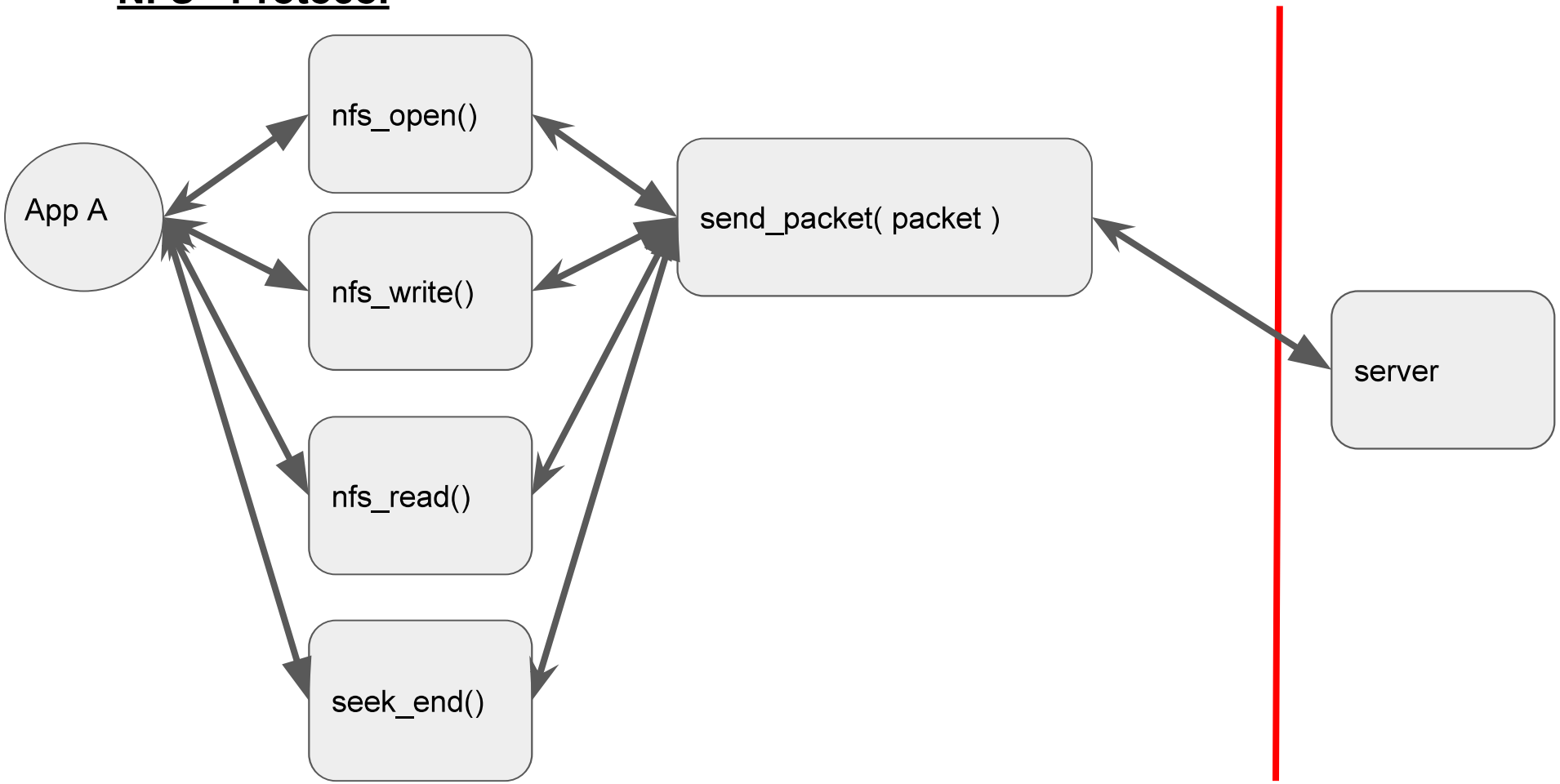
