DESARROLLO DE UN MECANISMO INTEGRADO DE PROTECCIÓN CONTRA FALLAS EN REDES MPLS (MULTIPROTOCOL LABEL SWITCHING). ANEXO

LEYLA EUNICE JOAQUÍ CHIMACHANÁ
EDWIN MARCELO BURBANO ANACONA

UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
GRUPO DE NUEVAS TECNOLOGÍAS EN TELECOMUNICACIONES
POPAYÁN
2005

DESARROLLO DE UN MECANISMO INTEGRADO DE PROTECCIÓN CONTRA FALLAS EN REDES MPLS (MULTIPROTOCOL LABEL SWITCHING). ANEXO

LEYLA EUNICE JOAQUÍ CHIMACHANÁ
EDWIN MARCELO BURBANO ANACONA

Monografía para optar al título de
Ingeniero en Electrónica y Telecomunicaciones

Director
Ing. Esp. OSCAR J. CALDERÓN CORTÉS

UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
GRUPO DE NUEVAS TECNOLOGÍAS EN TELECOMUNICACIONES
POPAYÁN
2005

TABLA DE CONTENIDO

ANEXO A - NETWORK SIMULATOR (NS-2)

PARTE 1. DESCARGA E INSTALACIÓN DEL NS EN WINDOWS 9X/2000/XP 4

PARTE 2. DESCARGA E INSTALACIÓN DEL NS Y MNS EN LINUX 6

PARTE 3. MNS VER2.0 MANUAL 7

ANEXO A - NETWORK SIMULATOR (NS-2)

PARTE 1. DESCARGA E INSTALACIÓN DEL NS EN WINDOWS 9x/2000/XP

Lo primero que se hace es descargar los archivos binarios ejecutables del NS y del NAM. Se accesa a la página: http://www.isi.edu/nsnam/ns/ns-build.html ó http://www.it.uc3m.es.

En esta página se busca la sección de descargar versiones de NS que se puede hacer de dos formas, por html y por ftp. Se accesa por html, y se escoge el directorio que dice binary. En este subdirectorio se encuentran todas las versiones disponibles tanto del NS como del NAM; lo importante aquí, es que cada versión de NS debe corresponder con la versión de NAM.

Esta instalación esta basada en la versión compilada para cygwin de ns, aunque no es necesaria la instalación del entorno de ejecución cygwin para emplear NS.

Los pasos para ejecutar el network simulator son los siguientes:

1. **Instalar Active TCL versión 8.4.7**, disponible en:

   http://downloads.activestate.com/ActiveTcl/Windows/8.4.7/ActiveTcl8.4.7.0-win32-ix86-108887.exe

2. Reiniciar el computador.

3. Crear un directorio en el que se guardan los ejecutables de NS, por ejemplo: **C:\ns**.

4. Descomprimir el fichero *ns-allinone-2.27-cygwin-binaries.zip* en el directorio creado en el paso anterior. El fichero esta disponible en http://www.it.uc3m.es/rcalzada/ns2/ns-allinone-2.27-cygwin-binaries.zip

5. Copiar el fichero **cygwin1.dll** en el directorio:

   c:\ns\usr\bin

   El fichero está disponible en http://www.it.uc3m.es/rcalzada/ns2/cygwin1.dll.

6. Copiar el fichero **nam.exe** en el directorio

   c:\ns\usr\bin

   El fichero está disponible en http://www.it.uc3m.es/rcalzada/ns2/nam.exe (en este paso se sobreescribe el fichero original).

7. Incluir en el PATH el directorio **c:\ns\usr\bin**

   c:\> PATH=%PATH%;c:\ns\usr\bin

8. Lo demás corresponde a la visualización en NAM y la interpretación de archivos de salida que es independiente de este proceso.

PARTE 2. DESCARGA E INSTALACIÓN DEL NS Y MNS EN LINUX

Igual que el procedimiento anterior, primero se descargan los archivos binarios ejecutables del NS y del NAM. Se accesa a la página: http://www.it.uc3m.es/~rcalzada/ns/.

