AFS

bintec-Dm 786-I

Copyright© Version 11.0H bintec-elmeg

Legal Notice

Warranty

This publication is subject to change.

bintec offers no warranty whatsoever for information contained in this manual.

bintec is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

I Related Documents

bintec-Dm 702-I TCP-IP

bintec-Dm 715-I BRS

bintec-Dm 720-I NAT Feature

bintec-Dm 735-I NAPT Facility

bintec-Dm 739-I IPSEC

bintec-Dm 752-I Access Control

bintec-Dm 754-I NSLA

bintec-Dm 755-I Dynamic NAT

bintec-Dm 764-I Route Mapping

bintec-Dm 788-I New NAT

bintec-Dm 795-I Policy Map-Class Map

bintec-Dm 808-I IPv6 Access Control

Chapter 1 Introduction

1.1 AFS System: Description

AFS stands for Advanced Firewall System. This system, developed by bintec, classifies and handles IP packets. To activate the AFS system, you need to deactivate static NAT (see manual *bintec Dm720-I NAT Feature*), dynamic NAT (see manual *bintec Dm755-I Dynamic NAT*), PAT (see manual *bintec Dm735-I NAPT Facility*) and IP access controls (see manual *bintec Dm702-I TCP-IP*). The AFS system replaces all the aforementioned systems.

When the AFS system is activated, IP (IPv4/IPv6) packets processed by the router are automatically associated with a session. When a packet is processed for the first time, the router tries to find its corresponding session among existing ones. If it can't, a new session is created and assigned to said packet.

Sessions are defined by source IP, destination IP and the protocol encapsulated in IP. If, for example, said protocol is TCP or UDP, the session is also defined by source and destination ports. Consequently, a session can be an RTP flow pertaining to a voice conversation, a Telnet session between two devices, or a simple ping.

When all IP packets are linked to a session, you can establish packet selection criteria based on the state, or type, of the associated session. Thus, you can establish selection criteria based on whether this is a new session, one that had already been established, or one with a response from the remote end that has not yet been received.

An AppId, as described in RFC 6759, is linked to each session. For non-TCP or UDP protocols, the AppId will be IANA-L3 (1) and will follow the protocol number. For TCP or UDP sessions, the AppId will be IANA-L4 (3) and will follow the destination port number of the packet that initiated the session (i.e. the server port). The ALG feature will override this AppId assignment and allocate dynamically created sessions, e.g. assign L4:21 (FTP) AppId to both control and data sessions. Moreover, the application detection feature (app-detect) may, if enabled, override the normal AppId assignment for the applications detected. For instance, the L4:80 (HTTP) AppId can be assigned to a session connecting to a 8080 server port if HTTP application is detected. Custom AppIds with a USER-defined (6) classification engine can be assigned to any session using Policy Based Routing (route-map). The AppId can be used as a selection criteria to match the use of stateful access-lists.

In the case of IPv4, additional information on IP address (NAT) handling is stored in each session. This way, the same translation is applied to all packets belonging to said session. This means NAT rules only need to be checked for the first packet in each session (greatly accelerating NAT application for complex, multiple-rule configurations).

Please bear in mind the order in which systems are applied when AFS is enabled.

An IP packet is processed as follows:

(1) The packet reaches code managing IP packet processing, the latter checks the IP checksum is correct and associates said packet to an AFS session. Application detection (app-detect), if enabled, is performed as soon as there is a session.

(2) IPSEC: if a packet is encapsulated in IPsec, it's decapsulated before continuing (see manual *bintec Dm739-I IPSEC*). This is only available for IPv4.

(3) Input Post-Filter (only valid if the **post-nat** command is enabled for the current VRF): this checks whether the packet matches a filter criterion for input interface filtering. If it does, the packet is passed or dropped accordingly (please see manuals *bintec Dm752-I Access Control* and *bintec Dm808-I IPv6 Access Control*). The possibility of discarding packets is important, since filtering is executed before NAT at destination. The filtered address is the source IP packet destination address, without translation.

(4) Destination NAT: if the destination IP has to be changed, the necessary translation is applied at this point (see manual *bintec Dm788-I New NAT*). This is only available for IPv4.

(5) Input Pre-Filter (only if **pre-nat** is enabled for the current VRF): similar to the destination Post-Filter. Like in the previous case, filtering is executed after NAT translation at destination (i.e. the filtering must be carried out based on the translated destination address and not over the source IP packet).

(6) ROUTE-MAP: the route map associated to the input interface is verified just before the IP routing table is checked (see manual *bintec Dm764-I Route Mapping*).

(7) The routing table is checked and the output interface determined.

(8) Output Pre-Filter (only if **pre-nat** is enabled for the current VRF): this checks the output filtering rules for the corresponding interface, dropping the packets specified by the filtering rules (please see manuals *bintec Dm752-I Access Control* and *bintec Dm808-I IPv6 Access Control*). Please note, as filtering is executed before NAT translation at source, the filtered address is the non-translated IP packet source address.

(9) Source NAT: if the source IP has to be changed, the necessary translation is applied at this point (see manual *bintec Dm788-I New NAT*). This is only available for IPv4.

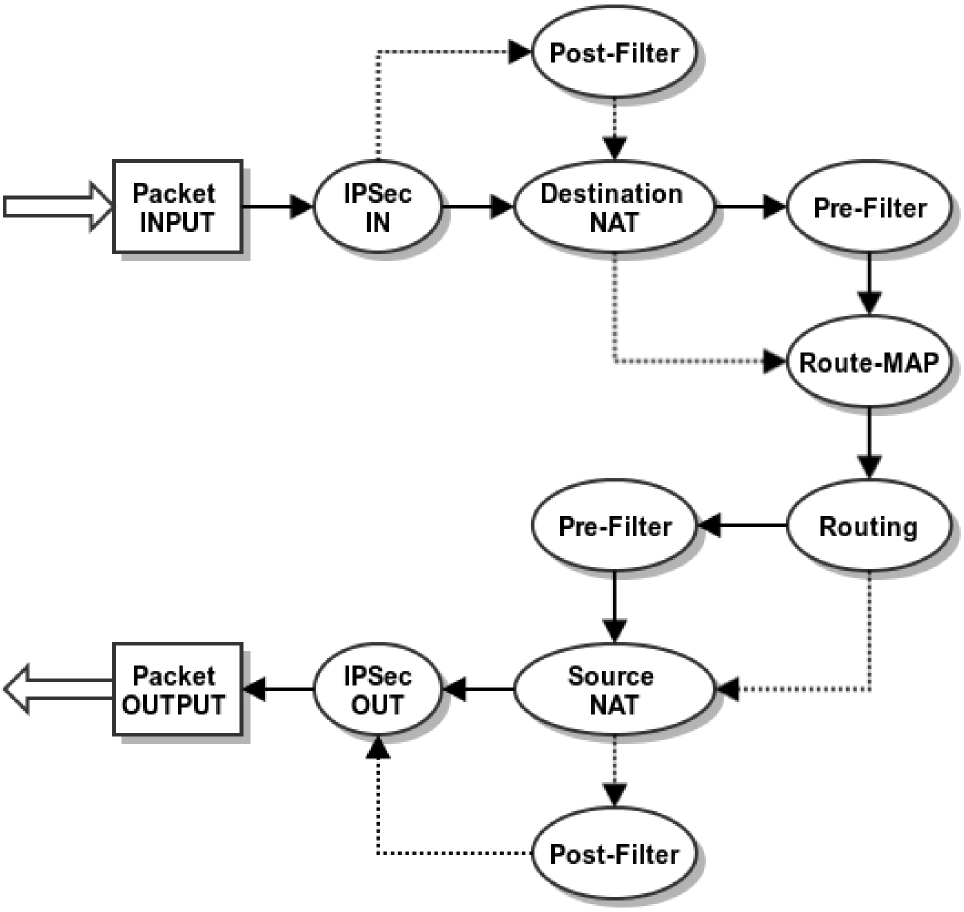
(10) Output Post-Filter (only if **post-nat** is enabled for the current VRF): similar to the output Pre-Filter. Like in the previous case, filtering is executed after NAT translation at source (i.e. the filtering must be carried out based on the translated source address and not over the source IP packet).

(11) IPSEC: if a packet has to be encapsulated, it is encapsulated at this point. This packet is then treated like a new packet and goes through all of the above-described stages again (see manual *bintec Dm739-I IPSEC*). This is only available for IPv4.

(12) The packet is sent through the output interface.

Please bear in mind that dynamic changes, executed in the device configuration of certain packets associated to a given session, do not take effect while said session is open (e.g. enabling NAT). For these changes to be effective, you must end the session and create a new one (deactivating AFS and reactivating the **no enable** and **enable** commands).

The following diagram shows how. Please note that the dotted arrows show the path for alternative packets when **post-nat** is enabled.



Chapter 2 Configuring the AFS System

2.1 Configuring the AFS System

To configure the AFS system, enter **feature afs** from the main configuration menu.

Syntax:

```
Config>feature afs
AFS config>
```

If you want to configure a specific VRF, access the AFS menu first (as explained above) and then access the VRF you want to configure. This feature is only available for IPv4. For instance, to configure the AFS system in the bintec VRF:

Example:

```
Config>feature afs
AFS config>vrf bintec
AFS vrf config>
```

AFS system configuration is identical for both the main VRF and the secondary VRFs. The following explanation applies to all.

2.1.1 [NO] ALG

Applications that include information on IP addressing and/or TCP/UDP ports outside of the corresponding headers are known as *venomous* applications. These must be handled in a specific way to run properly in a device executing NAT. Examples of these applications are FTP, SIP, etc.

To solve this problem in AFS, *application level gateways* (ALGs) are implemented. These are software modules that handle the content of IP packets associated to *venomous* applications and adjust the IP addresses and ports as necessary.

