APPENDIX D

Protocol Compliance Statements for the CSG 3.1(3)C6(2)

This appendix provides protocol compliance statements for the CSG 3.1(3)C6(2). Any RFCs that are not explicitly listed are not supported.

Layer 4 Inspection (accounting type=other)

The CSG differentiates TCP and UDP, and classifies all other protocols simply as IP. All protocols can be billed in this manner if further protocol-specific processing is not desired (or if deeper inspection for such protocols is not supported).

- IP—Compliant with RFC791. To avoid leakage, the CSG drops packets on a service for a prepaid user while reconciling the user's quota for that service. The frequency depends on how quickly the user is consuming quota on that service and generally amounts to a few packets. This is controlled by setting the CSG_BASIS_BYTE_RESERVED_MAX variable up to a setting of 256000. Settings above this value have no effect.

  The CSG volume counters wrap at 0xFFFFFFFF (268435455 bytes). The volume counters are 32 bits unsigned.

  The CSG supports IP fragmentation for generic Layer 4 flows, regardless of protocol and regardless of the order in which the flows arrive.
- UDP—Compliant with RFC768.
- TCP—Compliant with standard TCP (RFC3168).

Layer 7 Inspection (accounting type=specific protocol)

- IP—Compliant with RFC791.

  The CSG supports IP fragmentation for HTTP, WAP2.0, and WAP1.x, regardless of the order in which the flows arrive. The CSG does not support IP fragmentation for SMTP, POP3, IMAP4, FTP, and RTSP control connection, nor for RADIUS flows. The CSG drops IP fragments for those unsupported protocols.

To avoid leakage, the CSG drops packets on a service for a prepaid user while reconciling the user's quota for that service. The frequency depends on how quickly the user is consuming quota on that service and generally amounts to a few packets. This is controlled by setting the CSG_BASIS_BYTE_RESERVED_MAX variable up to a setting of 256000. Settings above this value have no effect.

The CSG volume counters wrap at 0xFFFFFFFF (268435455 bytes). The volume counters are 32 bits unsigned.

- UDP—Compliant with RFC768.

  See IP compliance for further restrictions.

- TCP—Compliant with standard TCP (RFC793), with the following exceptions:

  - The CSG does not pass TCP header options when performing deep packet inspection, other than MSS.
  - When performing Layer 7 inspection of TCP-based protocols, the CSG drops packets that arrive out of order, relying on retransmission to provide them again after the missing packets are received. Those packets that require inspection, and that are therefore subject to this dropping, vary per protocol. For HTTP, the CSG parses only HTTP headers, so the packets carrying HTTP headers must be in order. After the header parsing is complete and the CSG has detected the body, the CSG no longer enforces the order of the packets. This repeats for each transaction in a persistent or pipelined connection. There are two important exceptions to this: for chunked encoding, and for multipart content, every packet must be analyzed and therefore is dropped if not in order. For e-mail protocols (that is, SMTP, POP3, and IMAP4), all packets must be in order.

  See IP compliance for further restrictions.

- WP-TCP—Compliant with mandatory elements of Wireless Profiled TCP (WP-TCP) (WAP-225-TCP-20010331-a at http://www.wapforum.org/what/technical.htm), with the exception of SACK (RFC2018).

  Impact: End-user latency for lossy transmissions.

  See TCP compliance for further restrictions.

- HTTP—Compliant with RFC1945 (HTTP 1.0) and RFC2616 (HTTP 1.1), with the following exceptions:

  1. Each HTTP method must be initiated by the same endpoint that initiated the TCP connection (that is, by the same side that sent the TCP SYN).

     Impact: Client requests transfer no data (that is, they hang). See TO-TCP in MMS for WAP 2.0 compliance for an example.

  2. The maximum HTTP transaction volume is 268435455 bytes. If this length is exceeded, the CSG invokes Layer 4 billing for the remainder of the connection.
  3. HTTP request parsing is limited to 64000 bytes.

     Impact: Any headers beyond this limit are not recognized and therefore are not used in matching URL or header maps.

  4. The CSG supports up to 65535 concurrent HTTP TCP connections.
  5. If the HTTP server or client causes improper parsing, the CSG reverts to Layer 4 billing for the remainder of the TCP connection.

     For example, the CSG requires that all HTTP responses begin with the string "HTTP". If an HTTP response does not begin with "HTTP", the CSG increments a Layer 7 error statistic, "HTTP invalid msgs", and invokes Layer 4 billing.

Also, when parsing the response for an HTTP return code, the CSG accepts only ASCII decimal digits (0x30 - 0x39). If the response contains any other characters, the CSG increments the "HTTP invalid msgs" statistic and invokes Layer 4 billing.

If the CSG cannot parse the HTTP HEAD method, it invokes Layer 4 billing for all subsequent traffic.

