

DCCP Connection Establishment Through Middleboxes

Gerrit Renker & Gorry Fairhurst
University of Aberdeen
Scotland, UK

IETF-70, December 2007, Vancouver.

draft-fairhurst-dccp-behave-update-01

Talk Outline

- Motivation
- Solution Space for DCCP
- DCCP Traversal for Traditional NAT

Motivation

- NAT / middleboxes are here to stay
- No traversal \Leftrightarrow No deployment!
- Standard DCCP only works with NAT relay
- Simpler solution desirable (like for TCP)

NAT/middlebox Traversal

- “Hole Punching” Principle
 - peers connect *nearly simultaneously*
 - connection attempts *create middlebox state*

NAT/middlebox Traversal

- “Hole Punching” Principle
 - peers connect nearly simultaneously
 - connection attempts create middlebox state
- TCP NAT Traversal
 - relies on *simultaneous-open*
 - otherwise complex solutions (relay, TURN)

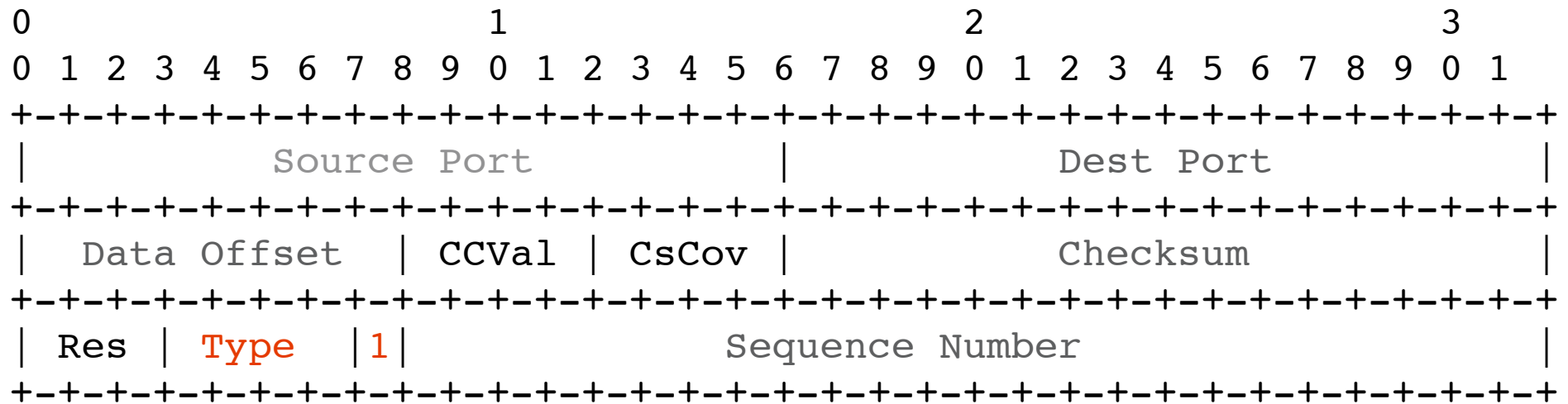
NAT/middlebox Traversal

- “Hole Punching” Principle
 - peers connect nearly simultaneously
 - connection attempts create middlebox state
- TCP NAT Traversal
 - relies on simultaneous-open
 - otherwise complex solutions (relay, TURN)
- DCCP NAT Traversal
 - similar problem as in TCP
 - but no simultaneous-open

DCCP Solution Space

- **Add internal states:**
 - not good; creates more complexity
 - simultaneous-open not an alternative
- **Role-reversal:**
 - one endpoint first active, then passive
 - complex and counter-intuitive
- **Use a type of “*Invite*” message:**
 - tied to condition (fully-specified socket)
 - needs only one conditional transition