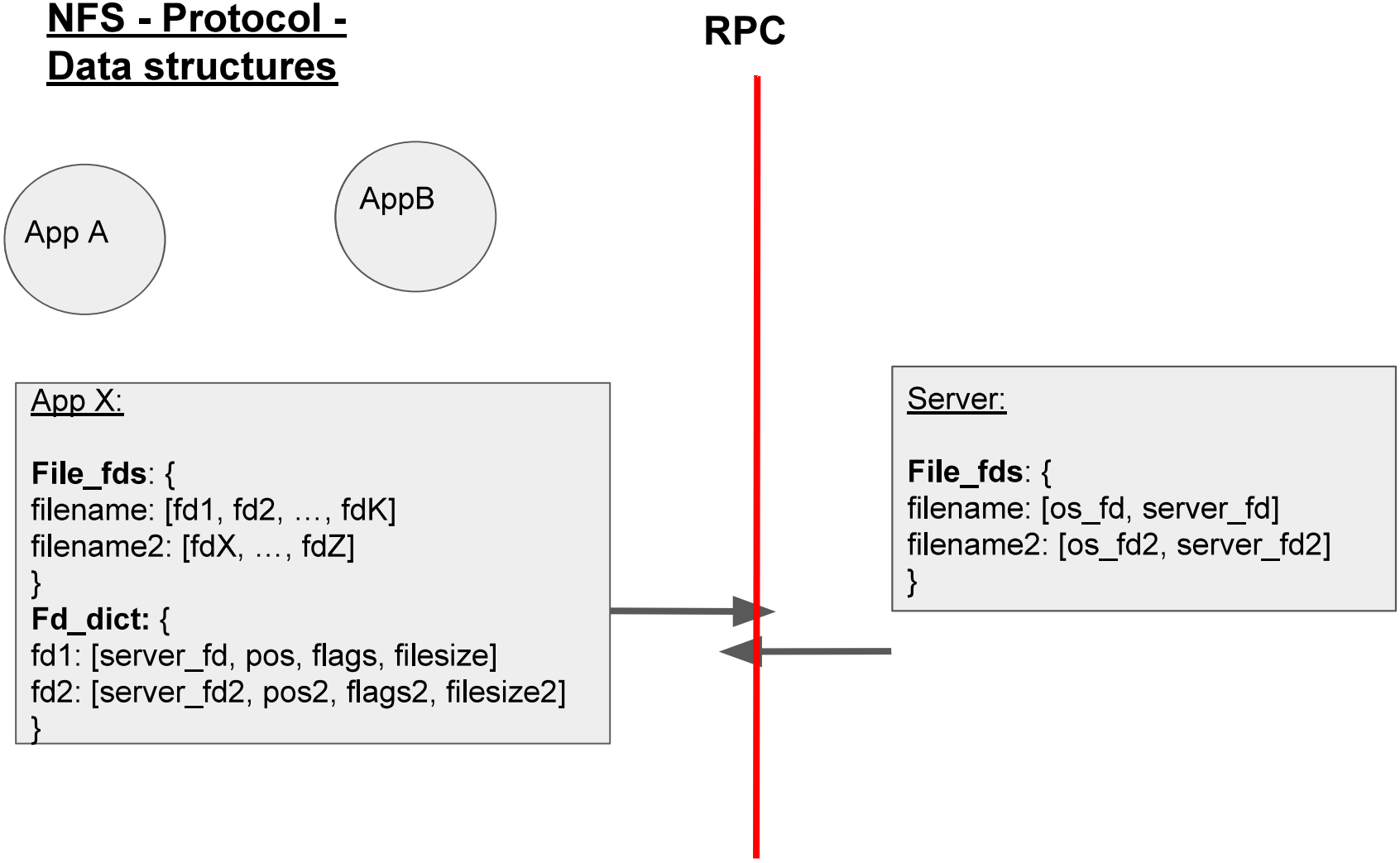
