



Surveillance réseau sur NetFPGA

Benoit Fontaine

Tristan Groléat

Franziska Hubert

Présentation orale, 1er mars 2010





Plan

- 1 Introduction
- 2 Transformation du NetFPGA en hub Ethernet
- 3 Détection des paquets TCP SYN
- 4 Gestion des registres du NetFPGA
- 5 Conclusion



Plan

Introduction

- 1 Introduction
- 2 Transformation du NetFPGA en hub Ethernet
- 3 Détection des paquets TCP SYN
- 4 Gestion des registres du NetFPGA
- 5 Conclusion



La surveillance d'un trafic qui explose

Introduction ► Contexte

Le trafic en cœur de réseau ne cesse de croître :

- **Nouvelles applications** : peer-to-peer, streaming vidéo (Youtube, Dailymotion)
- Actuellement : 10Mb/s, 100Mb/s (Fast Ethernet), 1Gb/s (Gigabit Ethernet) et 10 Gbit/s
- À venir : Ethernet à 40Gb/s, 100Gb/s

La surveillance de trafic :

- Pour la **sécurité**
- Pour faire des **statistiques**
- Pour la **facturation...**



Le matériel dont nous disposons

Introduction ► Contexte

- Un ordinateur puissant (sous Ubuntu) avec :
 - **La carte NetFPGA** sur le port PCI
 - **Deux cartes réseau** Gigabit Ethernet
- Un deuxième ordinateur moins puissant (sous un liveCD Fedora) avec **une carte réseau** Gigabit Ethernet





Accélération de surveillance de trafic en utilisant une carte NetFPGA :

- Prise en main du NetFPGA
- Compréhension de l'architecture du NetFPGA
 - la composition des modules : le hub éthernet
 - la création et l'accès aux registres : détection de paquets SYN
- Implémentation d'un algorithme de surveillance



Plan

Transformation du NetFPGA en hub Ethernet

- 1 Introduction
- 2 Transformation du NetFPGA en hub Ethernet**
- 3 Détection des paquets TCP SYN
- 4 Gestion des registres du NetFPGA
- 5 Conclusion



Organisation en modules

Transformation du NetFPGA en hub Ethernet

- Séquence de **modules indépendants**
- Les données circulent entre les modules comme un **pipeline**
- Les modules sont reliés par les signaux suivants :
 - Data (64 bits)
 - Control (8 bits)
 - Write
 - Ready
- Transfert d'information entre modules en ajoutant des **headers** aux paquets
- Le flot de données peut-être **stoppé**

Organisation en modules

Transformation du NetFPGA en hub Ethernet

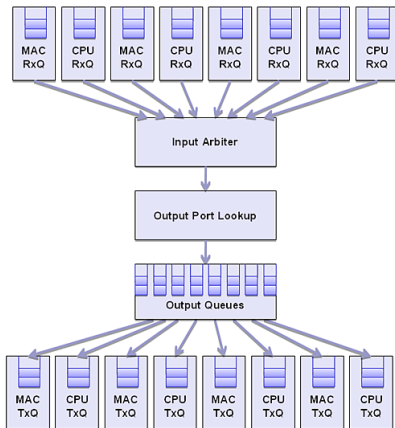


FIG.: Modules du Reference NIC



Du Reference NIC au Hub Ethernet

Transformation du NetFPGA en hub Ethernet

- On garde l'architecture du Reference NIC
- Mais on modifie le module **Output Port Lookup**



Le module Output Port Lookup

Transformation du NetFPGA en hub Ethernet

- Décide par quelles interfaces les paquets sortent
- **Utilisation** de l'en-tête créé par le module **Input Queue** pour déterminer l'**interface d'origine** des paquets
- **Modification** de cet en-tête pour définir l'**interface de sortie**

Bits	Purpose
15 :0	Packet length in bytes
31 :16	Input port
47 :32	Packet length in words
63 :48	Output port.



Plan

Détection des paquets TCP SYN

- 1 Introduction
- 2 Transformation du NetFPGA en hub Ethernet
- 3 Détection des paquets TCP SYN**
- 4 Gestion des registres du NetFPGA
- 5 Conclusion



Qu'est-ce qu'un paquet TCP SYN ?

Détection des paquets TCP SYN

Un paquet **TCP** :

- Un protocole pour fiabiliser les communications IP
- Pile classique de protocoles : **TCP/IP/Ethernet**

De type **SYN** :

- Sert à initialiser une connexion TCP
- Impose au destinataire d'allouer un espace mémoire à la connexion
- Un SYN envoyé par chaque machine au début d'une connexion



Comment identifier un paquet TCP SYN ?

