

Packets were corrupted and protocol was able to recover from it.

```
Terminal - HW3
PS C:\Users\Geoffrey\Classes\Networks\HW3> python server.py --host 127.0.0.1 --port 12000
[SERVER] listening on 127.0.0.1:12000
[SERVER] new peer ('127.0.0.1', 10001)
[RDY ('127.0.0.1', 10001)] -> ACK 0
[RDY ('127.0.0.1', 10001)] DELIVER seq=0 (130)
[SERVER] sending test.txt (1928) to ('127.0.0.1', 10001)
[RDY ('127.0.0.1', 10001)] TX seq=0 len=32
[RDY ('127.0.0.1', 10001)] TX seq=1 len=32
[RDY ('127.0.0.1', 10001)] TX seq=2 len=32
[RDY ('127.0.0.1', 10001)] TX seq=3 len=32
[RDY ('127.0.0.1', 10001)] RETX seq=0
[RDY ('127.0.0.1', 10001)] TX seq=4 len=32
[RDY ('127.0.0.1', 10001)] RETX seq=1
[RDY ('127.0.0.1', 10001)] TX seq=5 len=32
[RDY ('127.0.0.1', 10001)] RETX seq=2
[RDY ('127.0.0.1', 10001)] ACK for seq=0
[RDY ('127.0.0.1', 10001)] ACK for seq=1
[RDY ('127.0.0.1', 10001)] ACK for seq=2
[RDY ('127.0.0.1', 10001)] ACK for seq=3
[RDY ('127.0.0.1', 10001)] -> ACK 0
[RDY ('127.0.0.1', 10001)] ACK for seq=4
[RDY ('127.0.0.1', 10001)] ACK for seq=5
[]

PS C:\Users\Geoffrey\Classes\Networks\HW3> python emulator.py --listen-a 127.0.0.1:10000
--listen-b 127.0.0.1:10001 --server 127.0.0.1:12000 --corrupt 0.20
[EMUL] A('127.0.0.1', 10000) B('127.0.0.1', 10001) server=('127.0.0.1', 12000)
[EMUL] loss=0.00, corrupt=0.20, reorder=0.00, dup=0.00
[EMUL] CORRUPT Server->A
[]

PS C:\Users\Geoffrey\Classes\Networks\HW3> python client.py --server 127.0.0.1:12000 --em
ulator 127.0.0.1:10000 GET test.txt
[RDY ('127.0.0.1', 10000)] TX seq=0 len=13
[RDY ('127.0.0.1', 10000)] checksum BAD (drop)
[RDY ('127.0.0.1', 10000)] -> ACK 0
[RDY ('127.0.0.1', 10000)] DELIVER seq=0 (328)
[RDY ('127.0.0.1', 10000)] -> ACK 1
[RDY ('127.0.0.1', 10000)] DELIVER seq=1 (328)
[RDY ('127.0.0.1', 10000)] -> ACK 2
[RDY ('127.0.0.1', 10000)] DELIVER seq=2 (328)
[RDY ('127.0.0.1', 10000)] -> ACK 3
[RDY ('127.0.0.1', 10000)] DELIVER seq=3 (328)
[RDY ('127.0.0.1', 10000)] RETX seq=0
[RDY ('127.0.0.1', 10000)] -> ACK 0
[RDY ('127.0.0.1', 10000)] DELIVER seq=4 (328)
[RDY ('127.0.0.1', 10000)] -> ACK 1
[RDY ('127.0.0.1', 10000)] -> ACK 5
[RDY ('127.0.0.1', 10000)] DELIVER seq=5 (328)
[RDY ('127.0.0.1', 10000)] -> ACK 2
[RDY ('127.0.0.1', 10000)] ACK for seq=0
[CLIENT] wrote download_test.txt (192 bytes)
PS C:\Users\Geoffrey\Classes\Networks\HW3> []
```

Packets were lost and protocol was able to recover from it.