6. HTTP status 101 (switching protocols) is not supported. The CSG expects all subsequent requests to be unencrypted and parsable by HTTP rules (see HTTPS compliance for further restrictions).

   Impact: The user TCP connection might hang until the content idle timer expires, or until the connection closes for some other reason.
7. Error codes 204, 205, and 304 do not require a body. If a response contains one of these error codes, the CSG ignores "Content-Type:", "Content-Length:", and "Transfer-Encoding:chunked" headers that might be present in error.
8. The CSG does not support the CLOSING or TIME-WAIT states for TCP connections. After the end-points exchange FIN_ACK messages, the connection is terminated immediately, and the CSG does not process any out-of-order packets for the connection.
9. The existence of a Head method in a persistent HTTP TCP connection causes the CSG to invoke Layer 4 billing for the remainder of the connection. This Layer 4 charging is reported via the HTTPstatistics CDR for the Connect transaction. The CSG will not discern any additional transactions after the Head method is detected. If a Method Map is configured for the Connect method, the traffic is charged against the matching Policy. If no Policy exists with the Method Map, the CSG passes the traffic without charge.
10. Multipart content causes the CSG to invoke Layer 4 billing for the remainder of the connection.

Compliant with RFC2774 (HTTP Extension Framework), subject to the restrictions above.

See TCP compliance for further restrictions.

- HTTPS—Because HTTPS URLs and other headers are encrypted, the CSG cannot provide Layer 7 information for HTTPS requests.

  Also, switching from HTTP to HTTPS within the same persistent connection is subject to the following restrictions:

  – Switching via the Connect method (RFC2817) is supported. The CSG detects the Connect method and invokes Layer 4 billing for the remainder of the TCP connection. This Layer 4 charging is reported via the HTTPstatistics CDR for the Connect transaction. The CSG will not discern any additional transactions after the Connect method is detected. If a Method Map is configured for the Connect method, the traffic is charged against the matching Policy. If no Policy exists with the Method Map, the CSG passes the traffic without charge.

  – Switching via the "Upgrade" header (RFC2817) is ignored. The CSG attempts to parse the traffic as normal HTTP. When parsing fails, the CSG invokes Layer 4 billing for all subsequent traffic on the TCP connection, charging against the last matching Policy.

  See HTTP compliance for further restrictions.

- WAP 2.0 (HTTP over WP-TCP transport)—The CSG supports the billing of WAP 2.0 over clear text HTTP and the differential billing of MMS over WAP 2.0 over clear text HTTP (see MMS for WAP 2.0 compliance for details) as specified by the WapForum (wapforum.org - http://www1.wapforum.org/tech/terms.asp?doc=Technical_WAP2_0-20021106.zip), with the following exceptions:

  - There are two variants of Push OTA-HTTP: TO-TCP and PO-TCP. The CSG does not support TO-TCP, as described in WAP-235-PushOTA-20010425-a, for flows billed at Layer 7 (that is, those with HTTP policies). PO-TCP can be configured but requires more complex configuration (see MMS for WAP 2.0 compliance for details).
  - The CSG cannot bill TLS (encrypted connections) as WAP 2.0 flows. In WAP-235-PushOTA-20010425-a, TLS is referenced as OTA-HTTP-TLS.
  - See HTTPS compliance for restrictions regarding switching from HTTP to HTTPS within the same persistent connection. WAP-219-TLS-20010411-a specifies that only the Connect method is supported (that is, portions of RFC2817 pertaining to Upgrade requests or responses are not supported by WAP 2.0 clients).
  - Because the CSG does not currently pass TCP options, the CSG does not support the <WAP-GW-STD-11>, <WAP-GW-STD-13>, <WAP-GW-STD-14>, <WAP-GW-STD-15>, and <WAP-GW-STD-17> standards.

  See HTTP 1.1, HTTPS, and WP-TCP compliance for further restrictions.

- MMS for WAP 2.0 (HTTP transport)—At the current time, the MMS standard is very incomplete.
  - For MMS differentiation, the CSG requires that the “Content-Type” header in the request be set to “application/vnd.wap.mms-message” on all MMS/WAP2/HTTP exchanges, other than message retrieval.
  - For message retrieval, the “Content-Type” header is not present in the GET request, so the CSG uses the URL in the GET request and ignores the “Content-Type” header in the response. This method provides reasonable differentiation, although examining the “Content-Type” in the response would be the canonical technique for MMS differentiation per the standard.

MMS over WAP 2.0 allows three types of notification:

1. SMS-based notification carrying the URI for the MMS. The handset then initiates a GET request to that URI to retrieve the information.
2. TO-TCP (Terminal-Originated TCP). TO-TCP starts with SMS but provides only the IP address of the PPG. The terminal must then open a TCP connection and wait for an HTTP request from the PPG. This HTTP request is an OPTIONS method and must succeed before the handset can retrieve the notification.
3. PO-TCP (PPG-Originated TCP). PO-TCP is similar to TO-TCP, except the TCP connection is opened by the PPG and is followed by the OPTIONS method.