Détection des paquets TCP SYN

1. On sait qu'on est sur Ethernet
2. On lit le champ "Ethertype" de l'en-tête Ethernet : 0x800 pour IPv4
3. On lit le champ "Protocole" de l'en-tête IPv4 : 6 pour TCP
4. On lit le 5ème drapeau (SYN) de l'en-tête TCP : 1 pour un paquet SYN

Comment identifier un paquet TCP SYN ?

Détection des paquets TCP SYN

1. On sait qu'on est sur Ethernet
2. On lit le champ "Ethertype" de l'en-tête Ethernet : 0x800 pour IPv4

En octets

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14 ... 1513	1514	1515	1516	1517
Adresse MAC destination						Adresse MAC source						Type de protocole		Données		FCS/CRC		

3. On lit le champ "Protocole" de l'en-tête IPv4 : 6 pour TCP
4. On lit le 5ème drapeau (SYN) de l'en-tête TCP : 1 pour un paquet SYN

Comment identifier un paquet TCP SYN ?

Détection des paquets TCP SYN

1. On sait qu'on est sur Ethernet
2. On lit le champ "Ethertype" de l'en-tête Ethernet : 0x800 pour IPv4
3. On lit le champ "Protocole" de l'en-tête IPv4 : 6 pour TCP

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Version d'IP		Longueur de l'en-tête (en mots de 32 bits)				Type de service				Longueur totale en octets																					
Identification (pour les fragments)										Flags (pour les fragments)				Fragment offset																	
Durée de vie (TTL Time To Live)						Protocole				Somme de contrôle de l'en-tête																					
Adresse source																															
Adresse destination																															
Option(s) + bourrage																															

4. On lit le 5ème drapeau (SYN) de l'en-tête TCP : 1 pour un paquet SYN

Comment identifier un paquet TCP SYN ?

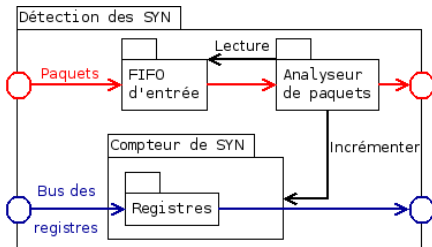
Détection des paquets TCP SYN

1. On sait qu'on est sur Ethernet
2. On lit le champ "Ethertype" de l'en-tête Ethernet : 0x800 pour IPv4
3. On lit le champ "Protocole" de l'en-tête IPv4 : 6 pour TCP
4. On lit le 5ème drapeau (SYN) de l'en-tête TCP : 1 pour un paquet SYN

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Port Source 2 octets																Port destination 2 octets															
Numéro de séquence																															
Numéro d'acquittement																															
Taille de l'en-tête		réservé		ECN		URG		ACK		PSH		RST		SYN		FIN		Fenêtre													
Somme de contrôle																Pointeur de données urgentes															
Options																								Remplissage							
Données																															

La réalisation matérielle de la détection

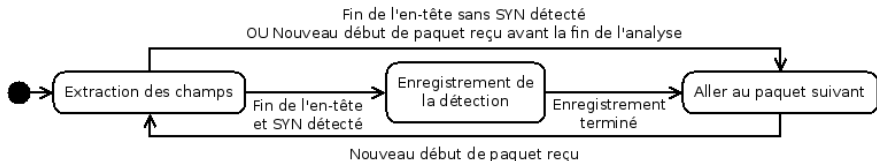
Détection des paquets TCP SYN



- FIFO d'entrée : `small_fifo` définie par NetFPGA
- Registres : `generic_register` défini par NetFPGA

Comportement de l'analyseur de paquets

Détection des paquets TCP SYN



- En-tête parcourue grâce à un compteur de mots de 64 bits (taille du bus)
- Début du paquet : donné par un signal de contrôle du bus
- Enregistrement actuel : incrémentation d'un compteur
- FIFO d'entrée fermée pendant l'enregistrement



Plan

Gestion des registres du NetFPGA

- 1 Introduction
- 2 Transformation du NetFPGA en hub Ethernet
- 3 Détection des paquets TCP SYN
- 4 Gestion des registres du NetFPGA**
- 5 Conclusion



La gestion des registres

Gestion des registres du NetFPGA

Pour chaque projet :

- Écrire les parties en Verilog
- Définir les registres qui leur sont associés
- Insertion des nouveaux modules dans le user data path

Le système de registres du NetFPGA est **basé sur XML** :

- Déclaration des registres
- Allocation automatique des adresses



La gestion des registres

Gestion des registres du NetFPGA

Pour chaque projet :

- Écrire les parties en Verilog
- Définir les registres qui leur sont associés
- Insertion des nouveaux modules dans le user data path

Le système de registres du NetFPGA est **basé sur XML** :

