

DCCP Simultaneous-Open Technique:

A journey through middleboxes

Gerrit Renker

Godred Fairhurst

draft-ietf-dccp-simul-open-00

Outline

1. Background Information (brief)
2. Changes Since Last Revision
3. Further Work

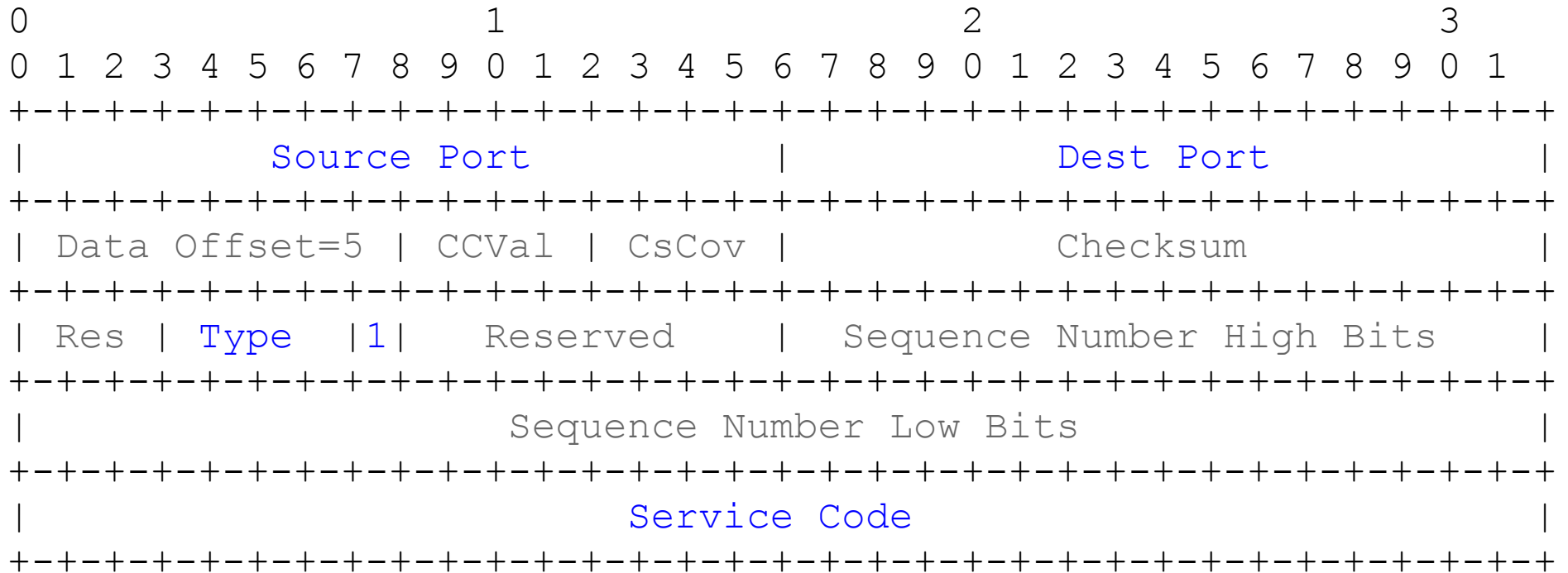
Background Information

- existing DCCP:
 - NAT traversal similar to TCP
 - but no simultaneous-open
 - hence has to rely on relay or tunnel
- this draft:
 - produces *effect* of simultaneous-open
 - without changes to standard DCCP operation
 - *indicator packet* to establish middlebox state
 - may help with *traversal of other middleboxes*

Server initialisation

- server initially needs to know ...
 - client IP-address:port and
 - client's chosen service code
- possible ways: SDP, RTSP, SIP, etc.
- to populate the 5-tuple ...
<src-IP, src-port, Service Code, dst-IP, dst-port>
- ... which is then communicated to NAT