```
Terminal - HW3
PS C:\Users\Geoffrey\Classes\Networks\HW3> python server.py --host 127.0.0.1 --port 12000
[SERVER] listening on 127.0.0.1:12000
[SERVER] new peer ('127.0.0.1', 10001)
[RDY ('127.0.0.1', 10001)] -> ACK 0
[RDY ('127.0.0.1', 10001)] DELIVER seq=0 (130)
[SERVER] sending test.txt (1928) to ('127.0.0.1', 10001)
[RDY ('127.0.0.1', 10001)] TX seq=0 len=32
[RDY ('127.0.0.1', 10001)] TX seq=1 len=32
[RDY ('127.0.0.1', 10001)] TX seq=2 len=32
[RDY ('127.0.0.1', 10001)] TX seq=3 len=32
[RDY ('127.0.0.1', 10001)] RETX seq=0
[RDY ('127.0.0.1', 10001)] TX seq=4 len=32
[RDY ('127.0.0.1', 10001)] TX seq=5 len=32
[RDY ('127.0.0.1', 10001)] RETX seq=1
[RDY ('127.0.0.1', 10001)] RETX seq=2
[RDY ('127.0.0.1', 10001)] ACK for seq=0
[RDY ('127.0.0.1', 10001)] ACK for seq=3
[RDY ('127.0.0.1', 10001)] -> ACK 0
[RDY ('127.0.0.1', 10001)] ACK for seq=4
[RDY ('127.0.0.1', 10001)] ACK for seq=5
[RDY ('127.0.0.1', 10001)] RETX seq=1
[RDY ('127.0.0.1', 10001)] ACK for seq=1
[RDY ('127.0.0.1', 10001)] RETX seq=2
[RDY ('127.0.0.1', 10001)] ACK for seq=2
[]

PS C:\Users\Geoffrey\Classes\Networks\HW3> python emulator.py --listen-a 127.0.0.1:10000
--listen-b 127.0.0.1:10001 --server 127.0.0.1:12000 --loss 0.20
[EMUL] A('127.0.0.1', 10000) B('127.0.0.1', 10001) server=('127.0.0.1', 12000)
[EMUL] loss=0.20, corrupt=0.00, reorder=0.00, dup=0.00
[EMUL] DROP Server->A len=11
[EMUL] DROP Server->A len=43
[EMUL] DROP Server->A len=43
[EMUL] DROP Server->A len=43
[EMUL] DROP Server->A len=43
[]

PS C:\Users\Geoffrey\Classes\Networks\HW3> python client.py --server 127.0.0.1:12000 --em
ulator 127.0.0.1:10000 GET test.txt
[RDY ('127.0.0.1', 10000)] TX seq=0 len=13
[RDY ('127.0.0.1', 10000)] -> ACK 0
[RDY ('127.0.0.1', 10000)] DELIVER seq=0 (328)
[RDY ('127.0.0.1', 10000)] RETX seq=0
[RDY ('127.0.0.1', 10000)] -> ACK 0
[RDY ('127.0.0.1', 10000)] -> ACK 4
[RDY ('127.0.0.1', 10000)] -> ACK 5
[RDY ('127.0.0.1', 10000)] ACK for seq=0
[RDY ('127.0.0.1', 10000)] -> ACK 1
[RDY ('127.0.0.1', 10000)] DELIVER seq=1 (328)
[RDY ('127.0.0.1', 10000)] -> ACK 2
[RDY ('127.0.0.1', 10000)] DELIVER seq=2 (328)
[RDY ('127.0.0.1', 10000)] DELIVER seq=3 (328)
[RDY ('127.0.0.1', 10000)] DELIVER seq=4 (328)
[RDY ('127.0.0.1', 10000)] DELIVER seq=5 (328)
[CLIENT] wrote download_test.txt (192 bytes)
PS C:\Users\Geoffrey\Classes\Networks\HW3> []
```

Packets were reordered and protocol was able to recover from it.

```
Terminal - HW3
PS C:\Users\Geoffrey\Classes\Networks\HW3> python server.py --host 127.0.0.1 --port 12000
[SERVER] listening on 127.0.0.1:12000
[SERVER] new peer ('127.0.0.1', 10001)
[RD] ('127.0.0.1', 10001) -> ACK 0
[RD] ('127.0.0.1', 10001) DELIVER seq=0 (130)
[SERVER] sending test.txt (1928) to ('127.0.0.1', 10001)
[RD] ('127.0.0.1', 10001) TX seq=0 len=32
[RD] ('127.0.0.1', 10001) TX seq=1 len=32
[RD] ('127.0.0.1', 10001) TX seq=2 len=32
[RD] ('127.0.0.1', 10001) TX seq=3 len=32
[RD] ('127.0.0.1', 10001) RETX seq=0
[RD] ('127.0.0.1', 10001) TX seq=4 len=32
[RD] ('127.0.0.1', 10001) RETX seq=1
[RD] ('127.0.0.1', 10001) TX seq=5 len=32
[RD] ('127.0.0.1', 10001) RETX seq=2
[RD] ('127.0.0.1', 10001) ACK for seq=0
[RD] ('127.0.0.1', 10001) ACK for seq=1
[RD] ('127.0.0.1', 10001) ACK for seq=2
[RD] ('127.0.0.1', 10001) ACK for seq=3
[RD] ('127.0.0.1', 10001) ACK for seq=4
[RD] ('127.0.0.1', 10001) ACK for seq=5
[]

PS C:\Users\Geoffrey\Classes\Networks\HW3> python emulator.py --listen-a 127.0.0.1:10000
--listen-b 127.0.0.1:10001 --server 127.0.0.1:12000 --reorder 0.40