Please note that the most common ports for the available algorithms are enabled by default. These ports are not shown when you list the subsystem configuration. To disable an algorithm, use the **disabled** option described in the following sections.

2.1.1.1 [NO] ALG FTP DISABLED

Disables the use of the FTP algorithm (including the default port). This command takes priority over the ALG FTP PORT entries, meaning their presence no longer has an impact.

Syntax:

```
AFS config>ALG FTP DISABLED
AFS config>
```

2.1.1.2 [NO] ALG FTP PORT

Enables the FTP algorithm for the TCP port indicated. Port 21 is enabled by default.

Syntax:

```
AFS config>ALG FTP PORT <port-number>
AFS config>
```

2.1.1.3 [NO] ALG PPTP DISABLED

Disables the use of the PPTP algorithm (including the default port). This command takes priority over the ALG PPTP PORT entries, meaning their presence no longer has an impact.

Syntax:

```
AFS config>ALG PPTP DISABLED
AFS config>
```

2.1.1.4 [NO] ALG PPTP PORT

Enables the PPTP algorithm for the TCP port indicated. Port 1723 is enabled by default.

Syntax:

```
AFS config>ALG PPTP PORT <port-number>
AFS config>
```

2.1.1.5 [NO] ALG SIP DISABLED

Disables the use of the SIP algorithm (including the default port). This command takes priority over the ALG SIP PORT entries, meaning their presence no longer has an impact.

Syntax:

```
AFS config>ALG SIP DISABLED
AFS config>
```

2.1.1.6 [NO] ALG SIP PORT

Enables the SIP algorithm for the UDP port indicated. Port 5060 is enabled by default.

This ALG only operates when the SIP protocol is encapsulated over UDP.

Syntax:

```
AFS config>ALG SIP PORT <port-number>
AFS config>
```

2.1.2 [NO] APP-DETECT

Configures the Application Detection subsystem within AFS. Several application detection capabilities can be enabled. Application detection is performed through Deep Packet Inspection (DPI) of the first packets of a session. Once detection has taken place, be it positive (detected) or negative (not detected), the session is classified, the information extracted, and subsequent session packets are not inspected further. While DPI is running in the first packets belonging to a session, an associated state (app-detecting) can be set as selection criteria in stateful access-lists.

For detection to work with fragmented packets, IP fragments are reassembled before applying DPI. Also, if the application detector so requires, TCP session packets of up to 8000 bytes are also reassembled.

Fields extracted from application detectors can be up to 200 characters long. Larger fields are reduced to that size.

2.1.2.1 [NO] APP-DETECT HTTP

Enables HTTP detection capabilities. When activated, the command analyzes TCP sessions looking for traces of the HTTP protocol. Versions 1.0 and 1.1 are both detected. Only the initial session request is inspected, so persistent connections only extract HTTP fields belonging to the first request. GET, PUT, POST, HEAD, TRACE, DELETE, OPTIONS and CONNECT are supported HTTP request methods. On HTTP detection, the L4:80 AppID is assigned to the session (regardless of the server-side port used for connection purposes).

Syntax:

```
AFS config>APP-DETECT HTTP
```

Command history:

Release	Modification
11.01.01	The "*app-detect http*" command was introduced as of version 11.01.01.

2.1.2.2 [NO] APP-DETECT HTTP HOST

Allows for HTTP Host header domain names to be extracted from the first session request. The information obtained can be displayed using the *list sessions* monitor command, exported via IPFIX, or used as match criteria for stateful access-lists.

Optionally, domain name filtering can be performed configuring the number of top level domains to keep and the number of top level domains to skip.

Syntax:

```
AFS config>APP-DETECT HTTP HOST ?
  <cr>
  keep-lvls  <1..10>   Number of domain levels to keep
     <cr>
     skip-lvls  <1..10>   Number of top levels to skip
```

Example:

These are the values extracted from detected host *play.google.com* that correspond to different domain filtering configurations:

Command	Extracted value
app-detect http host	play.google.com
app-detect http host keep-lvls 1 skip-lvls 1	google
app-detect http host keep-lvls 2	google.com

Command history:

Release	Modification
11.01.01	The "*app-detect http host*" command was introduced as of version 11.01.01.

2.1.2.3 [NO] APP-DETECT HTTP REFERER

Allows for HTTP Referer headers to be extracted from the first session request. The information obtained can be displayed using the *list sessions* monitor command, exported via IPFIX, or used as match criteria for stateful access-lists.

Optionally, a maximum size can be configured to limit the length of the extracted value.

The *host-only* mode extracts hostnames from the Referer header and falls back to Host header hostname in case no Referer header is present. The Referer hostname may be a more useful piece of information than the Host header, since one website access usually needs several resources for different servers and domains and these requests share the Referer header of the original website. After extracting the Referer hostname, all HTTP connections generated for the same website are assigned the hostname of the accessed site.

Also, optionally, domain name filtering can be performed in *host-only* mode (just like with the*app-detect http host* command).

Syntax:

```
AFS config>APP-DETECT HTTP REFERER ?
  <cr>         Enable http referer header detection
  max-size  <1..200>   Maximum referer size to extract
  host-only    Enable http referer host detection
     <cr>
     keep-lvls  <1..10>   Number of domain levels to keep
        <cr>
        skip-lvls  <1..10>   Number of top levels to skip
```

Command history:

Release	Modification
11.01.01	The "*app-detect http referer*" command was introduced as of version 11.01.01.

2.1.2.4 [NO] APP-DETECT HTTP URL

Allows for HTTP URLs to be extracted from the first session request. The information obtained can be displayed using the *list sessions* monitor command, exported via IPFIX, or used as match criteria for stateful access-lists.

Optionally, a maximum size can be configured to limit the length of the extracted value.

Syntax:

```
AFS config>APP-DETECT HTTP URL ?
  <cr>
  max-size  <1..200>   Maximum URL size to extract
```

Command history:

Release	Modification
11.01.01	The "*app-detect http url*" command was introduced as of version 11.01.01.

2.1.2.5 [NO] APP-DETECT HTTP USER-AGENT

Allows for the HTTP User-agent header to be extracted from the first session request. The information obtained can be displayed using the *list sessions* monitor command, exported via IPFIX, or used as match criteria for stateful access-lists.

Optionally, a maximum size can be configured to limit the length of the extracted value.

Syntax:

```
AFS config>APP-DETECT HTTP USER-AGENT ?
  <cr>
  max-size  <1..200>   Maximum user-agent size to extract
```

Command history:

Release	Modification
11.01.01	The "*app-detect http user-agent*" command was introduced as of version 11.01.01.

2.1.2.6 [NO] APP-DETECT SSL

Enables SSL/TLS detection capabilities.When activated, the command analyzes TCP sessions in search for SSL/TLS handshakes. Both SSL and TLS are detected, regardless of the version. Only the initial Client Hello of the session is inspected. When SSL/TLS is detected, no specific AppID is assigned to the session as SSL/TLS is not a final application but a transport, so the server-side port number carries more information regarding the application (e.g. 443 for HTTPS or 5061 for SIPS). Packets of detected SSL/TLS sessions can be selected using stateful access-lists with an *app-detect ssl* match command.

Syntax:

```
AFS config>APP-DETECT SSL
```

Command history:

Release	Modification
11.01.01	The "*app-detect ssl*" command was introduced as of version 11.01.01.

2.1.2.7 [NO] APP-DETECT SSL HOST

Allows for SSL/TLS server host domain names to be extracted from the Sever Name Indication (SNI) extension linked to Client Hello. The information obtained can be displayed using the *list sessions* monitor command, exported via IPFIX, or used as match criteria for stateful access-lists.

Optionally, domain name filtering can be configured by adding a number of top level domains to keep and a number of top level domains to skip, just like with the *app-detect http host* command.

Syntax:

```
AFS config>APP-DETECT SSL HOST ?
  <cr>
  keep-lvls  <1..10>   Number of domain levels to keep
     <cr>
     skip-lvls  <1..10>   Number of top levels to skip
```

Command history:

Release	Modification
11.01.01	The "*app-detect ssl host*" command was introduced as of version 11.01.01.

2.1.3 [NO] ENABLE

Enables the AFS system. The system is deactivated by default and must be activated via this command. This command is only available for the main VRF. Once the AFS system has been enabled, all the VRFs configured in the device are enabled too.

Syntax:

```
AFS config>ENABLE
AFS config>
```

2.1.4 [NO] FILTER

By using this command, the user can define whether filtering is carried out before or after NAT translation. This creates two possibilities:

(1) Pre-NAT. Filtering is executed with the untranslated addresses i.e. with the original IP packet addresses.

(2) Post-NAT. Filtering is executed with the addresses translated.

Please note that the type of filtering is determined by the place where source NAT is executed. Since destination NAT is the opposite of source NAT, the position of pre-NAT and post-NAT are reversed.

Syntax:

```
AFS config>filter type {pre-nat}|{post-nat}
AFS config>
```

pre-nat	Default value. Filtering is executed before NAT at source and, subsequently, at destination.
post-nat	Filtering is executed after NAT at source and before this at destination.

2.1.5 [NO] GLOBAL

Allows you to configure AFS global parameters. Currently, you can configure the maximum number of simultaneous sessions.

Syntax:

```
AFS config>GLOBAL MAX-SESSIONS {<max> | unlimited}
AFS config>
```

<max>	Maximum number of simultaneous sessions.
unlimited	Unlimited number of simultaneous sessions.

2.1.6 [NO] MAX-FRGMNT-ENTRIES

Configures the maximum number of fragmentation control entries used in the equipment. A control entry is created when an IP fragment with an unknown ID is received. The entry is used to store the fragments until the complete IP packet is received, in order to be processed and routed by the AFS. If the maximum number of entries is reached, the oldest entry is deleted (and the associated IP fragments dropped) before a new entry is created.