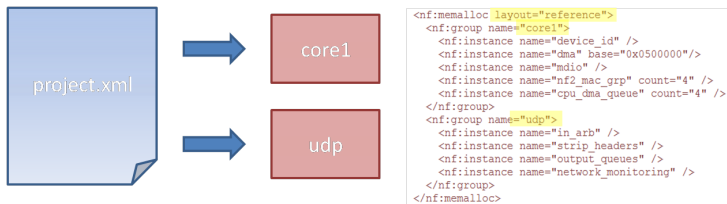
- Déclaration des registres
- Allocation automatique des adresses

La gestion des adresses est basée sur XML

Gestion des registres du NetFPGA

Deux tâches principales :

- Définition des bibliothèques utilisées
- Définition du "layout" des registres



Le nouveau module - définition XML

Gestion des registres du NetFPGA

project.xml

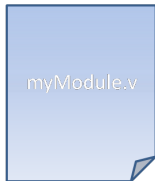
myModule.xml

```
<nf:registers>
  <nf:register>
    <nf:name>counter</nf:name>
    <nf:description>Number of packets dropped</nf:description>
    <nf:type>counter32</nf:type>
  </nf:register>
  <nf:register>
    <nf:name>packet_len_field</nf:name>
    <nf:description>Last packet_len_field value</nf:description>
    <nf:type>generic_hardware32</nf:type>
  </nf:register>
</nf:registers>
```

Le nouveau module - définition Verilog

Gestion des registres du NetFPGA

- Définition des nombres et des types de registres
- Mapping des entrées et sorties



```
generic regs
#(.UDP_REG_SRC_WIDTH (UDP_REG_SRC_WIDTH),
 .TAG (`NET_MON_BLOCK_ADDR),
 .REG_ADDR_WIDTH (`NET_MON_REG_ADDR_WIDTH),
 .NUM_COUNTERS (1),
 .NUM_SOFTWARE_REGS (0),
 .NUM_HARDWARE_REGS (13),
 .COUNTER_INPUT_WIDTH (1))
```

```
.hardware_regs ((step2[63:32],
                 step2[31:0],
                 step1[63:32],
                 step1[31:0],
                 step0[63:32],
                 step0[31:0],
                 16'h0, tcp_dst_port,
                 16'h0, tcp_src_port,
                 ipdst,
                 ipsrc,
                 24'h0, tcpflags,
                 24'h0, protocol_field,
                 16'h0, packet_len_field)),
```



Plan

Conclusion

- 1 Introduction
- 2 Transformation du NetFPGA en hub Ethernet
- 3 Détection des paquets TCP SYN
- 4 Gestion des registres du NetFPGA
- 5 Conclusion**



Objectifs atteints

Conclusion

- Prendre en main le NetFPGA
 - **Environnement de développement** installé
 - Étapes d'installation consultables sur le **wiki** et le **SVN**
 - **Station de travail** utilisable pour de futurs projets
- Analyser et utiliser l'architecture du NetFPGA
 - **Analyse** des modules existants
 - **Création et insertion** de nouveaux modules
 - **Documentation** sur le wiki
- Implémenter un algorithme de surveillance de trafic
 - Détection de **SYN flooding**



Objectifs atteints

Conclusion

- Prendre en main le NetFPGA
 - **Environnement de développement** installé
 - Étapes d'installation consultables sur le **wiki** et le **SVN**
 - **Station de travail** utilisable pour de futurs projets
- Analyser et utiliser l'architecture du NetFPGA
 - **Analyse** des modules existants
 - **Création et insertion** de nouveaux modules
 - **Documentation** sur le wiki
- Implémenter un algorithme de surveillance de trafic
 - Détection de **SYN flooding**



Objectifs atteints

Conclusion

- Prendre en main le NetFPGA
 - **Environnement de développement** installé
 - Étapes d'installation consultables sur le **wiki** et le **SVN**
 - **Station de travail** utilisable pour de futurs projets
- Analyser et utiliser l'architecture du NetFPGA
 - **Analyse** des modules existants
 - **Création et insertion** de nouveaux modules
 - **Documentation** sur le wiki
- Implémenter un algorithme de surveillance de trafic
 - Détection de **SYN flooding**



Objectifs à accomplir

Conclusion

- Implémentation de l'**algorithme CMS**
- **Frontend** de détection d'attaques SYN flooding



Acquis

Conclusion

- **Expérience** en système embarqué
- Utilisation d'un **FPGA** dans un environnement complexe
- Utilisation de la suite d'outils **Xilinx**
- Apprentissage du langage **Verilog**



Bénéfices pour l'école

Conclusion

- **Phase d'installation** maîtrisée
- **Phase de développement** maîtrisée
- Possibilité d'implémenter des **fonctionnalités poussées**
- **Grand catalogue de connaissances** sur notre wiki constamment mis à jour



Questions

Conclusion

