
RAPPORT DE PROJET

SIMULATEUR RESEAU

Du 11 Avril au 12 Mai 2016



Tuteur : M. REDON Xavier

DACOSTA NETO Charlène, IMA4

Année scolaire 2015-2016

SOMMAIRE

REMERCIEMENTS.....	3
INTRODUCTION.....	4
I – LA PRESENTATION DU PROJET	5
A-LE SCENARIO ROUTAGE.....	5
B- LE ROUTAGE COMMUTATEUR.....	6
II- LE TRAVAIL REALISE	7
A-LA REPRESENTATION DES PAQUETS	7
B-LA CREATION DE SCENARIOS.....	7
C-LE DOM HTML.....	8
CONCLUSION.....	10
ANNEXES.....	11

REMERCIEMENTS

Je tiens à remercier tout l'ensemble des tuteurs de projet pour leur disponibilité. Plus particulièrement, je remercie M. Redon Xavier, pour m'avoir permis de reprendre un nouveau projet.

Un merci à M. Rojo, directeur de département IMA qui est toujours à l'écoute de ses étudiants.

Pour finir, je remercie les responsables de notre formation nous avoir encadré.

INTRODUCTION

Au cours de ma 2^{ème} année de cycle ingénieur en Informatique Microélectronique Automatique, je devais réaliser un projet avec un collègue, qui n'a pas forcément abouti cause de problèmes personnels. On m'a alors confié un mini projet sur lequel j'ai bien travaillé et qui m'a permis de découvrir d'autres outils.

Ce projet, Simulateur Réseau, s'est fait du 11 Avril au 12 Mai ; contrairement aux autres qui avaient tout le semestre. Ce dernier montre le fonctionnement de la commutation Ethernet et du routage IPv4.

Au cours de ce projet quelques missions m'ont été confiées, lesquelles n'ont pas toutes été validées. J'ai tout d'abord pris un moment pour comprendre la structure des paquets, le fonctionnement des différents scénarios et enfin « appris » en quelque sorte le JavaScript ainsi que le Html qui n'étaient pas dans le programme.

Il a fallu d'abord, différencier les paquets Ethernet des paquets Ip ; ensuite vérifier que la page HTML était bien sur la dernière version et pour finir pouvoir permettre à un utilisateur de créer ses propres scénarios sans taper du code.

Dans un premier temps, je présenterai le projet avec les différents scénarios. Ensuite j'expliquerai les missions sur lesquelles j'ai travaillé ainsi qu'une légère explication du DOM Html. Enfin je vous soumettrai une conclusion globale de mon projet en passant par les péripéties.