Syntax:

```
AFS config>MAX-FRGMNT-ENTRIES <24..128>
AFS config>
```

The default value for this parameter is 48. Use the **no max-frgmnt-entries** command to set the default value.

Command history:

Release	Modification
11.00.07	The "*max-frgmnt-entries*" command was introduced as of version 11.00.07.
11.01.02	The "*max-frgmnt-entries*" command was introduced as of version 11.01.02.

2.1.7 [NO] SKIP-CHECKING-TCP-HANDSHAKE

Allows you to disable **Handshake** on TCP packet checking. By default, when AFS is enabled and a NAT rule configured for a given VRF, all TCP packets in such VRF are checked. If **Handshake** is not complete, these packets are discarded and undesired TCP sessions avoided (for instance, when a TCP packet with SYN+ACK flags is received without the SYN packet being sent).

Syntax:

```
AFS config>SKIP-CHECKING-TCP-HANDSHAKE
AFS config>
```

Note

For security reasons, we recommend enabling **Handshake** on TCP packet checking. However, in some scenarios, disabling this monitoring option may be necessary. (E.g. when TCP control session packets do not follow the same path).

Command history:

Release	Modification
10.09.26	The "*skip-checking-tcp-handshake*" command was introduced as of version 10.09.26.
11.00.05	The "*skip-checking-tcp-handshake*" command was introduced as of version 11.00.05.
11.01.00	The "*skip-checking-tcp-handshake*" command was introduced as of version 11.01.00.
11.00.07	The "*skip-checking-tcp-handshake*" command becomes unnecessary for VRFs where NAT is not configured.
11.01.06	The "*skip-checking-tcp-handshake*" command becomes unnecessary for VRFs where NAT is not configured.

2.1.8 [NO] TIMEOUT

Thanks to the packets processed by the router, sessions are dynamically created in the AFS system the moment they are detected. After a certain period of time, during which the router does not process traffic associated to them, said sessions are eliminated. Waiting times change depending on the protocol and the status of the session, and are configured through this command.

Similarly, the access-list timeout is used to handle active sessions that match the access-list configuration.

2.1.8.1 [NO] TIME-OUT GRE

Configures the timeout for GRE sessions.

Syntax:

```
AFS config>TIME-OUT GRE ?
  steady       Time-out for a bidirectional packet flow
  unreplied    Time-out for an unidirectional packet flow
AFS config>
```

unreplied	Timeout for GRE sessions where only traffic in the session source direction has been processed, i.e. no return packet has been processed.
steady	Timeout for already established GRE sessions, i.e. at least one packet for both directions has been processed.

Command history:

Release	Modification
11.01.06	As of version 11.01.06, values can be introduced without having to specify they are measured in seconds (e.g. both "8s" and "8m" are valid).

2.1.8.2 [NO] TIME-OUT ICMP

Configures the timeout for ICMP sessions.

Syntax:

```
AFS config>TIME-OUT ICMP ?
  unreplied <to>   Icmp unreplied pending session time-out
AFS config>
```

unreplied	Timeout for ICMP sessions where the ICMP petition has been processed, but not the reply.

Command history:

Release	Modification
11.01.06	As of version 11.01.06, values can be introduced without having to specify they are measured in seconds (e.g. both "8s" and "8m" are valid).

2.1.8.3 [NO] TIME-OUT OTHERS

Configures the timeout for the remaining protocols, i.e. those that aren’t GRE, ICMP, TCP or UDP.

Syntax:

```
AFS config>TIME-OUT OTHERS <to>
AFS config>
```

Command history:

Release	Modification
11.01.06	As of version 11.01.06, values can be introduced without having to specify they are measured in seconds (e.g. both "8s" and "8m" are valid).

2.1.8.4 [NO] TIME-OUT TCP

Configures the timeout for TCP sessions, depending on the state the session is in. These states are defined in the RFC 793 Transmission Control Protocol.

Syntax:

```
AFS config>TIME-OUT TCP ?
  close <to>           Close tcp state time-out
  close-wait <to>      Close-wait tcp state time-out
  established <to>     Established tcp state time-out
  fin-wait <to>        Fin-wait tcp state time-out
  last-ack <to>        Last-ack tcp state time-out
  max-retries <to>     Max-retries tcp state time-out
  syn-received <to>    Syn-received tcp state time-out
  syn-sent <to>        Syn-sent tcp state time-out
  time-wait <to>       Time-wait tcp state time-out
AFS config>
```

close	Time-out for TCP sessions in close state.
close-wait	Time-out for TCP sessions in close-wait state.
established	Time-out for TCP sessions in established state.
fin-wait	Time-out for TCP sessions in fin-wait state.
last-ack	Time-out for TCP sessions in last-ack state.
max-retries	Time-out for TCP sessions in max-retries state.
syn-received	Time-out for TCP sessions in syn-received state.
syn-sent	Time-out for TCP sessions in syn-sent state.
time-wait	Time-out for TCP sessions in time-wait state.

Command history:

Release	Modification
11.01.06	As of version 11.01.06, values can be introduced without having to specify they are measured in seconds (e.g. both "8s" and "8m" are valid).

2.1.8.5 [NO] TIME-OUT UDP

Configures the timeout for UDP sessions.

Syntax:

```
AFS config>TIME-OUT UDP ?
  unreplied <to>    Time-out for an unidirectional packet flow
  steady <to>         Time-out for a bidirectional packet flow
AFS config>
```

unreplied	Timeout for UDP sessions where only traffic in the session source direction has been processed, i.e. no return packet has been processed.
steady	Timeout for already established UDP sessions, i.e. at least one packet for both directions has been processed.

Command history:

Release	**Modification**
11.01.06	As of version 11.01.06, values can be introduced without having to specify they are measured in seconds (e.g. both "8s" and "8m" are valid).

2.1.8.6 [NO] TIME-OUT ACCESS-LIST

Configures a timeout for AFS sessions that use extended and stateful access lists. Commands shall be run in the order in which they are entered.

An access list to be exclusively used by the AFS timeout must be created. This list cannot be used with any other protocol.

Syntax:

```
AFS config>TIME-OUT ACCESS-LIST ?
  <100..1999>    Extended Access List number (100-1999)
  <100..1999>    Stateful Access List number (5000-9999)
AFS config>TIME-OUT ACCESS-LIST 100 <to>
AFS config>
```

Command history:

Release	**Modification**
11.01.06	The "*time-out access-list*" command was introduced as of version 11.01.06.

2.2 Configuring AFS-based multipath routing

As explained in manual bintec *Dm702-I TCP IP,* , multipath routing allows you to apply a payload balance system based on the relative weights configured in the output interfaces for paths included in the multipath. Next hop selection, used to transmit a certain packet, depends on the occupation factor of the interfaces involved. This, in turn, is directly based on parameters such as relative weight or bandwidth configured.

This section details the AFS-based multipath routing options available in bintec routers.

Example:

```
IP config>multipath ?
  per-destination    Enable per source and destination multipath routing
  per-packet         Enable per packet multipath routing
  per-afs-session    Per AFS session multipath
```

The types of multipath routing available are shown above, where **multipath per-afs-session** uses AFS on routing.

Command history:

Release	**Modification**
11.00.05	The "*multipath per-afs-session*" command was introduced as of version 11.00.05.
11.01.00	The "*multipath per-afs-session*" command was introduced as of version 11.01.00.

2.2.1 Multipath per-afs-session

As described in section 1, when a packet is processed for the first time, AFS tries to find its corresponding session among the existing ones. If it can't, a new session is created and assigned to said packet. Furthermore, if **multipath per-afs-session** is enabled and multiple paths exist to reach an equal cost destination, the router assigns a next hop to each new AFS session in a circular queue (Round-robin mode).

Activate **multipath per-afs-session** as follows.

```
IP config>multipath per-afs-session
```

If a static route configured with **multipath per-afs-session** is deleted during operation, and there were AFS sessions assigned to its next hop, the following occurs: when a packet for each session is rerouted, a new hop belonging to the remaining routes in the multipath is assigned (in Round-robin mode) for said sessions.

The **multipath per-afs-session** option offers recursive routing. When indirect routes are installed in the routing table, multipath per-afs-session is able to find the path across the routes until a next-hop that belongs to a directly connected network is found.

Command history:

Release	Modification
11.01.07	Recursive multipath per-afs-session has been implemented as of version 11.01.07.

2.2.2 Reactive multipath per-AFS-session

Current implementation of **multipath per-afs-session** behaves reactively when assigning a new hop to a new AFS session if *mp-congestion* and/or *mp-disable* are configured for IPv4 static routes in the multipath. This is described in manual bintec *Dm754-I NSLA*.

```
IP config>ROUTE <address> <mask> <gateway> [<cost>] [TRACK NSLA-ADVISOR <advisor-id>] \
          [mp-congestion <advisor-id>] [mp-disable <advisor-id>]
```

Both *mp-congestion* and *mp-disable* behave in a similar way. They can mark a route with the congestion and disabling flags (respectively), depending on the result of configured advisors. The flags are structured hierarchically and can be marked together.

Depending on the flags set, a route can be identified with the states of unmarked, congested or disabled. A route simultaneously marked with congestion and disabling flags is considered disabled. Disabled is a more restrictive state than congested.

The next figure shows the state of routes configured via the **multipath per-afs-session** command, as well as possible transitions between states.

