Tiempo_env=0 Terminal rcb=2 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00046ba54a Retardo=7 trgt_row=0 trgt_col=4 mode=0 pyld=6ba54a Terminal_env=0 Tiempo_env=0 Terminal rcb=3 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00100bffaaf Retardo=7 trgt_row=1 trgt_col=0 mode=0 pyld=0bffaaf Terminal_env=0 Tiempo_env=0 Terminal rcb=4 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0020327db8 Retardo=0 trgt_row=2 trgt_col=0 mode=0 pyld=327db8 Terminal_env=0 Tiempo_env=0 Terminal rcb=5 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00304790f4 Retardo=12 trgt_row=3 trgt_col=0 mode=0 pyld=4790f4 Terminal_env=0 Tiempo_env=0 Terminal rcb=6 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00400de5df Retardo=2 trgt_row=4 trgt_col=0 mode=0 pyld=0de5df Terminal_env=0 Tiempo_env=0 Terminal rcb=7 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=005111a8ed Retardo=0 trgt_row=5 trgt_col=0 mode=0 pyld=11a8ed Terminal_env=0 Tiempo_env=0 Terminal rcb=8 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00523a0244 Retardo=4 trgt_row=5 trgt_col=2 mode=0 pyld=3a0244 Terminal_env=0 Tiempo_env=0 Terminal rcb=9 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0053253c94 Retardo=12 trgt_row=5 trgt_col=3 mode=0 pyld=253c94 Terminal_env=0 Tiempo_env=0 Terminal rcb=10 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=005456c36d Retardo=20 trgt_row=5 trgt_col=4 mode=0 pyld=56c36d Terminal_env=0 Tiempo_env=0 Terminal rcb=11 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00150a6e6c Retardo=9 trgt_row=1 trgt_col=5 mode=0 pyld=0a6e6c Terminal_env=0 Tiempo_env=0 Terminal rcb=12 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=002569ecb4 Retardo=13 trgt_row=2 trgt_col=5 mode=0 pyld=69ecb4 Terminal_env=0 Tiempo_env=0 Terminal rcb=13 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0035267aba Retardo=10 trgt_row=3 trgt_col=5 mode=0 pyld=267aba Terminal_env=0 Tiempo_env=0 Terminal rcb=14 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=004545329e Retardo=5 trgt_row=4 trgt_col=5 mode=0 pyld=45329e Terminal_env=0 Tiempo_env=0 Terminal rcb=15 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0001724a1e Retardo=2 trgt_row=0 trgt_col=1 mode=0 pyld=724a1e Terminal_env=1 Tiempo_env=0 Terminal rcb=0 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00033d6a4c Retardo=14 trgt_row=0 trgt_col=3 mode=0 pyld=3d6a4c Terminal_env=1 Tiempo_env=0 Terminal rcb=2 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=000461758c Retardo=17 trgt_row=0 trgt_col=4 mode=0 pyld=61758c Terminal_env=1 Tiempo_env=0 Terminal rcb=3 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00106bc213 Retardo=3 trgt_row=1 trgt_col=0 mode=0 pyld=6bc213 Terminal_env=1 Tiempo_env=0 Terminal rcb=4 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=002030e690 Retardo=1 trgt_row=2 trgt_col=0 mode=0 pyld=30e690 Terminal_env=1 Tiempo_env=0 Terminal rcb=5 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00306341a8 Retardo=7 trgt_row=3 trgt_col=0 mode=0 pyld=6341a8 Terminal_env=1 Tiempo_env=0 Terminal rcb=6 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0040496f98 Retardo=2 trgt_row=4 trgt_col=0 mode=0 pyld=496f98 Terminal_env=1 Tiempo_env=0 Terminal rcb=7 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00513f7e4f Retardo=8 trgt_row=5 trgt_col=0 mode=0 pyld=3f7e4f Terminal_env=1 Tiempo_env=0 Terminal rcb=8 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00520bb723 Retardo=8 trgt_row=5 trgt_col=2 mode=0 pyld=0bb723 Terminal_env=1 Tiempo_env=0 Terminal rcb=9 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00534ad0a1 Retardo=0 trgt_row=5 trgt_col=3 mode=0 pyld=4ad0a1 Terminal_env=1 Tiempo_env=0 Terminal rcb=10 