DCCP-Listen Packet



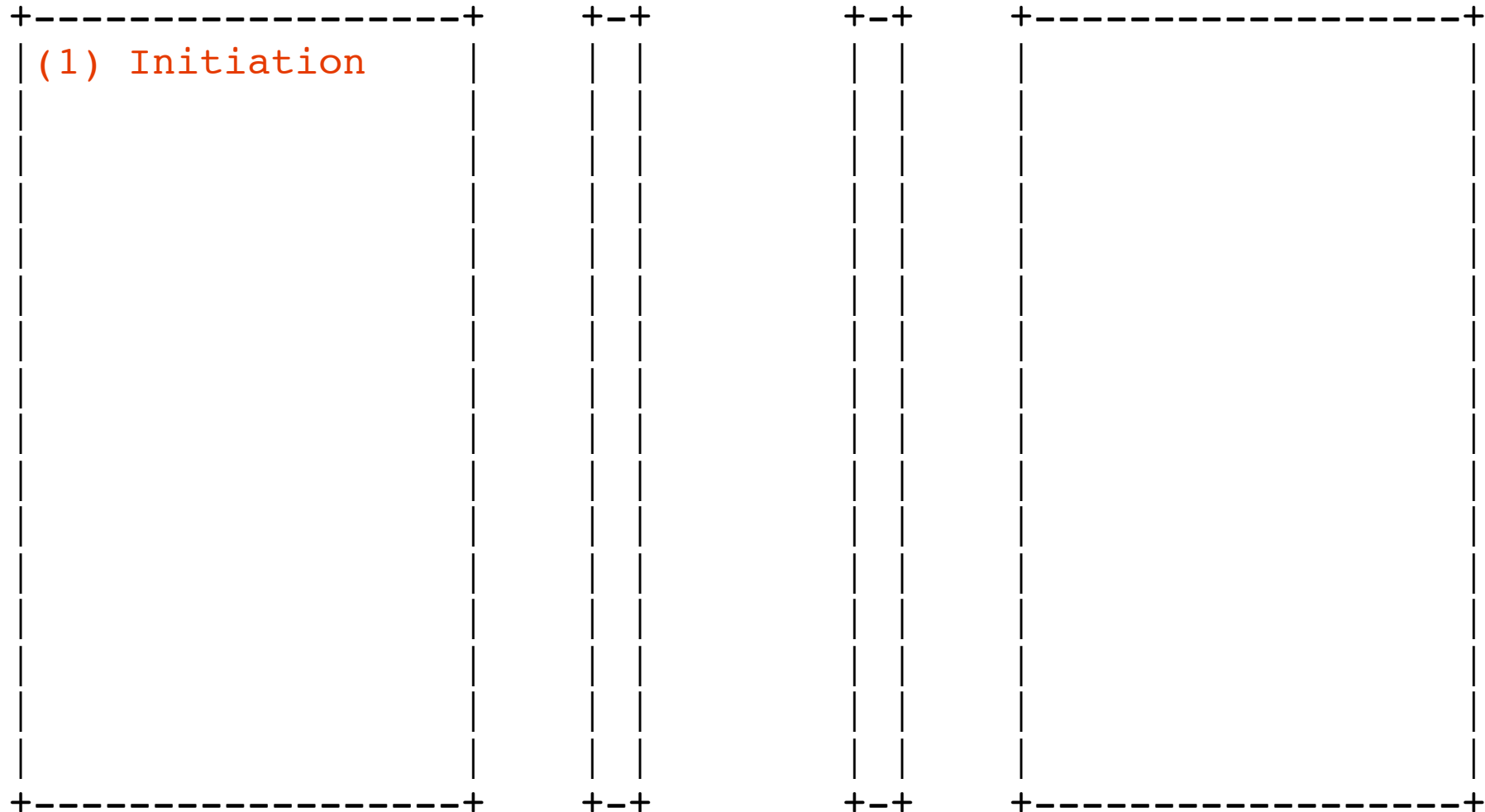
Client starts before server

DCCP A

DCCP B

NA

NB



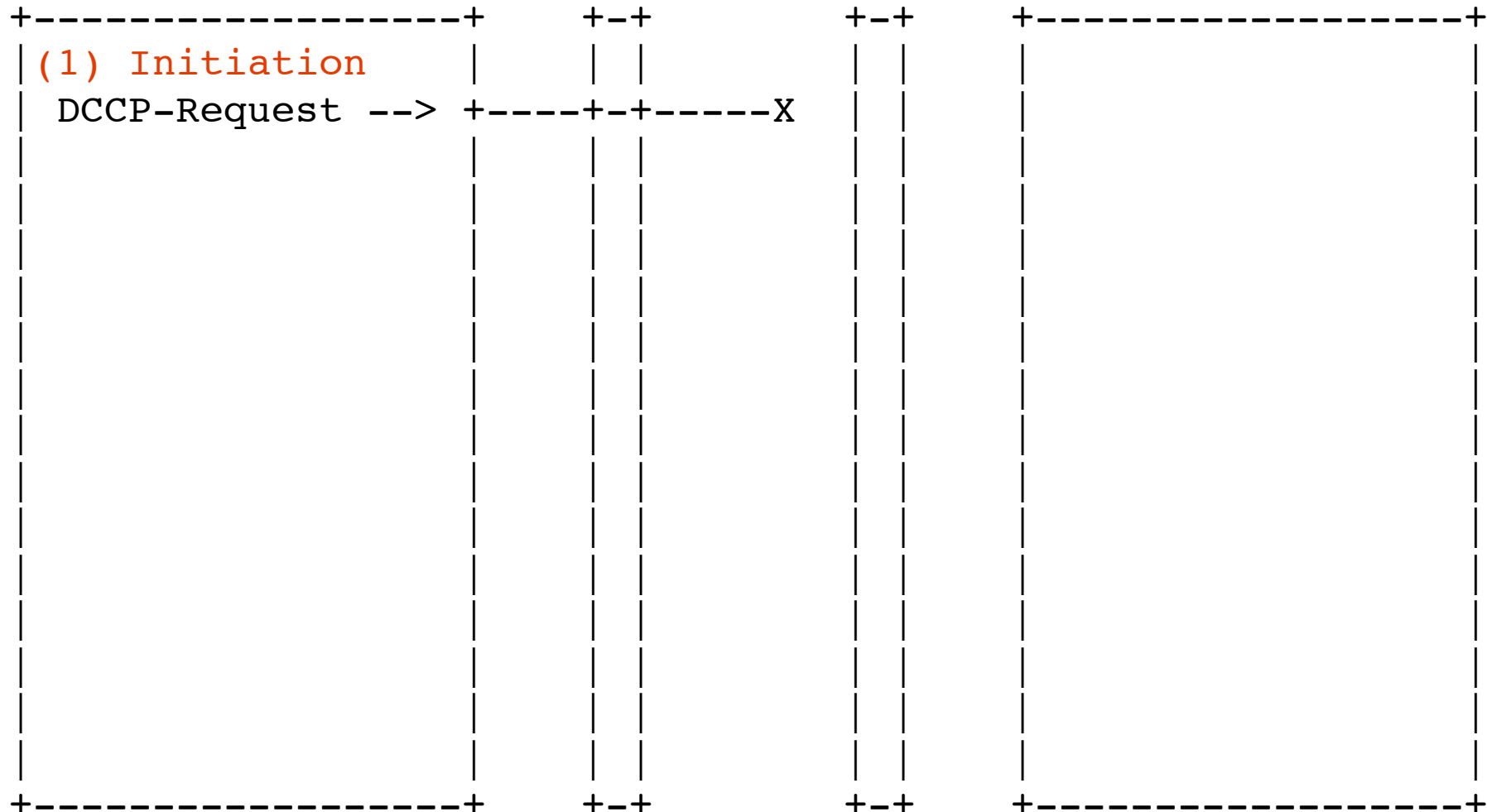
Client starts before server

DCCP A

DCCP B

NA

