

0.1. aggregate packets

Algorithm 1 aggregate packet by src and dst

input: *packets*

output: *groups*

```

1: function AGGREGATEPACKET(packets)
2:   groups  $\leftarrow$  dict()
3:   for each packet  $\in$  packets do
4:     if src  $\geq$  dst then
5:       key  $\leftarrow$  src + sport + dst + dport
6:     else
7:       key  $\leftarrow$  dst + dport + src + sport
8:     end if
9:     if key  $\in$  groups then
10:      groups[key].append(packet)
11:    else
12:      groups[key] = list()
13:      groups[key].append(packet)
14:    end if
15:  end for
16:  for each key  $\in$  groups do
17:    groups[key].sort()
18:  end for
19:  return groups
20: end function

```

0.2. split session

$$\sqrt{x} + \sqrt{x^2 + \sqrt{y}} = \sqrt[3]{k_i} - \frac{x}{m}$$

$$\lim_{x \rightarrow \infty} x_{22}^2 - \int_1^5 x dx + \sum_{n=1}^{20} n^2 = \prod_{j=1}^3 y_j + \lim_{x \rightarrow -2} \frac{x-2}{x}$$

$$y = \sum_{i=1}^n w_i \cdot x_i + \text{bias} \tag{5.1.1}$$

Algorithm 2 split group packets by syn and fin

input: *pktGroup***output:** *sessions*

```
1: function SPLITSESSION(pktGroup)
2:   sessions  $\leftarrow$  list()
3:   currentState  $\leftarrow$  ESTABLISHED
4:   syncnt  $\leftarrow$  0
5:   fincnt  $\leftarrow$  0
6:   stream  $\leftarrow$  list()
7:   for each packet  $\in$  pktGroup do
8:     if SYN & flags then
9:       if syncnt == 0 then
10:        currentState = HARF ESTABLISHED
11:        syncnt += 1
12:        sessions.append(newStream(stream))
13:        stream.clear()
14:      else if syncnt == 1 then
15:        currentState = ESTABLISHED
16:        syncnt += 1
17:      else
18:        currentState = ESTABLISHED
19:      end if
20:    end if
21:    if FIN & flags then
22:      if fincnt == 0 then
23:        currentState = HARF CLOSED
24:        fincnt += 1
25:      else if fincnt == 1 then
26:        currentState = CLOSED
27:        fincnt += 1
28:      else
29:        fincnt += 1
30:      end if
31:    end if
32:  end for
33:  return sessions
34: end function
```
