

计算机网络实验2 GBN sender实验

PB21081601 张芷苒

实验背景与目标

在这个实验中，需要将一个字符串中的每个字符封装为分组发送给接收方，并且遵循GBN协议：开始时发送滑动窗口内所有可以发送的分组。每发送一个分组，保存该分组在缓冲区中，表示已发送但还未被确认。缓冲区使用一个定时器，当定时器超时的时候，重新发送缓冲区中的所有分组。当收到接收方的确认后，判断该确认是否有效。如果无效的话，什么也不做；如果有效的话，采取累计确认，移动滑动窗口，将已经被确认的分组从缓冲区中删除，并且发送接下来可以发送的分组，重置定时器。

实验过程

实验要求完成 `run`, `put`, `timeout_callback` 三个函数的todo部分。

1. `run` 函数

功能： `run` 函数负责发送滑动窗口内所有可以发送的分组，并在发送后等待ACK或超时。

满足实验要求：

- **发送滑动窗口内的分组：** `run` 函数在一个循环中，只要缓冲区大小小于滑动窗口大小且还有字符未发送，就会发送新的分组。
- **保存分组到缓冲区：** 每发送一个分组，它就会被添加到 `self.outbound` 缓冲区中，表示已发送但未被确认。
- **等待ACK或超时：** `yield self.finish_channel.get()` 语句使得函数等待ACK的到来或超时。如果超时发生，`timeout_callback` 将被触发。

```
def run(self, env: Environment):  
    """  
    run函数：负责发送滑动窗口内所有可以发送的分组，并在发送后等待ACK或  
    超时。  
    """  
    while len(self.outbound) < self.window_size and self.absno  
< len(self.message):  
        packet = self.new_packet(self.seqno,  
self.message[self.absno])
```

```
        self.send_packet(packet)
        self.outbound.append(packet)
        self.seqno = (self.seqno + 1) % self.seqno_range
        self.absno += 1
    yield self.finish_channel.get()
```

2. put 函数

功能：put 函数处理从接收端收到的ACK，并进行相应的动作。

满足实验要求：

- **判断确认是否有效**：函数首先检查收到的ACK是否在 self.outbound 缓冲区中，这决定了ACK是否有效。
- **累计确认和移动滑动窗口**：如果ACK有效，函数则移除所有小于或等于该ACK号的分组，实现累计确认。同时， self.seqno_start 更新为最新确认的分组之后的值。
- **发送接下来可以发送的分组**：在累计确认之后，如果还有空间，在滑动窗口内继续发送新的分组。
- **重置定时器**：每次收到有效的ACK后，定时器都会被重置，以等待下一个超时周期。

```
def put(self, packet: Packet):
    """
    put函数：处理从接收端收到的ACK。
    """
    ackno = packet.packet_id
    self.dprint(f"receive ack {ackno}")
    if ackno in [pkt.packet_id for pkt in self.outbound]:
        while self.seqno_start != ackno:
            self.outbound.popleft()
            self.seqno_start = (self.seqno_start + 1) %
self.seqno_range
        self.outbound.popleft()
        self.seqno_start = (self.seqno_start + 1) %
self.seqno_range
        while len(self.outbound) < self.window_size and
self.absno < len(self.message):
            packet = self.new_packet(self.seqno,
self.message[self.absno])
            self.send_packet(packet)
            self.outbound.append(packet)
```

```
        self.seqno = (self.seqno + 1) % self.seqno_range
        self.absno += 1
        self.timer.restart(self.timeout)
        if self.absno ≥ len(self.message) and
len(self.outbound) == 0:
            self.finish_channel.put(True)
```

3. timeout_callback 函数

功能：timeout_callback 函数在定时器超时时被调用。

满足实验要求：

- **重新发送缓冲区中的所有分组**：当超时发生时，该函数将遍历 self.outbound 缓冲区中的所有分组，并重新发送它们。这符合GBN协议中超时后的行为要求。

```
def timeout_callback(self):
    """
    timeout_callback函数：在超时时被调用，重新发送缓冲区中的所有分组。
    """
    self.dprint("timeout")
    for packet in self.outbound:
        self.send_packet(packet)
```

实验结果

经测试，在本地和 openetlab 上，都能通过所有的样例，因此可以认为实验成功。

```
minerva@minerva-VirtualBox: ~/桌面/gbn/code
[sender](time: 8590.00) -> timeout
[sender](time: 8590.00) -> send  on seqno 3
[sender](time: 8590.00) -> send n on seqno 4
[sender](time: 8590.00) -> send u on seqno 5
[sender](time: 8590.00) -> send n on seqno 6
[sender](time: 8590.00) -> send c on seqno 7
[sender](time: 8590.00) -> send , on seqno 8
[sender](time: 8590.00) -> send
on seqno 9
[receiver](time: 8600.00) -> send ack 4
[receiver](time: 8600.00) -> send ack 5
[receiver](time: 8600.00) -> send ack 6
[receiver](time: 8600.00) -> send ack 7
[receiver](time: 8600.00) -> send ack 8
[receiver](time: 8600.00) -> send ack 9
[receiver](time: 8600.00) -> send ack 10
[sender](time: 8610.00) -> receive ack 3
[sender](time: 8610.00) -> receive ack 4
[sender](time: 8610.00) -> receive ack 5
[sender](time: 8610.00) -> receive ack 6
[sender](time: 8610.00) -> receive ack 7
[sender](time: 8610.00) -> receive ack 8
[sender](time: 8610.00) -> receive ack 9
minerva@minerva-VirtualBox:~/桌面/gbn/code$
```

2023-12-28 18:05:59

2e61b89bd110

ALL PASSED

gbn sender

100

Python3

张芷苒-pb21081601

反思与收获

1. **理解GBN协议的工作原理**：通过实际编写代码，我深入理解了GBN协议的工作机制，包括滑动窗口、序列号的管理、累计确认以及超时重传等核心概念。
2. **编程技能提升**：实验过程中，我不仅加深了对Python编程的理解，还学习到了如何在实际应用中处理网络通信问题，特别是在可靠数据传输方面。
3. **问题解决能力增强**：在处理无限循环、缓冲区管理和超时机制等问题时，我学会了如何逐步调试和解决复杂问题，这对提高我的问题解决能力非常有帮助。
4. **实践与理论的结合**：通过将理论知识应用于实践，我更加深刻地理解了这些概念，这将有助于我在未来的学习和工作中更好地运用这些知识。