Praktikum
Speicher-/ und Datennetze im IoT

QoS Eigenschaften von MQTT

Sommersemester 2018

Name: Schuhmacher, Stolz

Latenzzeit

Untersucht wird anhand geloggter TS die Latenzzeit der verschiedenen QoS-Modi.\ **Variiert** werden die Größenordnungen der Testdaten für Requests (jeweils 1, 10 und 100 Byte, KB und MB). \ Die im folgenden gelisteten Payloads, werden ggf. im Laufe der Experimente noch angepasst. Ebenfalls im Laufe der Experimente zu kennzeichnen, ist der Übergang wenn die versendeten Payloads die die MTU (Maximum Transmission Unit) des Netzwerks überschreiten und die übermittelten Nachrichten in mehrere Pakete aufgeteilt werden. Die voreingestellte MTU size auf den Clients und dem Broker beträgt 1500 Bytes. Siehe auch **??**.

```
setwd("/home/lisa/Darmstadt/05_Speicher und Datennetze IoT/Praktikum/Git/mqtt-qos-rountrip/logs/latenz-
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")
library("plyr")
library(kableExtra)

load("/home/lisa/Darmstadt/05_Speicher und Datennetze IoT/Praktikum/R_Analysis/latenzTC1mbps.Rda")
load("/home/lisa/Darmstadt/05_Speicher und Datennetze IoT/Praktikum/R_Analysis/latenzTc10kbps.Rda")
load("/home/lisa/Darmstadt/05_Speicher und Datennetze IoT/Praktikum/R_Analysis/latenzTc10mbps.Rda")
load("/home/lisa/Darmstadt/05_Speicher und Datennetze IoT/Praktikum/R_Analysis/latenzTc100kbps.Rda")
load("/home/lisa/Darmstadt/05_Speicher und Datennetze IoT/Praktikum/R_Analysis/latenztc100mbps.Rda")
```

```
#files <- list.files(pattern = "*bps.Rda", full.names = TRUE, recursive = FALSE)
files <- c("latenzTc100kbps", "latenztc100mbps", "latenzTc10kbps", "latenzTc10mbps", "latenzTC1mbps")
```

Übersicht und notwendige Anpassung der Messungen für die mit dem TC der Mayimale Durchsatz (File Name) angepasst wurde. Die tatsächlich versendete Paketgröße und Anzahl pro Sekunde ist dem jeweiligen Lognamen zu entnehmen

```
latenzTc100kbps$MaxDurc <- "100kbps"
latenztc100mbps$MaxDurc <- "100mbps"
latenzTc10kbps$MaxDurc <- "10kbps"
latenzTc10mbps$MaxDurc <- "10mbps"
latenzTC1mbps$MaxDurc <- "1mbps"

tcLogs <- rbind(latenzTc100kbps, latenztc100mbps, latenzTc10kbps, latenzTc10mbps, latenzTC1mbps)
tcLogs$Size <- "10KByte"

colnames<-colnames(tcLogs)
colnames[5]<-"PproSek"
colnames(tcLogs)<- colnames
```

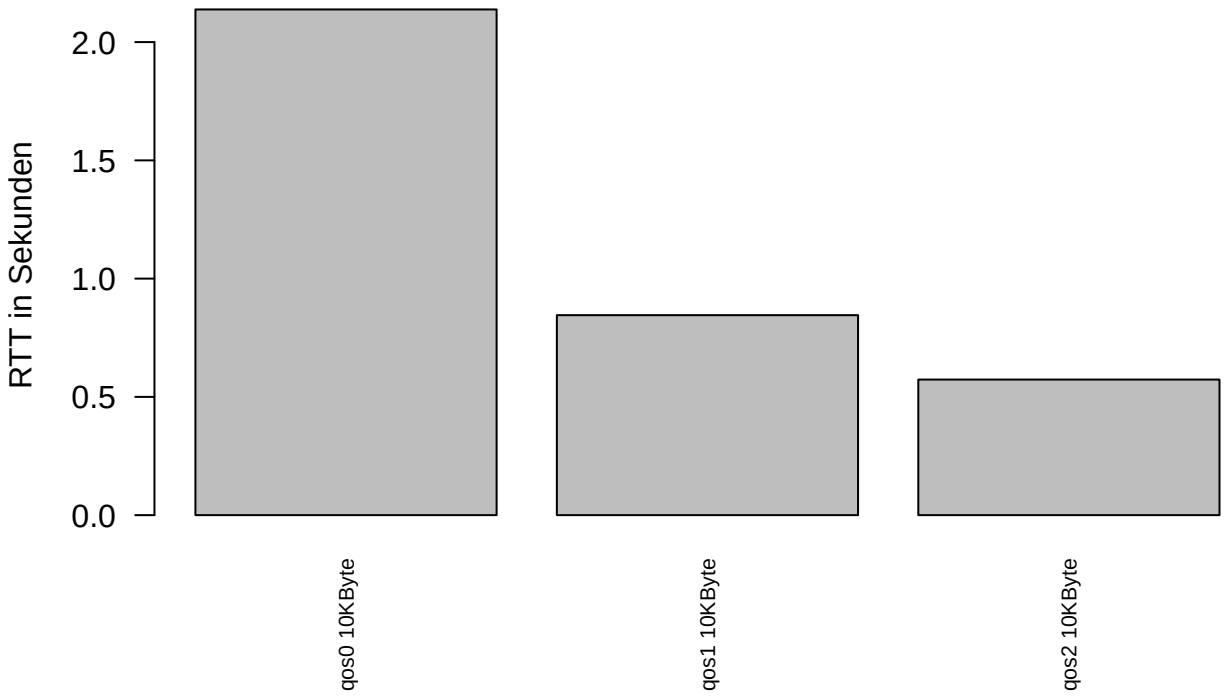
```
tcLogs$Byte<-tcLogs$Size
tcLogs$Byte[tcLogs$Byte == "1Byte"] <- 1
tcLogs$Byte[tcLogs$Byte == "10Byte"] <- 10
tcLogs$Byte[tcLogs$Byte == "100Byte"] <- 100
tcLogs$Byte[tcLogs$Byte == "1KByte"] <- 1000
tcLogs$Byte[tcLogs$Byte == "1500Byte"] <- 1500
tcLogs$Byte[tcLogs$Byte == "10KByte"] <- 10000
tcLogs$Byte[tcLogs$Byte == "100KByte"] <- 100000
tcLogs$Byte[tcLogs$Byte == "500KByte"] <- 500000
tcLogs$Byte[tcLogs$Byte == "1MByte"] <- 1000000
tcLogsSum <- summary(tcLogs)
tcLogsSum
#>       sent                          QoS                Size
#>  Min.   :2018-05-23 23:53:42.52   Length:34722       Length:34722
#>  1st Qu.:2018-05-24 00:28:49.04   Class :character   Class :character
#>  Median :2018-05-24 00:31:31.97   Mode  :character   Mode  :character
#>  Mean   :2018-05-24 00:29:49.74
#>  3rd Qu.:2018-05-24 00:33:27.54
#>  Max.   :2018-05-24 01:41:28.41
#>      Min              PproSek               rec
#>  Length:34722       Length:34722       Min.   :2018-05-23 23:53:42.53
#>  Class :character   Class :character   1st Qu.:2018-05-24 00:28:49.04
#>  Mode  :character   Mode  :character   Median :2018-05-24 00:31:31.98
#>                                        Mean   :2018-05-24 00:29:52.90
#>                                        3rd Qu.:2018-05-24 00:33:27.57
#>                                        Max.   :2018-05-24 01:43:25.93
#>    r_newid              rtt               id         MaxDurc
#>  Length:34722       Min.   : 0.001   Min.   :   1   Length:34722
#>  Class :character   1st Qu.: 0.002   1st Qu.: 342   Class :character
#>  Mode  :character   Median : 0.004   Median :1488   Mode  :character
#>                     Mean   : 1.352   Mean   :2043
#>                     3rd Qu.: 0.023   3rd Qu.:3658
#>                     Max.   :59.924   Max.   :5835
#>      Byte
#>  Length:34722
#>  Class :character
#>  Mode  :character
#>
#>
#>

tcLogs$ByteD<-tcLogs$MaxDurc
tcLogs$ByteD[tcLogs$ByteD == "1Byte"] <-   1
tcLogs$ByteD[tcLogs$ByteD == "10BByte"] <- 10
tcLogs$ByteD[tcLogs$ByteD == "100Byte"] <- 100
tcLogs$ByteD[tcLogs$ByteD == "1kbps"] <-   1000
tcLogs$ByteD[tcLogs$ByteD == "10kbps"] <-  10000
tcLogs$ByteD[tcLogs$ByteD == "100kbps"] <- 100000
tcLogs$ByteD[tcLogs$ByteD == "1mbps"] <-   1000000
tcLogs$ByteD[tcLogs$ByteD == "10mbps"] <-  10000000
tcLogs$ByteD[tcLogs$ByteD == "100mbps"] <- 100000000

tcLogs$ByteD<-as.numeric(tcLogs$ByteD)
```

Aggregation der Daten zur Beantwortung der Fragestellung bzgl. Latenzzeiten in Abhängigkeit zu QoS Level und Paketgröße.

