Configuring Application Assurance with CLI

This section provides information to configure Application Assurance entities using the command line interface. It is assumed that the user is familiar with basic configuration of policies.

Provisioning AA ISA MDA

The following illustrates syntax to provision AA ISA and configure ingress IOM QoS parameters. (The egress IOM QoS is configured in the **config>isa>application-assurance-grp>qos** context.)

CLI Syntax:

```
configure>card>mda mda-slot
 mda-type isa-aa
    network
       ingress
          pool
             slope-policy slope-policy-name
             resv-cbs percent-or-default
          queue-policy network-queue-policy-name
```

The following output displays AA ISA configuration example.

```
*A:cpm-a>config>app-assure# show mda 1/1
===============================================================================
MDA 1/1
===============================================================================
Slot  Mda   Provisioned           Equipped              Admin     Operational
            Mda-type              Mda-type              State     State
-------------------------------------------------------------------------------
1     1     isa-aa                isa-ms                up        up
===============================================================================
*A:cpm-a>config>app-assure#


*A:cpm-a>config>card# info
----------------------------------------------
        card-type iom-20g-b
        mda 1
            mda-type isa-aa
        exit
----------------------------------------------
*A:cpm-a>config>card#
```

Configuring an AA ISA Group

To enable AA on the router:

- Create an AA ISA group.
- Assign active and optional backup AA ISA(s) to an AA ISA group.
- Select the forwarding classes to divert.
- Enable the group.
- Optionally:
  - → Enable group policy partitioning
  - → Configure capacity cost threshold values
  - → Configure the number of transit prefix IP policies
  - → Configure IOM egress queues to the MS-ISA
  - → Enable overload cut through and configure the high and low watermarks values
  - → Configure performance statistics accounting

The following example illustrates AA ISA group configuration with:

- Primary AA ISA and warm redundancy provided by the backup AA ISA.
- “fail-to-wire” behavior configured on group failure.
- BE forwarding class selected for divert.
- Default IOM QoS for logical ISA egress ports. The ISA ingress QoS is configured as part of ISA provisioning (**config>card>mda>network>ingress>qos**).

The following commands illustrate AA ISA group configuration context.

CLI Syntax:

```
config>>isa>application-assurance-group isa-aa-group-id [aa-
      sub-scale {residential|vpn}] [create]
            backup mda-id
            description description
            divert-fc fc-name
            no fail-to-open
            isa-capacity-cost-high-threshold threshold
            isa-capacity-cost-low-threshold threshold
            partitions
            primary mda-id
            qos
               egress
                  from-subscriber
                     pool [pool-name]
                     resv-cbs percent-or-default
```

```
         slope-policy slope-policy-name
      port-scheduler-policy port-scheduler-policy-name
      queue-policy network-queue-policy-name
      to-subscriber
         pool [pool-name]
            resv-cbs percent-or-default
            slope-policy slope-policy-name
         port-scheduler-policy port-scheduler-policy-name
         queue-policy network-queue-policy-name
[no] shutdown
```

The following output displays an AA ISA group configuration example.

```
A:ALU-A>config>isa>aa-grp# info detail
 ----------------------------------------------
     no description
     primary 1/2
     backup  2/2
     no fail-to-open
     isa-capacity-cost-high-threshold 4294967295
     isa-capacity-cost-low-threshold 0
     no partitions
     divert-fc be
     qos
        egress
          from-subscriber
               pool
                    slope-policy "default"
                    resv-cbs default
               exit
               queue-policy "default"
               no port-scheduler-policy
          exit
          to-subscriber
               pool
                    slope-policy "default"
                    resv-cbs default
               exit
               queue-policy "default"
               no port-scheduler-policy
          exit
        exit
     exit
     no shutdown
----------------------------------------------
A:ALU-A>config>isa>aa-grp#
```

Configuring Watermark Parameters

Use the following CLI syntax to configure thresholds for logs and traps when under high consumption of the flow table. The flow table has a limited size and these thresholds can be established to alert the user that the table is approaching capacity. These flow table watermarks represent number of flow contexts allocated on the ISA, which will be slightly higher than the actual number of existing flows at the point when the watermark is reached.

The low threshold is used while the high threshold is used as an alarm.

CLI Syntax:

```
config>application-assurance
  flow-table-high-wmark high-watermark
  flow-table-low-wmark low-watermark
```

Configuring a Group Policy

Beginning and Committing a Policy Configuration

To enter the mode to create or edit Application Assurance policies, you must enter the **begin** keyword at the **`config>app-assure>group>policy`** prompt. The **commit** command saves changes made to policies during a session. Changes do not take affect in the system until they have performed the commit function. The **abort** command discards changes that have been made to policies during a session.

The following error message displays when creating or modifying a policy without entering **begin** first.

```
A:ALA-B>config>app-assure>group>policy#
MINOR: AA #1005 Invalid Set - Cannot proceed with changes when in non-
edit mode
```

There are no default policy options. All parameters must be explicitly configured.

Use the following CLI syntax to begin a policy configuration.

CLI Syntax:
```
config>app-assure# group group-id
 policy
    begin
```

Use the following CLI syntax to commit a policy configuration.

CLI Syntax:
```
config>app-assure# group group-id
 policy
    commit
```

Aborting a Policy Configuration

Use the following CLI syntax to abort a policy configuration.

CLI Syntax:
```
config>app-assure# group group-id
 policy
    abort
```

Configuring an IP Prefix List

An operator can use IP lists to define a list of IP addresses (along with any masks). This list can be later referenced in AQPs, application filters and/or session-filters.

Use the following CLI syntax to configure an application filter entry.

CLI Syntax:

```
Config>aa>group>policy>app-assurance>group <aa-group-
id>[:<partition>]
            ip-prefix-list <prefix-list-name> [create]
            no ip-prefix-list <prefix-list-name>
            description <description>
            no description
            prefix <address/mask> [name <prefix-name>]
            no prefix <address/mask>
```

```
*A:Dut-A>config>app-assure>group# ip-prefix-list AllowedLAN1Hosts create
*A:Dut-A>config>app-assure>group>pfx>$ description "allowed hosts"
*A:Dut-A>config>app-assure>group>pfx>$ prefix 10.10.8.2/32
*A:Dut-A>config>app-assure>group>pfx>$ prefix 10.10.8.180/32
*A:Dut-A>config>app-assure>group>pfx>$ prefix 10.10.8.231/32
*A:Dut-A>config>app-assure>group>pfx>$ exit
*A:Dut-A>config>app-assure>group#
```

```
*A:Dut-A>config>app-assure>group# ip-prefix-list "AllowedLan1Hosts"
*A:Dut-A>config>app-assure>group>pfx># info
----------------------------------------------
                description "allowed hosts"
                prefix 10.10.8.2/32
                prefix 10.10.8.180/32
                prefix 10.10.8.231/32
----------------------------------------------
*A:Dut-A>config>app-assure>group>pfx>#
```

Configuring AA Session Filters

Session filters can be configured to allow stateful firewall use-cases. Refer to AA Group Commands on page 216 for syntax and CLI descriptions.