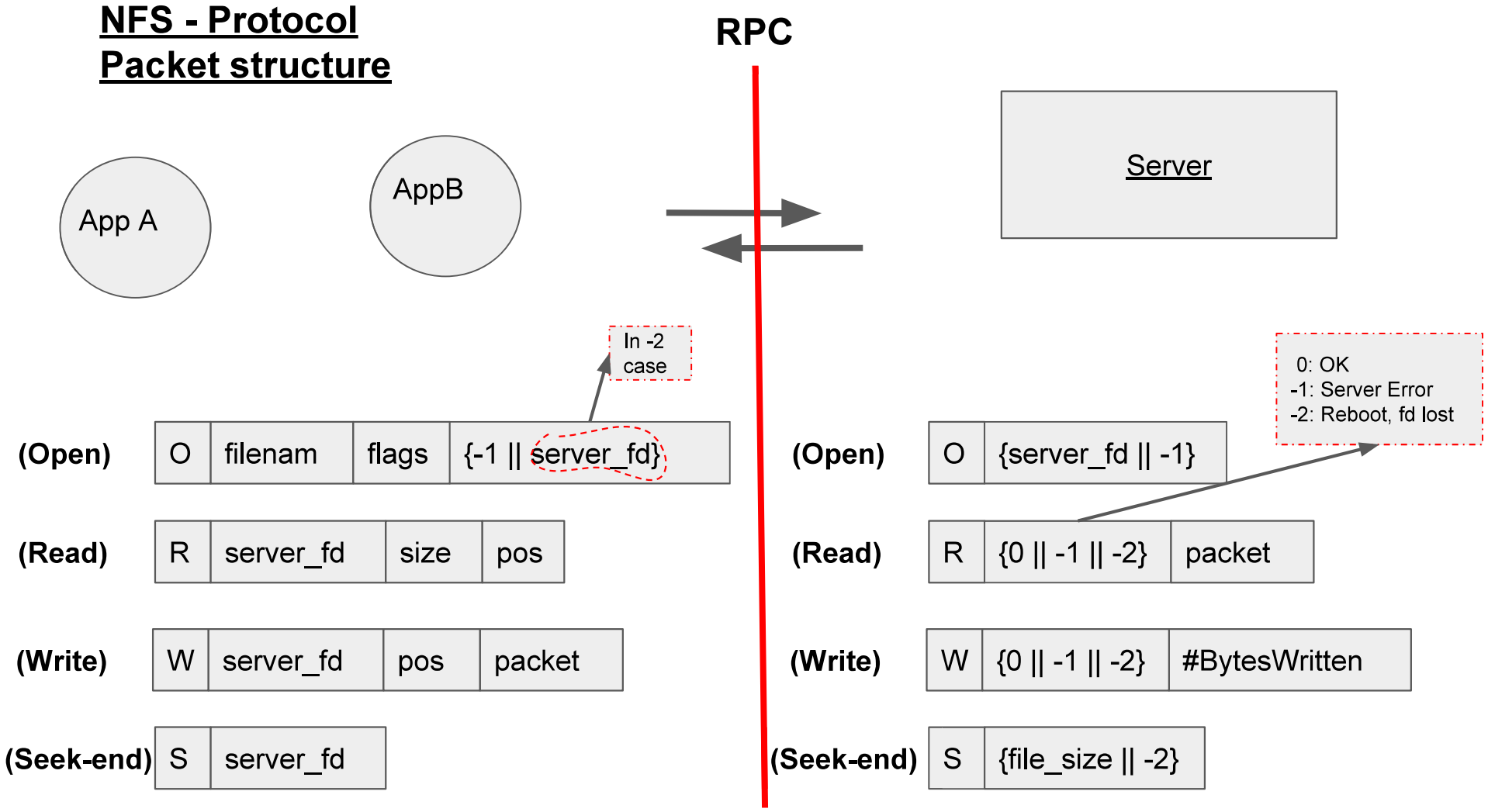
NFS - Protocol