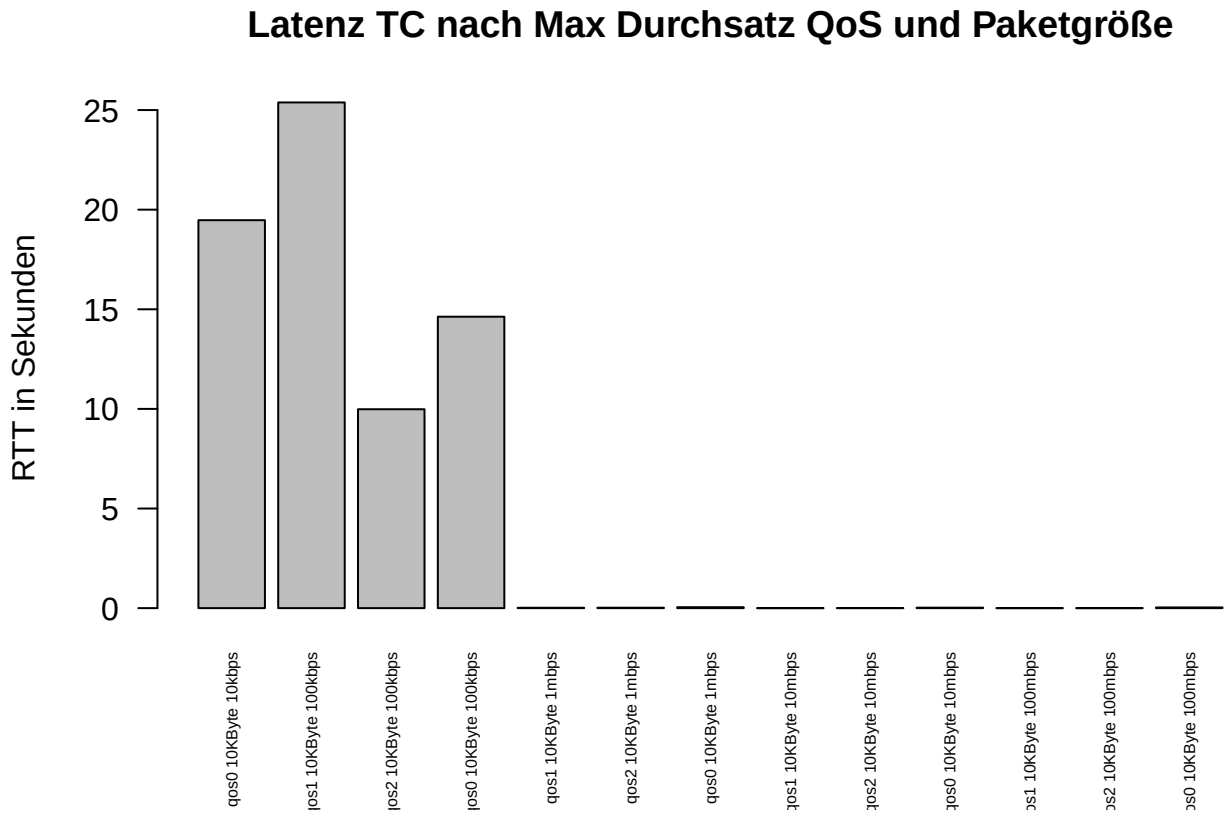
```
tcLogsAgg <- aggregate(tcLogs$rtt ~ tcLogs$QoS+tcLogs$Size+tcLogs$Byte, tcLogs, mean)
tcLogsAgg2 <- aggregate(tcLogs$rtt ~ tcLogs$QoS+tcLogs$Size+tcLogs$Byte + tcLogs$MaxDurc + tcLogs$ByteD

tcLogsAgg$`tcLogs$Byte`<-as.numeric(tcLogsAgg$`tcLogs$Byte`)
tcLogsAgg<-tcLogsAgg[order(tcLogsAgg$`tcLogs$Byte`),]

tcLogsAgg2$`tcLogs$Byte`<-as.numeric(tcLogsAgg2$`tcLogs$ByteD`)
tcLogsAgg2<-tcLogsAgg2[order(tcLogsAgg2$`tcLogs$ByteD`),]


tcLogsAgg2 %>%
  kable() %>%
  kable_styling()
```

tcLogs$QoS	tcLogs$Size	tcLogs$Byte	tcLogs$MaxDurc	tcLogs$ByteD	tcLogs$rtt
qos0	10KByte	1e+04	10kbps	1e+04	19.4711302
qos0	10KByte	1e+05	100kbps	1e+05	25.3815006
qos1	10KByte	1e+05	100kbps	1e+05	9.9820152
qos2	10KByte	1e+05	100kbps	1e+05	14.6274025
qos0	10KByte	1e+06	1mbps	1e+06	0.0190646
qos1	10KByte	1e+06	1mbps	1e+06	0.0211253
qos2	10KByte	1e+06	1mbps	1e+06	0.0474118
qos0	10KByte	1e+07	10mbps	1e+07	0.0021778
qos1	10KByte	1e+07	10mbps	1e+07	0.0043793
qos2	10KByte	1e+07	10mbps	1e+07	0.0242875
qos0	10KByte	1e+08	100mbps	1e+08	0.0022214
qos1	10KByte	1e+08	100mbps	1e+08	0.0030982
qos2	10KByte	1e+08	100mbps	1e+08	0.0341508

```
tcLogsAgg$Names <- paste(tcLogsAgg$`tcLogs$QoS`, tcLogsAgg$`tcLogs$Size`)
tcLogsAgg<-tcLogsAgg[order(tcLogsAgg$`tcLogs$Byte`),]
barplot(tcLogsAgg$`tcLogs$rtt`, main = "Latenz TC nach QoS und Paketgröße", ylab = "RTT in Sekunden", a
```



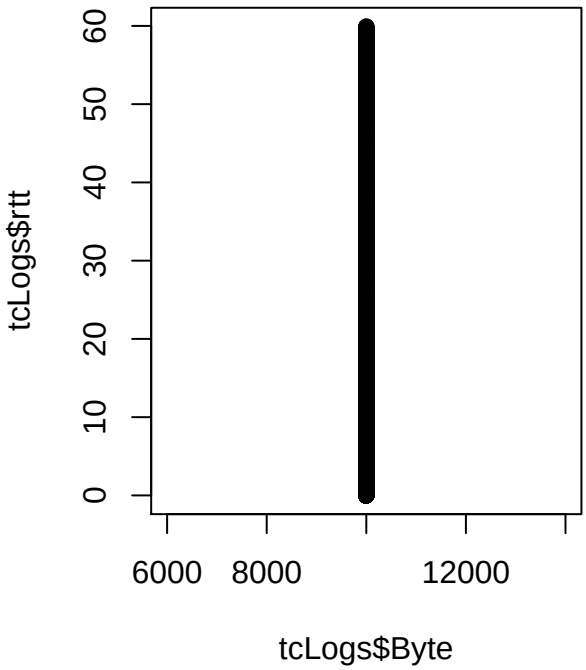

```
tcLogsAgg2$Names2 <- paste(tcLogsAgg$`tcLogs$QoS`, tcLogsAgg$`tcLogs$Size`, tcLogsAgg2$`tcLogs$MaxDurc`
tcLogsAgg<-tcLogsAgg[order(tcLogsAgg$`tcLogs$Byte`),]
barplot(tcLogsAgg2$`tcLogs$rtt`, main = "Latenz TC nach Max Durchsatz QoS und Paketgröße", ylab = "RTT
```

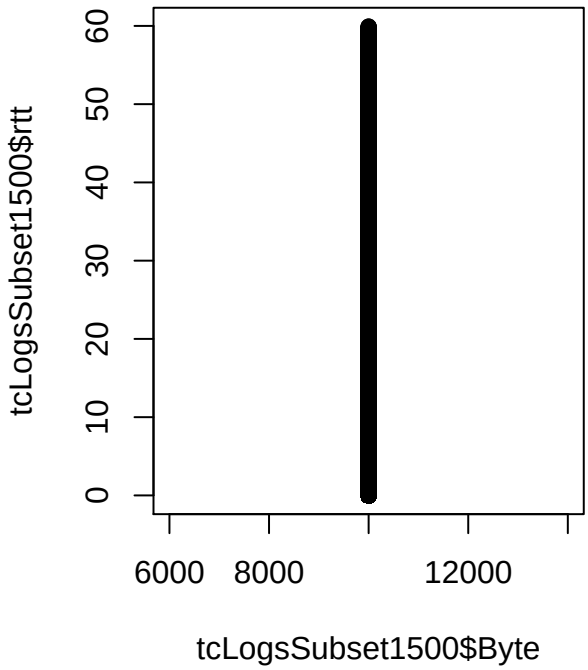


Im nächsten Schritt wird die statistische Abhngigkeit der rtt von QoS und Größe (Byte) untersucht. Im Falle einer einfachen linearen Regression sind nur qos2 und hohe Byte Zahlen signifikant.

```
reg_tcLogs <- lm(tcLogs$rtt~tcLogs$QoS+tcLogs$Byte+tcLogs$ByteD, data = tcLogs)
#> Error in `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]): contrasts can be applied only to
summary(reg_tcLogs)
#> Error in summary(reg_tcLogs): Objekt 'reg_tcLogs' nicht gefunden
reg_tcLogs2 <- lm(tcLogs$rtt~tcLogs$QoS+tcLogs$Byte+tcLogs$MaxDurc, data = tcLogs)
#> Error in `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]): contrasts can be applied only to
summary(reg_tcLogs2)
#> Error in summary(reg_tcLogs2): Objekt 'reg_tcLogs2' nicht gefunden
```