CLI Syntax:
```
*A:Dut-A>config>app-assure>group# session-filter <session-
filter-name> [create]
            default-action {permit|deny} [event-log <event-log-name>]
            description <description-string>
            entry <entry-id> [create]
                  action {permit|deny} [event-log <event-log-name>]
                  match
                     dst-ip <ip-address>
                     dst-ip ip-prefix-list <ip-prefix-list-name>
                     no dst-ip
                     dst-port {eq|gt|lt} <port-num>
                     dst-port range <start-port-num> <end-port-num>
                     no dst-port
                     ip-protocol-num <ip-protocol-number>
                     no ip-protocol-num
                     src-ip <ip-address>
                     no src-ip
                     src-ip ip-prefix-list <ip-prefix-list>
                     src-port {eq|gt|lt} <port-num>
                     src-port range <start-port-num> <end-port-num>
                     no src-port
```

```
*A:Dut-A>config>app-assure>group# session-filter " denyUnsolictedwMgntCntrl" create
       description “S-FW opted-in sub - allow ISP access"
       default-action deny event-log “FW_log”
      entry 10 create
         description "allow ICMP access from ISP LAN#1"
         match
           ip-protocol-num icmp
           src-ip 10.10.8.0/24
         exit
         action permit
        exit
       entry 30 create
         description "allow all TCP (e.g. FTP/telnet)access from ISP LAN#2"
         match
           ip-protocol-num tcp
           src-ip 192.168.0.0/24
         exit
         action permit
       entry 40 create
         description "allow TCP on port 80 /HTTP access from a IP List on ISP LAN#1"
         match
           ip-protocol-num tcp
           src-ip ip-prefix-list AllowedLAN1Hosts
           dst-port eq 80
         exit
         action permit

       exit


*A:Dut-A>config>app-assure>group>sess-fltr$ info
----------------------------------------------
                description "S-FW opted-in sub . allow ISP access"
                default-action deny event-log “FW_Log”
                entry 10 create
                    description "allow ICMP access from ISP LAN#1"
                    match
                        ip-protocol-num icmp
                        src-ip 10.10.8.0/24
                    exit
                    action permit
                exit
                entry 20 create
                    description "allow ICMP access from ISP LAN#2"
                    action deny
                exit
                entry 30 create
                    description "allow all TCP (e.g. FTP/telnet)access from ISP LAN#2"
                    match
                        ip-protocol-num tcp
                        src-ip 192.168.0.0/24
                    exit
                    action permit
                exit
                entry 40 create
                    description "allow TCP on port 80 /HTTP access from a IP List on ISP 
LAN#1"
                    match
                        ip-protocol-num tcp
                        src-ip ip-prefix-list "AllowedLan1Hosts"
                        dst-port eq 80
                    exit
                    action permit
                exit
----------------------------------------------
*A:Dut-A>config>app-assure>group>sess-fltr$


*A:Dut-A>config>app-assure>group>policy>eqp>
   entry 110 create
     description “FW for managed opted-in subs”
       match
         traffic-direction network-to-subscriber
       exit
       action
           session-filter “ denyUnsolictedwMgntCntrl "
          fragment-drop all event-log "FW_log"
         error-drop event-log “FW_log”
         overload-drop

       exit
   exit
```

```
*A:Dut-A>config>app-assure>group>policy>aqp>entry# info
----------------------------------------------
                        description "FW for managed opted-in subs."
                        match
                            traffic-direction network-to-subscriber
                        exit
                        action
                            session-filter "denyUnsolictedwMgntCntrl"
                            fragment-drop all event-log "FW_log"
                            error-drop event-log “FW_log”
                            overload-drop

                        exit
                        no shutdown
----------------------------------------------
*A:Dut-A>config>app-assure>group>policy>aqp>entry#
```

Configuring an Application Group

An operator can configure an application group to group multiple application into a single application assurance entity by referencing those applications in the group created.

Use the following CLI syntax to configure an application group.

CLI Syntax:

```
config>app-assure>group>policy# app-group application-group-name [create]
            description description
```

The following example displays an application group configuration.

```
*A:ALA-48>config>app-assure>group>policy# app-group "Peer to Peer" create
*A:ALA-48>config>app-assure>group>policy>app-grp# info
----------------------------------------------
                    description "Peer to Peer file sharing applications"
----------------------------------------------
*A:ALA-48>config>app-assure>group>policy>app-grp#
```

Configuring an Application

An operator can configure an application to group multiple protocols, clients or network applications into a single Application Assurance application by referencing it later in the created application filters as display in other sections of this guide.

Use the following CLI syntax to configure an application.

CLI Syntax:

```
config>app-assure>group>policy# application application-name
[create]
            app-group app-group-name
            description description
```

The following example displays an application configuration.

```
*A:ALA-48>config>app-assure>group>policy# application "SQL" create
*A:ALA-48>config>app-assure>group>policy>app# info
----------------------------------------------
                    description "SQL protocols"
                    app-group "Business Critical Applications"
----------------------------------------------
*A:ALA-48>config>app-assure>group>policy>app#
```

Configuring an Application Filter

An operator can use an application filter to define applications based on ALU protocol signatures and a set of configurable parameters like IP flow setup direction, IP protocol number, server IP address and server TCP/UDP port. An application filter references an application configured as previously shown.

Use the following CLI syntax to configure an application filter entry.