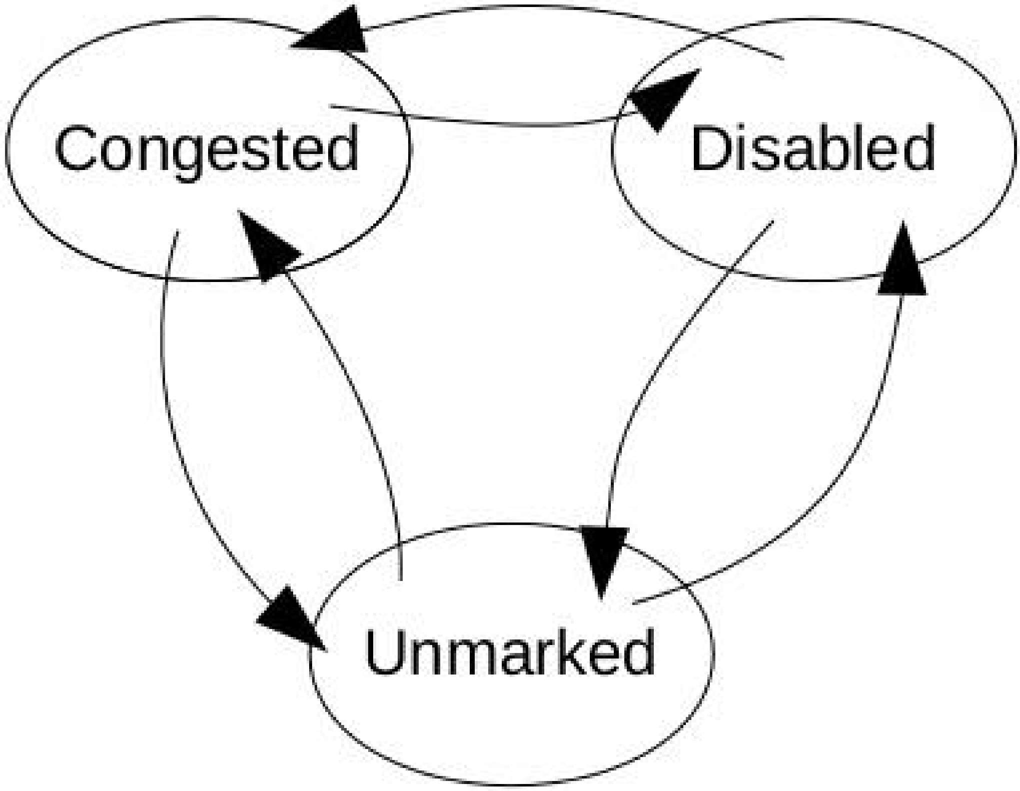


Fig. 1: State diagram

The *mp-congestion* option allows you to link a static route to an advisor. This way, when the advisor output is activated (TRUE), the route is marked as congested (allowing for unmarked routes to be selected in multipath routing instead).

If all routes in a multipath routing were marked as congested, one of them would still be selected.

The *mp-disable* option allows you to link a static route to an advisor. This way, when the advisor output is activated (TRUE), the route is marked as disabled (allowing for congested or unmarked routes to be selected in multipath routing instead).

If all routes in a multipath routing were marked as disabled, one of them would still be selected.

If *mp-congestion* and *mp-disable* are configured together for a static route in multipath, each configured advisor determines the state of each respective flag for said route, and, as a result, the state of the route in the multipath.

The same advisor can be configured for several routes. Moreover, *mp-congestion* and *mp-disable* can be active at the same time.

In brief, the **multipath per-afs-session** subsystem behaves as follows: firstly, it tries to select an unmarked route in the multipath; secondly, if no unmarked route is found, the subsystem searches for congested routes; and finally, if no unmarked or congested routes are found, the system selects a disabled route.

An example of configuration that uses reactive **multipath per-afs-session** can be found in the last chapter.

Command history:

Release	Modification
11.00.05	The "*mp congestion*" and "*mp-disable*" command options were introduced as of version 11.00.05.
11.01.00	The "*mp congestion*" and "*mp-disable*" command options were introduced as of version 11.01.00.

Chapter 3 Monitoring the AFS System

3.1 AFS System Monitoring Commands

The AFS system monitoring commands must be entered in the monitoring menu associated to AFS (*AFS*+). Use **feature afs** to access this menu. Said command is located in the IP protocol monitoring menu.

```
+feature afs
-- AFS Monitor --
AFS+
```

If you want to monitor a specific VRF, access it through the AFS menu.

Example:

```
+feature AFS
-- AFS Monitor --
AFS+vrf bintec
-- AFS Monitor --
AFS bintec vrf+
```

AFS system monitoring is identical for both the main VRF and the secondary VRFs. The following explanation applies to both.

The options presented in the AFS system monitoring menu are as follows:

```
AFS+?
  clear            Clear AFS general information
  filter           Define filter to match AFS sessions
  history-graph    History graph for sessions within the past 48 hours
  list             List AFS general information
  no               Negate a command or set its defaults
  top              List session traffic information
  vrf              Vrf AFS monitor
  exit
AFS+
```

3.1.1 CLEAR SESSIONS

Clears the AFS sessions that are currently active in the device.

Syntax:

```
AFS+ CLEAR SESSIONS
AFS+
```

Command history:

Release	Modification
11.00.05	The "*clear sessions*" command was introduced as of version 11.00.05.
11.01.00	The "*clear sessions*" command was introduced as of version 11.01.00.

3.1.2 LIST ALG

3.1.2.1 LIST ALG AWAITED-SESSIONS

Lists the sessions identified by the ALGs that the AFS system hasn't detected yet. For instance, if a control session requests a data session to be opened, the latter is listed under **awaited** until the AFS system detects the first packet belonging to said data session.

Syntax:

```
AFS+LIST ALG AWAITED-SESSIONS
AFS+
```

3.1.3 LIST FRAGMENTATION-CONTROL

Displays information on fragmentation controls implemented by the AFS session.

IP packets that are really fragments of a larger IP packet are classified and stored in a list displayed through this command.

Syntax:

```
AFS+LIST FRAGMENTATION-CONTROL
AFS+
```

3.1.4 LIST GLOBAL

Displays global statistics related to the AFS system. Unlike what happens with other commands, this one shows statistics for all VRFs configured in the device.

Syntax:

```
AFS+LIST GLOBAL
AFS+
```

Example:

```
AFS+LIST GLOBAL
          AFS Global status (All vrfs)
------------------------------------------------
Total active sessions ..................         1
Maximum number of sessions reached .....         2
Maximum number of sessions allowed .....     67104
Active IP fragmentation entries ........         0
Maximum IP fragmentation entries .......         0
Limit of IP fragmentation entries ......        48
Active application reassembly entries...         0
Maximum application reassembly entries..         1
AFS+
```

Command history:

Release	Modification
11.00.07	The "*list global*" command was introduced as of version 11.00.07.
11.01.02	The "*list global*" command was introduced as of version 11.01.02.

3.1.5 LIST NAT

Lists information on NAT rules.

Syntax:

```
AFS+LIST NAT <detail> rule <rule-id>
AFS+
```

low-detail	Less-detailed information.
high-detail	Very detailed information.
normal-detail	Normal details.
rule-id	Identifier for the rule you want displayed.

Example:

```
AFS+list nat high-detail rule 1
Packets processed 2, Bytes processed 104
Dynamic NAT, total entries 10 (1.1.1.1<->1.1.1.10), used entries 1
   Local IP 192.168.212.19 <-> Global IP 1.1.1.10  ( 2 sessions using it)
Protocol tcp(6) dying in 431999 secs, packets in router 1 established
Original 192.168.212.19:49544->192.168.1.2:21 Packets=12 (550 Bytes)
```

```
    Reply    192.168.1.2:21->1.1.1.10:49544 Packets=12 (791 Bytes) [STEADY] [SNAT rul 1]
Protocol tcp(6) dying in 118 secs, packets in router 0 time-wait
    Original 192.168.1.2:20->1.1.1.10:49545 Packets=5 (264 Bytes)
    Reply    192.168.212.19:49545->192.168.1.2:20 Packets=4 (164 Bytes) [STEADY]
[DNAT AWAITED] FROM [SNAT rul 1]
Summary:
  Number of sessions (match/total): 2/4
AFS+
```

The following information is provided:

(1) The amount of packets a session creates depending on the scenario selected.

(2) The total number of entries, local IP and global IP will only appear if a dynamic scenario without overload is created. Despite this change, the static and dynamic scenarios (with or without overload) will show the same data.

(3) In this example, we can see how 2 sessions have been created. Since the example illustrates a dynamic scenario without overload, the total number of entries is shown.

Command history:

Release	Modification
11.01.08	All the information is shown even when a non-dynamic overload scenario is created as per version 11.01.08.

3.1.6 LIST SESSIONS

Displays the sessions that are currently active in the device. It allows you to specify diverse *<filter>* selection criteria, linked through the *and* clause.