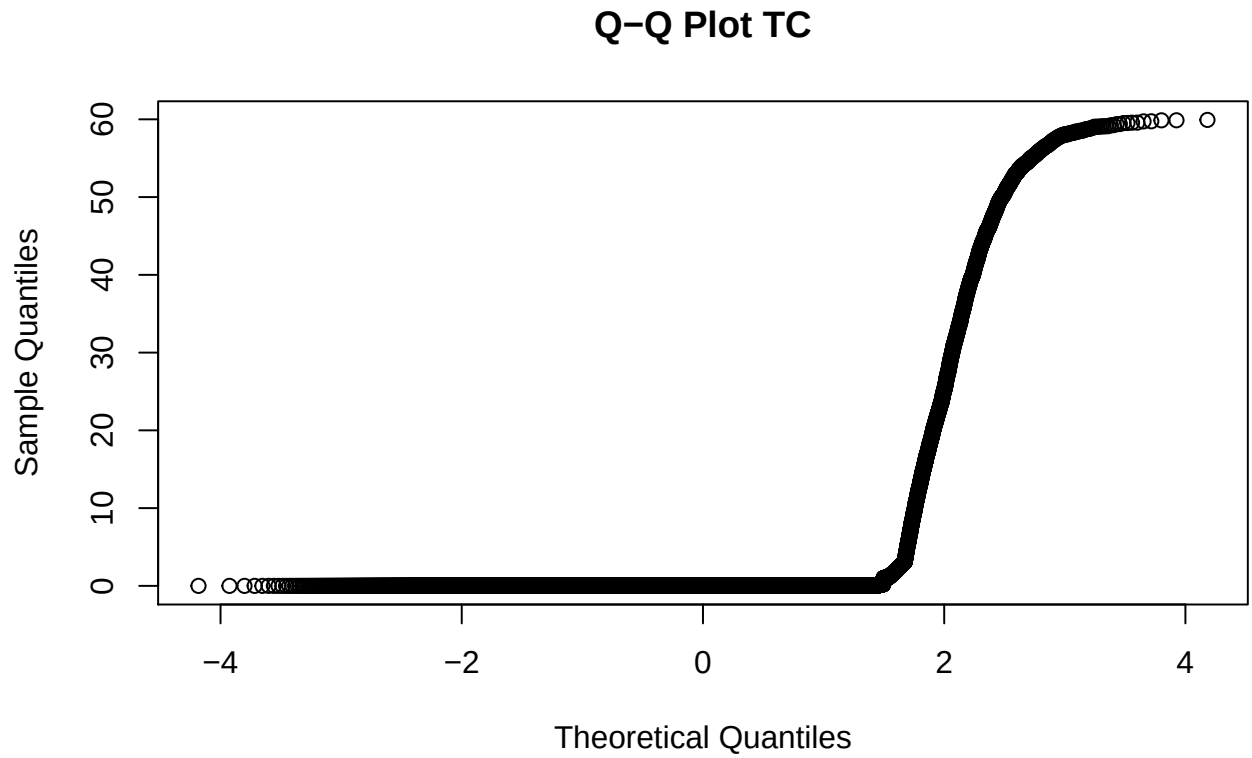
```
par(mfrow=c(1,2))
plot(tcLogs$Byte, tcLogs$rtt)
tcLogsSubset1500 <- tcLogs[tcLogs$Byte <= 1500, ]
plot(tcLogsSubset1500$Byte, tcLogsSubset1500$rtt)
```





Scatterplots bringen bei der Größe wenig Übersicht aus zwei Gründen: 1.) Logische Datenypen, d.h. alle Beobachtungen sind gehäuft in den geweiligen Klassen 2.) Ohne Standardisierung/ Transformation der Daten haben die extremen Werte (MByte) einen überproportionalen Anteil -> Im folgenden wird die Verteilung von rtt in QQ Plots betrachtet /.

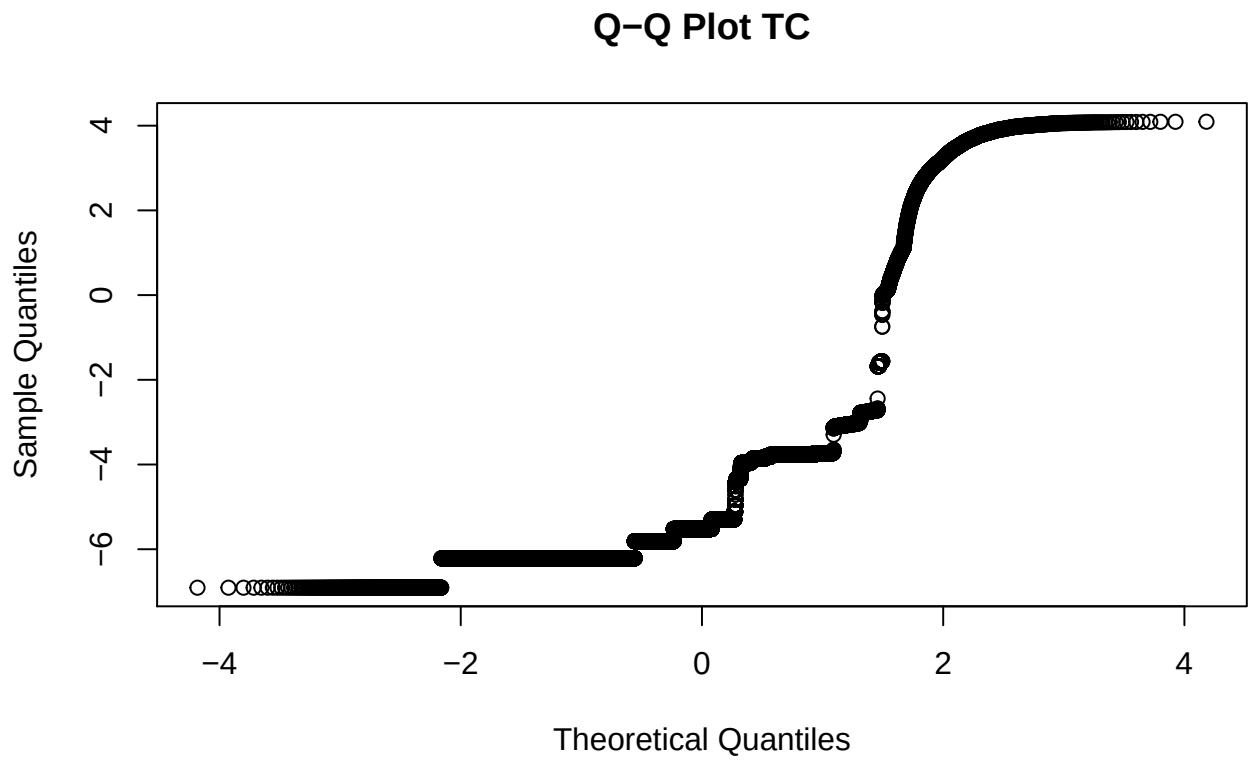
```
par(mfrow=c(1,1))
qqnorm(tcLogs$rtt, main = "Q-Q Plot TC")
```



Da rtt nicht normal verteilt ist, liefert die Lineare Regression keine zuverlässigen Ergebnisse. Nach der

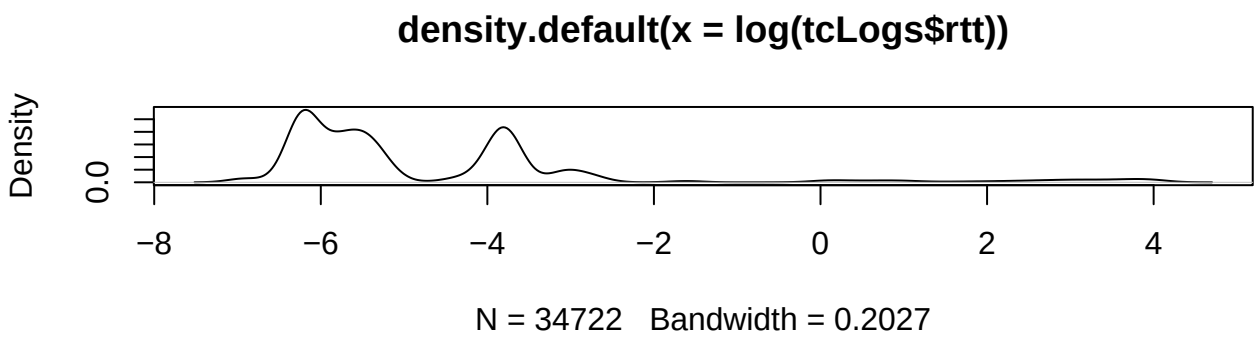
Transformation (logarithmierung) nähert sich die Verteilung der Variable rtt der Normalverteilung. (Normalverteilung ist erreicht, wenn die Sample Quantile den Theoretischen entsprechen - die Beobachtugen also auf einer Geraden liegen)

```
par(mfrow=c(1,1))
qqnorm(log(tcLogs$rtt), main = "Q-Q Plot TC")
```