CLI Syntax:

```
config>app-assure>group>policy# app-filter
 entry entry-id [create]
    application application-name
    description description-string
    expression expr-index expr-type {eq | neq} expr-string
    flow-setup-direction {subscriber-to-network | network-to-
       subscriber | both}
    http-match-all-requests
    ip-protocol-num {eq | neq} protocol-id
    network-address {eq | neq} ip-address
    network-address {eq | neq} ip-prefix-list ip-prefix-list-
       name
    protocol {eq | neq} protocol-signature-name
    server-address {eq | neq} ip-address
    server-address {eq | neq} dns-ip-cache dns-ip-cache-name
    server-address {eq | neq} ip-prefix-list ip-prefix-list-
       name
    server-port {eq | neq | gt | lt} server-port-number
    server-port {eq | neq} range start-port-num end-port-num
    server-port {eq} {port-num | range start-port-num end-
       port-num} first-packet-trusted|first-packet-validate}
    no shutdown
```

The following example displays an application filter configuration.

```
*A:ALA-48>config>app-assure>group>policy>app-filter# entry 30 create
*A:ALA-48>config>app-assure>group>policy>app-filter>entry# info
----------------------------------------------
                        description "DNS traffic to local server on expected port #53"
                        protocol eq "dns"
                        flow-setup-direction subscriber-to-network
                        ip-protocol-num eq *
                        server-address eq 192.0.2.0/32
                        server-port eq 53
                        application "DNS_Local"
                        no shutdown
----------------------------------------------
*A:ALA-48>config>app-assure>group>policy>app-filter>entry#
```

Configuring an Application Profile

Use the following CLI syntax to configure an application profile.

CLI Syntax:

```
config>app-assure>group>policy# app-profile app-profile-name
[create]
            characteristic characteristic-name value value-name
            description description-string
            [no] aa-sub-suppressible
            divert
```

The following example displays an application profile configuration.

```
*A:ALA-48>config>app-assure>group>policy# app-profile "Super" create
*A:ALA-48>config>app-assure>group>policy>app-prof# info
----------------------------------------------
                    description "Super User Application Profile"
                    divert
                    characteristic "Server" value "Prioritize"
                    characteristic "ServiceBw" value "SuperUser"
                    characteristic "Teleworker" value "Yes"
                    characteristic "VideoBoost" value "Priority"
----------------------------------------------
*A:ALA-48>config>app-assure>group>policy>app-prof#
```

Configuring Suppressible App-Profile with SRRP

For information about SRRP, refer to the 7750 SR OS Triple Play Guide.

In the context of an ESM SRRP deployment, the operator can define at the app-profile level if the subscriber will be diverted to the ISA-AA card per SRRP group interface. This can be useful to reduce the total number of ISA cards required in the event of a switch-over from a primary to backup SRRP node when AA is used as a value-add service for selected subscribers.

To configure the network for suppressible app-profiles in the context of SRRP the operator needs to:

- Enable the capability to suppress AA subscribers on a given SRRP group interface, typically by selecting backup SRRP group interfaces.
- ESM subscribers with a valid app-profile are diverted to AA by default, to suppress selected group of subscribers using AA for optional value-add services. The operator then specifies which app-profile will be suppressed and therefore not diverted to AA.

Use the following CLI syntax to enable the capability to suppress ESM subscribers from a backup SRRP group interface:

CLI Syntax:

```
config>service>vprn>sub-if>grp-if# suppress-aa-sub [create]
  characteristic characteristic-name value value-name
  description description-string
  [no] aa-sub-suppressible
  divert
```

The following example displays an application profile configuration used for premium subscribers, this type of subscriber will always be diverted to Application Assurance, it is also the default configuration:

```
7750>config>app-assure>group>policy# info
----------------------------------------------
                app-profile "Premium" create
                    characteristic "Parental-Control" eq "Yes"
                    divert
                exit
----------------------------------------------
```

The following example displays an application profile configuration for best effort / value-add subscribers not diverted to Application Assurance on the SRRP group interface configured with “suppress-aa-sub”:

```
7750>config>app-assure>group>policy# info
----------------------------------------------
                app-profile "1-default" create
                    divert
                    aa-sub-suppressible
                exit
----------------------------------------------
```

Configuring Application Service Options

Use the following CLI syntax to configure application service options.

CLI Syntax:

```
config>app-assure>group>policy# app-service-options
  characteristic characteristic-name [create]
     default-value value-name
     value value-name
```

The following example displays an application service options configuration.

```
*A:ALA-48>config>app-assure>group>policy>aso# info
----------------------------------------------
                    characteristic "Server" create
                        value "Block"
                        value "Permit"
                        value "Prioritize"
                        default-value "Block"
                    exit
                    characteristic "ServiceBw" create
                        value "Lite_128k"
                        value "Power_5M"
                        value "Reg_1M"
                        value "SuperUser"
                        default-value "Reg_1M"
                    exit
                    characteristic "Teleworker" create
                        value "No"
                        value "Yes"
                        default-value "No"
                    exit
                    characteristic "VideoBoost" create
                        value "No"
                        value "Priority"
                        default-value "No"
                    exit
----------------------------------------------
*A:ALA-48>config>app-assure>group>policy>aso#
```

Configuring a Policer

Use the following CLI syntax to configure a policer.

CLI Syntax:

```
config>app-assure>group>policy# policer policer-name type type
granularity granularity create
           action {priority-mark | permit-deny}
           adaptation-rule pir adaptation-rule
           description description-string
           mbs maximum burst size
           rate pir-rate
           tod-override tod-override-id [create]
```

The following example displays an Application Assurance policer configuration.

```
*A:ALA-48>config>app-assure>group# policer "RegDown_Policer" type dual-bucket-bandwidth
granularity subscriber create

*A:ALA-48>config>app-assure>group>policer# info
----------------------------------------------
                description "Control the downstream aggregate bandwidth for Regular 1Mbps
subscribers"
                rate 1000 cir 500
                mbs 100
                cbs 50
----------------------------------------------
*A:ALA-48>config>app-assure>group>policer#
```

Configuring an Application QoS Policy

Use the following CLI syntax to configure an application QoS policy.

CLI Syntax:

```
config>app-assure>group>policy# app-qos-policy
 entry entry-id [create]
    action
       bandwidth-policer policer-name
       drop
       error-drop [event-log event-log-name]
       flow-count-limit policer-name
       flow-rate-limit policer-name
       fragment-drop {all | out-of-order} [event-log event-
         log-name]
       http-error-redirect redirect-name
       mirror-source [all-inclusive] mirror-service-id
       overload-drop [event-log event-log-name]
       remark
          dscp in-profile dscp-name out-profile dscp-name
          fc fc-name
          priority priority-level
       url-filter url-filter-name characteristic characteris-
         tic-name
    description description-string
    match
       aa-sub sap {eq | neq} sap-id
       aa-sub esm {eq | neq} sub-ident-string
       aa-sub spoke-sdp {eq | neq} sdp-id:vc-id
       app-group {eq | neq} application-group-name
       application {eq | neq} application-name
       characteristic characteristic-name {eq} value-name
       dscp {eq | neq} dscp-name
       dst-ip {eq | neq} ip-address[/mask]
       dst-ip {eq | neq} ip-prefix-list ip-prefix-list-name
       dst-port {eq | neq} port-num
       dst-port {eq | neq} range start-port-num end-port-num
       src-ip {eq | neq} ip-address[/mask]
       src-ip {eq | neq} ip-prefix-list ip-prefix-list-name
       src-port {eq | neq} port-num
       src-port {eq | neq} range start-port-num end-port-num
       traffic-direction {subscriber-to-network | network-to-
         subscriber | both}
    no shutdown
```

