

Technologie Sieciowe Sprawozdanie 4

Michał Fajarewicz

Do programu Z2Receiver zostały wprowadzone modyfikacje pozwalające wypisywać otrzymane dane w kolejności ich identyfikatorów. Otrzymane pakiety zapisywane są do tablicy pakietów, i wypisywane dopiero gdy wszystkie pakiety o niższym identyfikatorze zostały wypisane.

```
for (int i=0; i<packetsReceived.size(); i++){
    if(packetsReceived.get(i).getIntAt( idx: 0) == counter){
        Z2Packet p = packetsReceived.get(i);
        System.out.println("R:" + p.getIntAt( idx: 0) + ": " + (char) p.data[4]);
        stringBuilder.append((char) p.data[4]);
        System.out.println(stringBuilder.toString());
        counter++;
        packetsReceived.remove(packetsReceived.get(i));
        i=0;
    }
}
```

Program Z2Sender został zmodyfikowany tak aby powtarzać wysyłanie niepotwierdzonych pakietów. Pakiety do wysłania zapisywane są w tablicy i wysyłane, po czym program czeka 5 sekund w czasie których oczekuje na pakiety potwierdzające i usuwa odpowiadające im pakiety z tablicy. Po upływie zadanego czasu program ponawia transmisję pozostałych elementów z tablicy. Gdy wszystkie pakiety zostaną potwierdzone program wypisuje komunikat „transmission confirmed”.

```
synchronized (datagramPackets) {
    while (!datagramPackets.isEmpty()) {
        for (DatagramPacket datagramPacket : datagramPackets) {
            try {
                socket.send(datagramPacket);
                sleep(sleepTime);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        try {
            datagramPackets.wait( timeout 5000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    System.out.println("transmission confirmed");
}
```

```
synchronized (datagramPackets) {
    Iterator<DatagramPacket> datagramIterator = datagramPackets.iterator();
    while (datagramIterator.hasNext()) {
        DatagramPacket datagramPacket = datagramIterator.next();
        pSent = new Z2Packet(datagramPacket.getData());
        if (pSent.getIntAt( idx: 0) == p.getIntAt( idx: 0)) {
            datagramIterator.remove();
        }
    }
}
```

Program Z2Forwarder nie został zmieniony.

Wywołanie programu dla danych „ALA MA KOTA AKOT MA ALE”

```
R:0: A ALA MA KOTA A K
A R:15: O
R:1: L ALA MA KOTA A KO
AL R:16: T
R:2: A ALA MA KOTA A KOT
ALA R:17:
S:12: A ALA MA KOTA A KOT
S:10: A R:18: M
S:0: A ALA MA KOTA A KOT M
S:1: L R:19: A
S:19: A ALA MA KOTA A KOT MA
S:0: A R:20:
S:0: A ALA MA KOTA A KOT MA
S:8: O R:21: A
S:8: O ALA MA KOTA A KOT MA A
R:3: R:22: L
ALA ALA MA KOTA A KOT MA AL
R:4: M R:23: E
ALA M ALA MA KOTA A KOT MA ALE
R:5: A S:17:
ALA MA S:6:
S:16: T S:4: M
S:5: A S:2: A
S:19: A S:7: K
S:21: A transmission confirmed
S:18: M S:6:
S:9: T S:7: K
S:18: M S:2: A
S:19: A S:7: K
S:14: K
S:3:
S:23: E
S:11:
S:20:
S:13:
S:22: L
S:9: T
S:16: T
S:18: M
S:20:
S:15: O
S:18: M
S:18: M
S:14: K
R:6:
ALA MA
R:7: K
ALA MA K
R:8: O
ALA MA KO
R:9: T
ALA MA KOT
S:23: E
R:10: A
ALA MA KOTA
R:11:
ALA MA KOTA
S:4: M
R:12: A
ALA MA KOTA A
R:13:
ALA MA KOTA A
R:14: K
ALA MA KOTA A K
R:15: O
```

Program Z2Receiver w trakcie działania wypisuje kompletną pierwszą część otrzymanego zdania.

Otrzymane dane po komunikacji transmisji to wysłane pakiety potwierdzające które dotarły z opóźnieniem.