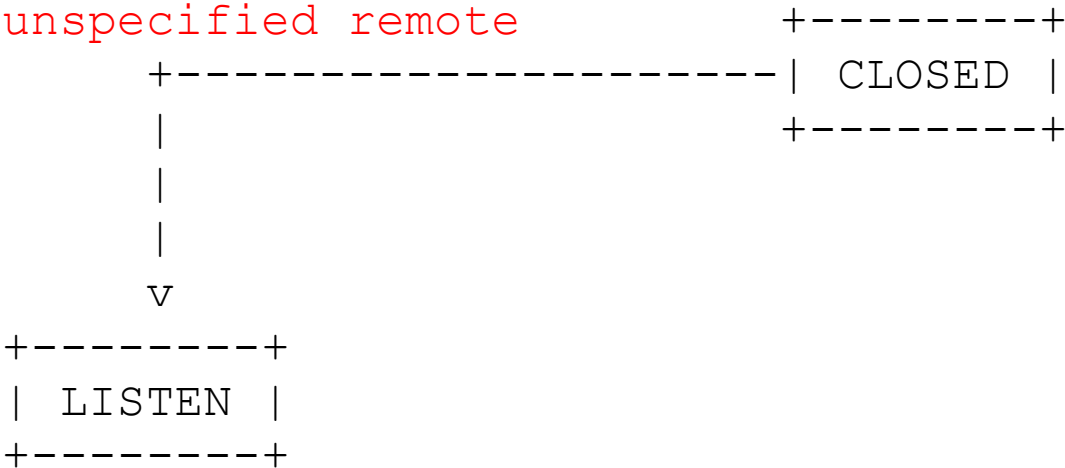
DCCP-Listen Packet



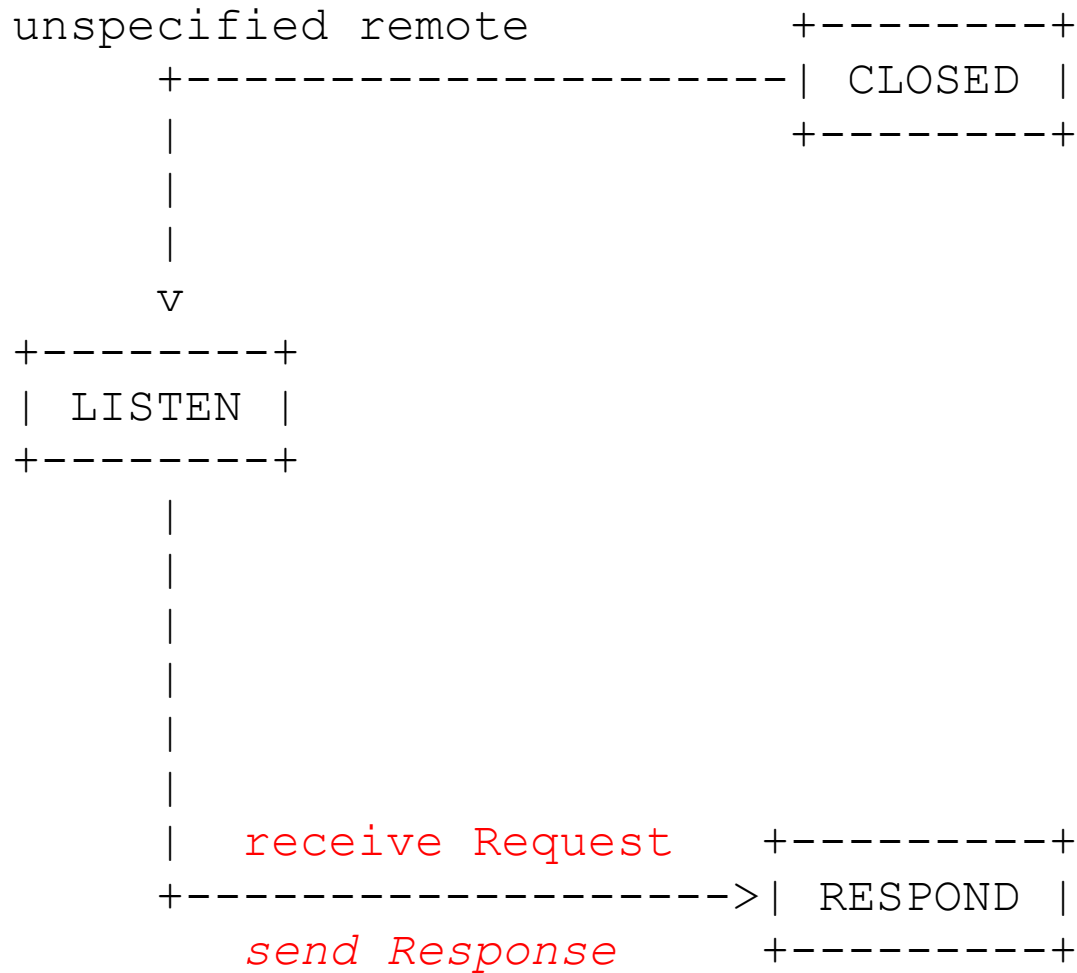
DCCP-Listen Packet

- intended for NAT, ignored by endpoints
- reuses DCCP-Request layout
- different **Type** value, because ...
 - avoid *semantic overloading* of packet types
 - RFC4340-clients may reply with Reset or Sync
 - would require *complicated processing rules*
- **Format:**
 - no payload / options carried on DCCP-Listen
 - *reuses 48-bit generic-header variant (X=1)*
 - sequence number values are ignored