The following example displays an application QoS policy configuration.

```
*A:ALA-48>config>app-assure>group>policy>aqp# entry 20 create
----------------------------------------------
                        description "Limit downstream bandwidth to Reg_1M subscribers"
                        match
                            traffic-direction network-to-subscriber
                            characteristic "ServiceBw" eq "Reg_1M"
                        exit
                        action
                            bandwidth-policer "RegDown_Policer"
                        exit
                        no shutdown
----------------------------------------------
*A:ALA-48>config>app-assure>group>policy>aqp#
```

The following example display an AQP entry configuration to mirror all positively identified only P2P traffic (AppGroup P2P) for a subset of subscribers with ASO characteristic **aa-sub-mirror** enabled.

```
A:ALA-48>config>app-assure>group>policy>aqp#
-------------------------------------------------------------------------------
   entry 100 create
     match
          app-group eq P2P
          characteristic aa-sub-mirror eq enabled
     exit
     action                       # mirror to an existing mirror service id
          mirror-source 100
     exit
   no shutdown
   exit
-------------------------------------------------------------------------------
A:ALA-48>config>app-assure>group>policy>aqp#
```

The following example displays an AQP entry to mirror all P2P traffic (all positively identified P2P traffic and any unidentified traffic that may or may not be P2P - AppGroup P2P) for a subset of subscribers with ASO characteristic **aa-sub-mirror** enabled (the order is significant):

```
A:ALA-48>config>app-assure>group>policy>aqp>entry#
-------------------------------------------------------------------------------
   entry 100 create
     match
          app-group eq P2P
          characteristic aa-sub-mirror value enabled
     exit
     action
          mirror-source all-inclusive 100
     exit
   no shutdown
   exit
-------------------------------------------------------------------------------
A:ALA-48>config>app-assure>group>policy>aqp#
```

Configuring an Application and DNS IP Cache for URL Content Charging Strengthening

In the context of URL content charging, also known as zero rating, the DNS IP cache (**dns-ip-cache** command) feature ensures that only legitimate traffic is classified in a given application and charging-group. Subscribers' DNS responses matching a list of domain names used for content charging populate the DNS IP cache. The system can then be configured to create app-filters matching HTTP or HTTPS expressions as well as the IP cache ensuring that traffic is properly classified.

To configure the system for URL content charging strengthening with a dns-ip-cache the operator needs to:

- Create an application of interest and its related app-filter's URL expressions. This application is typically mapped into a charging-group.
- Create a **dns-ip-cache**. Configure parameters so the IP cache is populated by the domain names from the application mapped to the zero rating charging group and specify which DNS server IP addresses the IP cache will listen from.
- Configure a AQP to enable the dns-ip-cache.

Use the following CLI syntax to create a dns-ip-cache:

CLI Syntax:

```
config>app-assure>group#
  dns-ip-cache dns-ip-cache-name [create]
     dns-match
        description <description-string>
        no description
        domain <domain-name> expression <expression>
        no domain <domain-name>
        server-address <server-address> [name <server-name>]
        no server-address <server-address>
     ip-cache
        size <cache-size>
        high-watermark <percent>
        low-watermark <percent>
     [no] shutdown
```

The following example displays a configuration for a **dns-ip-cache** configured to snoop DNS responses for two different domains “*.domain1.com” and “*domain2.com” which are zero rated or charged specifically by the operator. The configuration only uses DNS responses from the DNS server addresses configured within the **dns-match** to populate the **ip-cache**:

```
7750>config>app-assure>group# info
---------------------------------------------------------
dns-ip-cache "dns-ip-cache1" create
                description "DNS IP Cache #1"
                dns-match
                    domain "Sponsor#1-Domain#1" expression "*.domain1.com$"
                    domain "Sponsor#1-Domain#2" expression "*.domain2.com$"
                    server-address 8.8.4.4 name "Google"
                    server-address 8.8.8.8 name "Google"
                    server-address 192.168.100.11 name "OperatorX-DNS1"
                    server-address 192.168.100.12 name "OperatorX-DNS2"
                exit
                ip-cache
                    size 1000
                    high-wmark 90
                    low-wmark 80
                exit
                no shutdown
            exit
----------------------------------------------
```

The domains configured in the dns-ip-cache must match the app-filter expressions for the application(s) zero rated or charged specifically by the operator. The following example displays the charging-group **Zero Rated** and application **Sponsor Content #1** configuration:

```
7750>config>app-assure>group>policy# info
----------------------------------------------
                charging-group "Zero Rated" create
                    description "Zero Rated Content"
                    export-id 10
                exit
                app-group "Web" create
                exit
                application "Sponsor Content #1" create
                    description "Application#1 - Content Zero Rated"
                    app-group "Web"
                    charging-group "Zero Rated"
                exit
                app-filter
                    entry 100 create
                        expression 1 http-host eq "*.sponsor1-domain1.com$"
                        server-address eq dns-ip-cache "dns-ip-cache1"
                        application "Sponsor Content #1"
                        no shutdown
                    exit
                    entry 110 create
                        expression 1 http-host eq "*.domain2.com$"
                        server-address eq dns-ip-cache "dns-ip-cache1"
                        application "Sponsor Content #1"
                        no shutdown
                    exit
                exit
---------------------------------------------------------------------------
```

The following example displays the AQP entry to enable the **dns-ip-cache** to snoop DNS responses; this can be optionally based on ASO characteristics:

```
A:7750>config>app-assure>group>policy>aqp# entry 100 create
        match
            characteristic "dns-ip-cache" eq "yes"
        exit
        action
            action dns-ip-cache "dns-ip-cache1"
        exit
        no shutdown
```

Configuring an HTTP Error Redirect

Use the following CLI syntax to configure an HTTP error redirect policy:

CLI Syntax:

```
config>app-assure>group>
  http-error-redirect redirect-name create
  no http-error-redirect redirect_name
  description description-string
  no description
  error-code error-code [custom-msg-size custom-msg-size]
  no error-code error-code
  http-host http-host // eg. www.demo.barefruit.com
  no http-host
  participant-id participant-id // 32-char string used by tem-
     plate 1
  no participant-id
  no] shutdown
  template template-id // {1, 2} one for Barefruit, 2= Xerocole
  no template
```

