2010/01/16 - Connection header adjustments depending on the transaction mode.

HTTP transactions supports 5 possible modes :

```
   WANT_TUN             : default, nothing changed
   WANT_TUN + httpclose : headers set for close in both dirs
   WANT_KAL             : keep-alive desired in both dirs
   WANT_SCL             : want close with the server and KA with the client
   WANT_CLO             : want close on both sides.
```

When only WANT_TUN is set, nothing is changed nor analysed, so for commodity below, we'll refer to WANT_TUN+httpclose as WANT_TUN.

The mode is adjusted in 3 steps :
   - configuration sets initial mode
   - request headers set required request mode
   - response headers set the final mode

1) Adjusting the initial mode via the configuration

```
   option httpclose         => TUN
   option http-keep-alive   => KAL
   option http-server-close => SCL
   option forceclose        => CLO
```

Note that option httpclose combined with any other option is equivalent to forceclose.

2) Adjusting the request mode once the request is parsed

If we cannot determine the body length from the headers, we set the mode to CLO but later we'll switch to tunnel mode once forwarding the body. That way, all parties are informed of the correct mode.

Depending on the request version and request Connection header, we may have to adjust the current transaction mode and to update the connection header.

mode	req_ver	req_hdr	new_mode	hdr_change
TUN	1.0	-	TUN	-
TUN	1.0	ka	TUN	del_ka
TUN	1.0	close	TUN	del_close
TUN	1.0	both	TUN	del_ka, del_close
TUN	1.1	-	TUN	add_close
TUN	1.1	ka	TUN	del_ka, add_close
TUN	1.1	close	TUN	-
TUN	1.1	both	TUN	del_ka
KAL	1.0	-	CLO	-
KAL	1.0	ka	KAL	-
KAL	1.0	close	CLO	del_close
KAL	1.0	both	CLO	del_ka, del_close
KAL	1.1	-	KAL	-
KAL	1.1	ka	KAL	del_ka
KAL	1.1	close	CLO	-
KAL	1.1	both	CLO	del_ka
SCL	1.0	-	CLO	-
SCL	1.0	ka	SCL	del_ka
SCL	1.0	close	CLO	del_close
SCL	1.0	both	CLO	del_ka, del_close
SCL	1.1	-	SCL	add_close
SCL	1.1	ka	SCL	del_ka, add_close
SCL	1.1	close	CLO	-
SCL	1.1	both	CLO	del_ka
CLO	1.0	-	CLO	-
CLO	1.0	ka	CLO	del_ka
CLO	1.0	close	CLO	del_close
CLO	1.0	both	CLO	del_ka, del_close
CLO	1.1	-	CLO	add_close
CLO	1.1	ka	CLO	del_ka, add_close
CLO	1.1	close	CLO	-
CLO	1.1	both	CLO	del_ka

=> Summary:
   - KAL and SCL are only possible with the same requests :
     - 1.0 + ka
     - 1.1 + ka or nothing

   - CLO is assumed for any non-TUN request which contains at least a close header, as well as for any 1.0 request without a keep-alive header.

   - del_ka is set whenever we want a CLO or SCL or TUN and req contains a KA, or when the req is 1.1 and contains a KA.

   - del_close is set whenever a 1.0 request contains a close.

   - add_close is set whenever a 1.1 request must be switched to TUN, SCL, CLO and did not have a close hdr.

Note that the request processing is performed in two passes, one with the frontend's config and a second one with the backend's config. It is only possible to "raise" the mode between them, so during the second pass, we have no reason to re-add a header that we previously removed. As an exception, the TUN mode is converted to CLO once combined because in fact it's an httpclose option set on a TUN mode connection :

```
                  BE (2)
          | TUN  KAL  SCL  CLO
      ----+----+----+----+----
      TUN | TUN  CLO  CLO  CLO
          +
      KAL | CLO  KAL  SCL  CLO
  FE      +
  (1) SCL | CLO  SCL  SCL  CLO
          +
      CLO | CLO  CLO  CLO  CLO
```

3) Adjusting the final mode once the response is parsed

This part becomes trickier. It is possible that the server responds with a version that the client does not necessarily understand. Obviously, 1.1 clients are asusmed to understand 1.0 responses. The problematic case is a 1.0 client receiving a 1.1 response without any Connection header. Some 1.0 clients might know that in 1.1 this means "keep-alive" while others might ignore the version and assume a "close". Since we know the version on both sides, we may have to adjust some responses to remove any ambiguous case. That's the reason why the following table considers both the request and the response version. If the response length cannot be determined, we switch to CLO mode.

mode	res_ver	res_hdr	req_ver	new_mode	hdr_change
TUN	1.0	-	any	TUN	-

TUN	1.0	ka	any	TUN	del_ka
TUN	1.0	close	any	TUN	del_close
TUN	1.0	both	any	TUN	del_ka, del_close
TUN	1.1	-	any	TUN	add_close
TUN	1.1	ka	any	TUN	del_ka, add_close
TUN	1.1	close	any	TUN	-
TUN	1.1	both	any	TUN	del_ka
KAL	1.0	-	any	SCL	add_ka
KAL	1.0	ka	any	KAL	-
KAL	1.0	close	any	SCL	del_close, add_ka
KAL	1.0	both	any	SCL	del_close
KAL	1.1	-	1.0	KAL	add_ka
KAL	1.1	-	1.1	KAL	-
KAL	1.1	ka	1.0	KAL	-
KAL	1.1	ka	1.1	KAL	del_ka
KAL	1.1	close	1.0	SCL	del_close, add_ka
KAL	1.1	close	1.1	SCL	del_close
KAL	1.1	both	1.0	SCL	del_close
KAL	1.1	both	1.1	SCL	del_ka, del_close
SCL	1.0	-	any	SCL	add_ka
SCL	1.0	ka	any	SCL	-
SCL	1.0	close	any	SCL	del_close, add_ka
SCL	1.0	both	any	SCL	del_close
SCL	1.1	-	1.0	SCL	add_ka
SCL	1.1	-	1.1	SCL	-
SCL	1.1	ka	1.0	SCL	-
SCL	1.1	ka	1.1	SCL	del_ka
SCL	1.1	close	1.0	SCL	del_close, add_ka
SCL	1.1	close	1.1	SCL	del_close
SCL	1.1	both	1.0	SCL	del_close
SCL	1.1	both	1.1	SCL	del_ka, del_close
CLO	1.0	-	any	CLO	-
CLO	1.0	ka	any	CLO	del_ka
CLO	1.0	close	any	CLO	del_close
CLO	1.0	both	any	CLO	del_ka, del_close
CLO	1.1	-	any	CLO	add_close
CLO	1.1	ka	any	CLO	del_ka, add_close
CLO	1.1	close	any	CLO	-
CLO	1.1	both	any	CLO	del_ka

=> in summary :

- the header operations do not depend on the initial mode, they only depend on versions and current connection header(s).

- both CLO and TUN modes work similarly, they need to set a close mode on the reponse. A 1.1 response will exclusively need the close header, while a 1.0 response will have it removed. Any keep-alive header is always removed when found.

- a KAL request where the server wants to close turns into an SCL response so that we release the server but still maintain the connection to the client.

- the KAL and SCL modes work the same way as we need to set keep-alive on the response. So a 1.0 response will only have the keep-alive header with any close header removed. A 1.1 response will have the keep-alive header added for 1.0 requests and the close header removed for all requests.

Note that the SCL and CLO modes will automatically cause the server connection

to be closed at the end of the data transfer.