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0054423e7c Retardo=17 trgt_row=5 trgt_col=4 mode=0 pyld=423e7c Terminal_env=1 Tiempo_env=0 Terminal rcb=11 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00154e1a84 Retardo=10 trgt_row=1 trgt_col=5 mode=0 pyld=4e1a84 Terminal_env=1 Tiempo_env=0 Terminal rcb=12 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=00254911ba Retardo=3 trgt_row=2 trgt_col=5 mode=0 pyld=4911ba Terminal_env=1 Tiempo_env=0 Terminal rcb=13 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=003575bd54 Retardo=4 trgt_row=3 trgt_col=5 mode=0 pyld=75bd54 Terminal_env=1 Tiempo_env=0 Terminal rcb=14 Tiempo_rcb=0
[500010] Generador: transacción creada pkt=0045616f74 Retardo=14 trgt_row=4 trgt_col=5 mode=0 pyld=616f74 Terminal_env=1 Tiempo_env=0 Terminal rcb=15 Tiempo_rcb=0
```

Fig. 9: Verificación de la transacción de envío de todas a todas modo 0

5.6. Llenar FIFOs

Esta prueba se generó con el fin de obtener los resultados más confiables de ancho de banda y retrasos promedio generales y por terminal. Ya que en este escenario se crea una cantidad de transacciones aleatorias por terminal igual a la profundidad de las FIFOs de los manejadores (drivers) con retardos de 0 ciclos del reloj. Los resultados se pueden observar en las Figs. 10 y 11.

```
[501400] Generador: ha recibido una instrucción
[501400] Generador: Instrucción: llenar_fifo
[501400] Generador: transacción creada pkt=0052761504 Retardo=0 trgt_row=5 trgt_col=2 mode=0 pyld=761504 Terminal_env=0 Tiempo_env=0 Terminal rcb=9 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=00152ff508 Retardo=0 trgt_row=1 trgt_col=5 mode=0 pyld=2ff508 Terminal_env=0 Tiempo_env=0 Terminal rcb=12 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=00450acfb7 Retardo=0 trgt_row=4 trgt_col=5 mode=0 pyld=0acfb7 Terminal_env=0 Tiempo_env=0 Terminal rcb=15 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=00203d76c4 Retardo=0 trgt_row=2 trgt_col=0 mode=0 pyld=3d76c4 Terminal_env=0 Tiempo_env=0 Terminal rcb=5 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=002025ebcc Retardo=0 trgt_row=2 trgt_col=0 mode=0 pyld=25ebcc Terminal_env=0 Tiempo_env=0 Terminal rcb=5 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=0010a5006c Retardo=0 trgt_row=1 trgt_col=0 mode=1 pyld=25006c Terminal_env=0 Tiempo_env=0 Terminal rcb=4 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=0045976794 Retardo=0 trgt_row=4 trgt_col=5 mode=1 pyld=176794 Terminal_env=0 Tiempo_env=0 Terminal rcb=15 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=0054d12ea9 Retardo=0 trgt_row=5 trgt_col=4 mode=1 pyld=512ea9 Terminal_env=0 Tiempo_env=0 Terminal rcb=11 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=0020f0dff9 Retardo=0 trgt_row=2 trgt_col=0 mode=1 pyld=70dff9 Terminal_env=0 Tiempo_env=0 Terminal rcb=5 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=0030809db6 Retardo=0 trgt_row=3 trgt_col=0 mode=1 pyld=009db6 Terminal_env=0 Tiempo_env=0 Terminal rcb=6 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=0051520a61 Retardo=0 trgt_row=5 trgt_col=1 mode=0 pyld=520a61 Terminal_env=1 Tiempo_env=0 Terminal rcb=8 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=0020c54eeb Retardo=0 trgt_row=2 trgt_col=0 mode=1 pyld=454eeb Terminal_env=1 Tiempo_env=0 Terminal rcb=5 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=002559da35 Retardo=0 trgt_row=2 trgt_col=5 mode=0 pyld=59da35 Terminal_env=1 Tiempo_env=0 Terminal rcb=13 Tiempo_rcb=0
[501400] Generador: transacción creada pkt=0040140e53 Retardo=0 trgt_row=4 trgt_col=0 mode=0 pyld=140e53 Terminal_env=1 Tiempo_env=0 Terminal rcb=7 Tiempo_rcb=0
```

Fig. 10: Generación de las transacciones aleatorias para llenar las FIFOs de entrada de los drivers.