I - LA PRESENTATION DU PROJET

A-LE SCENARIO ROUTAGE

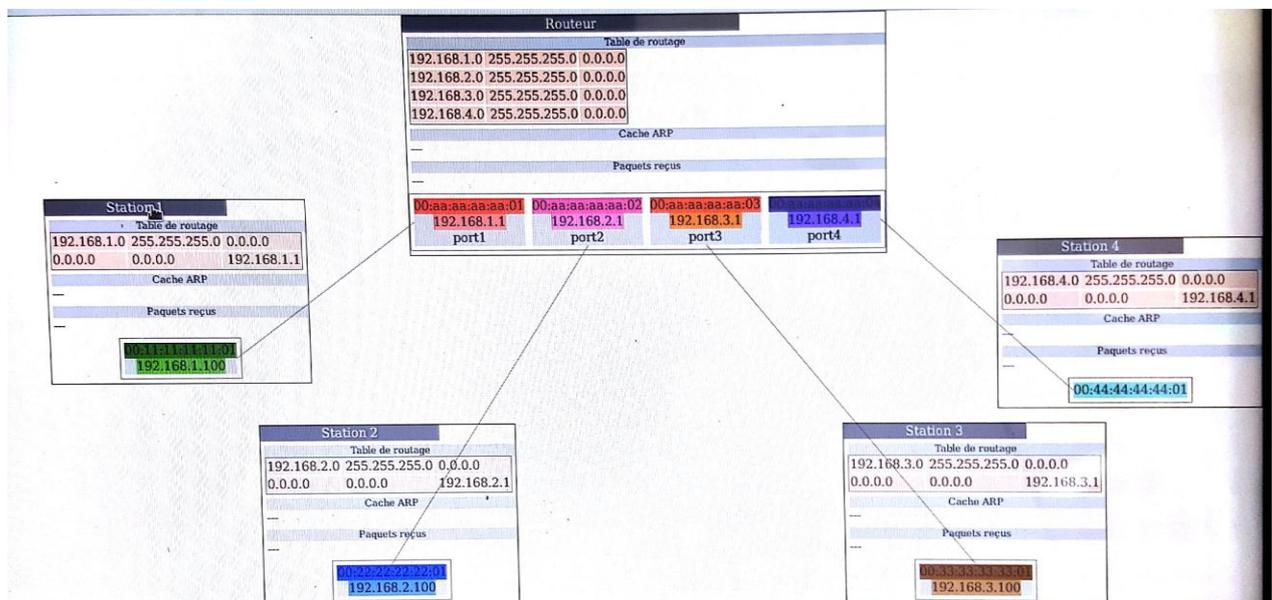
Au niveau du cache ARP, il y a une correspondance entre une adresse MAC et une adresse IP. Lorsqu'un paquet est envoyé dans le scénario 1, il passe systématiquement par le port relié à la station, si ce n'est pas la bonne cible le paquet est détruit. Donc les échanges ne peuvent se faire que s'il y a une liaison.

Le port du routeur étant sur le même réseau va envoyer un paquet à la station dont il connaît l'adresse IPv4 de destination, afin de demander son adresse MAC qui répond.

Le cache ARP est alors mis à jour avec l'adresse MAC et l'adresse IP de la station de destination.

La table de routage elle est une table de correspondance entre l'adresse de la station visée et le nœud suivant auquel le routeur doit délivrer le message.

Grâce à cette table, le routeur connaissant l'adresse du destinataire Ipv4 encapsulée dans le message va être capable de savoir sur quelle interface envoyer le message. Cela revient à savoir quelle carte réseau utiliser et à quel routeur, directement accessible sur un de ses réseaux locaux auquel cette carte est connectée, remettre le paquet.