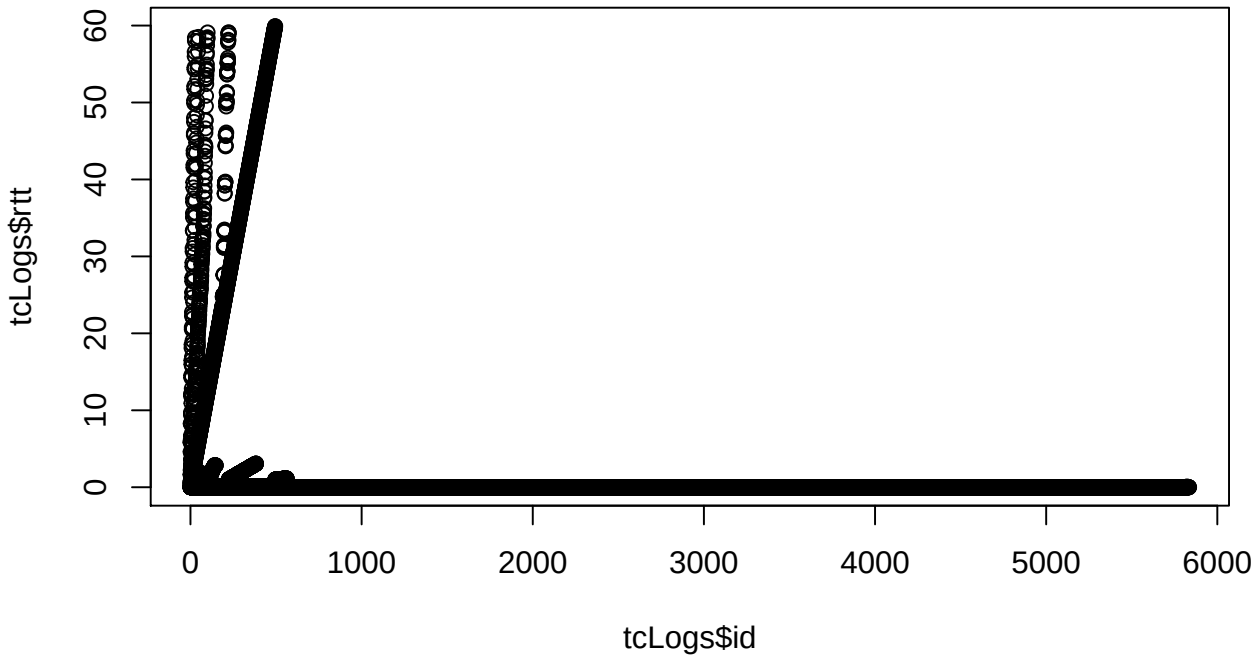
Trotz der Logarithmierung sind die Daten nicht perfekt Normalverteilt, jedoch annähernd.

```
par(mfrow=c(2,1))
plot(density(log(tcLogs$rtt)))
```

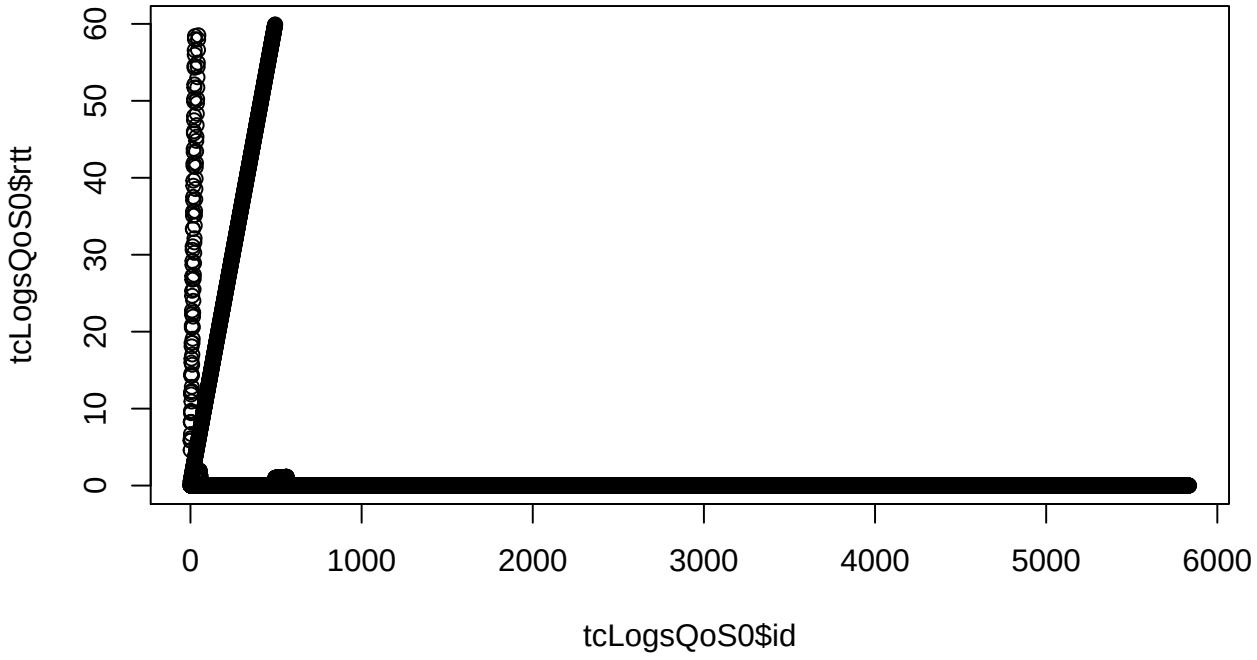



```
reg_tcLogs <- lm(log(tcLogs$rtt)~tcLogs$QoS+tcLogs$Byte, data = tcLogs)
#> Error in `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]): contrasts can be applied only to
summary(reg_tcLogs)
#> Error in summary(reg_tcLogs): Objekt 'reg_tcLogs' nicht gefunden
```