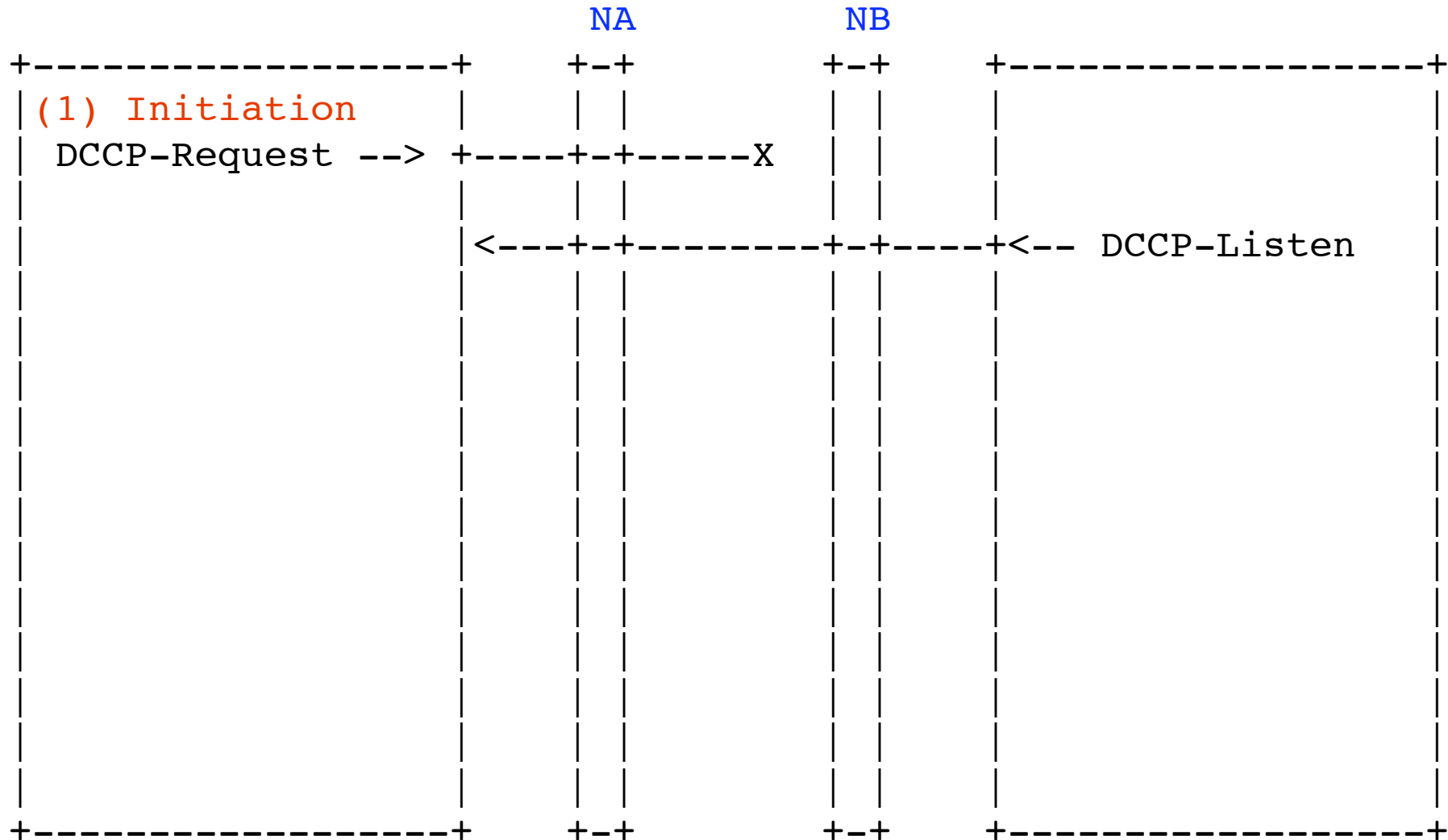
NB



Client starts before server

DCCP A

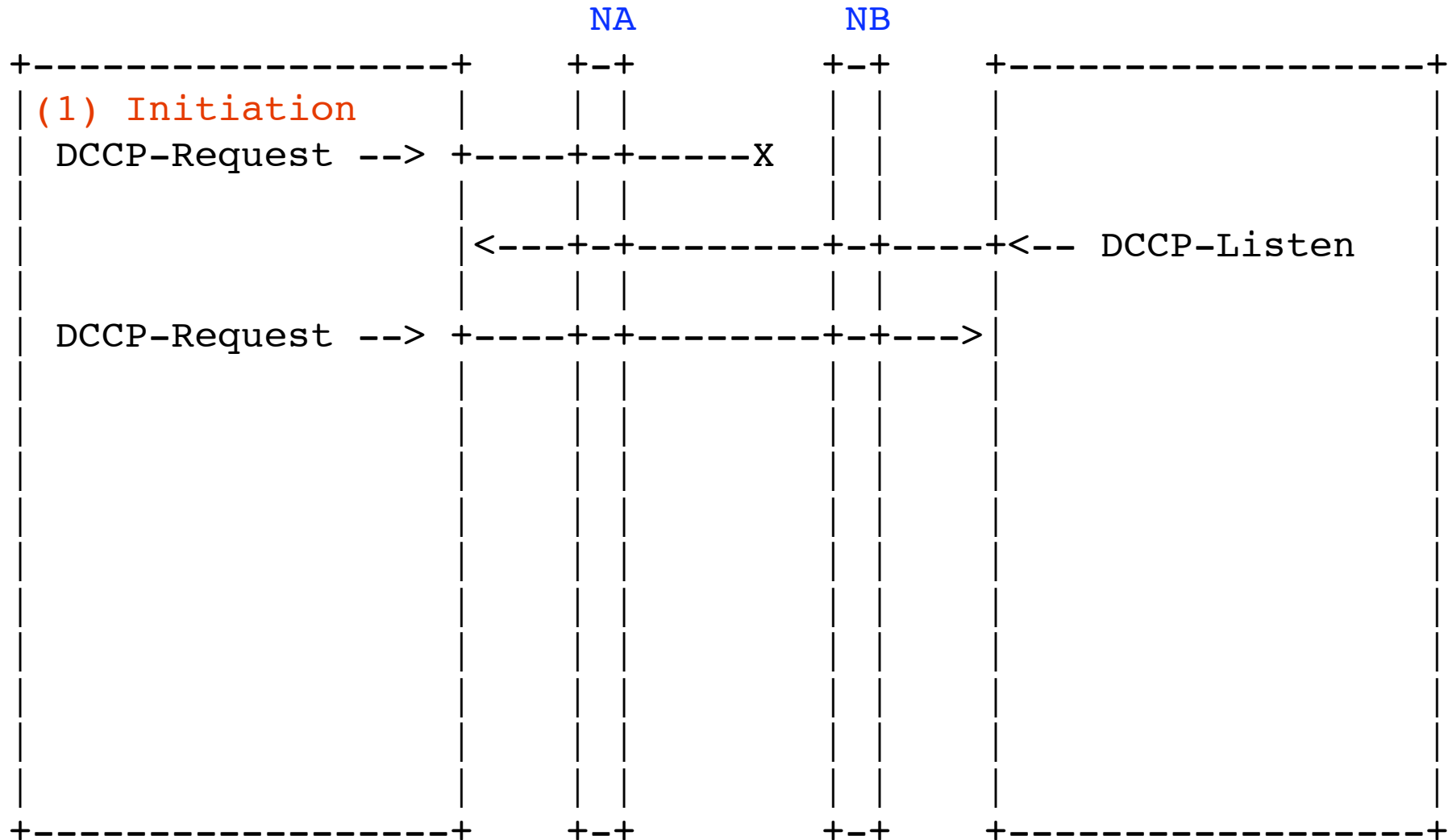
DCCP B



Client starts before server

DCCP A

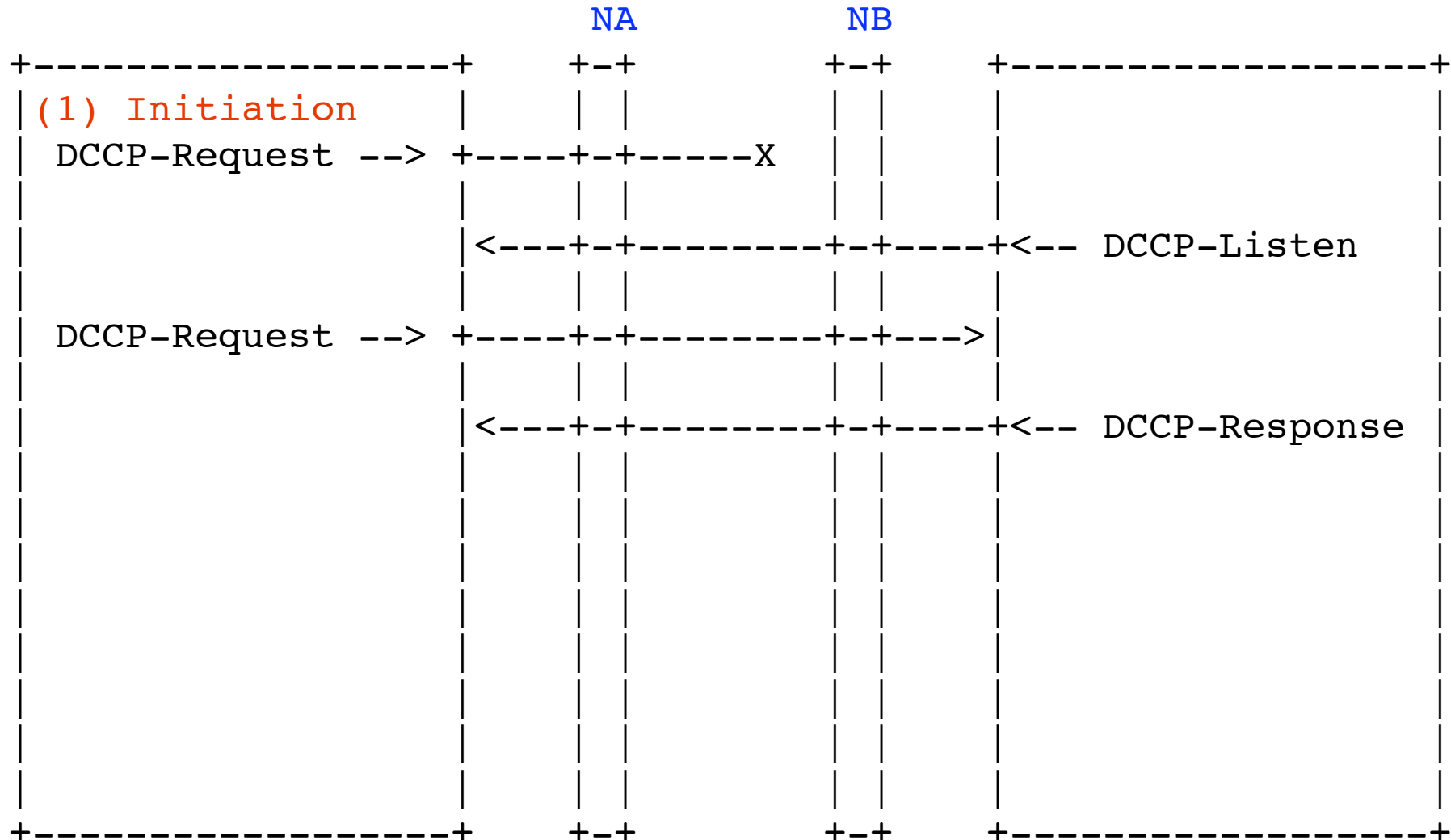
DCCP B



Client starts before server