```
[501690] Checker: El Paquete: 01152ff508 ha llegado a su destino por la ruta correcta
[501720] Checker: El Paquete: 0252761504 ha llegado a su destino por la ruta correcta
[501770] Checker: El Paquete: 0251520a61 ha llegado a su destino por la ruta correcta
[501790] Checker: El Paquete: 0320c54eeb ha llegado a su destino por la ruta correcta
[501870] Checker: El Paquete: 02203d76c4 ha llegado a su destino por la ruta correcta
[501880] Checker: El Paquete: 0104049292 ha llegado a su destino por la ruta correcta
[501910] Checker: El Paquete: 02452e53d6 ha llegado a su destino por la ruta correcta
[502010] Checker: El Paquete: 022025ebcc ha llegado a su destino por la ruta correcta
```

Fig. 11: Comprobación de la ruta de algunas de las transacciones utilizadas para llenar FIFOs de entrada.

Cabe resaltar que esta prueba fue la que generó más errores en la transmisión de paquetes en el DUT, ya que al tener muchas señales de pending en alto, el DUT parece no ser capaz de manejar la entrada de paquetes por sus 4 terminales en periodos cortos de tiempo. Como se puede ver en

la Fig. 12, el paquete 0x00534ab5bf llega a su destino haciendo una ruta incorrecta (e.g. no sigue la ruta según el modo), lo que ocasiona que el paquete se "duplica" y sea recibido dos veces por el mismo terminal.

```
[502710] Checker: El Paquete: 024057a962 ha llegado a su destino por la ruta incorrecta
[502740] Checker: El Paquete: 01048689fc ha llegado a su destino por la ruta correcta
[502780] Checker: El Paquete: 0303c36137 ha llegado a su destino por la ruta correcta
[502790] Checker: El Paquete: 021430a099 ha llegado a su destino por la ruta correcta
[502800] Checker: El Paquete: 00104837de ha llegado a su destino por la ruta correcta
[502800] Checker: El Paquete: 024057a962 ha llegado a su destino por la ruta correcta
[502800] Checker Error: el Paquete: 024057a962 recibido en ID: 43 nunca fue generado/se ha duplicado
[502810] Checker Error: el Paquete: 024057a962 recibido en ID: 33 nunca fue generado/se ha duplicado
[502820] Checker: Error el dato de la transacción no calza con el esperado pkt=0003c36137 Retardo=0 trgt_row=0 trgt_col=3 mode=1 pyId=436137 Terminal_env=0 Tiempo_env=0 Terminal_rcb=2 Tiempo_rcb=502805
[502840] Checker: El Paquete: 0314c1cc09 ha llegado a su destino por la ruta correcta
[502850] Checker: El Paquete: 00104837de ha llegado a su destino por la ruta correcta
[502860] Checker: El Paquete: 0103fc2983 ha llegado a su destino por la ruta correcta
[502870] Checker: El Paquete: 00026ca87a ha llegado a su destino por la ruta correcta
[502890] Checker: El Paquete: 0141a4a016 ha llegado a su destino por la ruta correcta
[502920] Checker: El Paquete: 0010eb6b1b ha llegado a su destino por la ruta correcta
[502920] Checker: El Paquete: 003553221d ha llegado a su destino por la ruta correcta
[502940] Checker Error: el Paquete: 024057a962 recibido en ID: 43 nunca fue generado/se ha duplicado
[502950] Checker: El Paquete: 0220f50ff9 ha llegado a su destino por la ruta correcta
[502970] Checker: Error el dato de la transacción no calza con el esperado pkt=0141a4a016 Retardo=0 trgt_row=4 trgt_col=5 mode=0 pyId=1a4016 Terminal_env=0 Tiempo_env=0 Terminal_rcb=15 Tiempo_rcb=502955
[502980] Checker: El Paquete: 013534a412a ha llegado a su destino por la ruta correcta
[503000] Checker: El Paquete: 011592f5f8 ha llegado a su destino por la ruta correcta
[503000] Checker: Error el dato de la transacción no calza con el esperado pkt=024057a962 Retardo=0 trgt_row=5 trgt_col=3 mode=0 pyId=4ab5bf Terminal_env=0 Tiempo_env=0 Terminal_rcb=10 Tiempo_rcb=502985
```

Fig. 12: Error en la transmisión de algunas de las transacciones utilizadas para llenar FIFOs de entrada.