The following example displays an Application Assurance HTTP redirect configuration.

```
*A:ALA-48>config>app-assure>group# http-error-redirect "redirect-404"
create
        description “redirect policy of 404 to Barefruit servers”
        error-code 404
        http-host
          att.barefruit.com
        participant-id att-ISP
        template 1


*A:ALA-48>config>app-assure>group> http-error-redirect# redirect-404
info
----------------------------------------------
               description "redirect policy of 404 to Barefruit servers"
                template 1
                http-host "att.barefruit.com"
                participant-id "att-ISP"

                error-code 404

*A:ALA-48>config>app-assure>group>http-error-redirect#
```

Configuring HTTP Header Enrichment

Use the following CLI syntax to configure an HTTP header Enrichment policy:

CLI Syntax:

```
config>app-assure>group> http-enrich <http_enrich_name> [ create]
            [no] description <description-string>
            [no] shutdown
            [no] field <field_name> name <header_name>
                   // Where “Field name” can be:
                   // subscriber-ip: Header name for subscriber IP
                   // subscriber-id: Header name for the subscriber ID
                   // static-string: Header name for inserted string
            [no] http-enrich <http_enrich_name>
```

The following example displays an Application Assurance HTTP header enrichment configuration.

```
*A:BNG>config>app-assure>group# http-enrich enrich_example create
*A:BNG>config>app-assure>group>http-enrich$ description "enrich HTTP headers with
subscriber IP and subscriber ID"
*A:BNG>config>app-assure>group>http-enrich$ field "static-string" name "x-string"
*A:BNG>config>app-assure>group>http-enrich$ field "static-string" static-string "orange"
*A:BNG>config>app-assure>group>http-enrich$ field "subscriber-id" name "x-subID"
*A:BNG>config>app-assure>group>http-enrich$ field "subscriber-id" anti-spoof
*A:BNG>config>app-assure>group>http-enrich$ field "subscriber-ip" name x-subIP
*A:BNG>config>app-assure>group>http-enrich$ field "subscriber-ip" encode type md5 key
"secret10"
----------------------------------------------
*A:BNG>config>app-assure>group>http-enrich$ info
----------------------------------------------
                field "static-string"
                    name "x-string"
                    static-string "orange"
                exit
                field "subscriber-id"
                    name "x-subID"
                    anti-spoof
                exit
                field "subscriber-ip"
                    name "x-subIP"
                    encode type md5 key "bF0sZZDNT8DbZoVJHD1vrYr5mJaEggEqWbSvPhgIcP-
W6hym0sc08O." hash2
                exit
----------------------------------------------
*A:BNG>config>app-assure>group>http-enrich$
```

In addition, the following **show** routine provides visibility into the various HTTP enrichment-related statistics:

```
*A:BNG# show application-assurance group 1 http-enrich "enrich_example "

===============================================================================
Application Assurance Group 1 HTTP Enrichment " enrich_example "
===============================================================================
Description   : enrich HTTP headers with subscriber IP and subscriber ID
Admin Status  : Up
AQP Referenced: No
-------------------------------------------------------------------------------
Name                             Field                             Enabled
                                                                   Features
-------------------------------------------------------------------------------
static-string                    x-string
subscriber-id                    x-subid                           A
subscriber-ip                    x-srcIP                           M
-------------------------------------------------------------------------------
                                                      A=anti-spoof,M=encode-md5

----------------------------------------------------------------------
Group               Enriched                 Not Enriched
----------------------------------------------------------------------
1:1                 12587                    3
1:2                 0                        0
----------------------------------------------------------------------
Total               12587                    3
----------------------------------------------------------------------
```

Configuring an HTTP Redirect Policy

Use the following CLI syntax to configure an HTTP redirect policy:

```
CLI Syntax:  config>app-assure>group# http-redirect redirect-name [create]
                 description <description-string>
                 no description
                 template <template-id>
                 redirect-url URL // redirect URL e.g. www.isp.com/redi-
                    rect.html
                 no redirect-url
                 [no] shutdown
             no http-redirect <redirect-name>
```

The following example displays an AA HTTP redirect configuration.

```
*A:ALA-48>config>app-assure>group# http-redirect "redirect1" create
            description “redirect policy for blocked http content traffic without url 
parameters”
            template 3
            redirect-url http://www.isp.com/redirect.html
            no shutdown
```

The following example displays an Application Assurance **http-redirect** configuration using macro substitution to append url parameters within the redirect url:

```
*A:ALA-48>config>app-assure>group# http-redirect "redirect2" create
     description "redirect policy for blocked http content traffic with url parameters"
     template 5
     redirect-url "http://www.isp.com/redirect.html?requestedurl=$URL&sub
scriberid=$SUB&subscriberip=$IP&routerid=$RTRID&vsa=$URLPRM"
     no shutdown
```

The following example displays AQP entry to block all http gaming traffic (AppGroup BlockedContent) and perform redirect:

```
A:ALA-48>config>app-assure>group>policy>aqp>entry#
-------------------------------------------------------------------------------
            entry 100 create
               match
                   app-group eq BlockedContent
               exit
               action
                   drop
                   http-redirect redirectgaming
               exit
            no shutdown
            exit
-------------------------------------------------------------------------------
A:ALA-48>config>app-assure>group>policy>aqp#
```

Configuring a Captive Redirect HTTP Redirect Policy

The traditional HTTP redirect policy is used to redirect flows on the HTTP response packet, meaning the TCP three-way handshake and the original HTTP request are allowed by the 7750 SR to the Internet before the subscriber is redirected. The captive redirect HTTP redirect policy is used to redirect flows without sending any traffic to the Internet unless it matches a configurable whitelist by terminating TCP sessions in the ISA-AA cards, in which case HTTP flows are redirected to a predefined redirect URL while non-HTTP TCP flows are TCP reset.

A session-filter is used to define the criteria for permitting or redirecting flows using the captive redirect HTTP redirect policy. Typically the operator needs to permit UDP on port 53 for DNS and they can optionally permit other content based on IP address, port number, IP prefix list, or DNS IP cache thus allowing specific on-net of off-net applications through the captive redirect policy.