Una de las versiones del NS para Linux es el *ns-allinone-2.1b9a*, el cual incluye los parches para MPLS. Este puede descargarse desde:

- ✓ La copia local con el parche MPLS ya instalado, *ns-aulas.tgz*. Descomprimir el fichero en el directorio /usr/local e incluir el directorio /usr/local/ns/bin en el PATH.

  ```
  # tar zxf ns-aulas.tgz
  ```

- ✓ La copia local *ns-allinone-2.1b9a*, *ns-allinone-2.1b9a.tar.gz*.
- ✓ Copia local de los parches MPLS, *mns_v2.0-for-ns-2.1b9-light.tar.gz*. En este caso aplicar el parche siguiendo las instrucciones y luego ejecutar el programa install.

Para la instalación tanto del NS como del NAM, el usuario debe estar como *root*. En el caso de utilizar una versión diferente (por ejemplo, ns 2.28) aplicar el parche y luego ejecutar el programa install, siguiendo las instrucciones que se encuentran en el README del archivo. Además, se recomienda las plataformas de las distribuciones tanto de Fedora core 2.0 como de Debian para la instalación del simulador, debido a que en estos sistemas trabaja y se desempeña mejor el NS.

PARTE 3. MNS ver2.0 MANUAL

MNS simulator is not a final version. So it may have some bugs.

##

ns simulator

##

An instance of the Simulator object is created by the following command (see NS manual for more details)

set ns [new Simulator]

##

APIs for LSP/ER-LSP and LDP

##

$ns MPLSnode

Create a new MPLS node and return a handle to it.

set LSRO [$ns MPLSnode]

$ns configure-ldp-on-all-mpls-nodes

Configure LDP agents on all the created MPLS nodes.

$MPLSnode enable-control-driven

Let $MPLSnode operate as control-driven trigger

$MPLSnode enable-data-driven

Let $MPLSnode operate as data-driven trigger

$MPLSnode enable-on-demand

Let $MPLSnode operate as on-demand-mode

$MPLSnode enable-ordered-control

Let $MPLSnode operate as ordered-control-mode

$ns enable-control-driven

Let all the created MPLS nodes operate as control-driven trigger

$ns enable-data-driven

Let all the created MPLS nodes operate as data-driven trigger

$ns enable-on-demand

Let all the created MPLS nodes operate as on-demand-mode

$ns enable-ordered-control

Let all the created MPLS nodes operate as ordered-control-mode

$MPLSnode send-ldp-release-msg $fec

Send a LDP Release message toward the node with $fec in order to release the established LSP with the FEC $fec.

$MPLSnode send-ldp-withdraw-msg $fec

Send LDP Withdraw message toward the upstream LSRs in order to release the established LSP with the FEC $fec.

$MPLSnode aggregate-flows $fec1 $fec2

Aggregate a fine flow with the FEC $fec1 into a coarse flow with the FEC $fec1.

Configuration Parameter

- $fec1 : The FEC of a fine flow

- $fec2 : The FEC of a coarse flow

##

APIs for ER-LSP

##

$MPLSnode setup-erlsp $fec $er $lspid

Create an ER-LSP of which the FEC is $fec, the specified Explicit Route is $er, and the LSPID is $lspid.

By this command, a CR-LDP Request message is sent toward the node with the FEC $fec along the ER $er.

Configuration Parameter

- $fec : The value of FEC
- $er : Explicit Route
- $lspid : The value of LSPID

$LSR1 setup-erlsp 7 5_4_8_6_7 3000
Create along the ER(i.e. LSR 5, 4, 8, 6, 7) an ER-LSP of which the FEC is 7 and the LSPID is 3000.

$LSR2 setup-erlsp 9 2_L3000_8 4000
Create along the ER(i.e. 2, L3000, 8) an ER-LSP of which the FEC is 9 and the LSPID is 4000.
In the ER, L3000 means LSPID and is used to identify the tunnel ingress point as a next hop.
So, this is an example to establish an ER-LSP that includes the established LSP Tunnel(i.e. its LSPid is 3000).