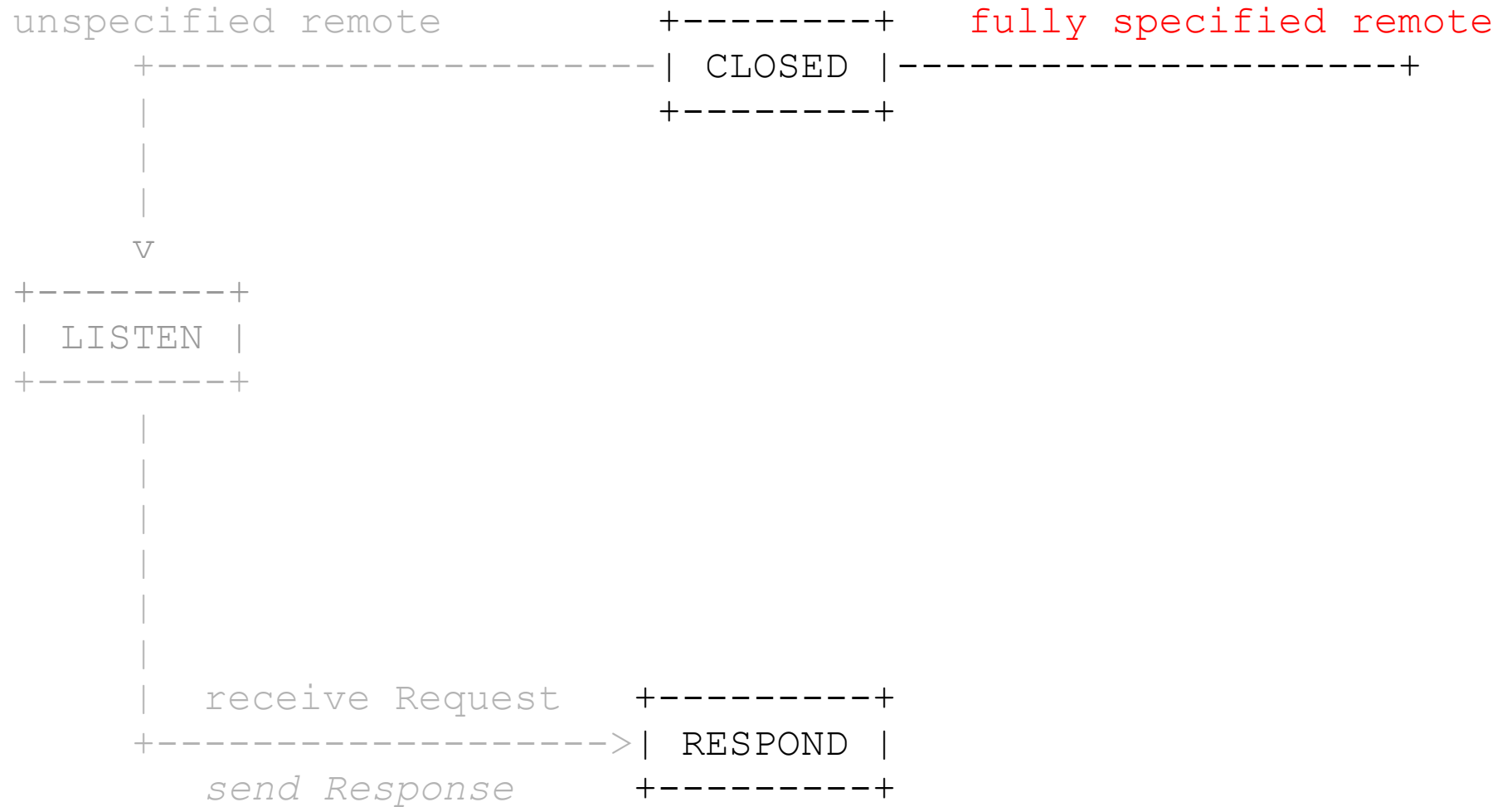
Standard DCCP passive-open



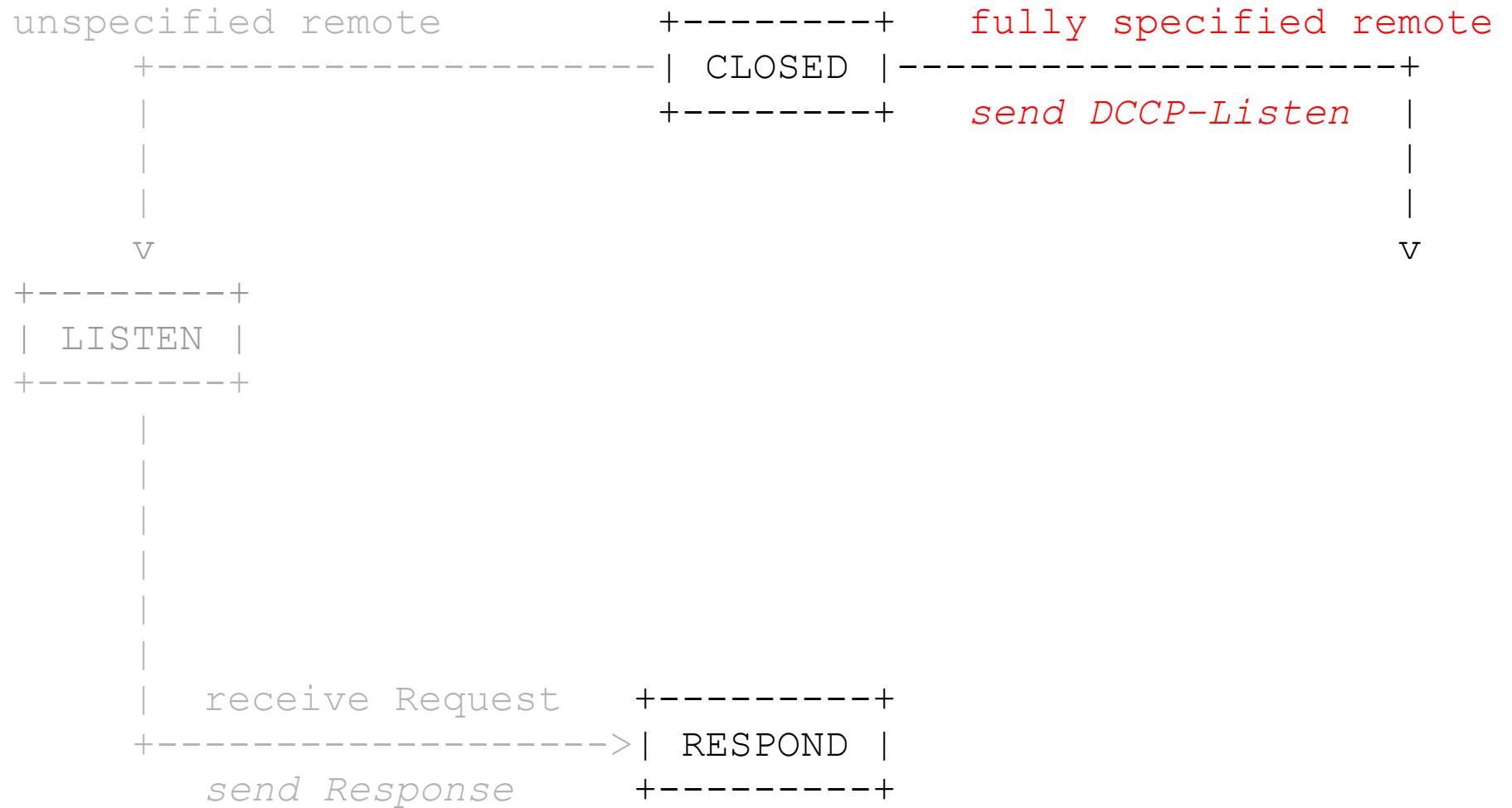