To configure the system for captive redirect HTTP redirect the operator needs to:

- Create an http-redirect policy. If the ISA group aa-sub-scale mode is configured for residential or VPN, then configure the http-redirect policy for captive-redirect and associate the appropriate VLAN id AA Interface (an aa-interface routable within the subscriber's service must be created for each ISA-AA card in the system). If the ISA group aa-sub-scale mode is configured for DSM, then there is no need to associate the http-redirect policy to a VLAN id and no need to create an AA Interface.
- Create a session filter policy to allow at the minimum UDP on port 53. Additional traffic can be whitelisted based on a statically defined IP prefix list or a dynamic DNS IP cache policy. The redirect landing page should be configured using IP prefixes.
- The last action of the session filter should be set to http-redirect the remaining flows using a predefined captive redirect HTTP redirect policy.

Use the following CLI syntax to create a captive redirect HTTP redirect policy:

CLI Syntax:
```
config>app-assure>group# http-redirect <redirect-name> [cre-
ate]
                    description <description-string>
                    no description
                    template <template-id>
                    no template
                    [no] tcp-client-reset
                    redirect-url <redirect-url>
                    no redirect-url
                    [no] shutdown
                    captive-redirect
                        vlan-id <service-port-vlan-id>
                        no vlan-id
                no http-redirect <redirect-name>
```

The following example displays a configuration for a session filter user in the context of captive redirect:

```
A:7750# configure application-assurance group 1:1 create
A:7750>config>app-assure>group# info
----------------------------------------------
            session-filter "wifi-unauthenticated" create
                default-action deny
                entry 5 create
                    match
                        ip-protocol-num udp
                        dst-port eq 53
                    exit
                    action permit
                exit
                entry 10 create
                    match
                        dst-ip dns-ip-cache "whitelist"
                    exit
                    action permit
                exit
                entry 15 create
                    description "Allow traffic to the redirect landing page server"
                    match
                        ip-protocol-num tcp
                        dst-port eq 80
                        dst-ip 172.16.70.100/32
                    exit
                    action permit
                exit
                entry 20 create
                    match
                        ip-protocol-num tcp
                    exit
                    action http-redirect "redirect-portal"
                exit
            exit
----------------------------------------------
```

The following example displays a configuration for the AA interface used by the captive redirect HTTP redirect policy for ESM Subscribers (DSM does not require the configuration of the AA Interface):

```
A:7750# configure service ies 1 customer 1 create
A:7750>config>service>ies# info
----------------------------------------------
            aa-interface "aa-if-captive-redirect-isa_1-2" create
                description "AA Interface for ISA-AA card 1/2"
                address 172.16.3.1/31
                sap 1/2/aa-svc:20 create
                    no shutdown
                exit
                no shutdown
            exit
----------------------------------------------
```

The following example displays a configuration for the HTTP redirect policy for ESM Subscribers (DSM does not require the configuration of the VLAN id):

```
A:7750# configure application-assurance group 1
A:7750>config>app-assure>group>http-redir# info
----------------------------------------------
                template 5
                tcp-client-reset
                redirect-url "http://172.16.70.100/Redirect/redirect-portal.html?Request-
edURL=$URL"
                captive-redirect
                    vlan-id 20
                exit
                no shutdown
----------------------------------------------
```

Configuring ICAP URL Filtering

To configure the system for ICAP URL Filtering, the operator needs to:

- Create an aa-interface and assign an ip address to each AA ISA within an IES or VPRN service. This routed interface is then used by the system to establish TCP communication with the ICAP server.
- Create an http-redirect policy (used by the url-filter to redirect http traffic).
- Create a url-filter, configure the icap server ip-address, redirect-policy, and default action.
- Verify that the aa-interface(s) and url-filter are operationally up.

Use the following CLI syntax to configure the aa-interfaces for each AA ISA:

CLI Syntax:

```
config>service>vprn# aa-interface <aa-if-name> [create]
 config>service>vprn>aa-if# aa-interface interface <ip-int-
    name> [create]
 description <description-string>
 no description
 address <ipv4_subnet/31>
 no address
    sap <card/mda/aa-svc:vlan> [create]
       description <description-string>
       no description
       egress
          [no] filter
          [no] qos
       exit
       ingress
          [no] qos
       exit
       [no] shutdown
    exit
```

The following examples displays an AA interface created for the ISA card 1/2 using IP address 172.16.2.1/31:

```
A:7750>config>service>ies# info
----------------------------------------------
            aa-interface "aa-if1" create
                address 172.16.2.1/31
                sap 1/2/aa-svc:10 create
                    egress
                        filter ip 10
                    exit
                    no shutdown
                exit
                no shutdown
            exit
```

In the example above, 172.16.2.1 is used by the IOM side of the interface while the ISA itself is automatically assigned 172.16.2.0 based on the /31 subnet.

Recommendations:

- More than one aa-interface can be configured per AA ISA card, however, the operator needs to use the same service vlan across all these interfaces for a given url-filter object.
- Configure an egress ip filter under the sap towards the ISA AA interface to only allow selected ip addresses or subnet (subnet examples: icap servers, network management).

Use the following CLI syntax to configure the url-filter:

CLI Syntax:

```
config>app-assure>group#
 url-filter <url-filter-name> [create]
    default-action {allow | block-all | block-http-redirect
       <redirect-name>}
    no default-action
    http-redirect <http-redirect-name>
    no http-redirect
    http-request-filtering {all | first}
    icap
       custom-x-header <x-header-name>
       [no] custom-x-header
       vlan-id <service-port-vlan-id>
       no vlan-id
       server <ip-address[:port]> [create]
          description <description-string>
          no description
          [no] shutdown
       no server <ip-address[:port]>
 no url-filter <url-filter-name>
```

The following examples displays a url-filter configuration:

```
*A:7750>config>app-assure>group# url-filter "filter1" create
    default-action block-http-redirect "http-redirect-portal"
    icap
       vlan-id 10
       server 172.16.1.101 create
           no shutdown
       exit
    exit
    no shutdown
```

The following examples displays the AQP entry to enable icap url-filtering for opted-in subscribers based on ASO characteristics:

```
A:7750>config>app-assure>group>policy>aqp# entry 100 create
        match
            characteristic "url-filter" eq "yes"
        exit
        action
            url-filter "filter1"
        exit
        no shutdown
```

Optionally the operator can add a custom-x-header to the ICAP request in order for the ICAP server to filter traffic based on this new x-header value instead of filtering based on subscriber names. This is done by defining a new ASO characteristic for the different ICAP filtering service packages used in the network and referring the characteristic name in the url-filter AQP action.