The CSG Layer 7 billing for MMS relies entirely on options 1 and 3. The CSG does not support TO-TCP. If a terminal reuses a persistent PO-TCP to initiate a new method request, the packets are dropped and the PO-TCP connection appears to be hung until TCP retry attempts expire.

See WAP 2.0 compliance for further restrictions.

- POP3—Compliant with RFC1939. The CSG reports the RFC2822 (Internet-Message Format) headers in the body of the POP3 message.

  See TCP compliance for further restrictions.

- IMAP4—Compliant with RFC3501.

  See TCP compliance for further restrictions.

- SMTP—Compliant with RFC 2821 - Simple Mail Transfer Protocol. Reports headers in the SMTP body formatted in accordance with RFC 2822 - Internet Message Format.

  The CSG does not support SMTP command pipelining as defined in RFC 2920 - SMTP Service Extension for Command Pipelining.

  Impact: Everything is charged for the first e-mail and either incomplete or no SMTP envelope and RFC 2822 headers are reported (depending on the e-mail content).

  See TCP compliance for further restrictions.

- FTP—Compliant with RFC959. The CSG requires that the control connection use port 21 on the server.

  See TCP compliance for further restrictions.

- RTSP—Compliant with RFC2326, except that the RFC allows RTSP control flows on either TCP or UDP, but the CSG supports RTSP control flows only on TCP. The CSG requires that the control connection use port 554 on the server, even though some servers allow other ports to be used. The CSG does not parse SMIL or SDP files, so correlation is not supported across multiple elements in the file.

  For Interleaved RTSP (Control and Stream both sharing the control connection), and for RTSP over HTTP:554 (with policy of type=rtsp), the CSG parses only the first SETUP command.

  See TCP compliance for further restrictions.

- WAP 1.x (WSP/WTP)—Compliant with the following specifications:

  1. WAP-100, Wireless Application Protocol Architecture Specification (WAP-100-WAPArch-19980430-a)
  2. WAP-165, Push Architectural Overview (WAP-165-PushArchOverview-19991108-a)
  3. WAP-203, Wireless Session Protocol Specification (WAP-203-WSP-20000504-a)
  4. WAP-201, Wireless Transaction Protocol Specification (WAP-201-WTP-20000219-a)

MMS for WSP is identified via WSP Content Type values 0X3E or application/vnd.wap.mms-message.

See UDP compliance for further restrictions.

- RADIUS—Compliant with RFC2865 and RFC2866. The CSG can inspect RADIUS Access and RADIUS Accounting messages.

  For RADIUS inspection, the CSG does not support fragmented RADIUS messages nor messages that exceed an Ethernet frame size (approximately 1470 bytes). Also, the CSG does not police the attributes that it does not use.

  - Specific to RFC 2865—Base RADIUS specification:

    In order to parse information in the Access Accept message (from the real server), the CSG requires attribute 1 (User-Name) or 31 (Calling-Station-Id), as configured. Page 63 of RFC 2865 shows a summary of the attributes for each of the RADIUS messages. It shows that attribute 31 is not included in the RADIUS Access Accept message, while Attribute 1 can be. The description of attribute 31 says, “It is only used in Access-Request packets.” There is no mention of MUST/SHALL/etc.

    For VSA subattribute parsing, we require the String contents to be encoded as a sequence of vendor type / vendor length / value fields. This is a recommendation (SHOULD) on page 48 of RFC 2865. If subattribute parsing is not configured, this restriction does not apply.

  - Specific to RFC 2866—Accounting:

    When operating as a RADIUS Accounting Endpoint, the RADIUS Accounting-Response generated by the CSG does not include any attributes, as per page 9 of the RFC:

    “A RADIUS Accounting-Response is not required to have any attributes in it.”

    However, on page 5, step 3, of the RFC:

    “The remote server logs the accounting-request (if desired), copies all Proxy-State attributes in order and unmodified from the request to the response packet, and sends the accounting-response to the forwarding server.”

    The CSG is not compliant with this latter statement, though it is not clear if this is a required element of the RFC.

  - Specific to RFC 2882—Extended practices:

    The CSG supports the RADIUS Disconnect messages defined in this RFC:

    40 Disconnect Request

    41 Disconnect Ack

    42 Disconnect Nak

  - Specific to RFC 3576—Dynamic extensions:

    This RFC notes specific ports to which the Disconnect Request should be sent. The CSG allows the customer to configure the NAS port. Also, note specific actions to be taken when the Ack or Nak is received—The CSG uses the Ack or Nak only to determine whether it should send the Request. The CSG does not use, process, or report any attributes included in the Ack or Nak. Attributes that the CSG sends in the Request are defined by the customer.

    The CSG does not support any other message types in this RFC.

  See UDP compliance for further restrictions.