Syntax:

```
AFS+LIST SESSIONS [<filter> [and <filter> [and ...]]]
<filter>:
      ipv4 [destination-address <dstaddr> [<dstmask>]]
           [source-address <srcaddr> [<srcmask>]]
      ipv6 [destination-address {<dstaddr> | <dstpref>}]
           [source-address {<srcaddr> | <srcpref>}]
      match <pattern>
      nat [destination | source] [rule <rule>]
      protocol {<protonum> |
                icmp [type <icmptype>] [code <icmpcode>] |
                tcp [destination-port <dstport>] [source-port <srcport>] |
                udp [destination-port <dstport>] [source-port <srcport>]}
AFS+
```

ipv4		Only selects IPv4 protocol sessions.
	destination-address	Only selects sessions with a destination address within the indicated network.
	<dstaddr>	Destination IP address.
	<dstmask>	Destination IP network mask. If this is not specified, only sessions with a destination address <dstaddr> are selected.
	source-address	Only selects sessions with a source address within the indicated network.
	<srcaddr>	Source IP address.
	<srcmask>	Source IP network mask. If this is not specified, only sessions with a source address <srcaddr> are selected.
ipv6		Only selects IPv6 protocol sessions.
	destination-address	Only selects sessions with a destination address within the indicated network.
	<dstaddr>	Destination IP address.
	<dstpref>	Destination IP network.
	source-address	Only selects sessions with a source address within the indicated network.
	<srcaddr>	Source IP address.
	<srcpref>	Source IP network.

match <pattern>		Only selects sessions where the **<pattern>** text is going to be shown. This criteria penalizes the speed at which the command is executed. If there are thousands of sessions established, the search can take several seconds (or even minutes). However, you can cancel this, at any point, by pressing a key.
nat		Only selects sessions where NAT is applied.
	destination	Only selects sessions where destination NAT is carried out.
	source	Only selects sessions where source NAT is carried out.
	<rule>	Only selects sessions that correspond to the NAT rule indicated.
protocol		Only selects sessions for the indicated protocol.
	<protonum>	Only selects sessions for the numerically indicated protocol.
	icmp	Only selects ICMP protocol sessions.
	<icmptype>	Type of ICMP packet.
	<icmpcode>	ICMP packet code.
	tcp	Only selects TCP protocol sessions.
	udp	Only selects UDP protocol sessions.
	<dstport>	Destination port for TCP or UDP sessions.
	<srcport>	Source port for TCP or UDP sessions.

Example 1:

Lists all sessions.

```
AFS+lis sessions
Protocol udp(17) dying in 30 secs, packets in router 0, appId L4:520
    Original 172.26.1.1:520->172.26.0.0:520 Packets=453 (59796 Bytes) [UNREPLIED]
    Reply    172.26.0.0:520->172.26.1.1:520 Packets=0 (0 Bytes) mark=0
AFS+
```

Example 2:

Lists TCP sessions.

```
AFS+list sessions protocol tcp
Protocol tcp(6) dying in 73 secs, packets in router 0, appId L4:4662 TIME_WAIT
    Original 201.240.220.14:15107->83.55.11.83:4662 Packets=7 (470 Bytes)
    Reply    172.24.100.130:4662->201.240.220.14:15107 Packets=11 (673 Bytes) [STEADY] mark=0
Protocol tcp(6) dying in 431999 secs, packets in router 0, appId L4:56597 ESTABLISHED
    Original 172.24.100.130:2903->217.216.188.178:56597 Packets=45 (1955 Bytes)
    Reply    217.216.188.178:56597->83.55.11.83:2903 Packets=44 (1904 Bytes) [STEADY] mark=0
Protocol tcp(6) dying in 57 secs, packets in router 0, appId L4:80 TIME_WAIT
    Original 172.24.100.130:4817->193.110.128.210:80 Packets=5 (967 Bytes)
    Reply    193.110.128.210:80->83.55.11.83:4817 Packets=5 (413 Bytes) [STEADY] mark=0
AFS+
```

3.1.7 HISTORY GRAPH

The number of AFS sessions registered during the last 48 hours can be shown in a graph that allows the user to define the range of time in which these sessions are displayed. The command can be executed on its own, showing the maximum period of time available (up to 48 hours) since the AFS feature was enabled, or two options can be configured to display a specific period of time.

Syntax:

```
AFS+history-graph ?
  <1..48>    Oldest hour of the history to be showed
  <cr>
AFS+history-graph 48 ?
  <0..47>    Most recent hour of the history to be showed
  <cr>
AFS+history-graph 48 47 ?
  <cr>
AFS+
```

The graph displayed shows the number of sessions registered in the frame time defined. Here is an example:

```
AFS+history-graph 24 23
AFS sessions from 24 to 23 hours ago.

 - Maximum number of sessions registered:     60
 - Minimum number of sessions registered:      0

    60_
    57_| :: :: ::  :: :: :: :: .:: :: :: :: ::  :: :: :: ::  :: :: :
    54_| :: :: ::  :: :: :: :: ::: :: :: :: ::  :: :: :: ::  :: :: :
    51_| :: :: ::  :: :: :: :: ::: :: :: :: ::: :: :: :: ::  :: :: :
    48_| :: :: :: .:: :: :: :: ::: :: :: :: ::: :: :: :: :: ::: :: :
    45_| :: :: ::.::: :: :: :: ::: :: :: :: ::: :: :: :: :: ::: :: :
    42_| :: :: :::::: :: :: :: ::: :: :: :: ::: :: :: :: :::::: :: :
    39_| :: :: :::::: :: :: :: ::: :: :: :: :::::: :: :: :::::: :: :
    36_| :: :: :::::: :: :: :: ::: :: :: :: :::::: :: :: :::::: :: :
    33_| :: :: :::::: :: :: :::::: :: :: :: :::::: :: :: :::::: :: :
    30_| :: :: :::::: :: :: ::::::.:: :: :: :::::: :: :: :::::: :: :
    27_| :: :: :::::: :: :: ::::::::: :: :: :::::: :: :: :::::: :: :
    24_| :: :: :::::: :: :: ::::::::: :: ::::::::: :: :: :::::: :: :
    21_| :: :: :::::: :: :: ::::::::: :: ::::::::: :: :: ::::::::: :
    18_| :: :: ::::::::: :: ::::::::: :: ::::::::: :: :: ::::::::: :
    15_| :: :::::::::::: :: ::::::::: :: ::::::::: :: :::::::::::: :
    12_| :: :::::::::::: :: ::::::::: :: :::::::::.:: :::::::::::: :
     9_|::: :::::::::::: :: ::::::::: :: :::::::::::: :::::::::::: :
     6_|::: :::::::::::: :::::::::::: :: :::::::::::: :::::::::::: :
     3_|::: :::::::::::: :::::::::::: :: :::::::::::: :::::::::::: :
     0_|:::.::::::::::::.:::::::::::::::.::::::::::::.::::::::::::.:
   24h.                                                        23h

AFS+
```

Command history:

Release	Modification
11.01.08	The "*history-graph*" command was introduced as of version 11.01.08.

3.1.8 TOP SESSIONS

Allows you to measure AFS sessions and see which ones are generating the most traffic.

Syntax:

```
AFS config$ AFS+top sessions ?
  match       Session monitor criteria
  number      Number of sessions to monitor
  original    Original direction of the session
  packets     Match packets/seconds instead of bytes
  reverse     Reverse direction of the session
  seconds     Number of seconds to monitor
  <cr>        Start measuring
AFS+
```

match	Allows you to specify a string. Only sessions containing this string are taken into account for evaluation purposes.
number	Allows you to specify the number of sessions to print. Default is 10.
original	Only takes into account traffic transmitted from the session source to the destination when executing classification.
packets	If you specify this option, measuring is executed taking into account packets per second and not bytes per second.
reverse	Only takes into account traffic transmitted from the session destination to the source when executing classification.
seconds	Specifies the time during which the measurement is carried out. Default is 5 seconds.

Example:

Measuring 3 sessions to see which one transmitted the most traffic in 5 seconds.

```
AFS+top sessions number 3
Measuring traffic each 5 seconds: .....
1) Overall 10185/17 Original: 9945/11 Reverse: 240/6 (bps/pps)
Protocol tcp(6) dying in 432000 secs, packets in router 3, appId L4:15387 ESTABLISHED
    Original 172.24.100.130:4461->83.194.103.87:15387 Packets=56 (49728 Bytes)
    Reply    83.194.103.87:15387->83.55.11.83:4461 Packets=30 (1200 Bytes) [STEADY] mark=0
2) Overall 5360/6 Original: 4104/4 Reverse: 1256/2 (bps/pps)
Protocol tcp(6) dying in 432000 secs, packets in router 0, appId L4:58133 ESTABLISHED
    Original 172.24.100.130:4154->84.121.138.76:58133 Packets=20 (20520 Bytes)
    Reply    84.121.138.76:58133->83.55.11.83:4154 Packets=12 (6280 Bytes) [STEADY] mark=0
3) Overall 3720/3 Original: 1704/2 Reverse: 2016/1 (bps/pps)
Protocol tcp(6) dying in 431999 secs, packets in router 2, appId L4:4661 ESTABLISHED
    Original 172.24.100.130:4328->84.76.226.203:4661 Packets=10 (8520 Bytes)
    Reply    84.76.226.203:4661->83.55.11.83:4328 Packets=9 (10080 Bytes) [STEADY] mark=0
```

Chapter 4 Examples

4.1 Firewall

We want to configure a firewall that only allows outgoing (not incoming) connections through a PPP interface.

Use FTP by loading its ALG, so incoming data connections pertaining to FTP are acknowledged and permitted (as they are associated to an outgoing connection).