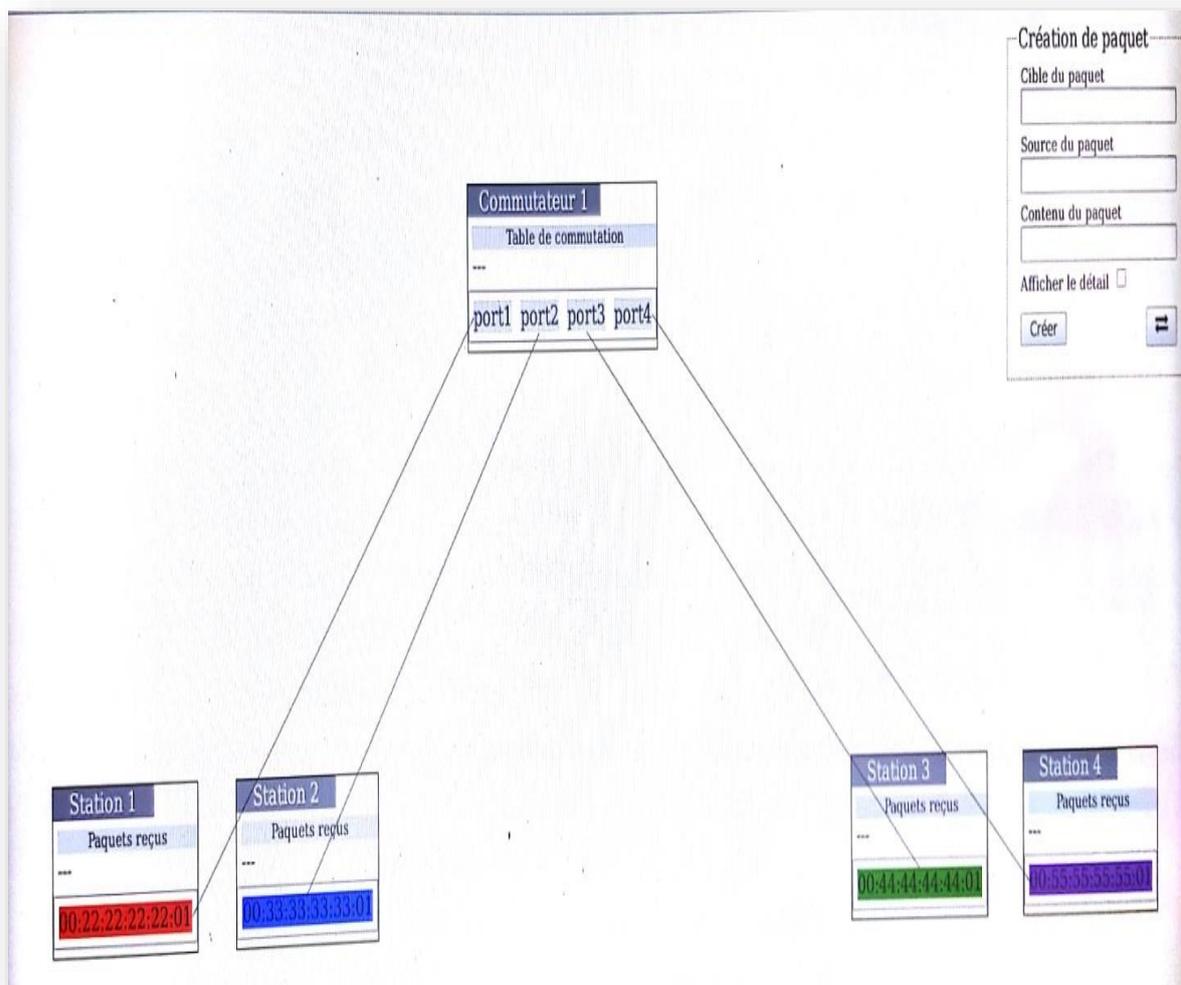


B- LE ROUTAGE COMMUTATEUR

Ce scénario permet de voir le fonctionnement du routage Ipv4. La table de commutation affiche l'adresse Ethernet source du paquet, son port et une durée de vie ; le commutateur apprend où se trouve la station source.

La station correspondante envoie au commutateur le paquet qui l'envoie à son tour aux autres stations et dépose le paquet à la station ayant pour adresse l'adresse de destination correspondante.

Si après avoir fait un premier envoi d'un paquet, on inverse destination et source, le commutateur ne diffuse plus à toutes les stations car connaissant déjà l'emplacement de la destination.



II- LE TRAVAIL REALISE

A-LA REPRESENTATION DES PAQUETS

Le but était ici de comprendre la structure d'un paquet, déjà par le CSS qui est la partie du code pour la mise en forme de la page Html. Après des simulations sur un scénario, je me suis aperçu que les paquets Ethernet n'étaient pas différenciables des paquets Ip.

Pour aboutir à ce résultat j'ai modifié des paramètres comme la hauteur des paquets et la bordure.

La deuxième étape était de vérifier si la page était compatible HTML5.

B-LA CREATION DE SCENARIOS

Cette étape devrait permettre à un utilisateur de créer son propre scénario sans pour autant accéder à l'interface du code.

Le client devient alors autonome (mode client).

J'ai créé un formulaire servant à l'utilisateur de rentrer le type de bloc qu'il veut ajouter, le nom et un bouton ajouter qui va appeler 3 sous codes (voir annexe) selon le type entré.

Ce formulaire est écrit dans un nouveau fichier et on le charge par la suite dans un scénario vide grâce à « load ».

Voici un exemple avec l'ajout de ports sur un commutateur.