[EMUL] A=('127.0.0.1', 10000), B=('127.0.0.1', 10001), server=('127.0.0.1', 12000)
[EMUL] loss=0.00, corrupt=0.00, reorder=0.40, dup=0.00
[EMUL] REORDER (hold) A->Server
[EMUL] SEND delayed A->Server
[EMUL] REORDER (hold) A->Server
[EMUL] REORDER (hold) A->Server
[EMUL] SEND delayed A->Server
[EMUL] REORDER (hold) A->Server
[EMUL] SEND delayed Server->A
[EMUL] REORDER (hold) Server->A
[EMUL] SEND delayed Server->A
[EMUL] REORDER (hold) A->Server
[EMUL] SEND delayed A->Server
[]

PS C:\Users\Geoffrey\Classes\Networks\HW3> python client.py --server 127.0.0.1:12000 --emulator
127.0.0.1:10000 GET test.txt
[RD] ('127.0.0.1', 10000) TX seq=0 len=13
[RD] ('127.0.0.1', 10000) ACK for seq=0
[RD] ('127.0.0.1', 10000) -> ACK 0
[RD] ('127.0.0.1', 10000) DELIVER seq=0 (328)
[RD] ('127.0.0.1', 10000) -> ACK 1
[RD] ('127.0.0.1', 10000) DELIVER seq=1 (328)
[RD] ('127.0.0.1', 10000) -> ACK 2
[RD] ('127.0.0.1', 10000) DELIVER seq=2 (328)
[RD] ('127.0.0.1', 10000) -> ACK 3
[RD] ('127.0.0.1', 10000) DELIVER seq=3 (328)
[RD] ('127.0.0.1', 10000) -> ACK 4
[RD] ('127.0.0.1', 10000) DELIVER seq=4 (328)
[RD] ('127.0.0.1', 10000) -> ACK 5
[RD] ('127.0.0.1', 10000) DELIVER seq=5 (328)
[RD] ('127.0.0.1', 10000) -> ACK 6
[CLIENT] wrote download_test.txt (192 bytes)
PS C:\Users\Geoffrey\Classes\Networks\HW3> []
```

Protocol successfully wrote a file to a remote server.

```
Terminal - HW3
PS C:\Users\Geoffrey\Classes\Networks\HW3> python server.py --host 127.0.0.1 --port 12000
[SERVER] listening on 127.0.0.1:12000
[SERVER] new peer ('127.0.0.1', 10001)
[RD] ('127.0.0.1', 10001) -> ACK 0
[RD] ('127.0.0.1', 10001) DELIVER seq=0 (130)
[SERVER] expecting upload -> test.txt from ('127.0.0.1', 10001)
[RD] ('127.0.0.1', 10001) -> ACK 1
[RD] ('127.0.0.1', 10001) DELIVER seq=1 (328)
[RD] ('127.0.0.1', 10001) -> ACK 2
[RD] ('127.0.0.1', 10001) DELIVER seq=2 (328)
[RD] ('127.0.0.1', 10001) -> ACK 3
[RD] ('127.0.0.1', 10001) DELIVER seq=3 (328)
[RD] ('127.0.0.1', 10001) -> ACK 4
[RD] ('127.0.0.1', 10001) DELIVER seq=4 (328)
[RD] ('127.0.0.1', 10001) -> ACK 5
[RD] ('127.0.0.1', 10001) DELIVER seq=5 (328)
[RD] ('127.0.0.1', 10001) -> ACK 6
[RD] ('127.0.0.1', 10001) DELIVER seq=6 (328)
[SERVER] stored test.txt (1928) from ('127.0.0.1', 10001)
[RD] ('127.0.0.1', 10001) TX seq=0 len=31
[RD] ('127.0.0.1', 10001) ACK for seq=0
[]

PS C:\Users\Geoffrey\Classes\Networks\HW3> python emulator.py --listen-a 127.0.0.1:10000 --listen-b 127.0.0.1:10001
[EMUL] A=('127.0.0.1', 10000), B=('127.0.0.1', 10001), server=('127.0.0.1', 12000)
[EMUL] loss=0.00, corrupt=0.00, reorder=0.00, dup=0.00
[]

PS C:\Users\Geoffrey\Classes\Networks\HW3> python client.py --server 127.0.0.1:12000 --emulator
127.0.0.1:10000 PUT test.txt
[RD] ('127.0.0.1', 10000) TX seq=0 len=13
[RD] ('127.0.0.1', 10000) ACK for seq=0
[RD] ('127.0.0.1', 10000) TX seq=1 len=32
[RD] ('127.0.0.1', 10000) ACK for seq=1
[RD] ('127.0.0.1', 10000) TX seq=2 len=32
[RD] ('127.0.0.1', 10000) ACK for seq=2
[RD] ('127.0.0.1', 10000) TX seq=3 len=32
[RD] ('127.0.0.1', 10000) ACK for seq=3
[RD] ('127.0.0.1', 10000) TX seq=4 len=32
[RD] ('127.0.0.1', 10000) ACK for seq=4
[RD] ('127.0.0.1', 10000) TX seq=5 len=32
[RD] ('127.0.0.1', 10000) ACK for seq=5
[RD] ('127.0.0.1', 10000) TX seq=6 len=32
[RD] ('127.0.0.1', 10000) ACK for seq=6
[RD] ('127.0.0.1', 10000) -> ACK 0
[RD] ('127.0.0.1', 10000) DELIVER seq=0 (318)
OK: Stored test.txt (192 bytes)
PS C:\Users\Geoffrey\Classes\Networks\HW3> []
```

```
test.txt
Hello! This is a test file for Reliable Data Transfer.