Standard DCCP passive-open



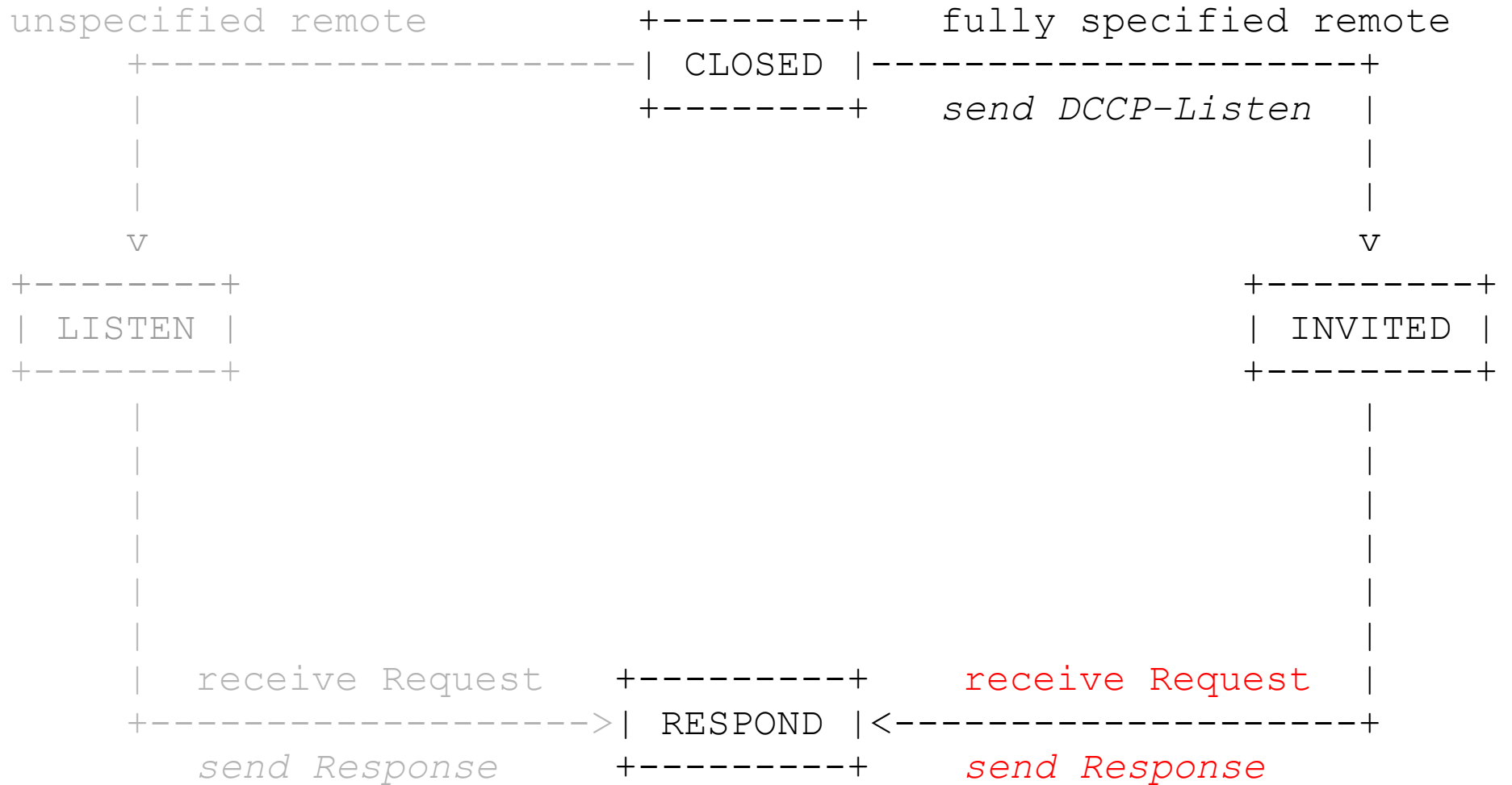
Fully-specified passive-open