The following example displays a url-filter configuration with the custom-x-header field added to the ICAP request:

```
A:7750>config>app-assure>group# url-filter "filter1" create
    default-action block-http-redirect "http-redirect-portal"
    http-redirect "http-redirect-portal"
    icap
        custom-x-header "Filtering-Policy"
        vlan-id 10
        server 172.16.1.101 create
            no shutdown
        exit
    exit
    no shutdown
```

The following example displays the App-Service-Option characteristic used to define the type of filtering policy available:

```
A:7750>config>app-assure>group>policy>aso# info
----------------------------------------------
                    characteristic "url-filter-policy" create
                        value "filtering-policy-1"  #less than 10 years old
                        value "filtering-policy-2"  # less than 16 years old
                        value "mcdonalds"
                        value "none"
                        value "starbucks"
                        default-value "none"
                    exit
----------------------------------------------
```

The following example displays the App-Qos-Policy action required to add the appropriate ASO value to the ICAP custom-x-header custom field:

```
A:7750>config>app-assure>group>policy>aqp# entry 100 create
        match
            characteristic "url-filter" eq "yes"
        exit
        action
            url-filter "filter1" characteristic "url-filter-policy"
        exit
        no shutdown
```

Configuring Local URL-List Filtering

To configure the system for local URL-list filtering, the operator needs to:

- Create a URL-list policy referencing a valid file located on the compact flash
- Create a url-filter policy for local-filtering by referencing this URL-list
- Create an AQP to apply this url-filter policy

Use the following CLI syntax to create a URL-list:

CLI Syntax:
```
config>app-assure>group# url-list <url-list-name> [create]
  description <description-string>
  no description
  decrypt-key <key|hash-key|hash2-key> [hash | hash2]
  no decrypt-key
  file <file-url>
  no file
  [no] shutdown
```

The decryption key is optional, if the decryption key is not specified the system will assume that the file is not encrypted. To encrypt a file in Linux using the supported encryption format use the follwoing command:

```
Linux# openssl des3 -nosalt -in <input-file.txt> -out <output.enc>
```

The following example displays a URL-list configuration:

```
A:7750>config>app-assure>group# url-list url-list1 create
---------------------------------------------
                description "Local List for URL Filtering"
                decrypt-key ".i84/P1uS0lMGoQkae7mAV2Oj10n726Z" hash2
                file "cf3:\url-list1.enc"
                no shutdown
---------------------------------------------
```

Use the following CLI syntax to create a url-filter policy for local-filtering:

CLI Syntax:
```
config>app-assure>group# url-filter <url-filter-name> [cre-
ate]
  url-filter <url-filter-name> [create]
  description <description-string>
  no description
  default-action {allow | block-all | block-http-redirect <re-
     direct-name>}
  no default-action
  [no] http-redirect <redirect-name>
  http-request-filtering {all|first}
```

```
local-filtering
[no] url-list <url-list-name>
[no] shutdown
```

The following example displays a url-filter configured for local-filtering:

```
A:7750>config>app-assure>group# url-filter "url-blacklist1" create
A:7750>config>app-assure>group>url-filter# info
----------------------------------------------
                default-action allow
                http-redirect "http-redirect-portal"
                local-filtering
                    url-list "url-list1"
                exit
                no shutdown
----------------------------------------------
```

Note that the default action should always be configured to “allow” when the url-filter is configured for local-filtering. The default-action in this context represents the action the system will take in case the local-list file is not accessible; this scenario may happen if the source file was corrupted or if the compact flash card was not accessible.

The following example displays the AQP entry to enable ICAP url-filtering for opted-in subscribers based on ASO characteristics:

```
A:7750>config>app-assure>group>policy>aqp# entry 100 create
        match
            characteristic "child-protection" eq "yes"
        exit
        action
            url-filter "url-blacklist1"
        exit
        no shutdown
```

Configuring HTTP Notification

Use the following CLI syntax to configure an HTTP Notification policy.

CLI Syntax:

```
config>app-assure>group#
 http-notification <http-notification-name> [create]
    description <description-string>
    no description
    script-url <script-url-name>
    no script-url
    interval {one-time | <minimum-interval>}
    template <template-id>
    no template
    [no] shutdown
 no http-notification <http-notification-name>
```

The following example displays an HTTP notification policy configured with a minimum interval of 5 minutes:

```
A:7750>config>app-assure>group# http-notification "in-browser-notification" create
A:7750>config>app-assure>group>http-notif# info
-----------------------------------------------
                description "In Browser Notification Example"
                template 1
                script-url "http://1.1.1.1/In-Browser-Notification/script.js"
                interval 5
                no shutdown
-----------------------------------------------
```

The operator then needs to enable the http-match-all-req feature for any HTTP request sent the messaging server domain which will be used to monitor HTTP notification success/failures. This is done by creating a new application and enabling http-match-all-req within the app-filter.

```
A:7750>config>app-assure>group>policy# application "IBN Messaging Server" create
A:7750>config>app-assure>group>policy>app$ app-group "Web"

A:7750>config>app-assure>group>policy# app-filter entry 100 create
A:7750>config>app-assure>group>policy>app-filter>entry$ info
-----------------------------------------------
                        expression 1 http-host eq "^1.1.1.1$"
                        http-match-all-req
                        application "IBN Messaging Server"
                        no shutdown
-----------------------------------------------
```

The following examples displays the AQP entry required to match this policy based on an ASO characteristic:

```
A:7750>config>app-assure>group>policy>aqp# info
----------------------------------------------
                    entry 200 create
                        match
                            characteristic "in-browser-notification" eq "yes"
                        exit
                        action
                            http-notification "in-browser-notification"
                        exit
                        no shutdown
                    exit
----------------------------------------------
```

Configuring AA Volume Accounting and Statistics

A network operator can configure AA volume statistic collection and accounting on both AA ISA system and subscriber levels.

The following commands illustrate the configuration of statistics collection and accounting policy on an AA group/partition aggregate level (without subscriber context).

CLI Syntax:
```
config>app-assure>group>statistics>app-group
  accounting-policy act-policy-id
  collect-stats
```

CLI Syntax:
```
config>app-assure>group>statistics>application
  accounting-policy act-policy-id
  collect-stats
```

CLI Syntax:
```
config>app-assure>group>statistics>protocol
  accounting-policy act-policy-id
  collect-stats
```

These commands illustrate the configuration of statistics collection and accounting policy for each AA subscriber in the system.

CLI Syntax:
```
config>app-assure>group>statistics>aa-sub
  accounting-policy acct-policy-id
  aggregate-stats
  app-group app-group-name export-using export-method [export-
     method...(upto 2 max)]
  application application-name export-using export-method [ex-
     port-method...(upto 2 max)]
  charging-group charging-group-name export-using export-meth-
     od [export-method...(upto 2 max)]
  collect-stats
  exclude-tcp-retrans
  max-throughput-stats
  protocol protocol-name export-using export-method
  radius-accounting-policy rad-acct-plcy-name
```

These commands illustrate configuration of special study mode for a subset of AA subscribers (configured) to collect all protocol and/or application statistics with an AA subscriber context.