$MPLSnode bind-flow-erlsp $fec $flowid $lspid

Bind a flow with the FEC $fec and flow id $flowid to the established ER-LSP with the LSPID $lspid

$LSR2 bind-flow-erlsp 9 100 3000"
Bind a flow that FEC is 9 and flow id is 100 to the established ER-LSP of which LSPID is 3000.

##
APIs for CR-LSP and CR-LDP
##

$ns cfg-cbq-for-SBTS $qlim $cbq_qtype $okborrow $bw $maxidle $extradelay

Configure CBQ on all MPLS nodes so as to support Simple Best-effort Traffic Service.

Configuration Parameters
- $qlim : The queue size in packets
- $cbq_qtype : The type of a Queue object into the compound CBQ
- $okborrow : A boolean indicating the class is permitted to borrow bandwidth from its parent
- $bw $maxidle : The máximum amount of time a class may be required to have its packets queued before they are permitted to be forwarded

- $extradelay : increase the delay experienced by a delayed class by the specified time

For more information about configuration parameters, see the CBQ manual provided in NS.

$ns cfg-cbq-for-HBTS $qlim $cbq_qtype $okborrow $bw $maxidle $extradelay

Configure CBQ on all MPLS nodes so as to support Higher priority Best-effort Traffic Service.

Configuration Parameters
see <$ns cfg-cbq-for-SBTS>

$ns cfg-cbq-for-STS $qlim $cbq_qtype $okborrow $bw $maxidle $extradelay

Configure CBQ on all MPLS nodes so as to support Signalling Traffic Service.

Configuration Parameters
see <$ns cfg-cbq-for-SBTS>

$ns cfg-cbq-for-RTS $qlim $cbq_qtype $okborrow $bw $maxidle $extradelay

Configure CBQ on all MPLS nodes so as to support Real-time Traffic Service.

Configuration Parameters
see <$ns cfg-cbq-for-SBTS>

$ns bind-flowid-to-SBTS $idl [$id2]

Cause packets containing flow id $id1 (or those in the range $id1 to $id2 inclusive) to be associated with SBTS service.

$ns bind-flowid-to-HBTS $id1 [$id2]

Cause packets containing flow id $id1 (or those in the range $id1 to $id2 inclusive) to be associated with HBTS service.

$ns bind-flowid-to-STS $id1 [$id2]

Cause packets containing flow id $id1 (or those in the range $id1 to $id2 inclusive) to be associated with STS service.

$ns bind-ldp-to-SBTS

Cause LDP packets to be associated with SBTS service.

$ns bind-ldp-to-HBTS

Cause LDP packets to be associated with HBTS service.

$ns bind-ldp-to-STS

Cause LDP packets to be associated with STS service.

$MPLSnode setup-crlsp $fec $er $lspid $TRate $BSize $PSize $SPrio $HPrio

Create an CR-LSP ofth which the FEC is $fec, the specified Explicit Route is $er, and the LSPID is $lspid. By this command, a CR-LDP Request message is sent toward the node with $fec along the $er.

Configuration Parameter

- $fec : The value of FEC
- $er : Explicit Route
- $lspid : The value of LSPID
- $TRate : The value of traffic rate
- $BSize : The value of buffer size
- $PSize : The value of packet size
- $SPrio : The value of setup priority
- $HPrio : The value of holding priority

$LSR2 setup-erlsp 4 2_3 5000 450K 400B 200B 7 3

Create along the ER(i.e LSR 2 and 8) an ER-LSP of which FEC is 4 and LSPID is 5000. This is an example to establish an CR-LSP that includes traffic parameters.

$MPLSnode send-crldp-release-msg $lspid

Send a CR-LDP Release message toward an upstream LSR in order to release the established ER-LSP/CR-LSP of which the LSPID is $lspid.