6. Bugs encontrados

Como se mencionó anteriormente, durante las pruebas se encontraron algunos bugs, que consisten en que el paquete se pierde en su proceso de transmisión por el DUT o que el dato se duplica y se recibe dos veces en el mismo terminal, sin embargo, de 500 transacciones, este error puede suceder por lo menos dos veces, una forma de evitar que esto suceda es aumentando el retardo mínimo de envío de los paquetes, es decir, darle más tiempo de procesamiento al DUT, los errores se pueden apreciar en las Figs. 13 y 14.

```
[502650] Checker: El Paquete: 01359cfd37 ha llegado a su destino por la ruta correcta
[502720] Checker Error: el Paquete: 0253588772 recibido en ID: 33 nunca fue generado/se ha duplicado
[502720] Checker: El Paquete: 0145a34211 ha llegado a su destino por la ruta correcta
[502730] Checker Error: el Paquete: 0253588772 recibido en ID: 43 nunca fue generado/se ha duplicado
```

Fig. 13: Bug de dato perdido

```
[502325] Checker: Error el dato de la transacción no calza con el esperado pkt=0330ae1c11
```

Fig. 14: Bug de dato que jamás llega a su destino

7. Ancho de banda y retraso promedio de la transmisión

A partir de los informes que genera el Scoreboard con cada prueba, se llevaron a cabo diversas pruebas con distinta profundidad de FIFOs, sin embargo, manteniendo constantemente el ancho de palabra en 40 bits. Por medio de los archivos .csv se generaron los gráficos de las Figs. 15, 16, 17 y 18.

La Fig. 15 corresponde a la tasa de bits máxima en la transmisión de paquetes de 40 bits con respecto a la profundidad de las FIFOs. Se observa que para profundidades mayores a 19, al ancho de banda máximo se mantiene constante en 666,67 kbps, así mismo el valor máximo se alcanza con FIFOs de profundidad de 18, siendo este de 1333,33 kbps aproximadamente.

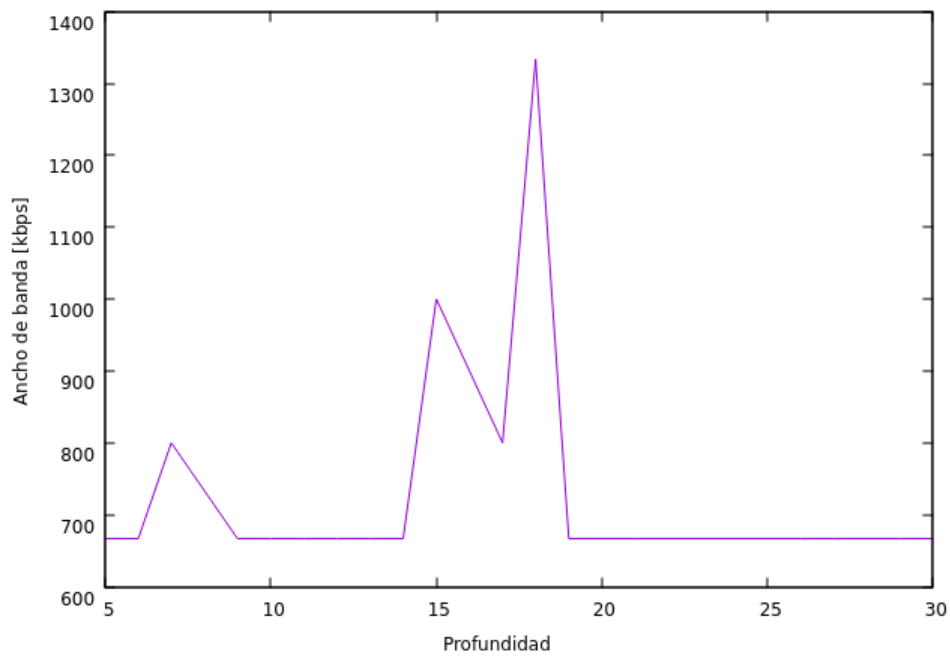


Fig. 15: Ancho de banda máximo

Con respecto al ancho de banda mínimo de la transmisión de 40 bits, se muestra en la Fig. 16 que dicho valor decrece conforme se va aumentando la profundidad de las FIFOs del DUT. Lo cual tiene sentido, ya que las FIFOs pueden almacenar más datos y por ende crece la latencia de la transmisión de los paquetes. Esta misma tendencia se puede ver reflejada en la Fig. 17, cuya gráfica corresponde al ancho de banda promedio de las transmisiones.