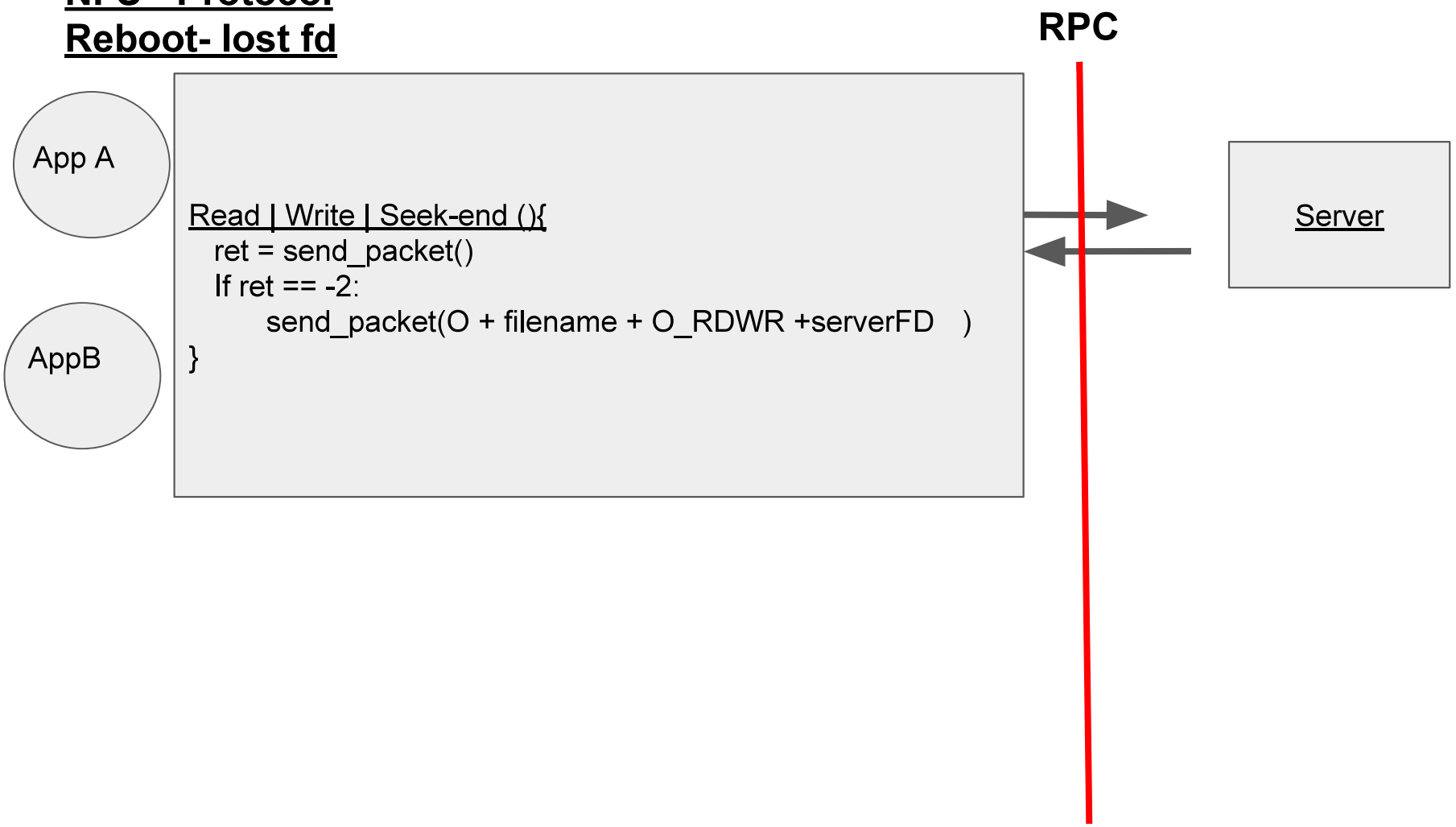


NFS - Protocol - Data structures

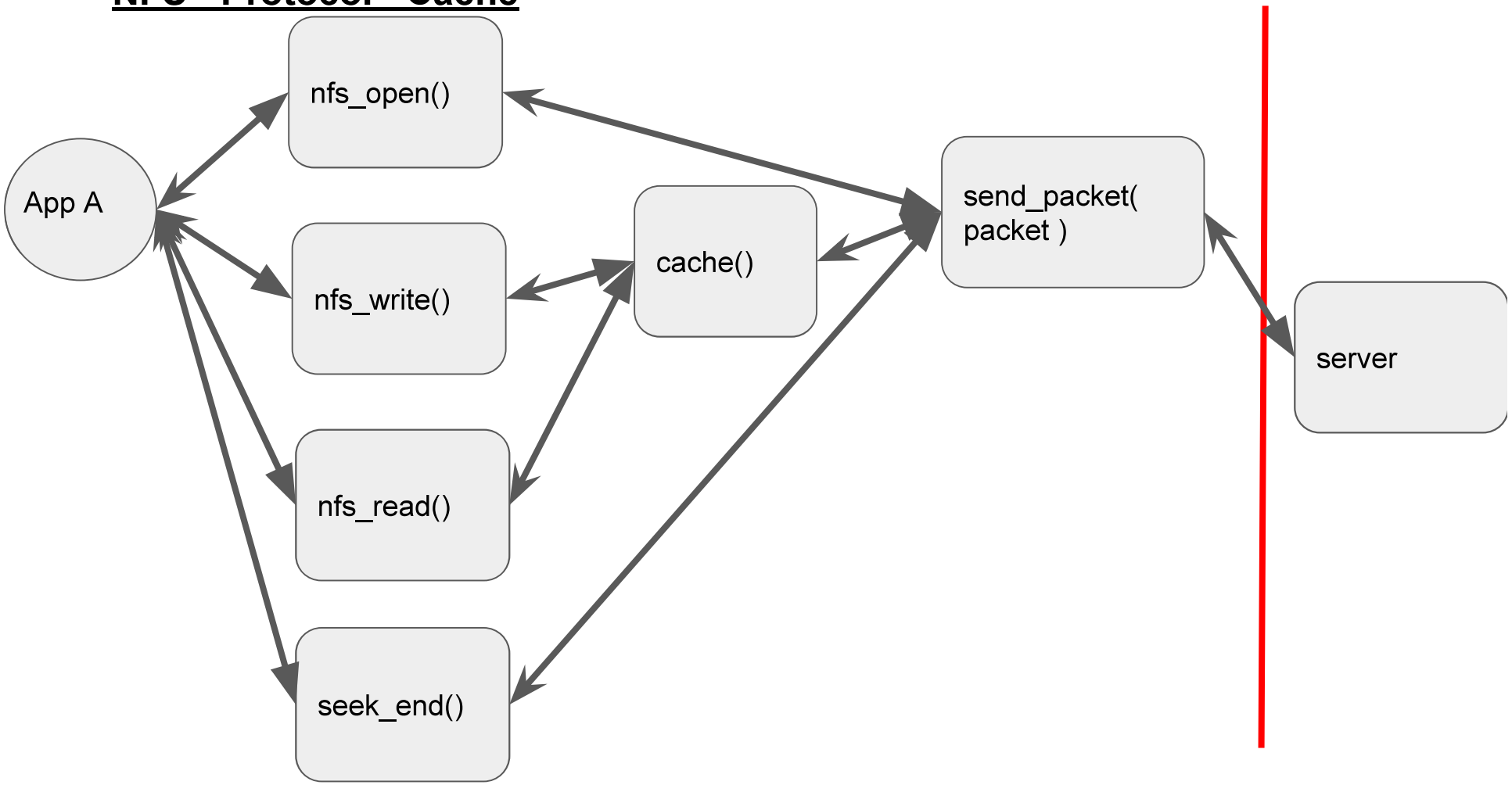



NFS - Protocol
Packet structure
RPC
App A
AppB
Server
In -2 case
0: OK
-1: Server Error
-2: Reboot, fd lost
(Open)
O
filenam
flags
{-1 || server_fd}
(Open)
O
{server_fd || -1}
(Read)
R
server_fd
size
pos
(Read)
R
{0 || -1 || -2}
packet
(Write)
W
server_fd
pos
packet
(Write)
W
{0 || -1 || -2}
#BytesWritten
(Seek-end)
S
server_fd
(Seek-end)
S
{file_size || -2}