$LSR1 send-crldp-release-msg 3000

release the ER-LSP with the LSPID 3000 by using CR-LDP Release message

$MPLSnode send-crldp-withdraw-msg $lspid

Send a CR-LDP Withdraw message toward downstream LSRs in order to release the established ER-LSP/CR-LSP of which the LSPID is $lspid.

$MPLSnode send-crldp-notifícation-msg $status $lspid $tr

Send a CR-LDP Notification message toward upstream LSRs in order to give them a notification information.

Configuration Parameter
$status : The status information defined in CR-LDP standards
$lspid : LSPID
$tr : Traffic information

$MPLSnode set-flow-prio $fec $flowid $priority

Let the traffic with FEC $fec and flow id $flowid have priority $priority $priority is 0 or 1. 1 means high priority.

There are three call-back functions in MNS

- proc notify-erlsp-setup {node lspid}

It is used to notify user that CR-LDP Mapping message with LSPID $lspid arrived at a node $node.

- proc notify-erlsp-fail {node status lspid tr}

It is used to notify user that CR-LDP Notification message with a status information $status, LSPID $lspid, and traffic information $tr arrived at a node $node.

- proc notify-erlsp-release {node lspid}

It is used to notify user that CR-LDP Release/Withdraw message with LSPID $lspid arrived at a node $node.

##
APIs for the constraint-based routing
##

$ns collect-resource-info $itime

Used to collect the resource information from all MPLS nodes periodically every the time interval $itime

$MPLSnode constraint-based-routing $dstid $bw

Used to calculate explicit route.
If O.K return an explicit route, else return -1.

Configuration Parameters
- $dstid : the ID of the destination LSR
- $bw : the required bandwidth

##
APIs for reroute simulation in MPLS Networks
##

$ns enable-reroute $option

let all MPLS nodes execute path restoration function.

There are five options as follows:
- drop : Not create any new alternative path.
- L3 : make use of L3 routing table.
- notify-prenegotiated : The node that detected a link failure transmits a LDP Notification message to its upstream LSRs.
- simple-dynamic : Create new alternative path between The node that detected a link failure and the PML if one does not exist.
- shortest-dynamic: Create new alternative path between The node that detected a link failure and the next node if one does not exist.

$ns set-protection-lsp $stime $itime $lspid

used to see whether a link failure occurs on all MPLS nodes.

Configuration Parameters
- $stime : The time to start detecting a link failure
- $itime : The time intervals
- $lspid : the LSPID of working LSP

$MPLSnode reroute-lsp-binding Sw_lspid a_lspid

used to bind working LSP to alternative LSP

Configuration Parameters

- $w_lspid : The LSPID of working LSP
- $a_lspid : The LSPID of alternative LSP

$MPLSnode enable-reroute-egress-lsr

let $MPLSnode operate as a Protection Merge LSR

```
#################################################################################
#   APIs for Trace of MPLS/LDP packets and Dump of Tables                       #
#################################################################################
```

$MPLSnode trace-mpls

Trace packets in a MPLS node, $MPLSnode

$MPLSnode trace-ldp

Trace LDP packets in a MPLS node, $MPLSnodel

$MPLSnode pft-dump

Display a table, PFT(Partial Forwarding Table) managed in a MPLS node, $MPLSnode

$MPLSnode erb-dump

Display a table, ERB(Explicit Route information Table) managed in a MPLS node, $MPLSnode

$MPLSnode lib-dump

Display a table, LIB(Label Information Base) managed in a MPLS node, $MPLSnode

```
#################################################################################
#   APIs for utility                                                            #
#################################################################################
```

$ns ldp-request-color $color

Set a color for LDP Request message.

$ns ldp-request-color $green

Set green color for LDP Request message.

$ns ldp-mapping-color $color

Set a color for LDP Mapping message

$ns ldp-withdraw-color $color

Set a color for LDP Withdraw message

$ns ldp-release-color $color

Set a color for LDP Release message

$ns ldp-notifícation-color $color

Set a color for LDP Notification message