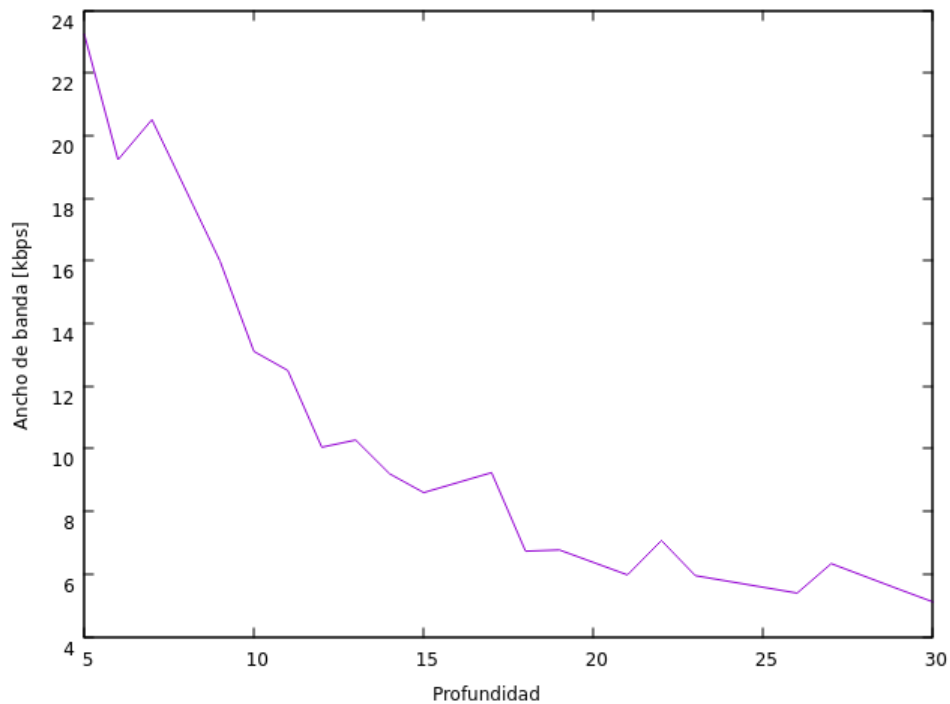


Fig. 16: Ancho de banda mínimo

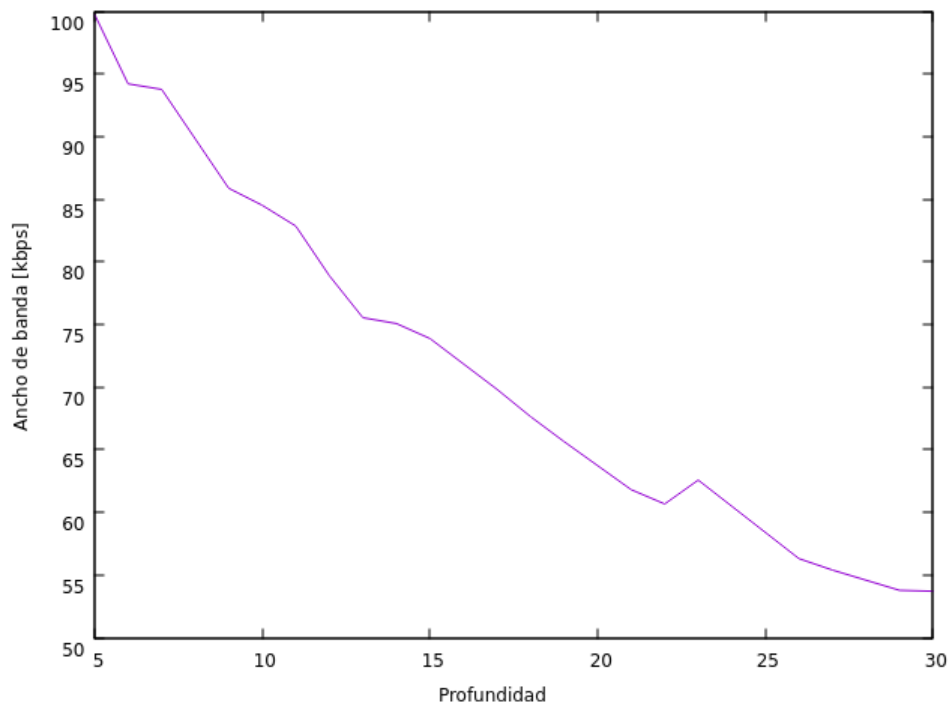


Fig. 17: Ancho de banda promedio

En la Fig. 18 se puede apreciar como al aumentar la profundidad de las FIFOs la transacción tarda más en llegar su destino, esto se debe a que deberá esperar en la cola por más tiempo hasta que los otros paquetes que se encuentran por delante sean enviados.