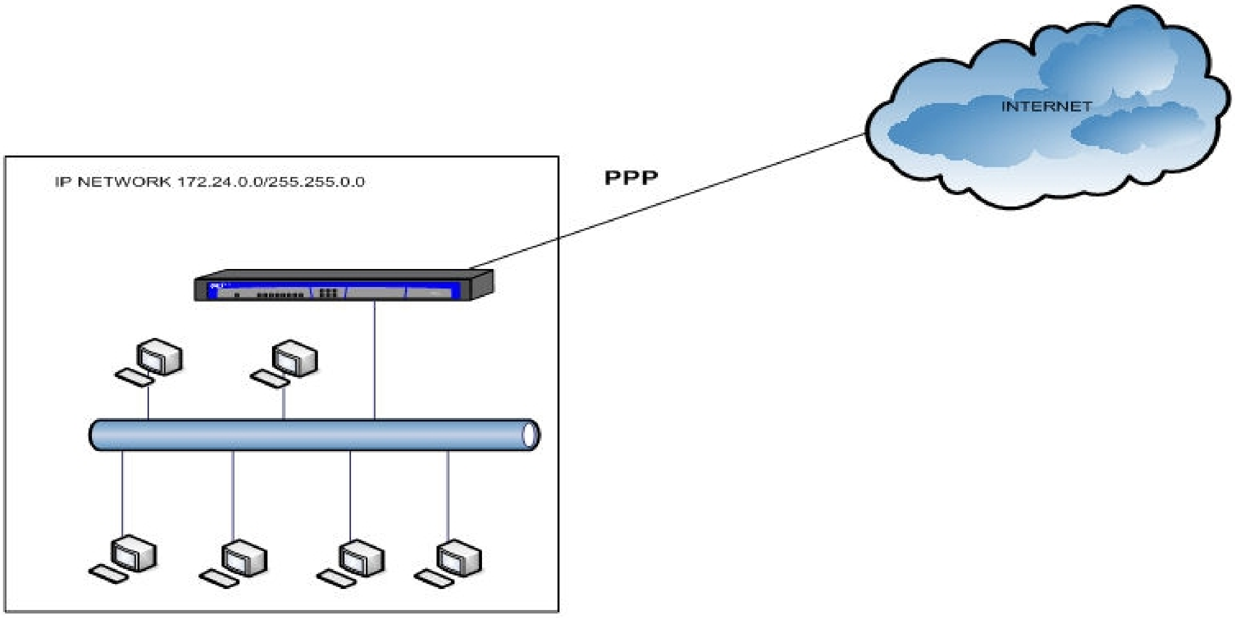


Fig. 2: Firewall configuration

Configuration:

```
   feature afs
      enable
   exit
;
   feature access-lists
; -- Access Lists user configuration --
      access-list 5000
         entry 1 deny
         entry 1 session state new
;
         entry 2 permit
;
      exit
;
   exit
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.24.1.21 255.255.0.0
;
;
;
;
;
   exit
;
   network serial0/0
; -- Interface Synchronous Serial Line. Configuration --
      mtu 1500
      speed 256000
   exit
;
```

```
;
   network ppp1
; -- Generic PPP User Configuration --
      ip access-group 5000 in
;
      ip address unnumbered
;
;
;
;
;
      ppp
; -- PPP Configuration --
         ipcp local address assigned
      exit
;
      base-interface
; -- Base Interface Configuration --
         base-interface serial0/0 link
;
      exit
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 ppp1
;
   exit
```

4.2 Prioritizing VoIP Packets

An office has IP telephones and softphones that use SIP to make external calls through the Internet. We want to prioritize VoIP traffic with identifier 20 in the IP header tos field above all other traffic.

To do this, activate the AFS system and configure a route-map (see manual *bintec Dm764-I Route Map*) that marks both the UDP packets with source port 5060 (SIP traffic) and the UDP flows the AFS system classifies as RTP.

Then, configure BRS in the outgoing PPP interface so that it prioritizes the aforementioned marked traffic (see manual *bintec Dm715-BRS*).

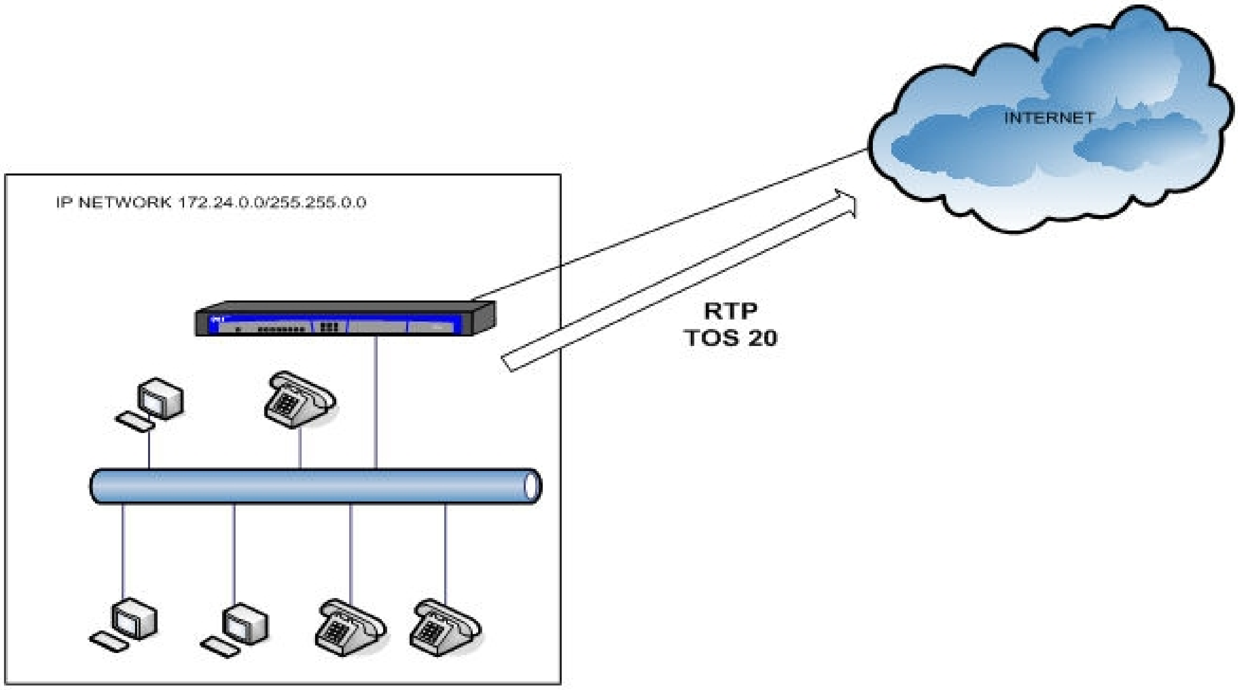


Fig. 3: Prioritizing VoIP packets

Configuration:

```
   feature afs
      enable
   exit
;
   feature access-lists
; -- Access Lists user configuration --
      access-list 201
         entry 1 default
         entry 1 permit
         entry 1 tos-octet 20
;
      exit
      access-list 5000
         entry 1 permit
         entry 1 rtp
;
         entry 2 permit
         entry 2 source udp port 5060
;
         entry 3 deny
      exit;
   exit
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.24.1.21 255.255.0.0
;
      ip policy route-map mark-rtp
;
   exit
;
   network serial0/0
; -- Interface Synchronous Serial Line. Configuration --
      mtu 1500
      speed 256000
   exit
;
;
   network ppp1
; -- Generic PPP User Configuration --
      ip address unnumbered
;
;
;
;
;
      ppp
; -- PPP Configuration --
         ipcp local address assigned
      exit
;
      base-interface
; -- Base Interface Configuration --
         base-interface serial0/0 link
;
      exit
;
   exit
;
   feature route-map
; -- Route maps user configuration --
      route-map "mark-rtp"
         entry 1 default
         entry 1 permit
         entry 1 match ip address 5000
         entry 1 set ip tos-octet 20
```

```
;
      exit
;
   exit
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 ppp1
;
   exit
   feature bandwidth-reservation
; -- Bandwidth Reservation user configuration --
      network ppp1
         enable
         class local 10
;
         class default 40
;
         class voip 100 real-time
;
;
         access-list 201 voip
;
         queue-length 32 5
      exit
;
   exit
```

4.3 Reactive load balancer

An office has two available lines for Internet connection. We want to connect the office network to both lines at the same time to use the whole Internet bandwidth.

Interface ethernet0/2 is connected to the LAN, while Interfaces ethernet0/0 and ethernet0/1 are connected to the Internet. In addition, NAT is configured for these two interfaces.

The link via ethernet0/0 supports a bandwidth of up to 10 Mbps and is considered primary. The one via ethernet0/1 supports a bandwidth of up to 3 Mbps and is considered secondary. In this hypothetical scenario, sending traffic via ethernet0/0 is preferred to sending it via ethernet0/1.

This is what happens in the above scenario:

- When the input bandwidth through the ethernet0/0 interface is lower than 8 Mbps, all new AFS sessions are selected to be sent through this interface.
- If the input bandwidth in ethernet0/0 ranges between 8 Mbps and 9 Mbps, and the input bandwidth in the ethernet0/1 interface is lower than 1 Mbps, all new AFS sessions are selected to be sent through both interfaces in Round-robin mode.
- When the input bandwidth in the ethernet0/0 interface exceeds 9 Mbps and the input bandwidth in the ethernet0/1 interface is lower than 1 Mbps, all new AFS sessions are selected to be sent through the latter.
- If input bandwidth in ethernet0/0 remains between 8 Mbps and 9 Mbps, and the input bandwidth in the ethernet0/1 interface increases until it ranges between 1 Mbps and 2 Mbps, all new AFS sessions are selected to be sent through the former.
- If the input bandwidth in the ethernet0/0 interface exceeds 9 Mbps and the input bandwidth in the ethernet0/1 interface ranges between 1 Mbps and 2 Mbps, all new AFS sessions are selected to be sent through both interfaces in Round-robin mode.
- Lastly, when the input bandwidth in the ethernet0/1 interface exceeds 2 Mbps, and is over 9 Mbps in the other one, all new AFS sessions are selected to be sent through the ethernet0/0 interface.

Information on how to measure bandwidth in an interface and how to report it can be found in manual *bintec Dm795-Policy Map-Class Map*.