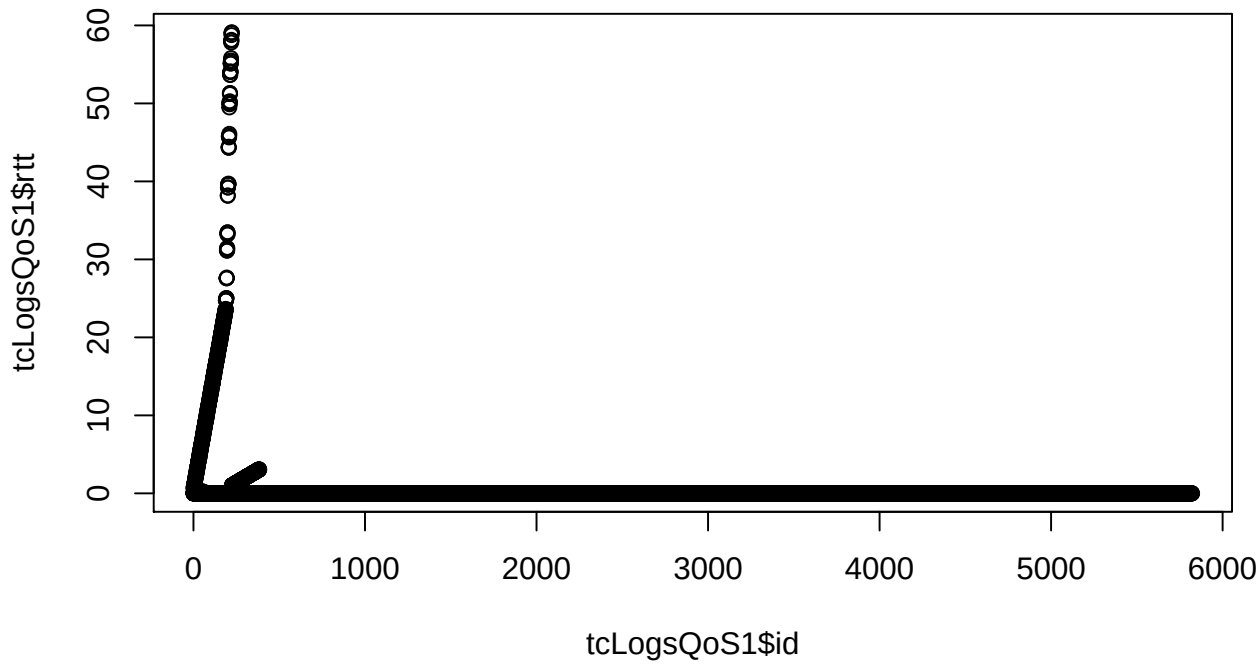
```
plot(tcLogs$id, tcLogs$rtt)
```



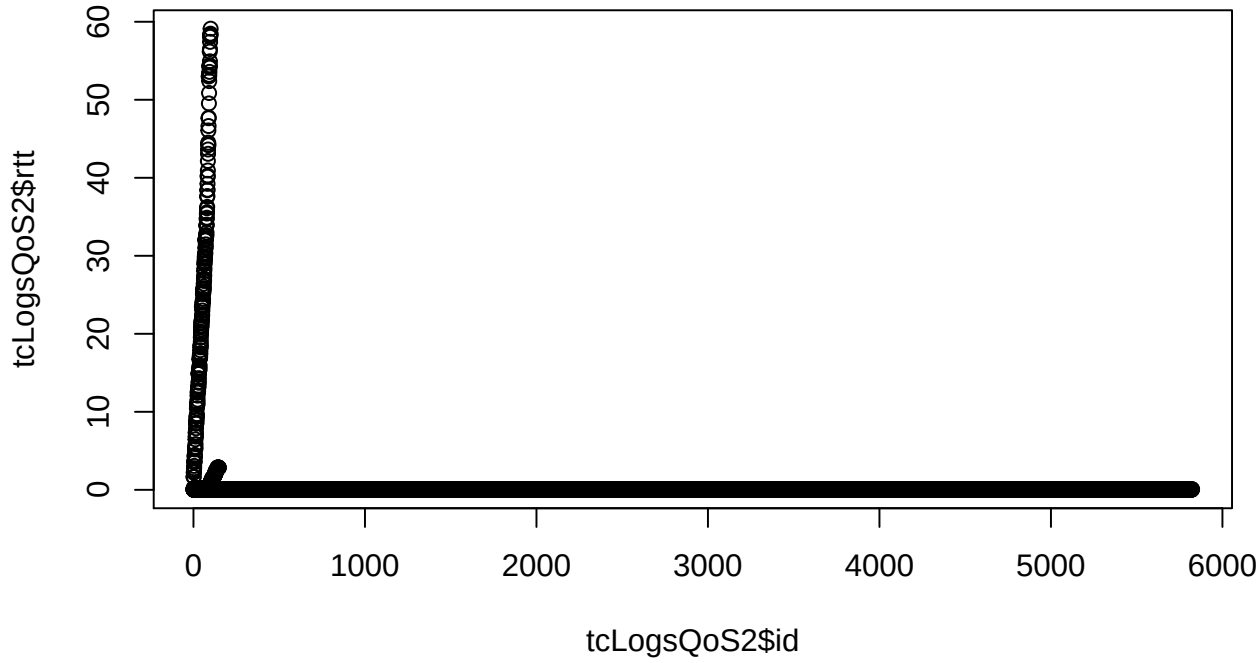

```
#########################
# Aufsplittung nach QoS #
#########################

tcLogsQoS0<-tcLogs[tcLogs$QoS == "qos0",]
tcLogsQoS1<-tcLogs[tcLogs$QoS == "qos1",]
tcLogsQoS2<-tcLogs[tcLogs$QoS == "qos2",]

plot(tcLogsQoS0$id, tcLogsQoS0$rtt)
```



```r
plot(tcLogsQoS1$id, tcLogsQoS1$rtt)
```

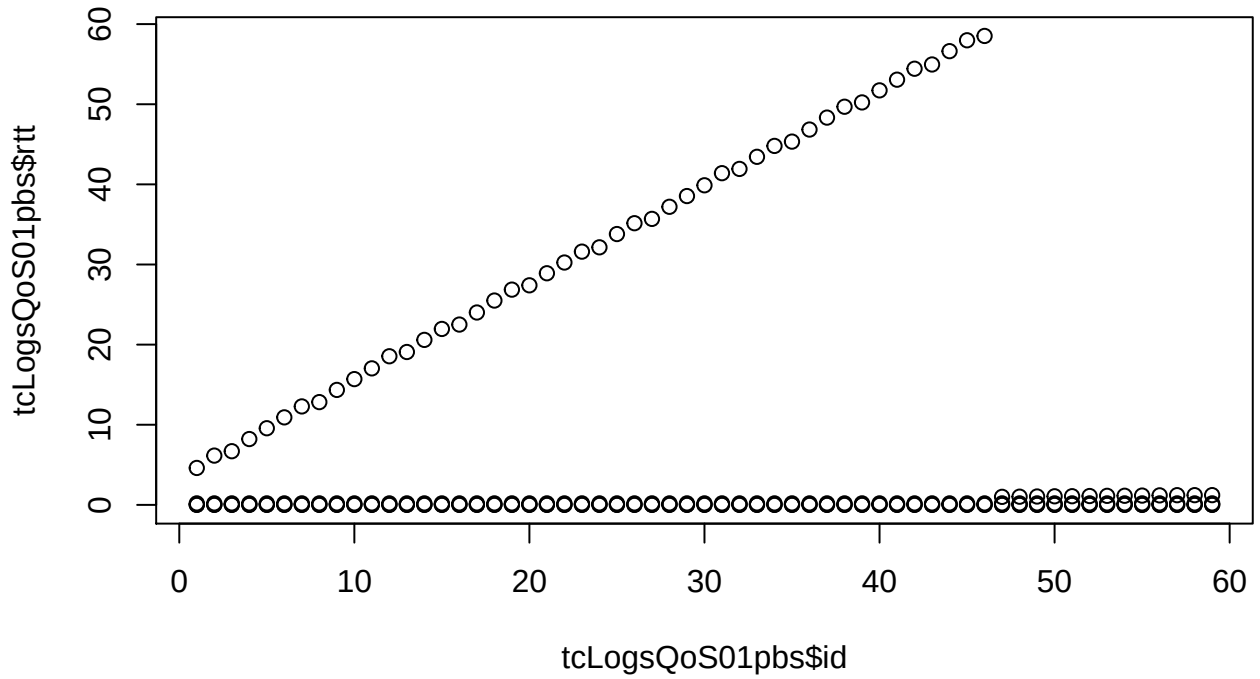



```r
plot(tcLogsQoS2$id, tcLogsQoS2$rtt)
```

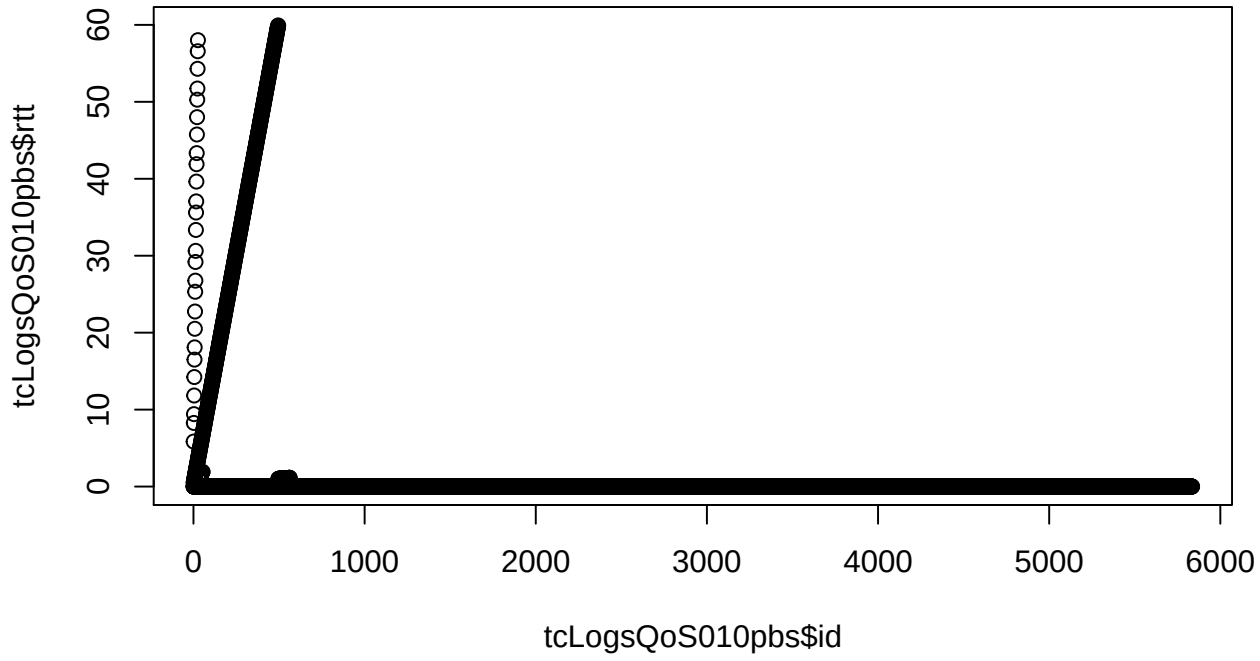



```r
############################################################
# QoS Level _ Aufsplittung nach Pakete pro Sekunde Qos #
############################################################
## QoS0 - Aufsplittung
tcLogsQoS01pbs<-tcLogsQoS0[tcLogsQoS0$PproSek == "1pbs",]
tcLogsQoS010pbs<-tcLogsQoS0[tcLogsQoS0$PproSek == "10pbs",]
tcLogsQoS0100pbs<-tcLogsQoS0[tcLogsQoS0$PproSek == "100pbs",]
```

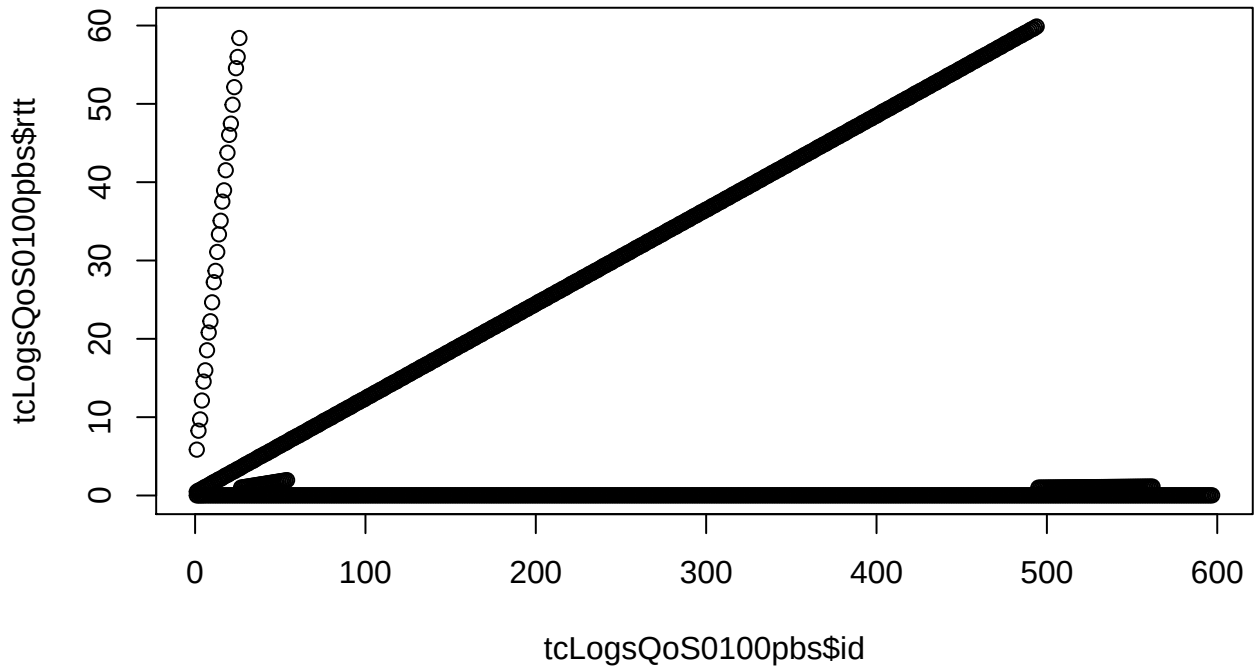
```
plot(tcLogsQoS01pbs$id, tcLogsQoS01pbs$rtt)
```



