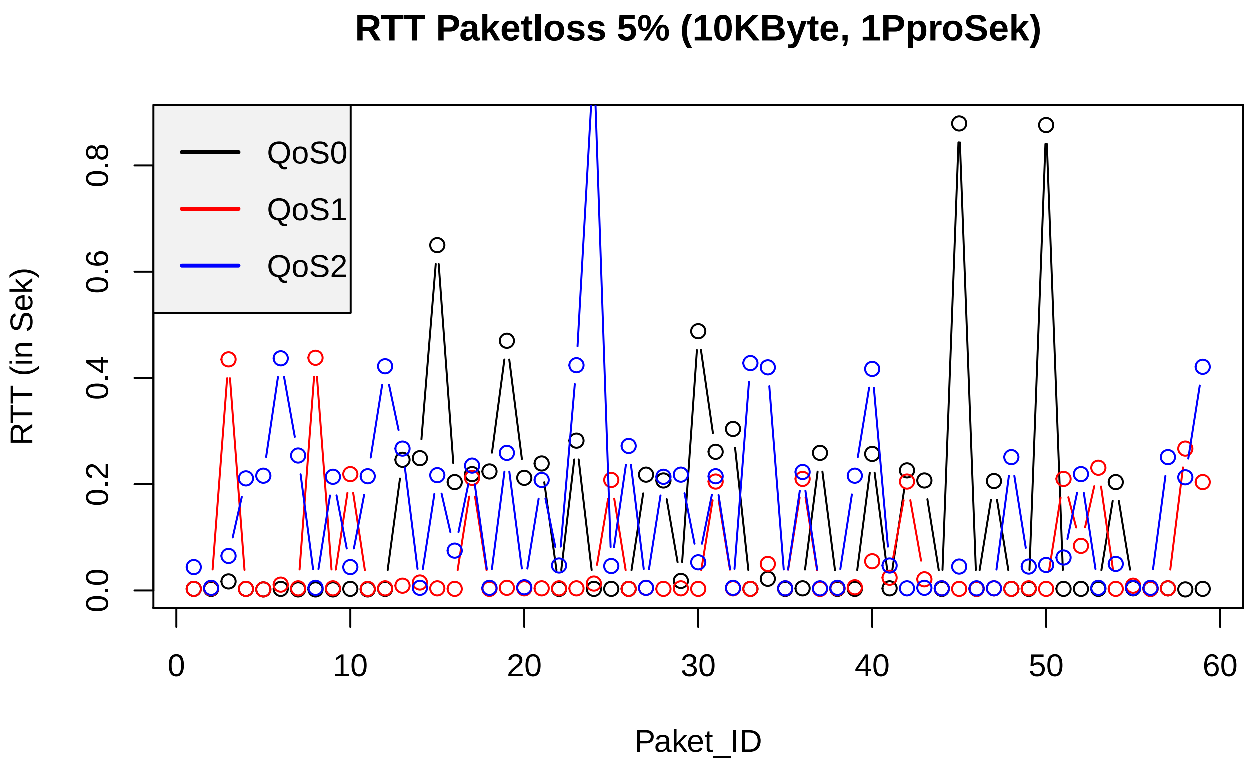
Anhang 1

Einlesen der Daten und RTT- Berechnung

Exemplarisch an einem 5% Paketloss file

```
############
# 5%-ploss #
############

setwd("/home/lisa/Darmstadt/05_Speicher und Datennetze IoT/Praktikum/Git/mqtt-qos-rountrip/logs/5%-ploss
options(digits.secs=3) # needs to be set from time to time - otherwise R doesn't allow for ms
library("data.table", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("h2o", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")
library("tidyr", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.4")

#Create the list of log files in the folder
files <- list.files(pattern = "*client1.log", full.names = TRUE, recursive = FALSE)
names <- substr(files, start = 18, stop = 60)

# Read the logs into dataFrames and bind
# df <- rbindlist(lapply(files, fread))

######################
# Create dataFrames #
######################
# Take Date + Time for adequate TS and formate to POSIXct

for (i in 1:length(files)) {
  Timestamp<-c(as.POSIXct("2018-05-18 14:01:41.264 CEST"))
  newID<-c()
  #x <- get(files[i])
  x<-rbindlist(lapply(files[i], fread))
  colnames(x)<- c("Date", "Time", "Action", "Topic", "QoS", "Size", "ID")

  for (j in 1:nrow(x)) {
    Timestamp[j]<-as.POSIXct(strptime(gsub(":", ".", paste(x[j,1],x[j,2])),"%Y-%m-%d %H.%M.%OS"))
    newID[j]<-paste(x[j,4], x[j,7])
  }
  x<-cbind(x, Timestamp, newID)
  assign(paste(names[i]),x)
  remove(newID, Timestamp, x)
}

#########################
# Create DF to hold RTT #
#########################
# name Vector
namesSent<-c()
namesRec<-c()
namesTime<-c()

# Split each set into sent and receive to substract in next step (each stored separately)
# Create name Vectors for Sent, Receive and Time to access in next step
for (i in 1:length(names)){
  sentTimes <- subset(get(paste(names[i])), Action=="sent")
  recTimes <- subset(get(paste(names[i])), Action=="received")
  assign(paste("sentTimes", names[i]), sentTimes)
  namesSent[i]<-paste("sentTimes", names[i])
  assign(paste("recTimes", names[i]), recTimes)
  namesRec[i]<-paste("recTimes", names[i])

  times<-as.data.frame(matrix(nrow=2000, ncol=6)) # Create times Matces to store RTT in next step
  colnames(times)<- c("sent", "s_newid", "rec", "r_newid", "rtt", "id")
  times[,1] <-as.POSIXct(strptime(times[, "sent"],"%Y-%m-%d %H.%M.%OS"))
  times[,3] <-as.POSIXct(strptime(times[, "rec"],"%Y-%m-%d %H.%M.%OS"))
  assign(paste("times", names[i]), times)
  namesTime[i]<-paste("times", names[i]) # Store Names of Time Matrices to access with get command