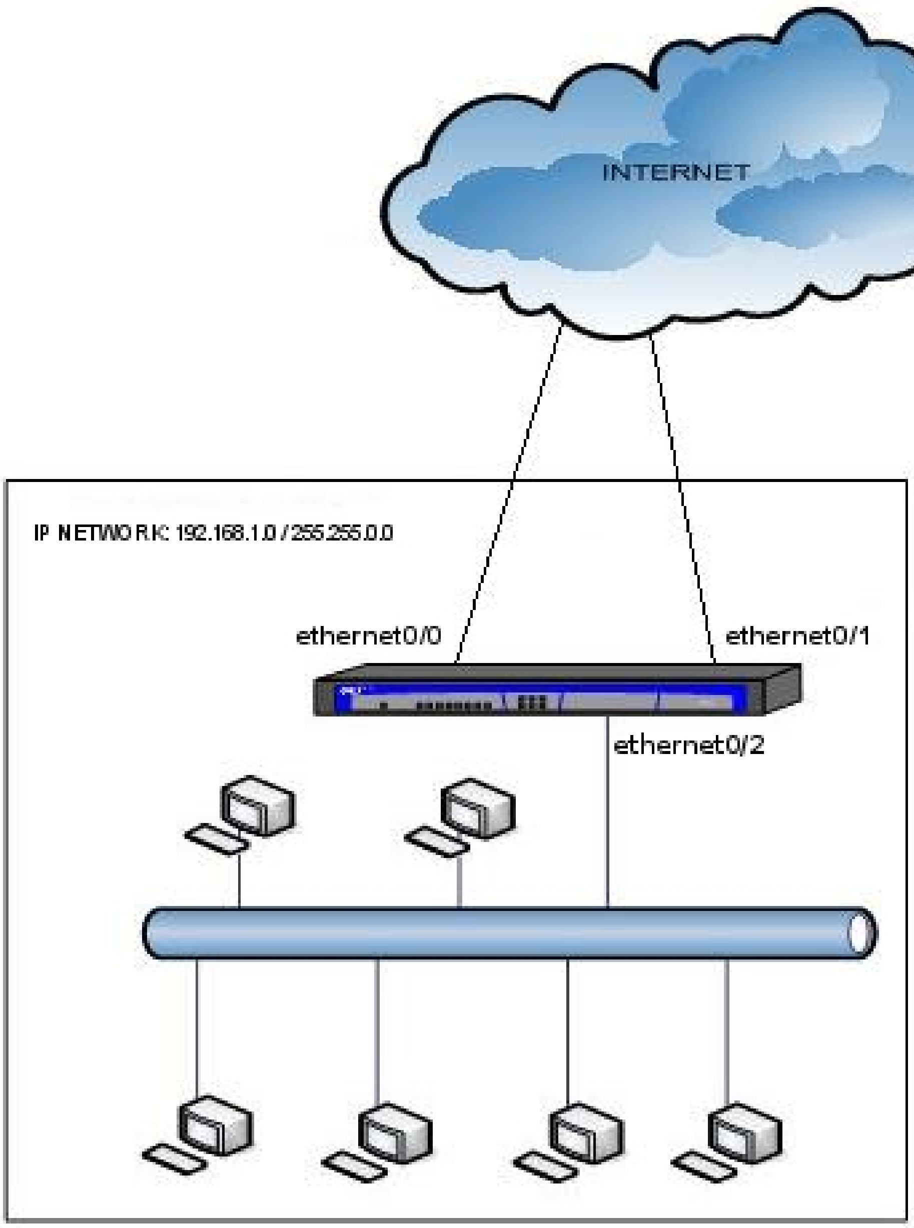


Fig. 4: Reactive load balancer

Configuration:

```
   feature afs
      enable
   exit
;
   feature class-map
; -- Class-Map Menu Configuration --
      class-map “INPUT-BANDWIDTH”
         entry 1 default
         entry 1 permit
;
      exit
;
   exit
;
   feature policy-map
; -- Policy-Map Menu Configuration --
      policy-map “ETH0-BANDWIDTH”
         class INPUT-BANDWIDTH configuration
            timer-bps 1
            report rate-kbps nsla-filter 1
            report rate-kbps nsla-filter 2
         exit
;
      exit
;
      policy-map “ETH1-BANDWIDTH”
         class INPUT-BANDWIDTH configuration
            timer-bps 1
            report rate-kbps nsla-filter 3
            report rate-kbps nsla-filter 4
         exit
;
      exit
;
   exit
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 87.32.128.17 255.255.255.252
;
      service-policy ETH0-BANDWIDTH input
;
;
   exit
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address ip address 212.14.2.30 255.255.255.252
;
      service-policy ETH1-BANDWIDTH input
;
   exit
;
   network ethernet0/2
; -- Ethernet Interface User Configuration --
      ip address 192.168.1.1 255.255.255.0
;
   exit
;
;
;
;
;
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 87.32.128.18 mp-congestion 1
      route 0.0.0.0 0.0.0.0 212.14.2.29  mp-congestion 2 mp-disable 3
;
      multipath per-afs-session
      nat
         rule 1 out ethernet0/0 dynamic overload
         rule 1 translation source interface ethernet0/0
;
         rule 2 out ethernet0/1 dynamic overload
         rule 2 translation source interface ethernet0/1
;
      exit
;
   exit
;
;
;
;
   feature nsla
; -- Feature Network Service Level Advisor --
```

```
      enable
;
      filter 1 description ETH0-BANDWIDTH-LOW
      filter 1 generic-input
      filter 1 significant-samples 1
      filter 1 activation threshold 8000
      filter 1 activation sensibility 100
      filter 1 activation stabilization-time 0
      filter 1 deactivation threshold 8000
      filter 1 deactivation sensibility 100
      filter 1 deactivation stabilization-time 0
;
      filter 2 description ETH0-BANDWIDTH-HIGH
      filter 2 generic-input
      filter 2 significant-samples 1
      filter 2 activation threshold 9000
      filter 2 activation sensibility 100
      filter 2 activation stabilization-time 0
      filter 2 deactivation threshold 9000
      filter 2 deactivation sensibility 100
      filter 2 deactivation stabilization-time 0
;
      filter 3 description ETH1-BANDWIDTH-LOW
      filter 3 generic-input
      filter 3 significant-samples 1
      filter 3 activation threshold 1000
      filter 3 activation sensibility 100
      filter 3 activation stabilization-time 0
      filter 3 deactivation threshold 1000
      filter 3 deactivation sensibility 100
      filter 3 deactivation stabilization-time 0
;
      filter 4 description ETH1-BANDWIDTH-HIGH
      filter 4 generic-input
      filter 4 significant-samples 1
      filter 4 activation threshold 2000
      filter 4 activation sensibility 100
      filter 4 activation stabilization-time 0
      filter 4 deactivation threshold 2000
      filter 4 deactivation sensibility 100
      filter 4 deactivation stabilization-time 0
;
      alarm 1 filter-id 1
;
      alarm 2 filter-id 2
;
      alarm 3 filter-id 3
;
      alarm 4 filter-id 4
;
      advisor 1 alarm-id 2
;
      advisor 2 alarm-id 3
;
      advisor 3 not alarm-id 1
      advisor 3 or alarm-id 4
;
   exit
```

The way configured advisors behave determines the reactivity of the load balancer.

At the beginning, when the router is switched on and no traffic has been transmitted yet, bandwidth through the ethernet0/0 interface is lower than 8 Mbps. This means that the output of advisor 1 (which represents congestion in the ethernet0/0 link) is false, whilst the output of advisor 3 (which represents the disabling of the second link to the Internet, i.e. ethernet0/1) is true. This means the first route is not congested and the second route is disabled and not congested. As a result, all new AFS sessions are created with IP 87.32.128.18 as next hop and sent through the ethernet0/0 link. [CASE 1]

After the router has been operating for a while, bandwidth through the ethernet0/0 interface may range between 8 Mbps and 9 Mbps. If this happens, the output of advisor 1 (which represents congestion in the ethernet0/0 link) remains false and the output of advisor 3 (which represents the disabling of the second link to the Internet, i.e. ethernet0/1) becomes false too. Moreover, since the bandwidth of the ethernet0/1 interface is lower than 1 Mbps and is polled by advisor 2 (which represents the congestion of the ethernet0/1 link) its output is false as well. This means that the first route is not congested and that the second route is neither disabled nor congested. As a result, each new AFS session created selects the next-hop to use in Round-Robin mode, with IPs 87.32.128.18 and 212.14.2.29. These sessions are sent through the ethernet0/0 and ethernet0/1 links alternatively. [CASE 2]

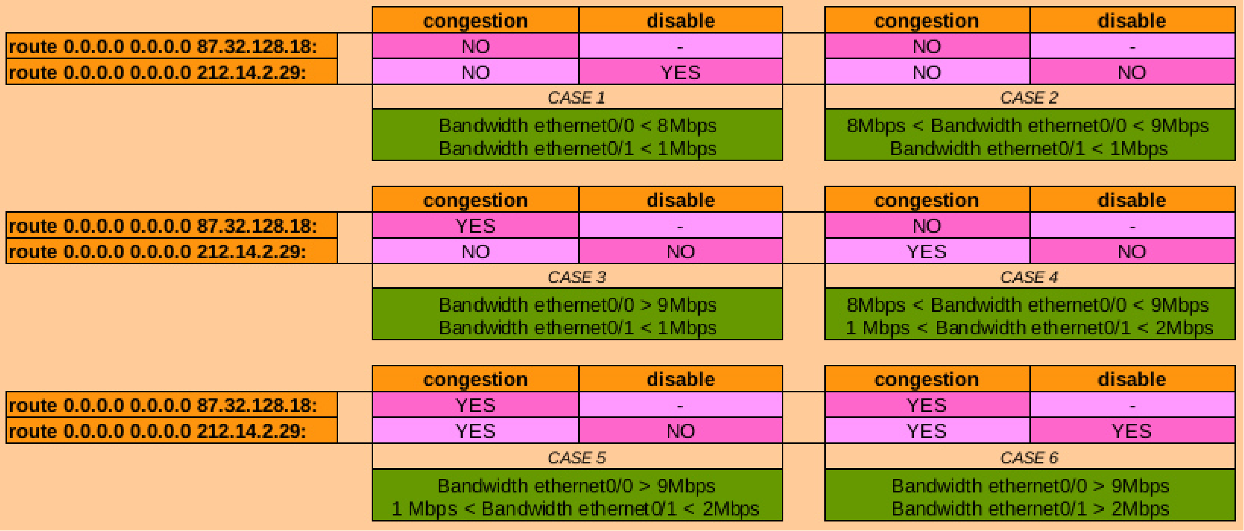
Some minutes later, bandwidth through the ethernet0/0 interface may exceed the breakpoint of 9 Mbps. If this happens, the output of advisor 1 (which represents congestion in the ethernet0/0 link) is true, whilst the output of advisor 3 (which represents the disabling of the second link to the Internet, i.e. ethernet0/1) remains false. Moreover, since the bandwidth of the ethernet0/1 interface is lower than 1 Mbps, the output of advisor 2 (which represents the congestion of the ethernet0/1 link) remains false as well. This means that the first route is congested and the second route is neither disabled nor congested. As a result, all new AFS sessions are created with IP 212.14.2.29 as next hop and sent through the ethernet0/1 link. [CASE 3]