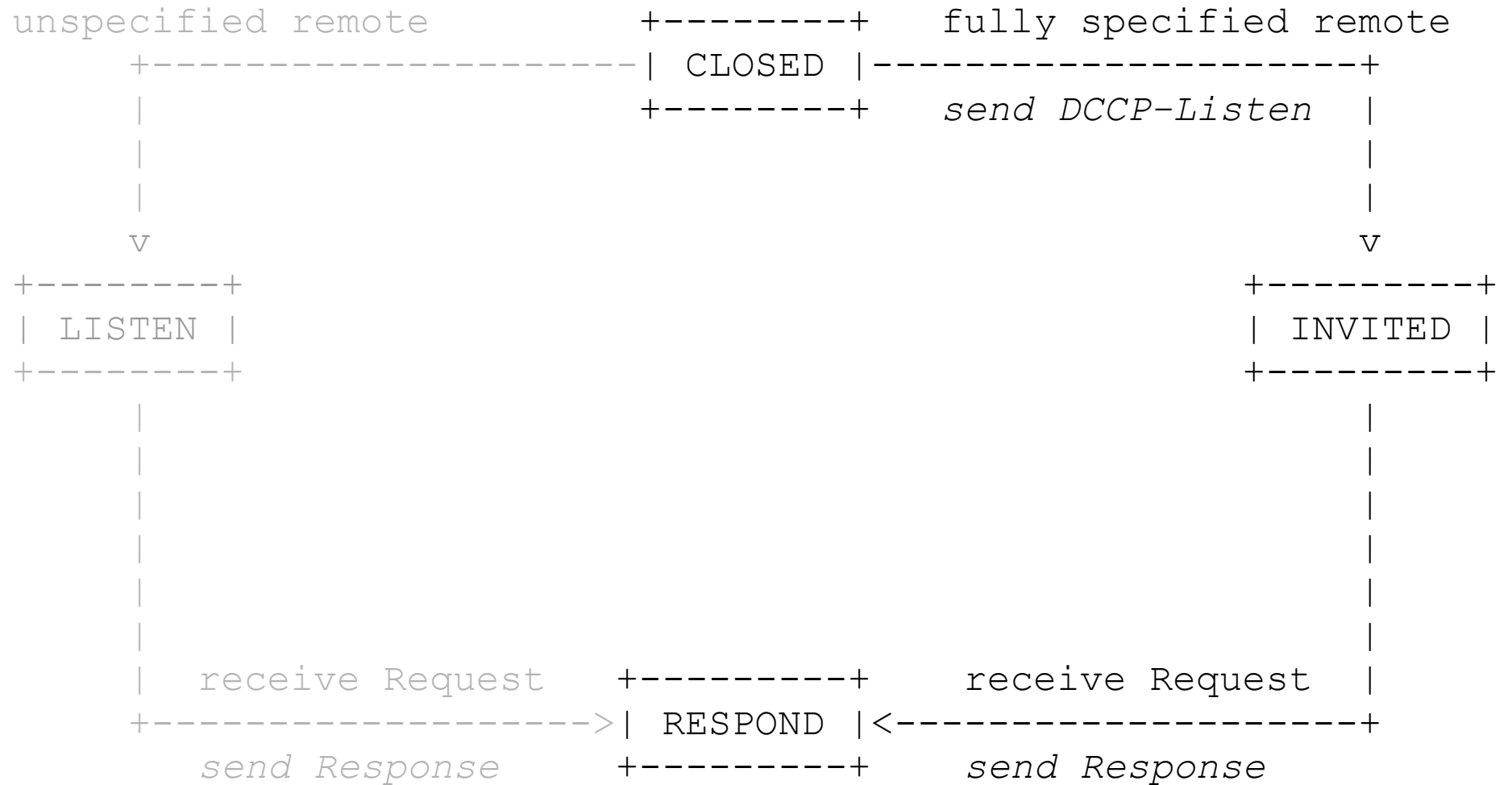
Fully-specified passive-open



Fully-specified passive-open



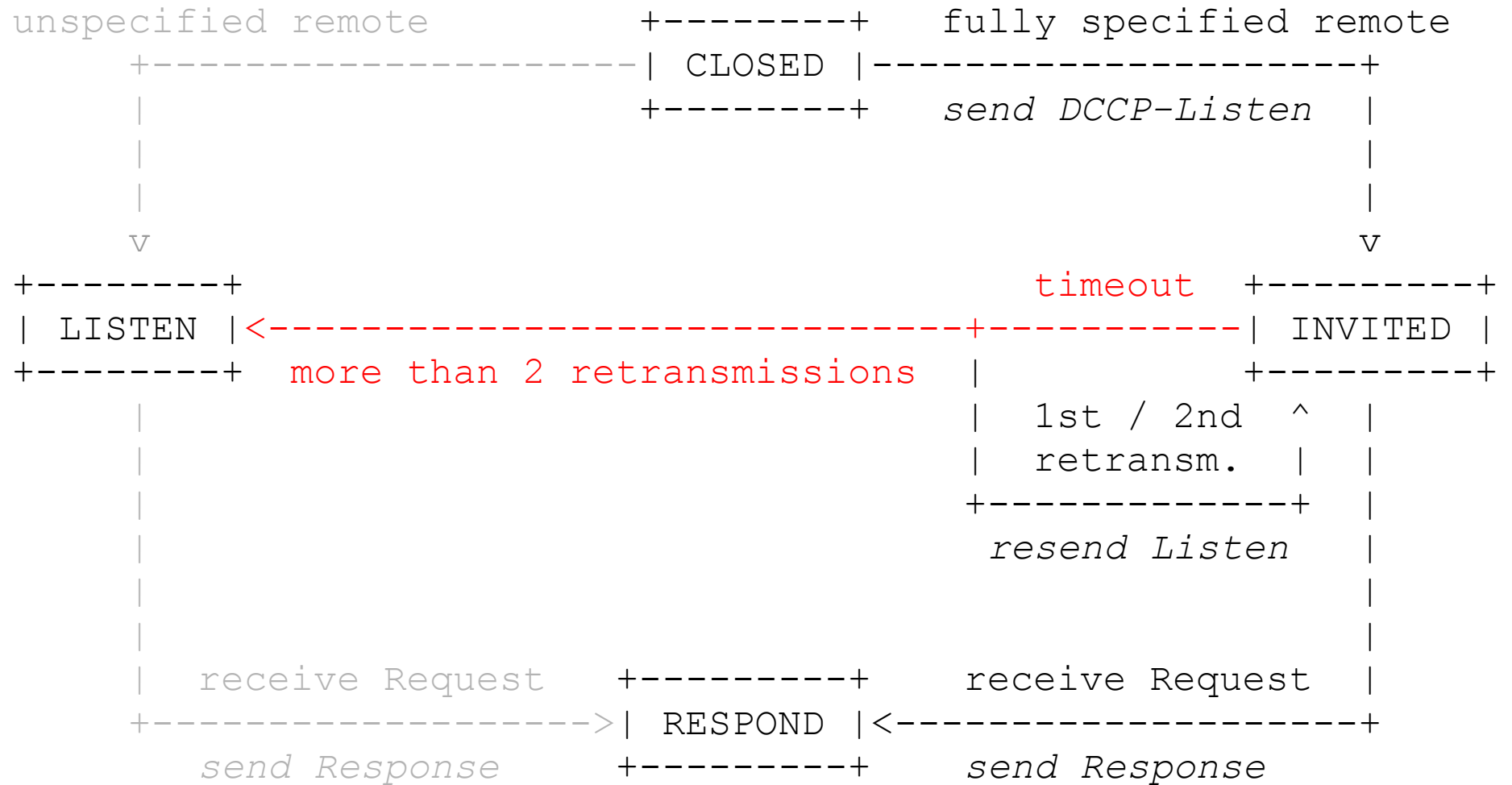