DCCP A

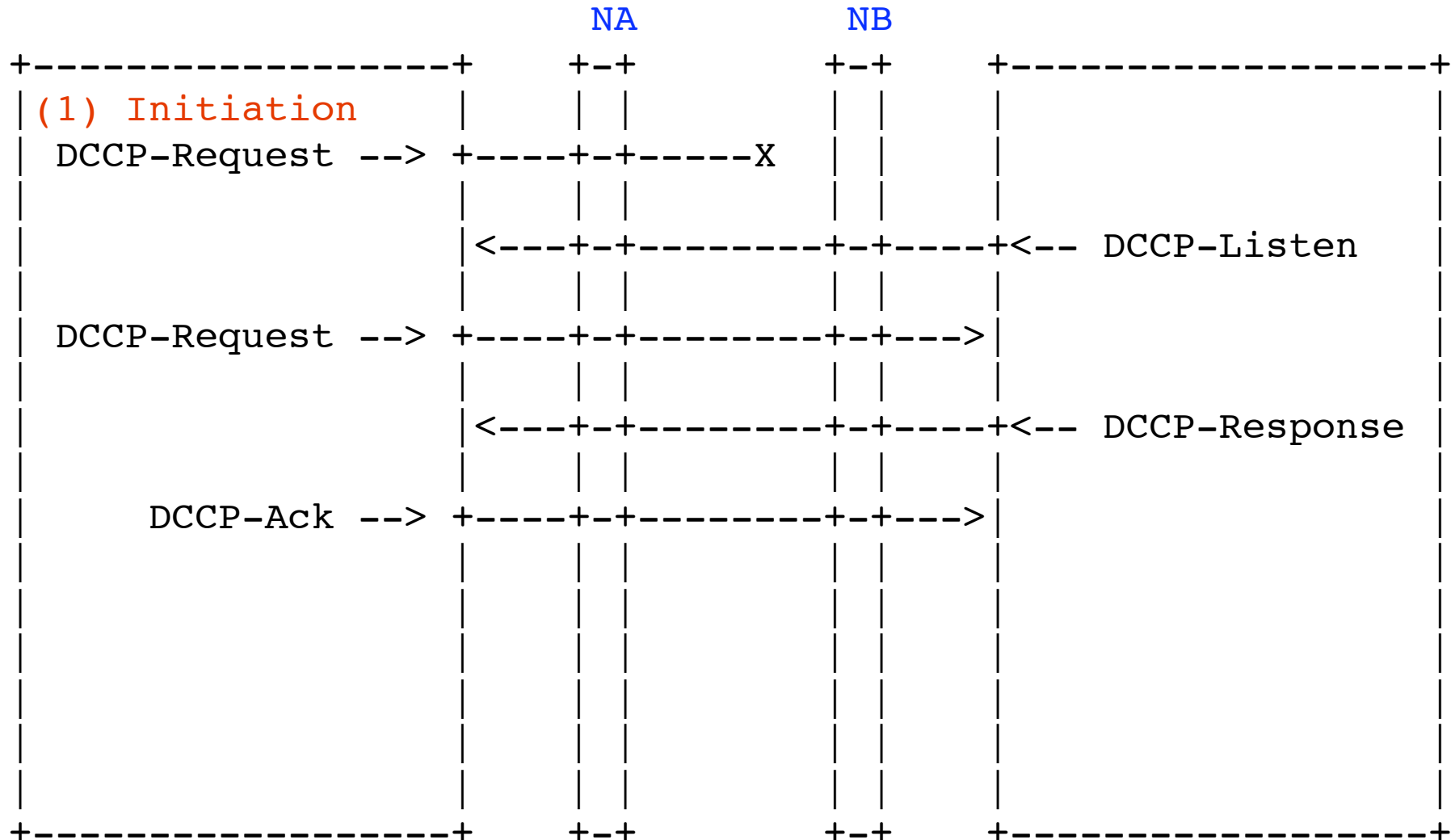
DCCP B



Client starts before server

DCCP A

DCCP B



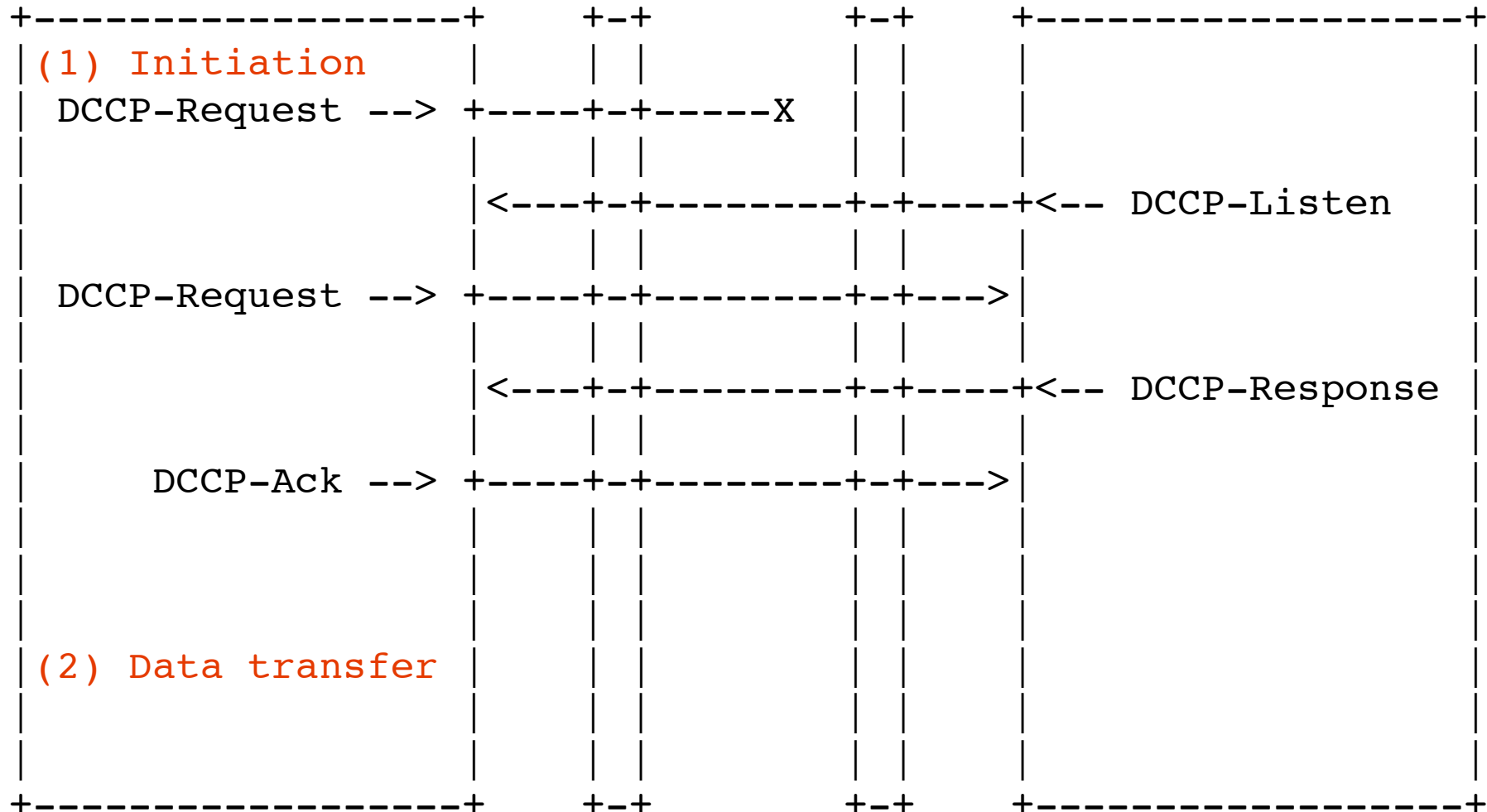
Client starts before server

DCCP A

DCCP B

NA

NB



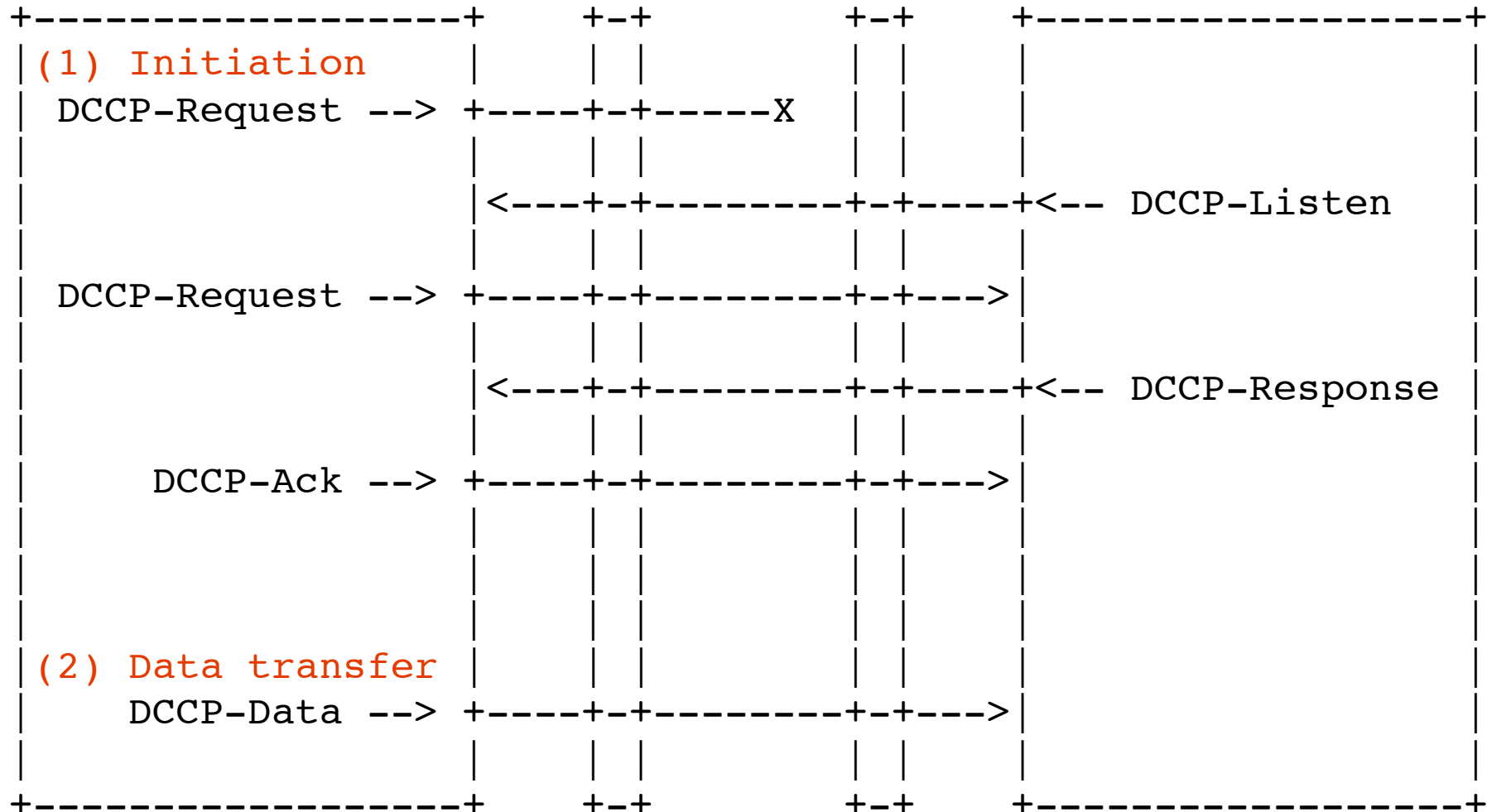