SEYE		
Table de commutation		

Port1	Port2	Port3
3	Port	Port

Le bouton « Port » va créer des ports supplémentaires. Et l'utilisateur choisi le numéro du port

C-LE DOM HTML

```
function add_network_appliance(form){
var type=form.children("select[name='type']");
var type_value=type.val();
var name=form.children("input[name='nom']");
var name_value=name.val();
var element_choisi;

// Dans le cas ou il s'agit d'un commutateur
if(type_value=="c"){

element_choisi="<div class='commutateur' ";
element_choisi+="style='top: 100px; left: 500px'> ";
element_choisi+="<div class='nom'>"+name_value+"</div> ";
element_choisi+="<div class='cadre_titre' onClick='table_toggle($('#commutateur1_mem'))';>Table de commutation</div> ";
element_choisi+="<div class='memcom fantome' id='commutateur1_mem'></div> ";
element_choisi+="<div class='ports' id='commutateur1_ports'> </div>";
element_choisi+="<input type= 'text'> <div class='buttonAjouts' <button class='formulaire_bouton' type='button' onClick = 'ajouterPort(this)';>Port</button></div>";
element_choisi+="</div>";
}
```

Voilà une partie des codes que j'ai pu écrire de la fonction de création des éléments. Cette fonction, `add_network_appliance ()`, va permettre en fonction du type choisi par l'utilisateur de pouvoir les ajouter. L'outil « `input` » est là pour récupérer le texte saisi par l'utilisateur au niveau du formulaire.

Sur l'exemple, je montre le cas du commutateur ; le `type_value` récupéré du choix de l'utilisateur est le « `c` » comme configuré dans le code du formulaire. Des `div` sont alors créés, notamment celui du bouton qui va appeler une fonction `AjouterPort(this)` lors d'un simple clic.

```

//Cette fonction va permettre de rajouter un Port dans le device selectionné
function ajouterPort(objectTriger){

    //on récupere la valeur choisie par l'utilisateur
    var valueDuPort = objectTriger.previousElementSibling.value;
    //On recupere l'element qui va contenir les ports ajoutés par l'utilisateur.
    var EllesPorts = objectTriger.previousElementSibling.previousElementSibling;
    //On va rajouter un element port
    var divPort = document.createElement("div");
    divPort.setAttribute("class", "port");
    divPort.setAttribute("id", "commutateur"+valueDuPort)
    EllesPorts.appendChild(divPort);

    var divNomint = document.createElement("div");
    divNomint.setAttribute("class", "nomint");
    divNomint.innerHTML = "Port"+valueDuPort;
    divPort.appendChild(divNomint);
}

```

La fonction nous permettra de rajouter des ports grâce à l'objet déclencheur qui est ici le simple clic.

Le dom Html est un ensemble d'outils déjà faits qui nous facilitent la manipulation des pages. Avec 'previousElementSibling' on peut remonter à l'élément précédent. Sur l'exemple on va récupérer la valeur choisie par l'utilisateur ainsi que l'élément qui va contenir les ports ajoutés par l'utilisateur.

Par la suite, grâce à « createElement » on va générer un div et lui attribuer des attributs tels que la classe et l'id qui sera le nom concaténé à la valeur choisie.

Un second div est créé la pour créer le port en lui-même.

Avec l'outil « appendChild », on peut alors ajouter le nœud divNomint à l'élément divPort.

CONCLUSION

Ce projet bien que court, m'a permis d'avoir quelques bases sur le Html et le JavaScript.

Avec plus de temps, j'aura peut-être pu ramener de meilleurs résultats et même arrivé au bout des perspectives.

Je remercie quand même mes encadrants de m'avoir permis de travailler sur un autre projet même si cela ne suffira pas à valider mon année ; je n'en sors que forgée.

ANNEXES

CODE DU FORMULAIRE

```
<html
>

  <head>

    <meta http-equiv="content-type" content="text/html; charset=utf-8" />

    <title>Formulaire utilisateur</title>

    <meta charset="UTF-8">

    <link href="reseau.css" rel="stylesheet" type="text/css">

    <link href="jquery-ui/jquery-ui.css" rel="stylesheet" type="text/css">

  </head>

  <body>

    <script type="text/javascript" src="jquery.js"></script>

    <script type="text/javascript" src="jquery-ui/jquery-ui.js"></script>

    <script type="text/javascript" src="reseau.js"></script>

    <script type="text/javascript" src="Reseau_dyn.js"></script>

    <fieldset id="form_dyn" class="formulaire_creation" style="top: 0px; left: 0px;" >

      <legend class="formulaire_titre">Ajout d'element reseau</legend>

      <label class="formulaire_label" for="element_type">Type d'element</label>

      <select class="formulaire_menu" id="element_type" name="type">

        <option value="r">routeur</option>

        <option value="c">commutateur</option>

        <option value="s">station</option>

      </select>

      <label class="formulaire_label" for="element_nom">Nom de l'element</label>
```

```
<input class="formulaire_champ" id="element_nom" type="text" name="nom"/>
</label>
<div class="formulaire_div">
<button class="formulaire_bouton" type="button"
onClick="add_network_appliance($('#form_dyn'))">Ajouter</button>
</div>
</fieldset>
```