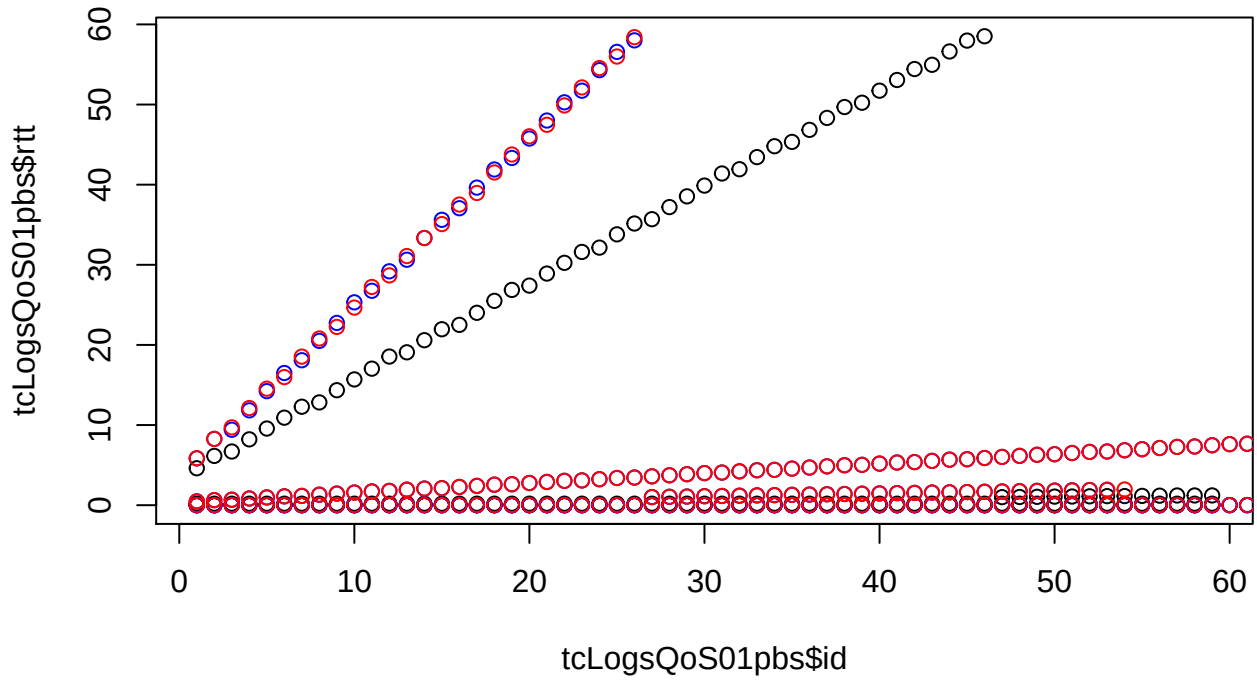

```
plot(tcLogsQoS010pbs$id, tcLogsQoS010pbs$rtt)
```




```
plot(tcLogsQoS0100pbs$id, tcLogsQoS0100pbs$rtt)
```

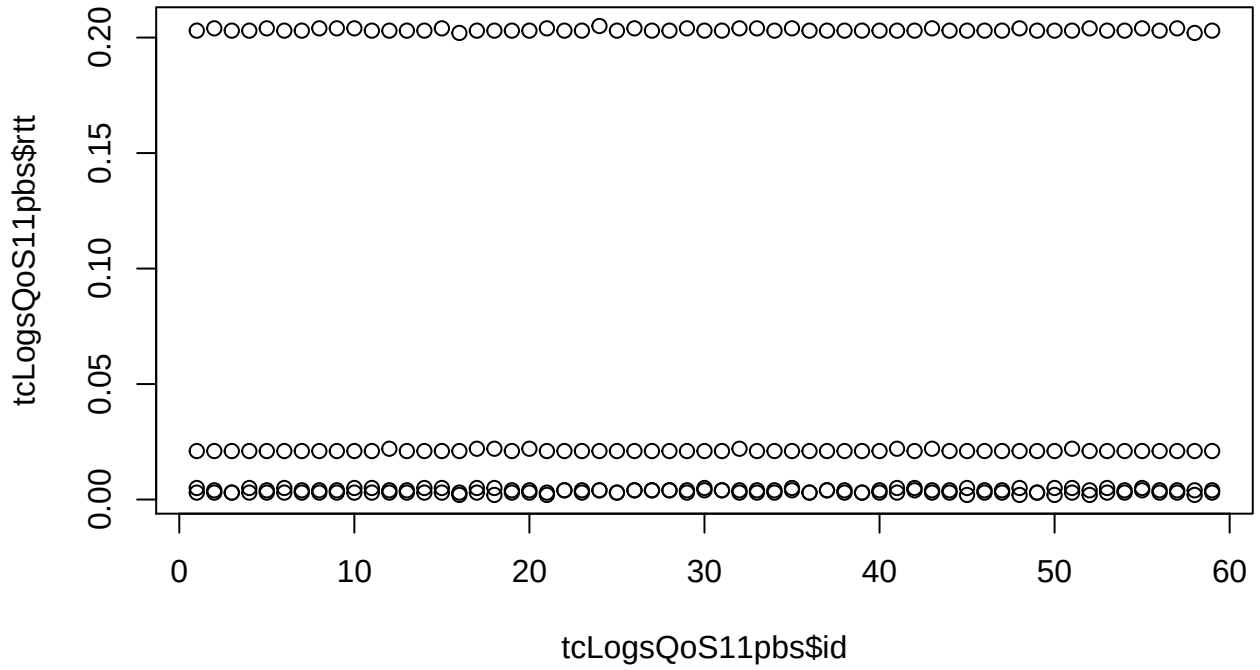



```
## QoS1 - Aufsplittung - Eine Grafik!
#par(mfrow=c(1,3))
plot(tcLogsQoS01pbs$id, tcLogsQoS01pbs$rtt)
points(tcLogsQoS010pbs$id, tcLogsQoS010pbs$rtt, col="blue")
points(tcLogsQoS0100pbs$id, tcLogsQoS0100pbs$rtt, col="red")
```

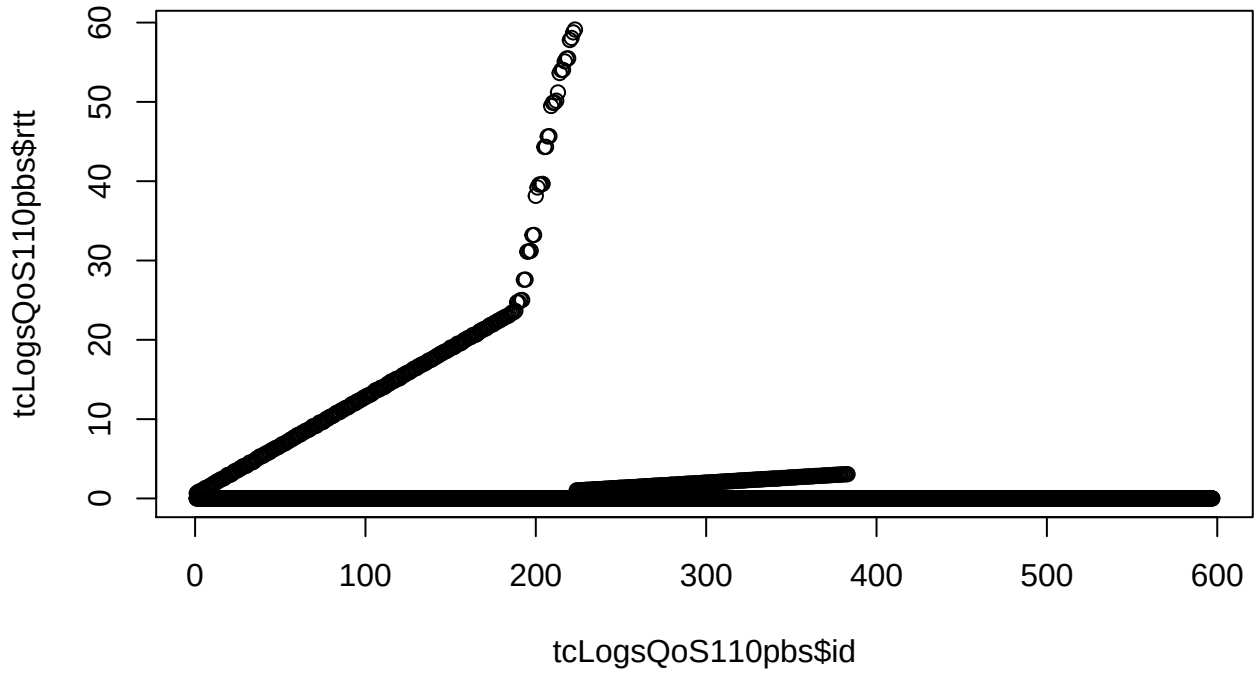



```
## QoS1 - Aufsplittung
tcLogsQoS11pbs<-tcLogsQoS1[tcLogsQoS1$PproSek == "1pbs",]
tcLogsQoS110pbs<-tcLogsQoS1[tcLogsQoS1$PproSek == "10pbs",]
tcLogsQoS1100pbs<-tcLogsQoS1[tcLogsQoS1$PproSek == "100pbs",]
```