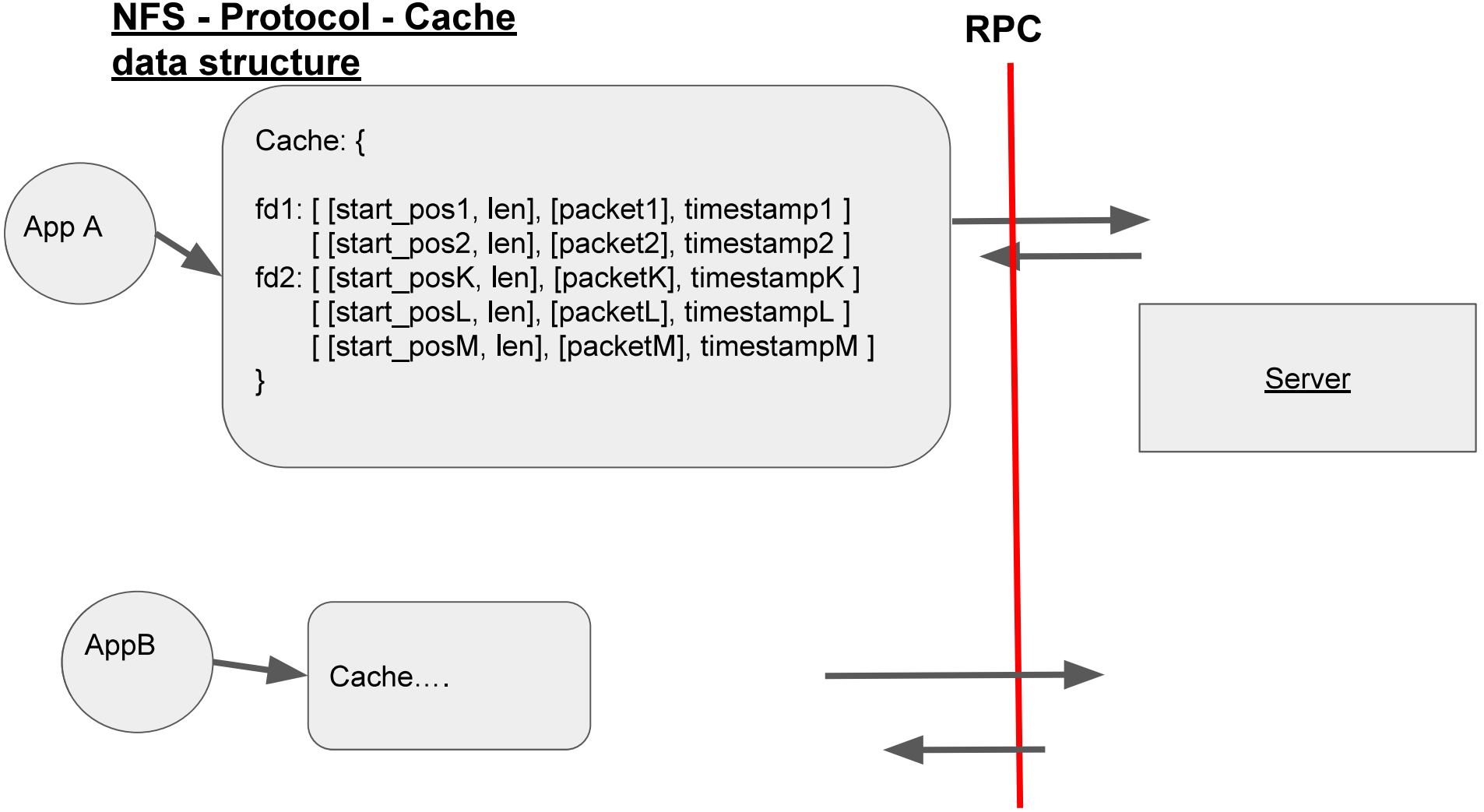
NFS - Protocol
Reboot- lost fd



NFS - Protocol - Cache

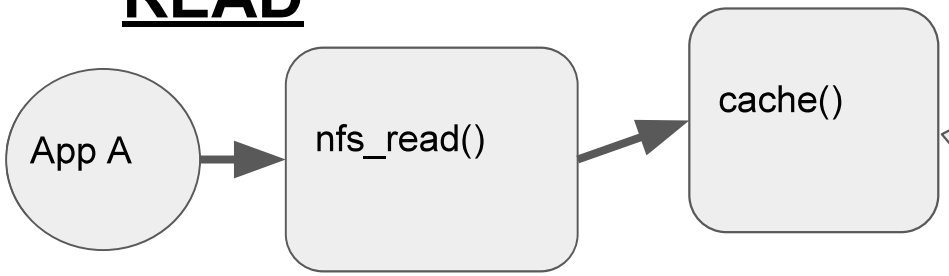


NFS - Protocol - Cache data structure



NFS - Implementation -

READ

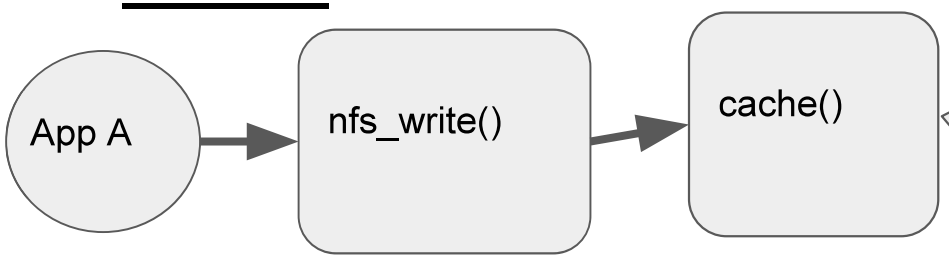



```
blocks = file_size / cache_block_size
while file_size > 0:
  If server_fd in cache:
    for each line:
      If file_pos in line:
        buffer.append(line)
      else:
         Find block_start
         ret= send_packet('R',server_fd,
block_size, bloc_start)
          ret=data
          Ret = read_cache(data)
          If ret == 1: #end offile
            buf.append(reast_packet)
   Else:
      Find block_start
         ret= send_packet('R',server_fd,
block_size, bloc_start)
          ret=data
          Ret = read_cache(data)
```

read_cache()

```
total_blocks=0
for x in cache_dic.keys():
   # Find total_blocks
if total_blocks > configuration.cacheSize:
   minLRU = time.time()
   Find least recem used block
   Delete block
if total_blocks <= configuration.cacheSize
  if server_fd in cache_dic::
    cache_dic[server_fd].insert(sortedpos, data)
else:   #an den exw fd in cache
  cache_dic[server_fd]=[[[file_pos, len(data)],[data],
time.time()]]
```

NFS - Implementation -
WRITE




```
 ret= send_packet()
If ret >0:
  blocks = file_size / cache_block_size
  while temp_len > 0:
  If server_fd in cache:
    if len(cache_dic[server_fd]) == 0:
        write_cache()
    for each line:
        If file_pos in line:
          Find block start
          Create block #me bash block_start kai
file_pos
          write_cache(server_fd, temp_block,
block_start, configuration.blockSize)
        Else:#an den uparxei o filepos
          Find block start
          cache_dic[server_fd].insert(sorted_pos,
tempblock)
  Else:#an den uparxei o fd
    write_cache(server_fd)
```

Write_cache()

```
total_blocks=0
for x in cache_dic.keys():
   # Find total_blocks
if total_blocks > configuration.cacheSize:
   minLRU = time.time()
   Find least recem used block
   Delete block
if total_blocks <= configuration.cacheSize
if server_fd in cache_dic::
  for x in cache_dic[server_fd]:
   If file_pos:
    #ananewsh thesewn sthn cache_dic
else:   #an den exw fd in cache
  cache_dic[server_fd]=[]
```