CODE SCENARIO

```
<!DOCTYPE
html>
```

```
<html>
<head>
<title>scenario_util</title>
<meta charset="UTF-8">
<link href="reseau.css" rel="stylesheet" type="text/css">
<link href="jquery-ui/jquery-ui.css" rel="stylesheet" type="text/css">
</head>
<body onLoad="init(); main();">
<script src="jquery.js"></script>
<script src="jquery-ui/jquery-ui.js"></script>
<script src="reseau.js"></script>
<script src="Reseau_dyn.js"></script>
```

```
<script>

function init(){

$("#formul_util").load("formul_util.html");

$("#formip4").load("formip4.html");

elements=[ ];

connexions=[

];

}

</script>

<div id="formul_util"></div>

<div id="formip4"></div>

</body>

</html>
```

CODE RESEAU_DYN

```
function add_network_appliance(form){
var type=form.children("select[name='type']");
var type_value=type.val();
var name=form.children("input[name='nom']");
var name_value=name.val();
var element_choisi;

// Dans le cas ou il s'agit d'un commutateur
if(type_value=="c"){

element_choisi= "<div class='commutateur' ";
element_choisi+="style='top: 100px; left: 500px'> ";
element_choisi+="<div class='nom'>"+name_value+"</div> ";
element_choisi+="<div class='cadre_titre' onClick='table_toggle($('#commutateur1_mem'))';>Table de
commutation</div> ";
element_choisi+="<div class='memcom fantome' id='commutateur1_mem'></div> ";
element_choisi+="<div class='ports' id='commutateur1_ports'> </div>";
element_choisi+="<input type= 'text'> <div class='buttonAjouts' <button class='formulaire_bouton'
type='button' onClick = 'ajouterPort(this);'>Port</button></div>";
element_choisi+="</div>";

    }

else if (type_value=="r"){
    element_choisi=('<div class="routeur" style="top: 0px; left: 600px;" /> <div
class="nom">'+name_value+'</div> <div class="ports"> <div class="port" > <div class="ethport"
style="background-color: red;" type="text"></div> <div class="ip4port" style="background-color:
```

```

salmon;"type="text"></div> <div class="nomint">port1</div> </div> </div>');

    }

else if(type_value=="s"){
    element_choisi=(' <div class="station" style="bottom: 250px; left: 230px;"> <div
class="nom">'+name_value+'</div> </div>');

    }

else{
}

$(element_choisi).uniqueId();

$('body').append($(element_choisi));

//On va ajouter l'element inseré dans la liste d'elements
elements = [];

elements.push($(element_choisi));

//On appelle ensuite la fonction main
}

//Cette fonction va permettre de rajouter un Port dans le device selectionné
function ajouterPort(objectTriger){

    //on récupere la valeur choisie par l'utilisateur
    var valueDuPort = objectTriger.previousElementSibling.value;
    //On recupere l'element qui va contenir les ports ajoutés par l'utilisateur.
    var EllesPorts = objectTriger.previousElementSibling.previousElementSibling;

```

```
//On va rajouter un element port
var divPort = document.createElement("div");
divPort.setAttribute("class", "port");
divPort.setAttribute("id", "commutateur"+valueDuPort)
EllesPorts.appendChild(divPort);

var divNomint = document.createElement("div");
divNomint.setAttribute("class", "nomint");
divNomint.innerHTML = "Port"+valueDuPort;
divPort.appendChild(divNomint);

}
```