The file will be transferred over UDP with loss and corruption.
The protocol should recover using retransmission and acknowledgments.
```

Protocol successfully read a file from a remote server.

```
Terminal Local x

PS C:\Users\Geoffrey\Classes\Networks\WM3> python server.py --host 127.0.0.1 --port 12000
[SERVER] listening on 127.0.0.1:12000
[SERVER] new peer ('127.0.0.1', 10001)
[RDPT ('127.0.0.1', 10001)] -> ACK 0
[RDPT ('127.0.0.1', 10001)] DELIVER seq=0 (138)
[SERVER] sending test.txt (1928) to ('127.0.0.1', 10001)
[RDPT ('127.0.0.1', 10001)] TX seq=0 len=32
[RDPT ('127.0.0.1', 10001)] TX seq=1 len=32
[RDPT ('127.0.0.1', 10001)] TX seq=2 len=32
[RDPT ('127.0.0.1', 10001)] TX seq=3 len=32
[RDPT ('127.0.0.1', 10001)] RETX seq=0
[RDPT ('127.0.0.1', 10001)] TX seq=4 len=32
[RDPT ('127.0.0.1', 10001)] RETX seq=1
[RDPT ('127.0.0.1', 10001)] TX seq=5 len=32
[RDPT ('127.0.0.1', 10001)] RETX seq=2
[RDPT ('127.0.0.1', 10001)] ACK for seq=0
[RDPT ('127.0.0.1', 10001)] ACK for seq=1
[RDPT ('127.0.0.1', 10001)] ACK for seq=2
[RDPT ('127.0.0.1', 10001)] ACK for seq=3
[RDPT ('127.0.0.1', 10001)] ACK for seq=4
[RDPT ('127.0.0.1', 10001)] ACK for seq=5

PS C:\Users\Geoffrey\Classes\Networks\WM3> python emulator.py --listen-s 127.0.0.1:10000
--listen-r 127.0.0.1:10001 --server 127.0.0.1:12000
[EMUL] A=('127.0.0.1', 10000), B=('127.0.0.1', 10001), server=('127.0.0.1', 12000)
[EMUL] loss=0.00, corrupt=0.00, reorder=0.00, dup=0.00
[]

PS C:\Users\Geoffrey\Classes\Networks\WM3> python client.py --server 127.0.0.1:12000 --em
ulator 127.0.0.1:10000 GET test.txt
[RDPT ('127.0.0.1', 10000)] TX seq=0 len=13
[RDPT ('127.0.0.1', 10000)] ACK for seq=0
[RDPT ('127.0.0.1', 10000)] -> ACK 0
[RDPT ('127.0.0.1', 10000)] DELIVER seq=0 (328)
[RDPT ('127.0.0.1', 10000)] -> ACK 1
[RDPT ('127.0.0.1', 10000)] DELIVER seq=1 (328)
[RDPT ('127.0.0.1', 10000)] -> ACK 2
[RDPT ('127.0.0.1', 10000)] DELIVER seq=2 (328)
[RDPT ('127.0.0.1', 10000)] -> ACK 3
[RDPT ('127.0.0.1', 10000)] DELIVER seq=3 (328)
[RDPT ('127.0.0.1', 10000)] -> ACK 4
[RDPT ('127.0.0.1', 10000)] -> ACK 5
[RDPT ('127.0.0.1', 10000)] DELIVER seq=4 (328)
[RDPT ('127.0.0.1', 10000)] -> ACK 6
[RDPT ('127.0.0.1', 10000)] DELIVER seq=5 (328)
[RDPT ('127.0.0.1', 10000)] -> ACK 7
[CLIENT] wrote download_test.txt (192 bytes)
PS C:\Users\Geoffrey\Classes\Networks\WM3> []
```