What if DCCP-Listen is lost?



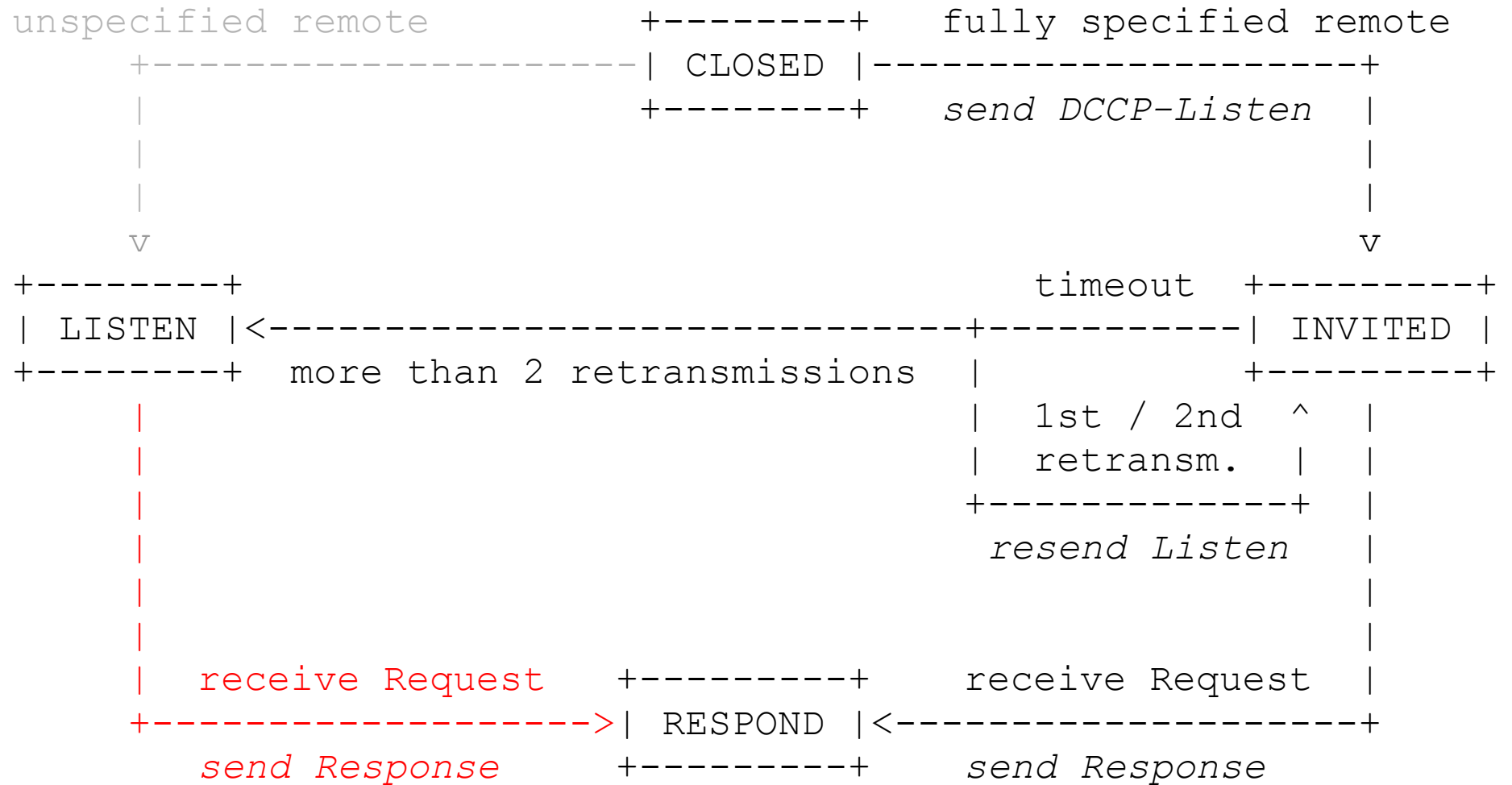
... or lost again?