}

#################
# Calculate RTT #
#################

# Fill times Data Frames with Sent TS and IDs
for(i in 1 : length(namesSent)){
  sentTimes<- get(paste(namesSent[i]))
  times<- get(paste(namesTime[i]))

  for (j in sentTimes$ID) {
    times[j, "sent"]<- sentTimes[which(sentTimes$ID == j),"Timestamp"]
    times[j, "id"]<- sentTimes[which(sentTimes$ID == j),"ID"]
    times[j, "s_newid"]<- sentTimes[which(sentTimes$ID == j),"newID"]
  }
  assign(paste("times", names[i]), times)
  #assign(times, paste("times", names[i]))
}

# Fill times Data Frames with Recieved TS and IDs
for(i in 1 : length(namesRec)){
  recTimes<- get(paste(namesRec[i]))
  times<- get(paste(namesTime[i]))

  for (j in recTimes$ID) {
    times[j, "rec"]<- recTimes[which(recTimes$ID == j),"Timestamp"]
    times[j, "id"]<- recTimes[which(recTimes$ID == j),"ID"]
    times[j, "r_newid"]<- recTimes[which(recTimes$ID == j),"newID"]
  }
  assign(paste("times", names[i]), times)
  #assign(times, paste("times", names[i]))
}


# Calculate Difference
for (i in 1:length(namesTime)){
  times<- get(paste(namesTime[i]))

  for (j in 1:nrow(times)) {
    times[j,"rtt"]<- difftime(times[j,3], times[j,1])
  }

  times <- na.omit(times)
  assign(paste("times", names[i]), times)
}


#####################
# Merge Data Frames #
#####################
latenzPL5proz <- merge(get(namesTime[1]), get(namesTime[2]))

for (i in 1:length(namesTime)){
latenzPL5proz <- rbind(latenzPL5proz, get(namesTime[i]))
}

####################
# Split Topic Name #
####################
#separate(latenzPL5proz$s_newid)

latenzPL5prozSep <- latenzPL5proz %>% separate(s_newid, c("n1", "n2", "QoS", "Size", "Min", "n3", "Spee
z <- c(-2, -3, -7, -9)
latenzPL5proz <- latenzPL5prozSep[,z]


################
# Plot Results #
################

rttQoS0<-get(namesTime[1])
rttQoS1<-get(namesTime[2])
rttQoS2<-get(namesTime[3])

par(mfrow = c(1, 1))
plot(rttQoS0$id, rttQoS0$rtt, main = "RTT Paketloss 5% (10KByte, 1PproSek)",
     ylab = "RTT (in Sek)", xlab = "Paket_ID", type = "b")
points(rttQoS1$id, rttQoS1$rtt, col = "red", type = "b")
points(rttQoS2$id, rttQoS2$rtt, col = "blue", type = "b")

legend("topleft", c("QoS0", "QoS1", "QoS2"), text.width = 4,
       col = c("black", "red", "blue"),
       text.col = "black", cex = 1  ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")
```


RTT Paketloss 5% (10KByte, 1PproSek)
QoS0
QoS1
QoS2
RTT (in Sek)
0.0
0.2
0.4
0.6
0.8
0
10
20
30
40
50
60
Paket_ID