As a deviation from the previous paragraph, let's imagine the bandwidth through the ethernet0/1 interface increases until it ranges between 1 Mbps and 2 Mbps, while the ethernet0/0 interface remains between 8 Mbps and 9 Mbps. If this happens, the output of advisor 1 (which represents congestion in the ethernet0/0 link) is false and the output of advisor 3 (which represents the disabling of the second link to the Internet, i.e. ethernet0/1) is false as well. However, since the bandwidth of the ethernet0/1 interface exceeds 1 Mbps, the output of advisor 2 (which represents the congestion of the ethernet0/1 link) is true. This means that the first route is not congested and that the second route is congested and not disabled. As a result, all new AFS sessions are created with IP 87.32.128.18 as next hop and are sent through the ethernet0/0 link. [CASE 4]

As time goes by, the bandwidth needs of the office may increase slightly. As a result, the bandwidth through the ethernet0/1 interface may range between 1 Mbps and 2 Mbps, while the ethernet0/0 interface exceeds the threshold of 9 Mbps. At this point, the output of advisor 1 (which represents congestion in the ethernet0/0 link) is true, whilst the output of advisor 3 (which represents the disabling of the second link to the Internet, i.e. ethernet0/1) remains false. Moreover, since the bandwidth of the ethernet0/1 interface exceeds 1 Mbps, the output of advisor 2 (which represents the congestion of the ethernet0/1 link) is true. This means that the first route is congested and the second route is congested and not disabled. As a result, each new AFS session created selects the next-hop to use in Round-robin mode, with IPs 87.32.128.18 and 212.14.2.29. These sessions are then sent through the ethernet0/0 and ethernet0/1 links alternatively. [CASE 5]

At some point, corporate bandwidth needs may reach peak value and the bandwidth through the ethernet0/1 interface can exceed the breakpoint of 2 Mbps, while the ethernet0/0 interface already surpasses the record of 9 Mbps. In this case, the output of advisor 1 (which represents congestion in ethernet0/0 link) is true while the output of advisor 3 (which represents the disabling of the second link to the Internet, i.e. ethernet0/1) also becomes true. Moreover, since the bandwidth of the ethernet0/1 interface exceeds 1 Mbps, the output of advisor 2 (which represents the congestion of the ethernet0/1 link) is true as well. This means that the first route is congested and that the second route is both congested and disabled. As a result, all new AFS sessions are created with IP 87.32.128.18 as next hop and are sent through the ethernet0/0 link. [CASE 6]

All of the above is summarized in the following table:



	congestion	disable	congestion	disable
route 0.0.0.0 0.0.0.0 87.32.128.18:	NO	-	NO	-
route 0.0.0.0 0.0.0.0 212.14.2.29:	NO	YES	NO	NO
	CASE 1		*CASE 2*	
	Bandwidth ethernet0/0 < 8Mbps Bandwidth ethernet0/1 < 1Mbps		8Mbps < Bandwidth ethernet0/0 < 9Mbps Bandwidth ethernet0/1 < 1Mbps	

	congestion	disable	congestion	disable
route 0.0.0.0 0.0.0.0 87.32.128.18:	YES	-	NO	-
route 0.0.0.0 0.0.0.0 212.14.2.29:	NO	NO	YES	NO
	CASE 3		*CASE 4*	
	Bandwidth ethernet0/0 > 9Mbps Bandwidth ethernet0/1 < 1Mbps		8Mbps < Bandwidth ethernet0/0 < 9Mbps 1 Mbps < Bandwidth ethernet0/1 < 2Mbps	

	congestion	disable	congestion	disable
route 0.0.0.0 0.0.0.0 87.32.128.18:	YES	-	YES	-
route 0.0.0.0 0.0.0.0 212.14.2.29:	YES	NO	YES	YES
	CASE 5		*CASE 6*	
	Bandwidth ethernet0/0 > 9Mbps 1 Mbps < Bandwidth ethernet0/1 < 2Mbps		Bandwidth ethernet0/0 > 9Mbps Bandwidth ethernet0/1 > 2Mbps	

Fig. 5: Cases 1 to 6

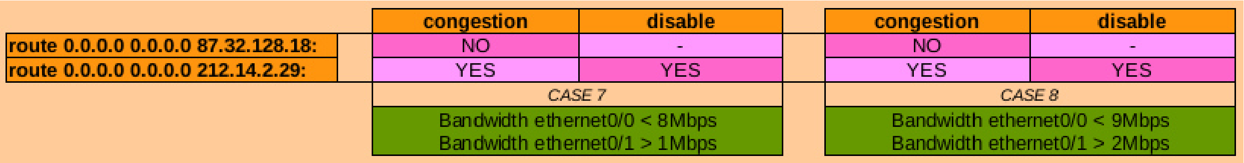
It is worth highlighting that, once a next-hop has been assigned to a session, the hop stays fixed regardless of the state of the other routes. I.e. If a new AFS session is allocated IP 212.14.2.29 as next hop because the route via 87.32.128.18 is congested and the one via 212.14.2.29 is neither disabled nor congested [CASE 3], and then the bandwidth in the ethernet0/0 link falls below 8 Mbps (meaning the route via 87.32.128.18 is no longer congested and the one via 212.14.2.29 is disabled) [CASE 1], this change on states will only affect future AFS sessions. The previously generated session, selected to be sent via 212.14.2.29, will not be moved.

In addition to the previously mentioned states, two more can occur. These last two states are the result of immovable hops (once an AFS session has been assigned a next-hop, the latter is never modified except when deleting a route in the multipath during operation) and unpredictable variations in bandwidth evolution (unexpected increases and decreases). A single high-bandwidth demanding AFS session could use up as much bandwidth as dozens of regular ones. As a result, in some cases, the secondary link's whole bandwidth can be used up while the primary link is not fully exploited.

The first of these states occurs when the bandwidth through which the ethernet0/1 interface unexpectedly exceeds 1 Mbps, or even surpasses the breakpoint of 2 Mbps, while the bandwidth in the ethernet0/0 interface falls under 8 Mbps. In this state, the output of advisor 1 (which represents congestion in the ethernet0/0 link) is false, while the output of advisor 3 (which represents the disabling of the second link, i.e. ethernet0/1) is true. Moreover, since the bandwidth of the ethernet0/1 interface exceeds 1 Mbps, the output of advisor 2 (which represents the congestion of the ethernet0/1 link) is true as well. This means that the first route is not congested and that the second route is both congested and disabled. As a result, all new AFS sessions are created with IP 87.32.128.18 as next hop and are sent through the ethernet0/0 link. [CASE 7]

The second of these states occurs when the bandwidth through the ethernet0/1 interface unexpectedly exceeds the breakpoint of 2 Mbps, while the bandwidth in the ethernet0/0 interface falls under 9 Mbps (or even 8 Mbps). At this point, the output of advisor 1 (which represents congestion in the ethernet0/0 link) is false, whilst the output of advisor 3 (which represents the disabling of the second link, i.e. ethernet0/1) is true. Moreover, since the bandwidth of the ethernet0/1 interface is over 1 Mbps, the output of advisor 2 (which represents the congestion of the ethernet0/1 link) is true as well. This means that the first route is not congested and the second route is both congested and disabled. As a result, all new AFS sessions are created with IP 87.32.128.18 as next hop and are sent through the ethernet0/0 link. [CASE 8]

These two possible states are summarized in the following table:



	congestion	disable	congestion	disable
route 0.0.0.0 0.0.0.0 87.32.128.18:	NO	-	NO	-
route 0.0.0.0 0.0.0.0 212.14.2.29:	YES	YES	YES	YES
	CASE 7		CASE 8	
	Bandwidth ethernet0/0 < 8Mbps Bandwidth ethernet0/1 > 1Mbps		Bandwidth ethernet0/0 < 9Mbps Bandwidth ethernet0/1 > 2Mbps	

Fig. 6: Cases 7 and 8

Finally, it is worth clarifying that, if one of the routes in the multipath is deleted during operation, the AFS sessions that belong to the next hop of this deleted route are assigned to the next-hop of one of the remaining routes when a packet of each session is rerouted. This means that, if a new AFS session is assigned to next-hop 212.14.2.29 because the route via 87.32.128.18 is congested and the one via 212.14.2.29 is neither disabled nor congested [CASE 3], and then the ethernet0/1 interface is shut down, when a packet belonging to said session and with destination address on the Internet reaches the router, the latter will realize next-hop 212.14.2.29 does not exist and will assign hop 87.32.128.18 to the session.