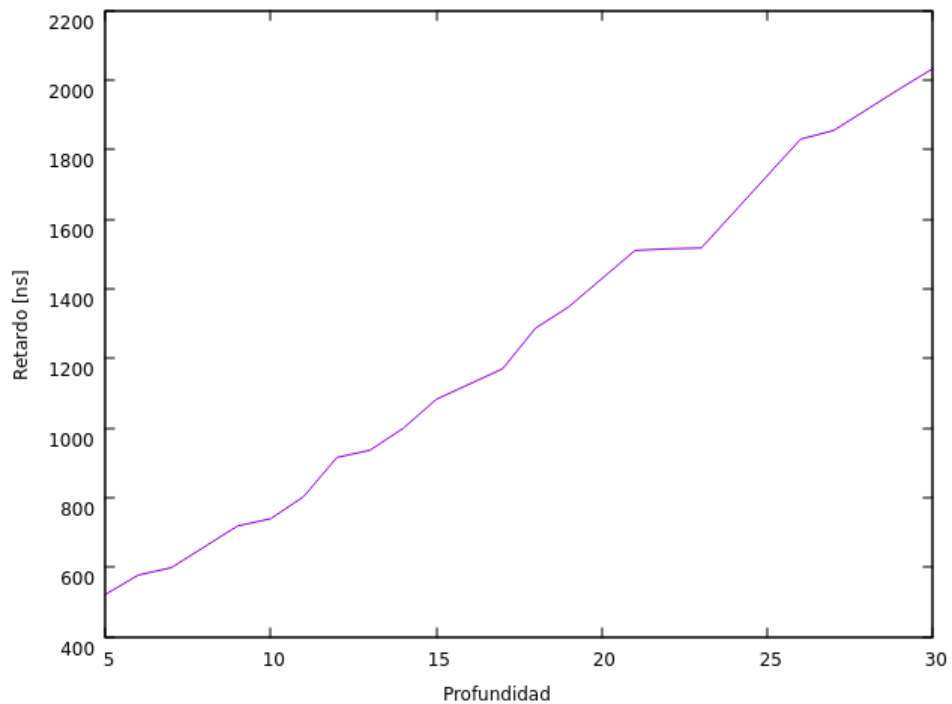


Fig. 18: Retardo promedio

8. Cobertura Funcional

Para la medición de la cobertura de las pruebas se definió un grupo de cobertura que se activa en el flanco de subida de la señal **popin** del DUT. En el grupo de cobertura se incluyó un *coverpoint*, el cual se utiliza para medir la cobertura de los *id* de los routers internos del DUT. Dentro de *coverpoint*, se definieron dos bins (contenedores) uno que representan valores específicos de los routers internos, con esto se logró rastrear cuántas veces se alcanzan estos valores durante la simulación.

El otro bin consiste en un *ignore_bins* que utiliza una función para ignorar ciertos bins de *coverpoint* en función de los valores del router id. Esto con el fin no tomar en cuenta para el *coverage* los posibles envíos a sí mismos, ya que esto se evita en las pruebas, por lo que si no se ignoran estos casos el *coverage* se vería influenciado.

En las Figs. 19 y 20 se observan los resultados respecto al *coverage* de los routers internos del DUT.