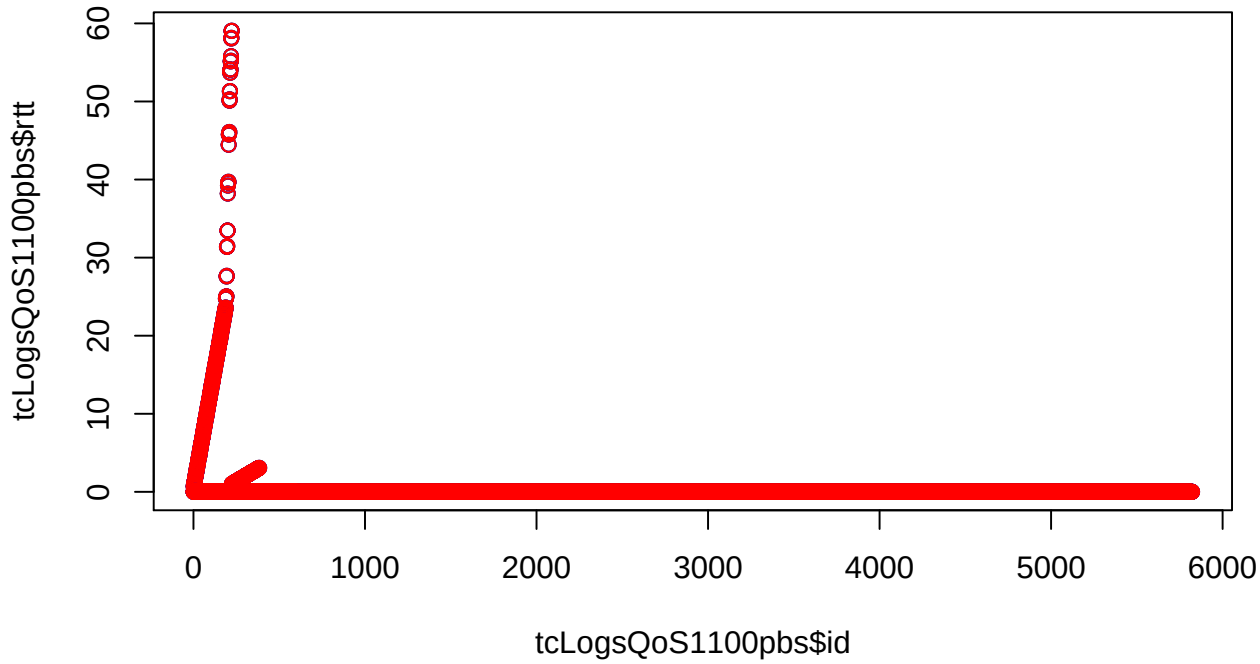
```
plot(tcLogsQoS11pbs$id, tcLogsQoS11pbs$rtt)
```



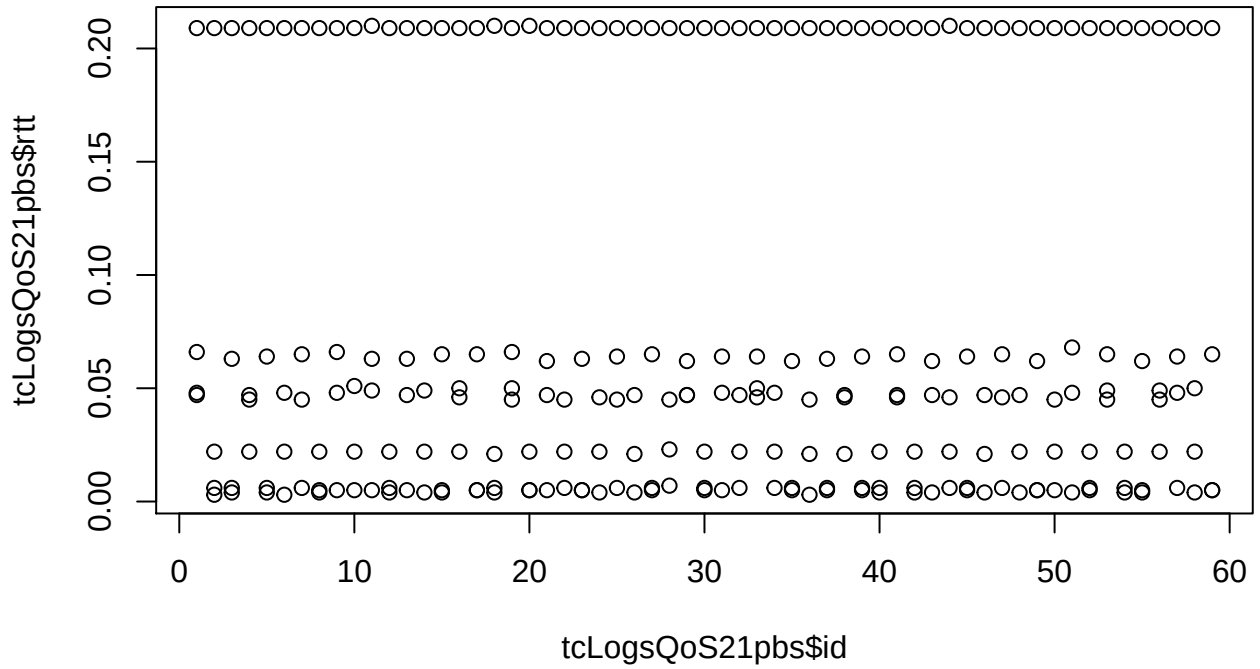

```
plot(tcLogsQoS110pbs$id, tcLogsQoS110pbs$rtt)
```



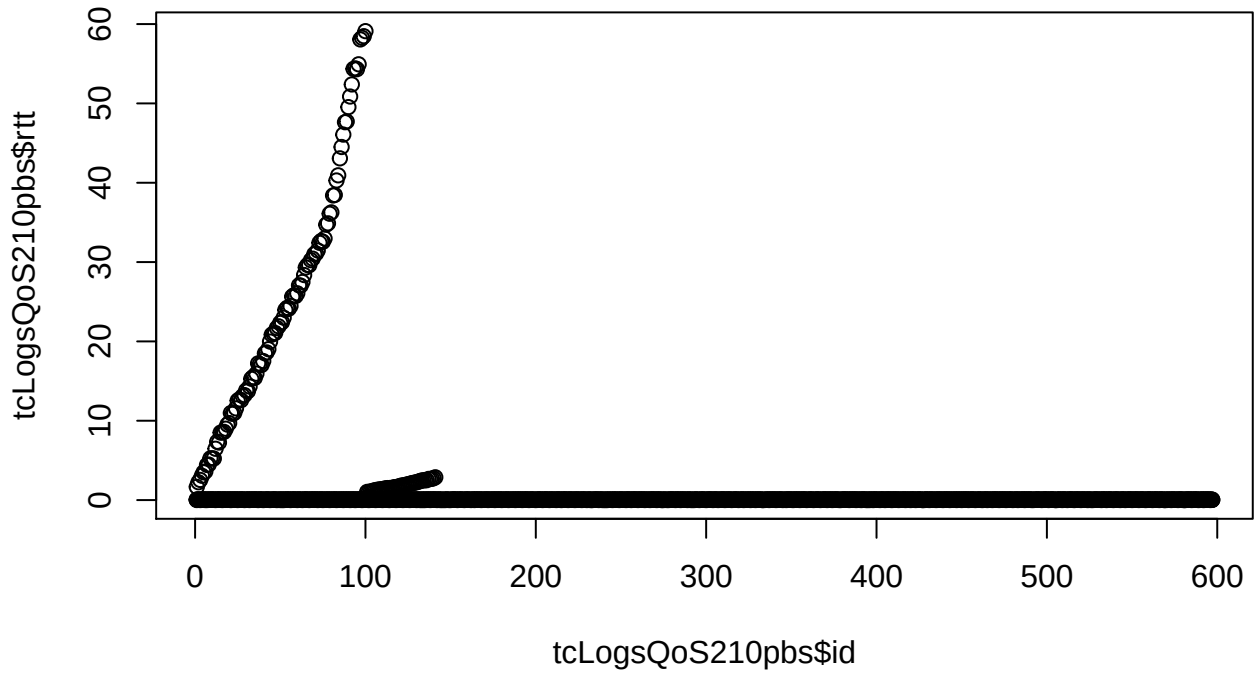

```
plot(tcLogsQoS1100pbs$id, tcLogsQoS1100pbs$rtt)

## QoS1 - Aufsplittung - eine Grafik!
plot(tcLogsQoS1100pbs$id, tcLogsQoS1100pbs$rtt)
points(tcLogsQoS1100pbs$id, tcLogsQoS1100pbs$rtt, col="blue")
points(tcLogsQoS1100pbs$id, tcLogsQoS1100pbs$rtt, col="red")
```