CLI Syntax:
```
config>app-assure>group>statistics# aa-sub-study {applica-
tion|protocol}
            accounting-policy acct-policy-id
            collect-stats
```

For details on accounting policy configuration (including among others AA record type selection and customized AA subscriber record configuration) refer to the OS System Management Guide.

The following output illustrates per AA-subscriber statistics configuration that elects statistic collection for a small subset of all application groups, applications, protocols:

```
*A:ALU-40>config>app-assure>group>statistics>aa-sub# info
----------------------------------------------
                    accounting-policy 4
                    collect-stats
                    app-group "File Transfer"
                    app-group "Infrastructure"
                    app-group "Instant Messaging"
                    app-group "Local Content"
                    app-group "Mail"
                    app-group "MultiMedia"
                    app-group "Business_Critical
                    app-group "Peer to Peer"
                    app-group "Premium Partner"
                    app-group "Remote Connectivity"
                    app-group "Tunneling"
                    app-group "Unknown"
                    app-group "VoIP"
                    app-group "Web"
                    app-group "Intranet"
                    application "BitTorrent"
                    application "eLearning"
                    application "GRE"
                    application "H323"
                    application "TLS"
                    application "HTTP"
                    application "HTTPS"
                    application "HTTPS_Server"
                    application "HTTP_Audio"
                    application "HTTP_Video"
                    application "eMail_Business"
                    application "eMail_Other"
                    application "Oracle"
                    application "Skype"
                    application "SAP"
                    application "SIP"
                    application "SMTP"
                    application "SQL_Alltypes"
                    application "TFTP"
                    protocol "bittorrent"
                    protocol "dns"
                    protocol "sap"
                    protocol "skype"
----------------------------------------------
*A:ALU-40>config>app-assure>group>statistics>aa-sub#
```

Configuring Cflowd Collector

The following output displays an Application Assurance cflowd collector configuration example:

Example:

```
*A:ALA-48# configure application-assurance group 1 cflowd
collector 138.120.131.149:55000 create
*A:ALA-48>config>app-assure>group>cflowd>collector$description
"cflowd_collector_NewYork"
*A:ALA-48>config>app-assure>group>cflowd>collector# no shutdown
*A:ALA-48>config>app-assure>group>cflowd>collector# exit
```

```
*A:ALA-48>config>app-assure>group>cflowd# info
----------------------------------------------
    collector 138.120.131.149:55000 create
         description "cflowd_collector_NewYork"
    no shutdown
----------------------------------------------
*A:ALA-48>config>app-assure>group>cflowd#
```

Configuring AA Volume, TCP and RTP Performance Reporting

CLI Syntax:
```
config>application-assurance>group isa-aa-group-id
 cflowd
 collector ip-address[:port] [create]
 no collector ip-address[:port]
 description description-string
 no description
    [no] shutdown
 rtp-performance
    flow-rate sample-rate
    no flow-rate
    flow-rate2 sample-rate2
    no flow-rate2
 tcp-performance
    flow-rate sample-rate
    no flow-rate
    flow-rate2 sample-rate2
    no flow-rate2
 template-retransmit seconds
 no template-retransmit
 [no] shutdown
 volume
    rate sample-rate
    no rate
    [no] shutdown
```

CLI Syntax:
```
config>application-assurance
 group isa-aa-group-id[:partition [create]]
 no group isa-aa-group-id[:partition
    cflowd
       volume
          [no] shutdown
       rtp-performance
          [no] app-group app-group-name [flow-rate|flow-rate
          2]
          [no] application application-name [flow-rate|flow-
          rate 2]
          [no] shutdown
       tcp-performance
          [no] app-group app-group-name [flow-rate|flow-rate
          2]
          [no] application application-name [flow-rate|flow-
          rate 2]
          [no] shutdown
```

Note: The default if flow-rate

The following example shows a configuration that:

- Enables per-flow volume stats for group 1, partition 1 and configures sampling rate to 1/1000.
- Enables per-flow TCP performance stats for web_traffic application within group 1, partition 1 and configures TCP sampling rate to 1/500.
- Enables per-flow TCP performance stats for citrix_traffic application within group 1, partition 1 using TCP sampling rate2 to 1/100.
- Enables per-flow RTP A/V performance stats for voip_traffic application within group 1, partition 1 and configures rtp sampling rate to 1/10.

```
*A:ALA-48# configure application-assurance group 1 cflowd
*A:ALA-48>config>app-assure>group>cflowd# volume rate 1000
*A:ALA-48>config>app-assure>group>cflowd# tcp-performance flow-rate 500
*A:ALA-48>config>app-assure>group>cflowd# tcp-performance flow-rate2 100
*A:ALA-48>config>app-assure>group>cflowd# rtp-performance   flow-rate 10
*A:ALA-48>config>app-assure>group>cflowd# no shutdown
*A:ALA-48>config>app-assure>group>cflowd# info
----------------------------------------------
                collector 138.120.131.149:55000 create
                    description "cflowd_collector_NewYork"
                exit
                volume
                    rate 1000
                exit
                tcp-performance
                    flow-rate 500
                    flow-rate 100
                rtp-performance
                    flow-rate 10
                exit
                no shutdown
----------------------------------------------
*A:ALA-48>config>app-assure>group>cflowd#
```

```
*A:ALA-48# configure application-assurance group 1:1 cflowd
*A:ALA-48>config>app-assure>group>cflowd#
*A:ALA-48>config>app-assure>group>cflowd# volume no shutdown
*A:ALA-48>config>app-assure>group>cflowd# tcp-performance application "web_traffic"
*A:ALA-48>config>app-assure>group>cflowd# tcp-performance application "citrix" [flow-
rate2]
*A:ALA-48>config>app-assure>group>cflowd# tcp-performance no shutdown
*A:ALA-48>config>app-assure>group>cflowd# rtp-performance application "voip_traffic"
*A:ALA-48>config>app-assure>group>cflowd# rtp-performance no shutdown
*A:ALA-48>config>app-assure>group>cflowd# info
----------------------------------------------
            volume
               no shutdown exit
            rtp-performance no shutdown
               application "voip_traffic"
            tcp-performance
               no shutdown
               application "web_traffic"
               application “citrix” flow-rate2
```

```
    exit
----------------------------------------------
*A:ALA-48>config>app-assure>group>cflowd#
```