```
Coverage = 100.00 %, router: 11, term: 00
Coverage = 37.50 %, router: 11, term: 01
Coverage = 37.50 %, router: 11, term: 02
Coverage = 100.00 %, router: 11, term: 03
Coverage = 100.00 %, router: 12, term: 00
Coverage = 50.00 %, router: 12, term: 01
Coverage = 37.50 %, router: 12, term: 02
Coverage = 62.50 %, router: 12, term: 03
Coverage = 100.00 %, router: 13, term: 00
Coverage = 62.50 %, router: 13, term: 01
Coverage = 37.50 %, router: 13, term: 02
Coverage = 50.00 %, router: 13, term: 03
Coverage = 100.00 %, router: 14, term: 00
Coverage = 100.00 %, router: 14, term: 01
Coverage = 37.50 %, router: 14, term: 02
Coverage = 37.50 %, router: 14, term: 03
Coverage = 62.50 %, router: 21, term: 00
Coverage = 37.50 %, router: 21, term: 01
Coverage = 50.00 %, router: 21, term: 02
Coverage = 100.00 %, router: 21, term: 03
Coverage = 62.50 %, router: 22, term: 00
Coverage = 50.00 %, router: 22, term: 01
Coverage = 50.00 %, router: 22, term: 02
Coverage = 62.50 %, router: 22, term: 03
Coverage = 62.50 %, router: 23, term: 00
Coverage = 62.50 %, router: 23, term: 01
Coverage = 50.00 %, router: 23, term: 02
Coverage = 50.00 %, router: 23, term: 03
Coverage = 62.50 %, router: 24, term: 00
Coverage = 100.00 %, router: 24, term: 01
Coverage = 50.00 %, router: 24, term: 02
Coverage = 37.50 %, router: 24, term: 03
```

Fig. 19: Resultados del *coverage*
(a)

```
Coverage = 50.00 %, router: 31, term: 00
Coverage = 37.50 %, router: 31, term: 01
Coverage = 62.50 %, router: 31, term: 02
Coverage = 100.00 %, router: 31, term: 03
Coverage = 50.00 %, router: 32, term: 00
Coverage = 50.00 %, router: 32, term: 01
Coverage = 62.50 %, router: 32, term: 02
Coverage = 62.50 %, router: 32, term: 03
Coverage = 50.00 %, router: 33, term: 00
Coverage = 62.50 %, router: 33, term: 01
Coverage = 62.50 %, router: 33, term: 02
Coverage = 50.00 %, router: 33, term: 03
Coverage = 50.00 %, router: 34, term: 00
Coverage = 100.00 %, router: 34, term: 01
Coverage = 62.50 %, router: 34, term: 02
Coverage = 37.50 %, router: 34, term: 03
Coverage = 37.50 %, router: 41, term: 00
Coverage = 37.50 %, router: 41, term: 01
Coverage = 100.00 %, router: 41, term: 02
Coverage = 100.00 %, router: 41, term: 03
Coverage = 37.50 %, router: 42, term: 00
Coverage = 50.00 %, router: 42, term: 01
Coverage = 100.00 %, router: 42, term: 02
Coverage = 62.50 %, router: 42, term: 03
Coverage = 37.50 %, router: 43, term: 00
Coverage = 62.50 %, router: 43, term: 01
Coverage = 100.00 %, router: 43, term: 02
Coverage = 50.00 %, router: 43, term: 03
Coverage = 37.50 %, router: 44, term: 00
Coverage = 100.00 %, router: 44, term: 01
Coverage = 100.00 %, router: 44, term: 02
Coverage = 37.50 %, router: 44, term: 03
```

Fig. 20: Resultados del *coverage*
(b)

9. Conclusiones

El entorno de verificación elaborado muestra ser capaz de ejercitar las capacidades del DUT de la manera más eficiente posible. Asimismo, el checker creado es capaz de calcular y hacer seguimiento de la ruta que debe seguir cada paquete a partir de su fuente, destino y modo, lo que colaboró en la búsqueda de los errores del diseño.

El DUT es capaz de transmitir un volumen alto de datos, sin embargo, es necesario establecer un retardo mínimo que le permita transmitir cada uno de los datos sin que alguno se pierda en el proceso, ya que internamente no usa máquinas de estados, por lo que cada dato necesita un periodo en el cual pueda ser recibido y transmitido sin interferencia de alguno otro dato.

Con el *coverage* que se muestra en la Fig. 19 y 20 se concluye que al menos todos los routers id que dan hacia alguna terminal externa tienen una cobertura funcional del 100 %, mientras que los routers id internos que no dan hacia alguna terminal externa tiene una cobertura menor al 100 % dado que no todas las rutas provocan un **popin** en estos routers.