Client starts before server

DCCP A

DCCP B

NA

NB



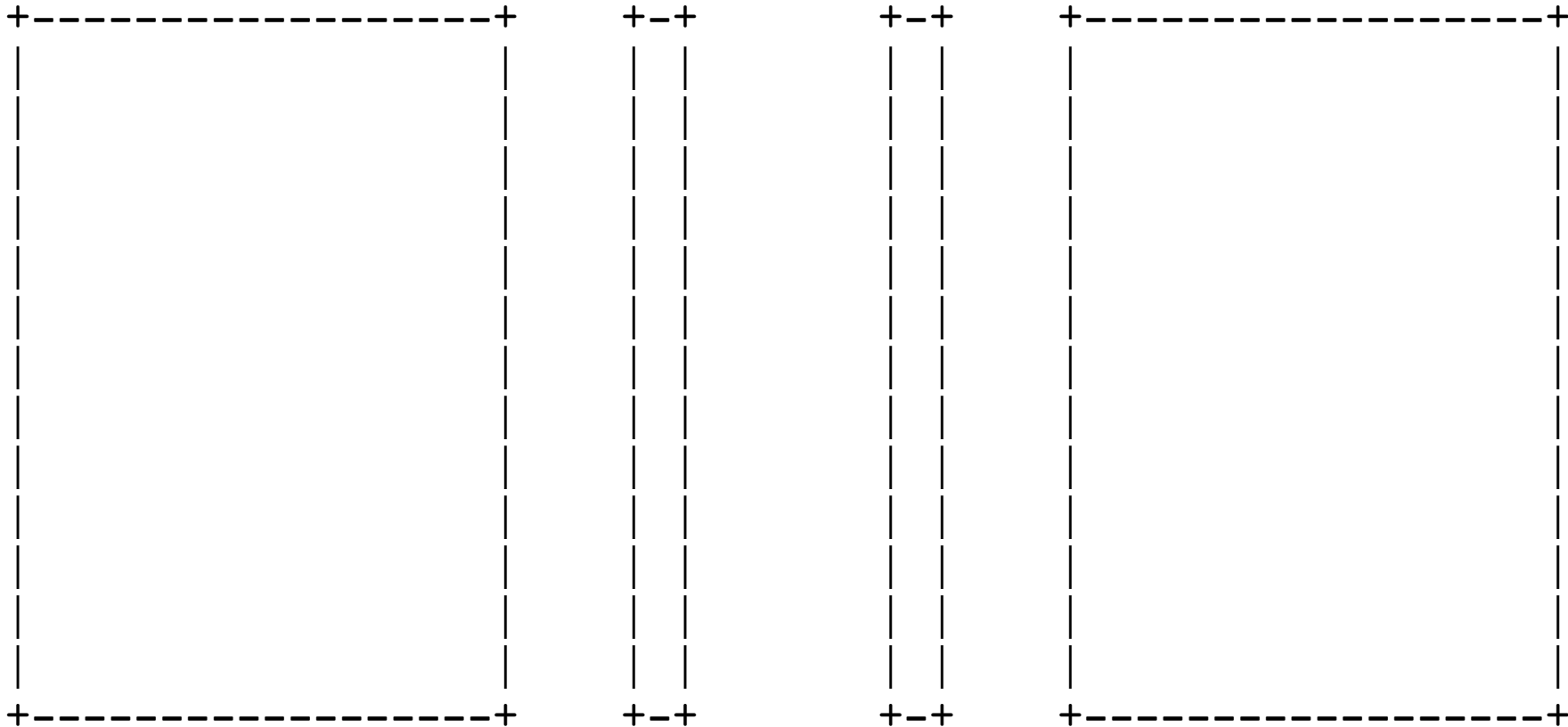
Server starts before client

DCCP A

DCCP B

NA

NB



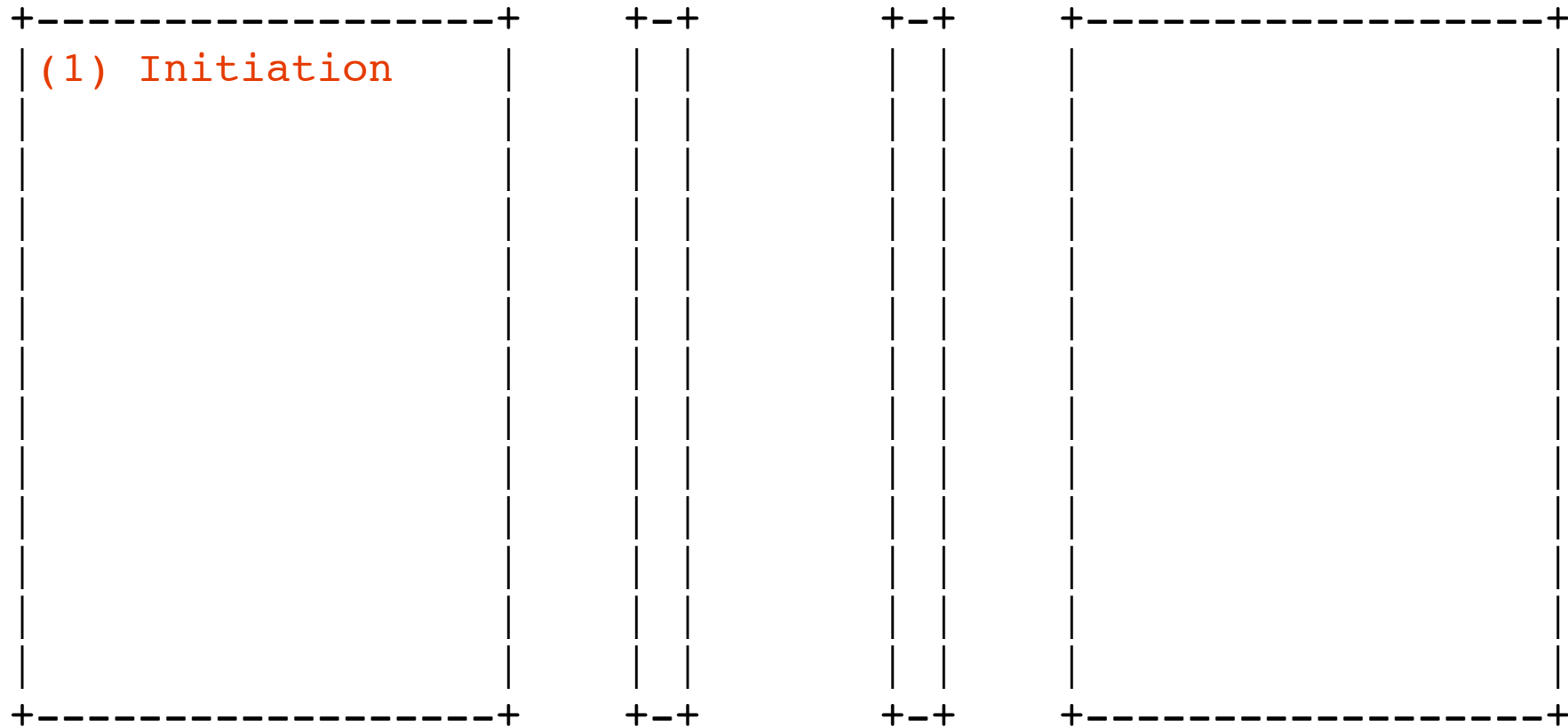
Server starts before client