Fallback after 2 retransmissions



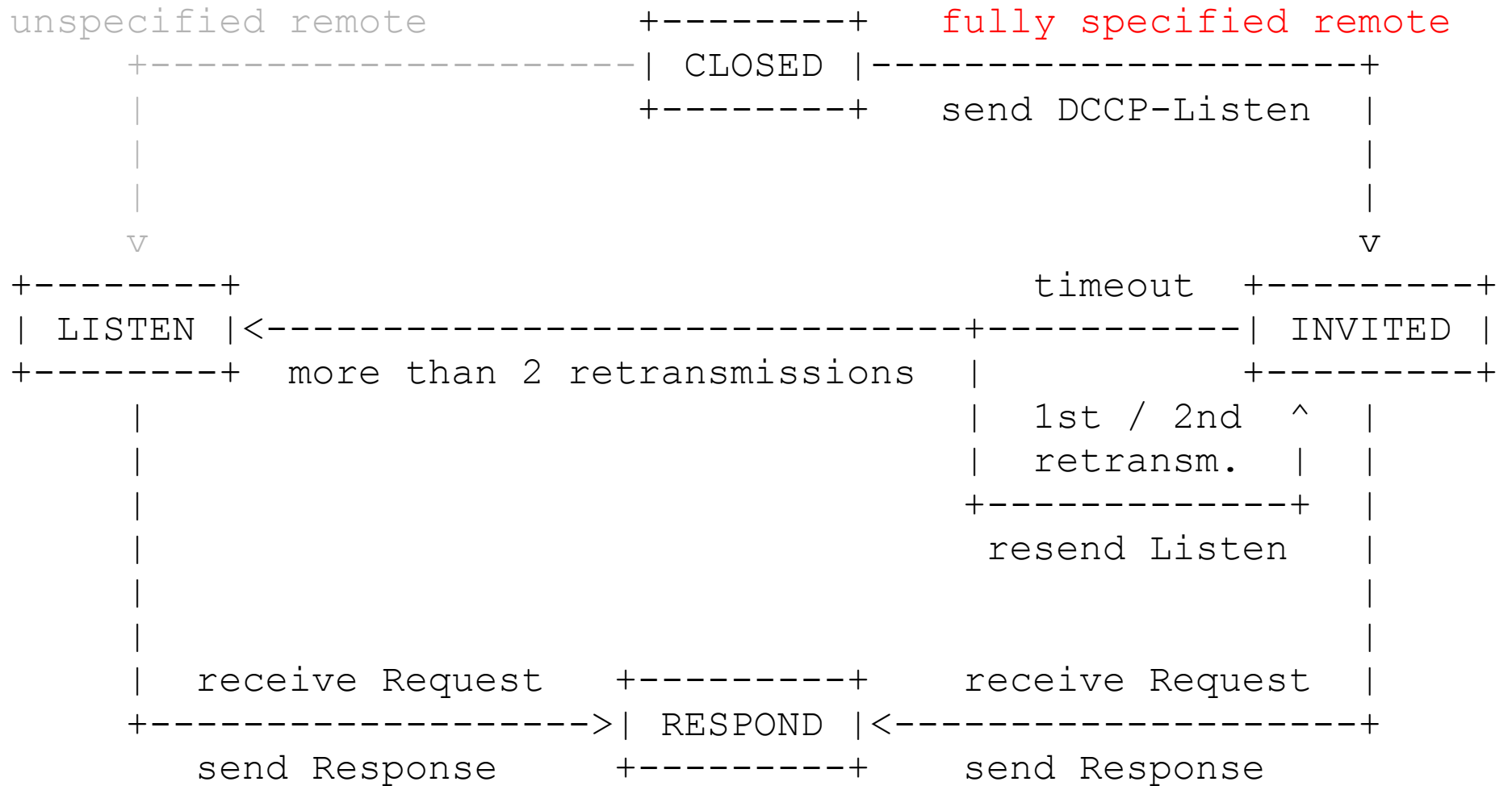
Fallback after 2 retransmissions



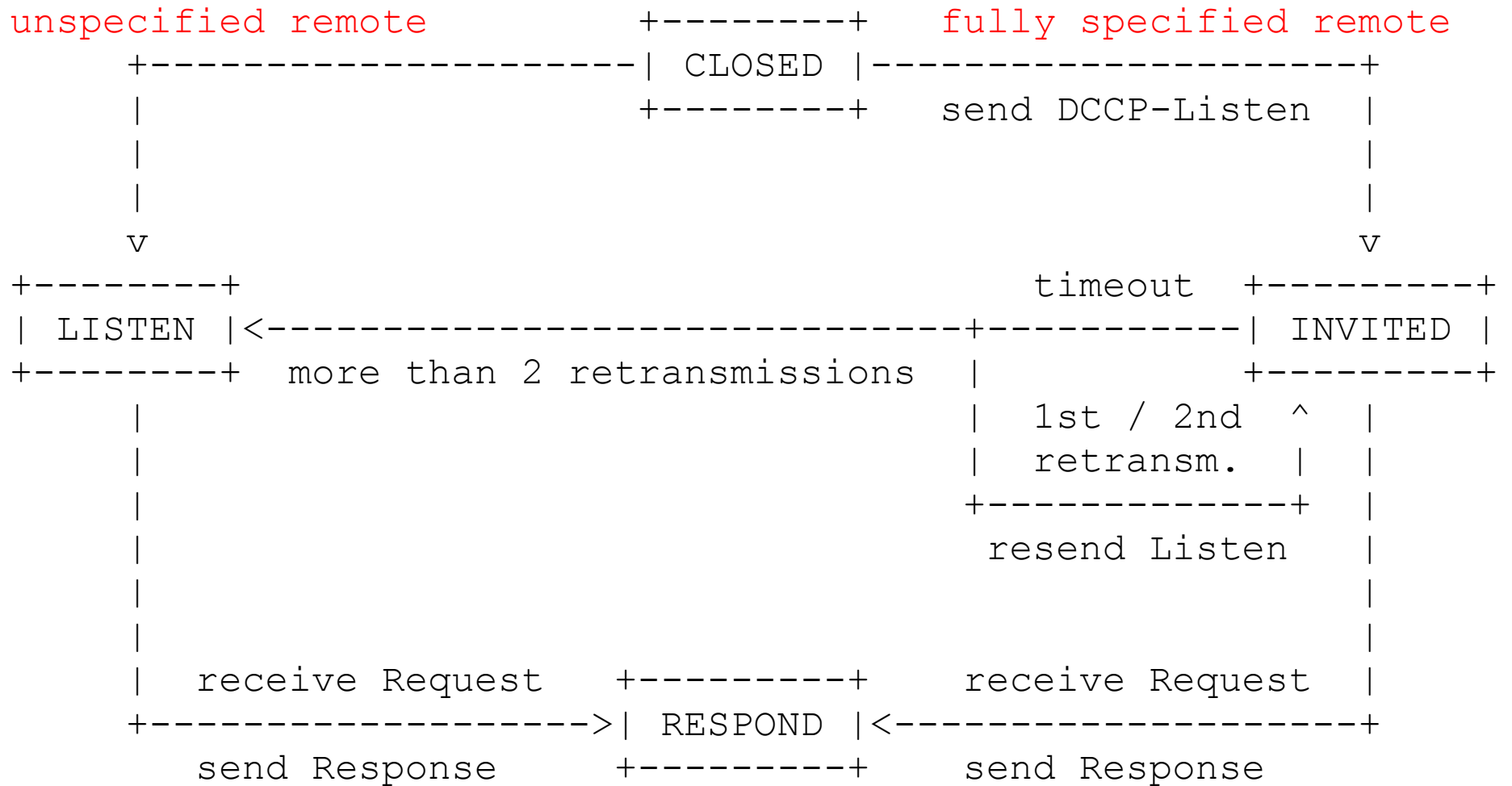
The extension is conditional



The extension is conditional



The extension is conditional



Retransmission of DCCP-Listen

- **Requirement:** *establish state before client retransmits DCCP-Request (1 second)*
- **Goals:**
 - 1) keep it simple (soft-state only)
 - 2) avoid correlated loss
 - 3) longer RTT: middlebox may not be next hop
 - 4) provide fall-back if everything else fails

Retransmission of DCCP-Listen

- **Requirement:** *establish state before client retransmits DCCP-Request (1 second)*
- **Goals:**
 - 1) keep it simple (soft-state only)
 - 2) avoid correlated loss
 - 3) longer RTT: middlebox may not be next hop
 - 4) provide fall-back if everything else fails
- **Chosen solution:**
 - 2 retransmissions, 200 msec apart
 - after 600 msec, fall back to standard LISTEN

Traversal Implementation

- **Linux DCCP NAT module**
 - current work-in-progress (Patrick McHardy)
 - quite likely available in upcoming 2.6.26
 - UDP-Lite NAT traversal already works!
- **Linux kernel support for DCCP-Listen**
 - seems straightforward
 - testing as soon as NAT module available

Traversal Implementation

- **Linux DCCP NAT module**
 - current work-in-progress (Patrick McHardy)
 - quite likely available in upcoming 2.6.26
 - UDP-Lite NAT traversal already works!
- **Linux kernel support for DCCP-Listen**
 - seems straightforward
 - testing as soon as NAT module available
- **We are interested in other / further implementations!**

Conclusions

- no change to standard DCCP operation
 - conditional extension
 - communicate all state to middlebox
 - simple retransmission scheme
- may work for other middleboxes, too:
 - interested in implementation
 - solicit feedback regarding wider applicability