DCCP A

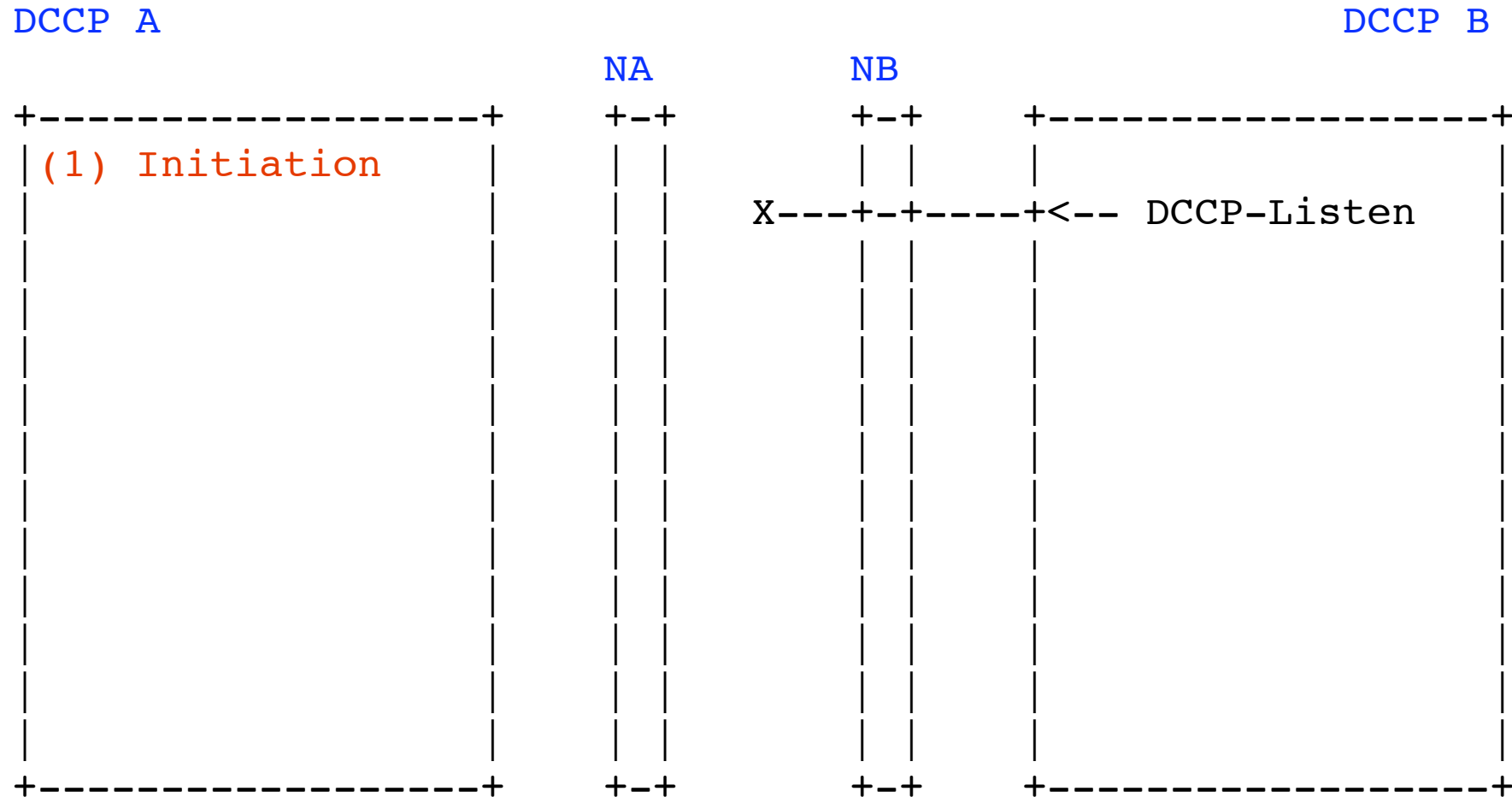
DCCP B

NA

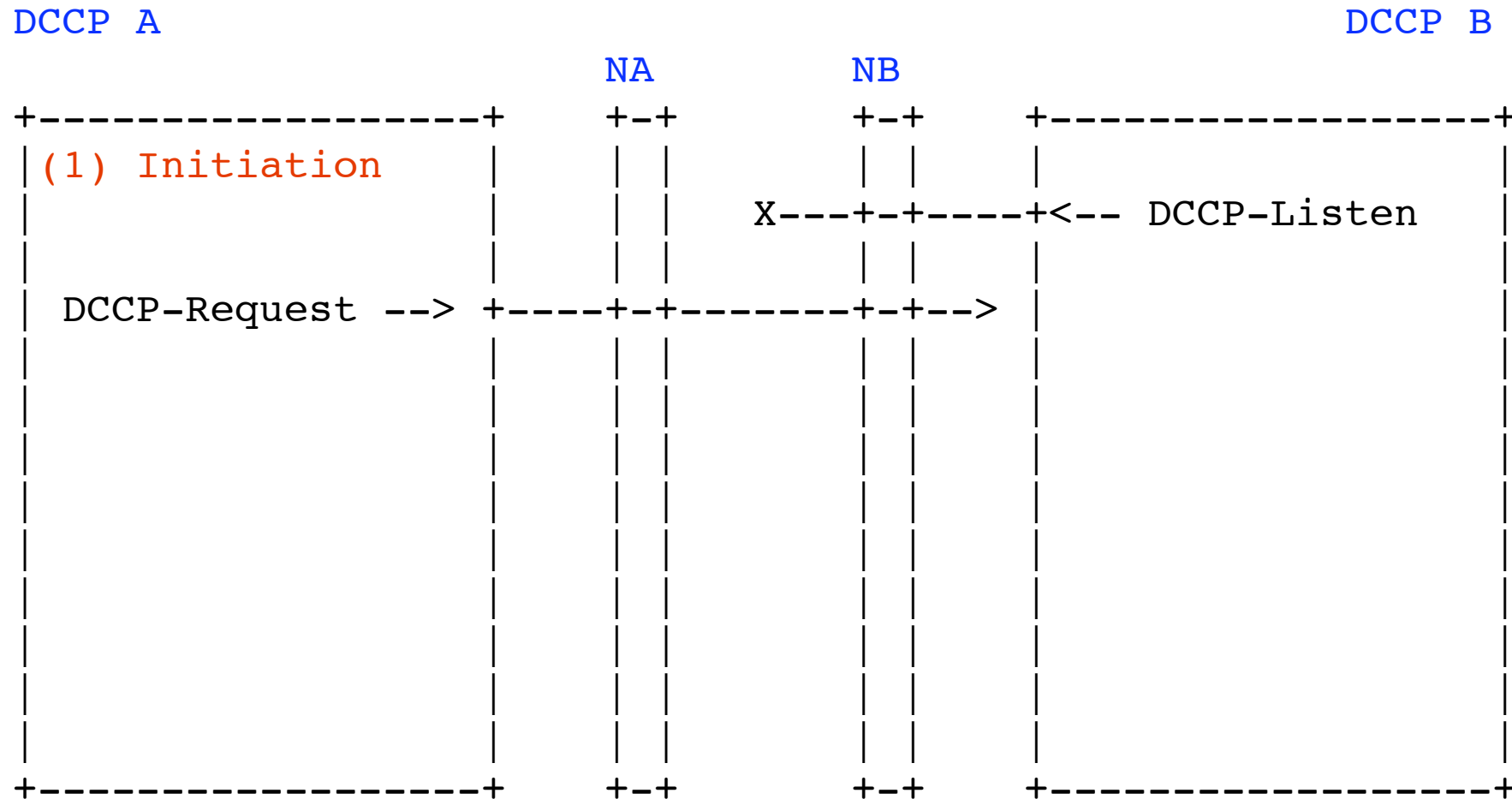
NB



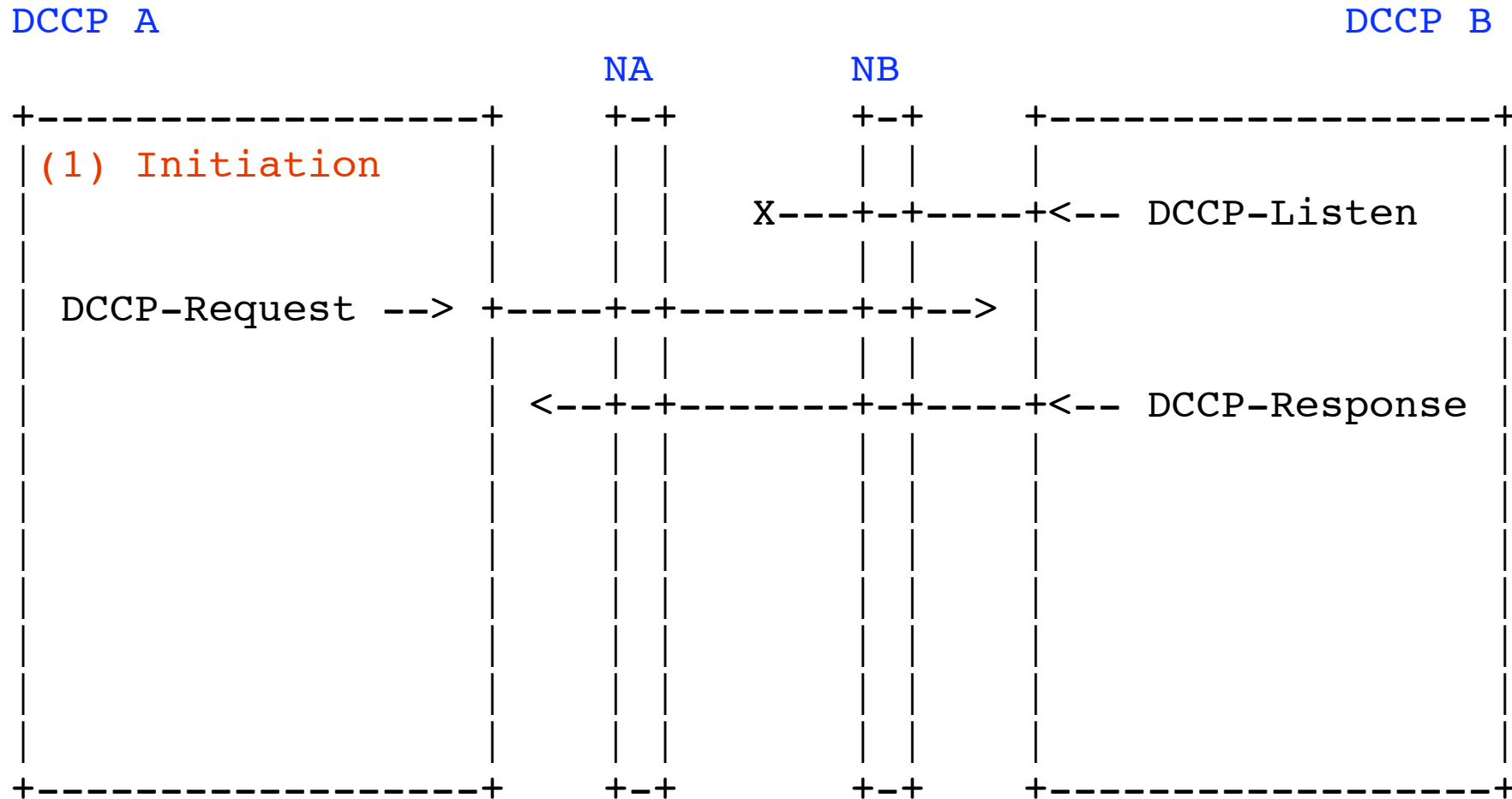
Server starts before client



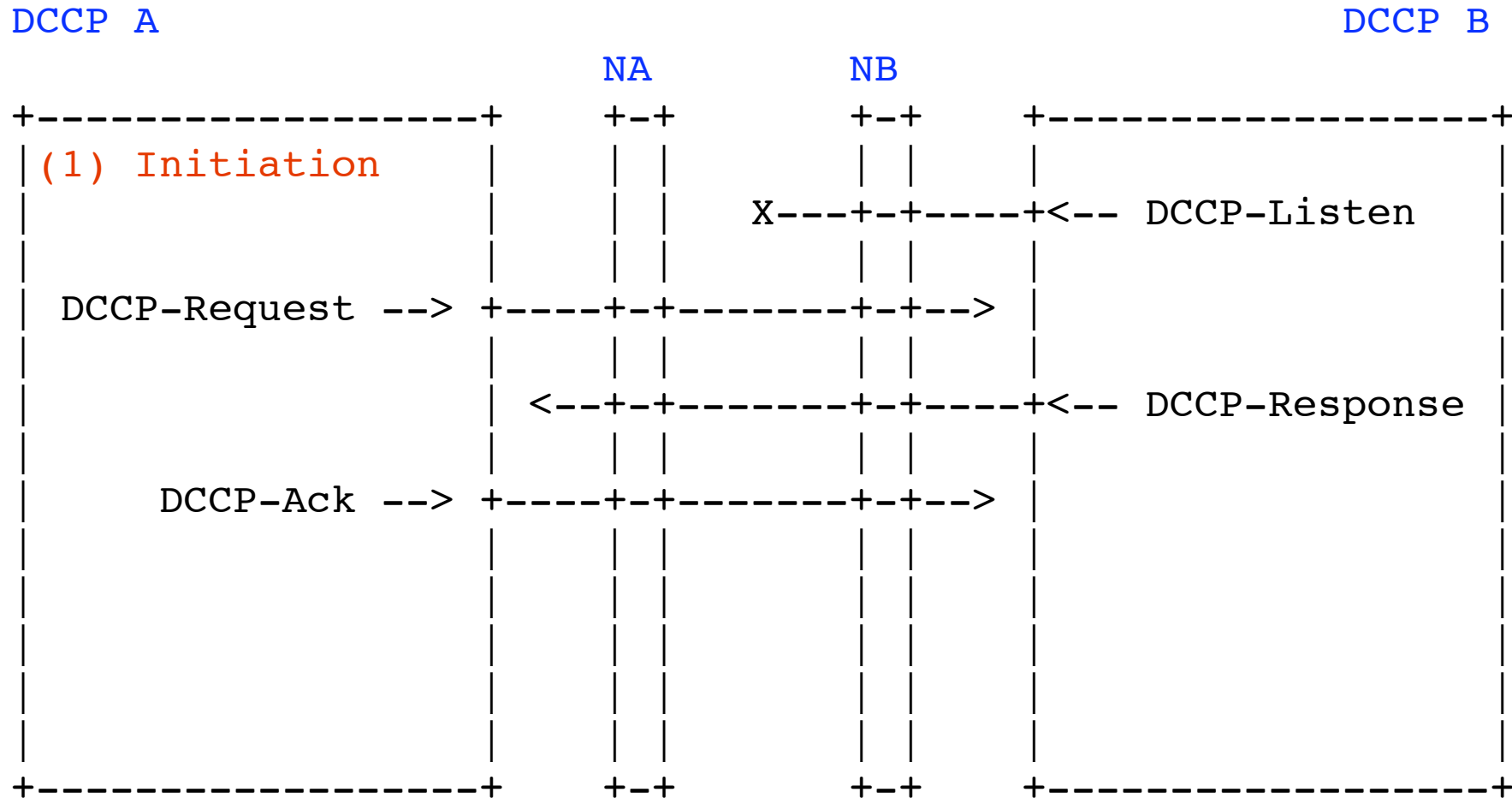
Server starts before client



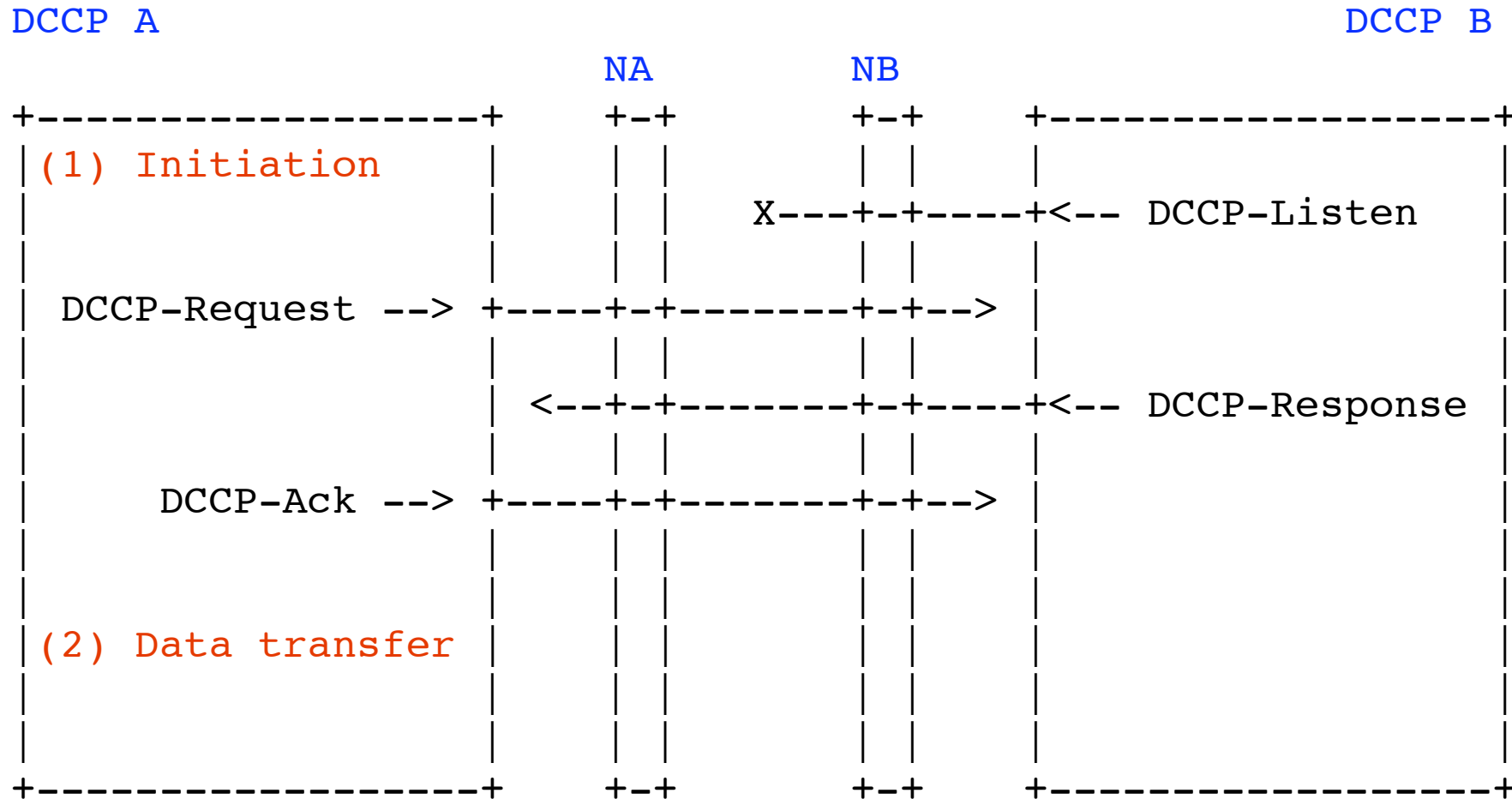
Server starts before client



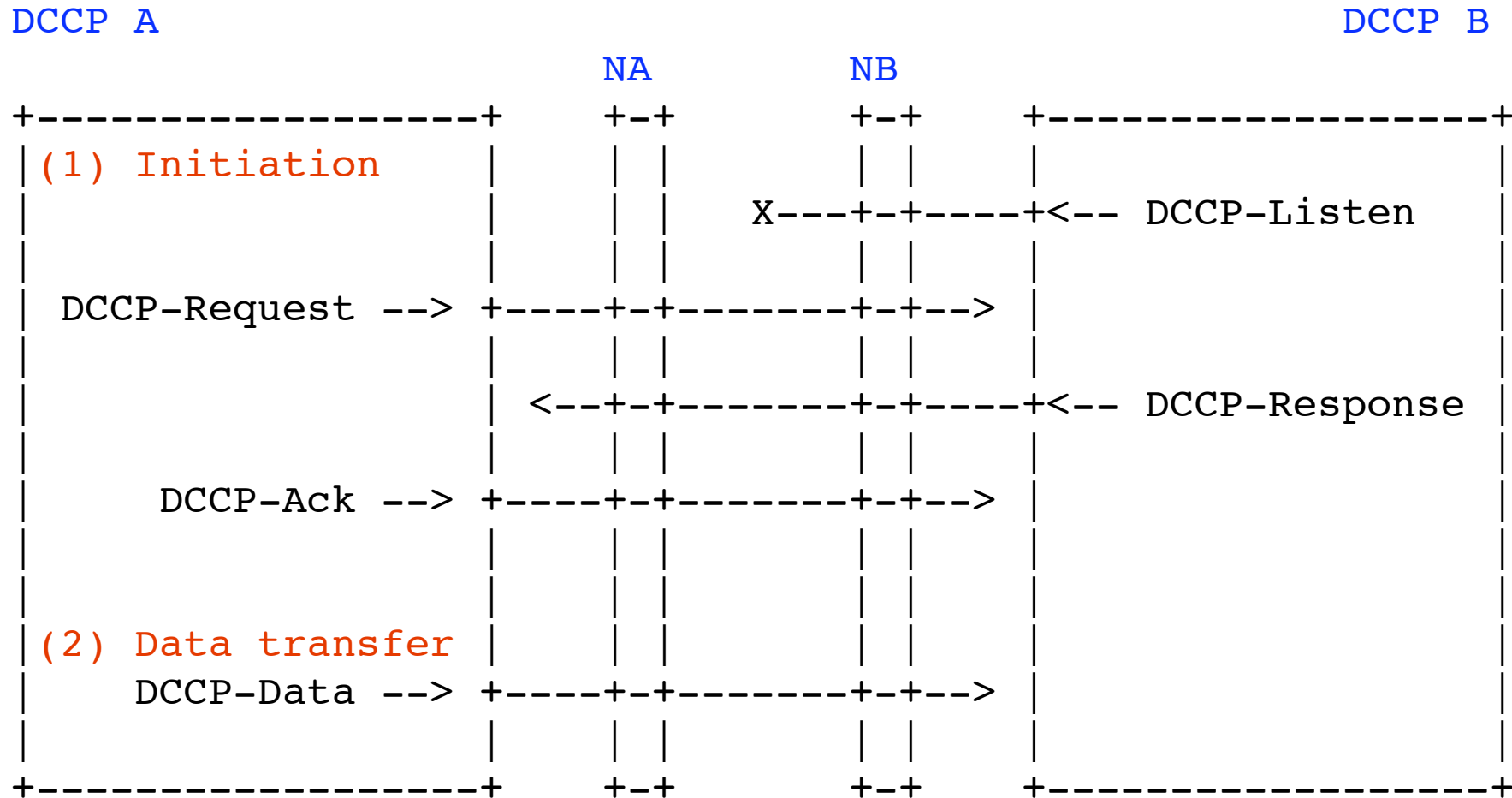
Server starts before client



Server starts before client



Server starts before client



Conclusions

- Built-in middlebox traversal support
- Added a single message
- A single state transition
- Compatible with normal DCCP procedure
- Simple(st?) possible solution

- Do we need to repeat the message?