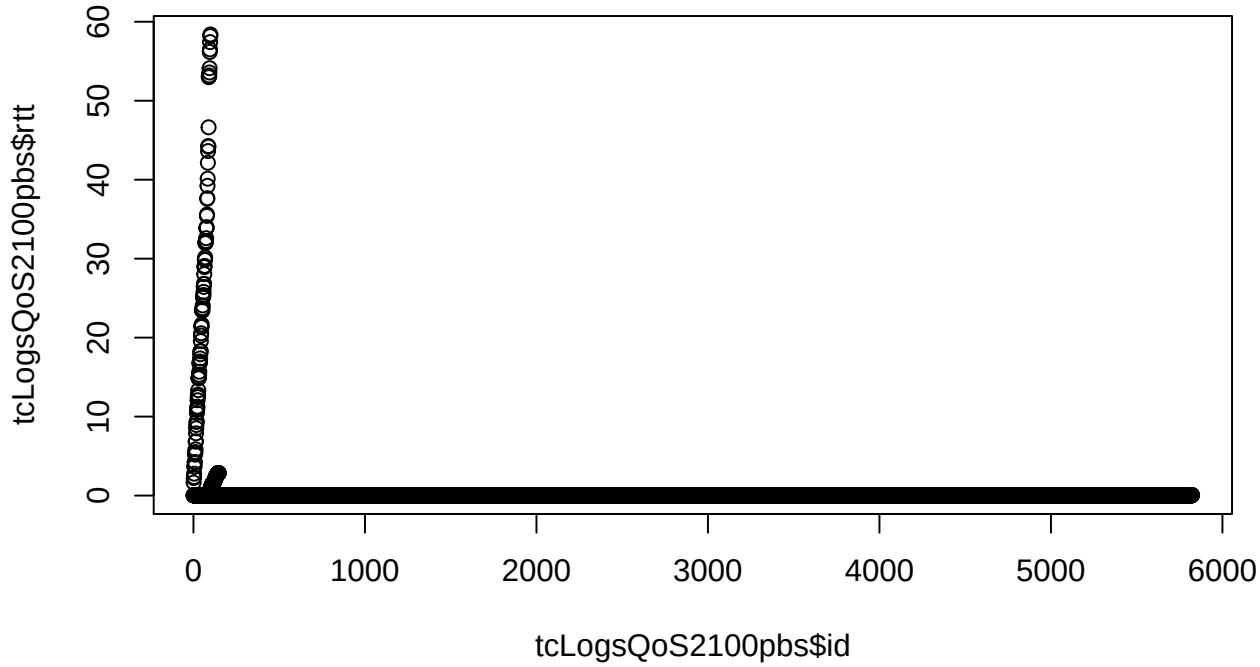

```
## QoS2 - Aufsplittung
tcLogsQoS21pbs<-tcLogsQoS2[tcLogsQoS2$PproSek == "1pbs",]
tcLogsQoS210pbs<-tcLogsQoS2[tcLogsQoS2$PproSek == "10pbs",]
tcLogsQoS2100pbs<-tcLogsQoS2[tcLogsQoS2$PproSek == "100pbs",]

plot(tcLogsQoS21pbs$id, tcLogsQoS21pbs$rtt)
```



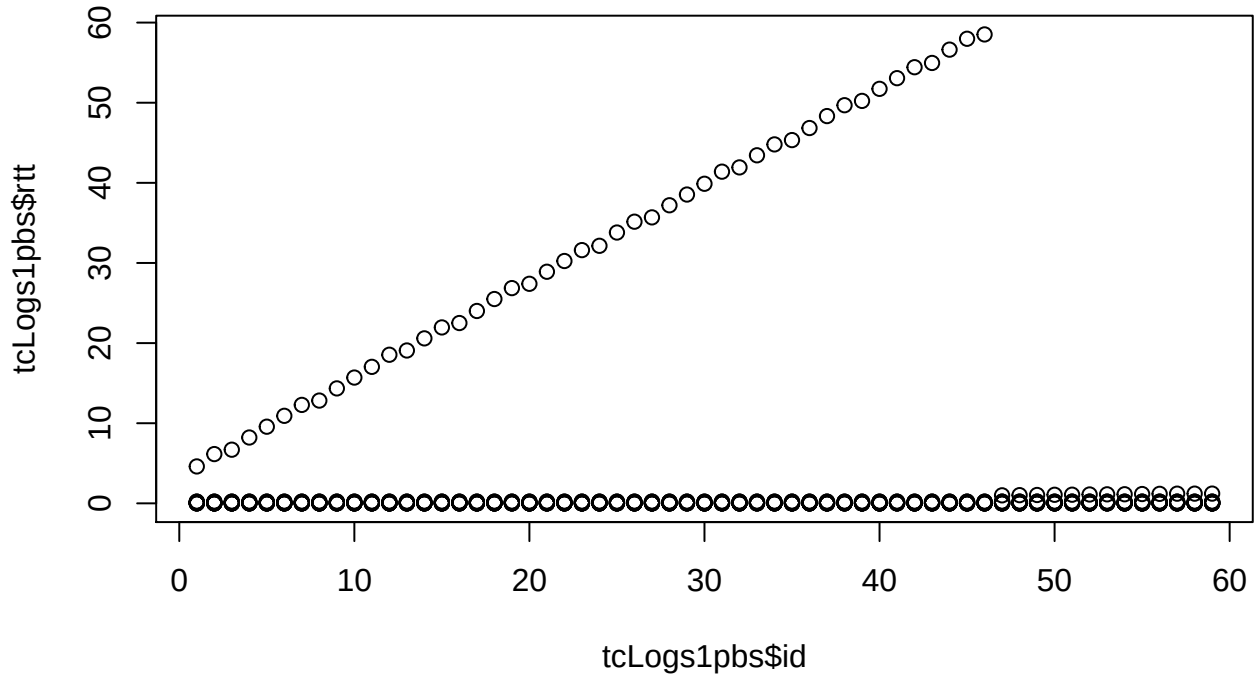

```
plot(tcLogsQoS210pbs$id, tcLogsQoS210pbs$rtt)
```



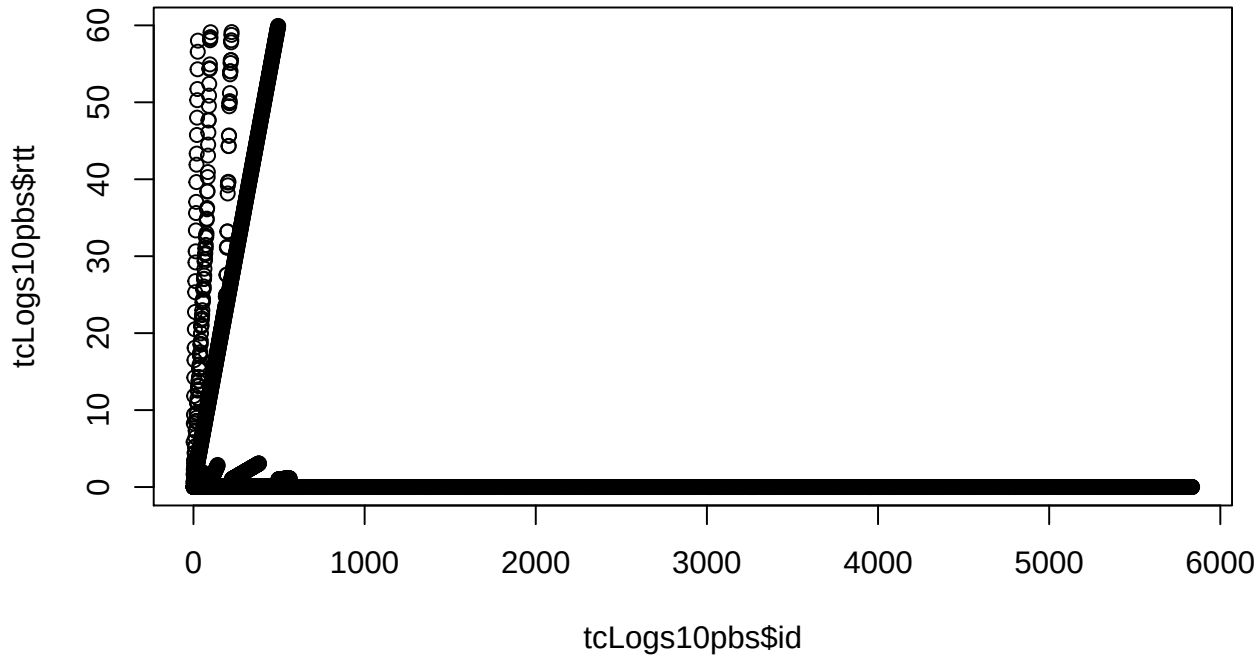

```
plot(tcLogsQoS2100pbs$id, tcLogsQoS2100pbs$rtt)
```



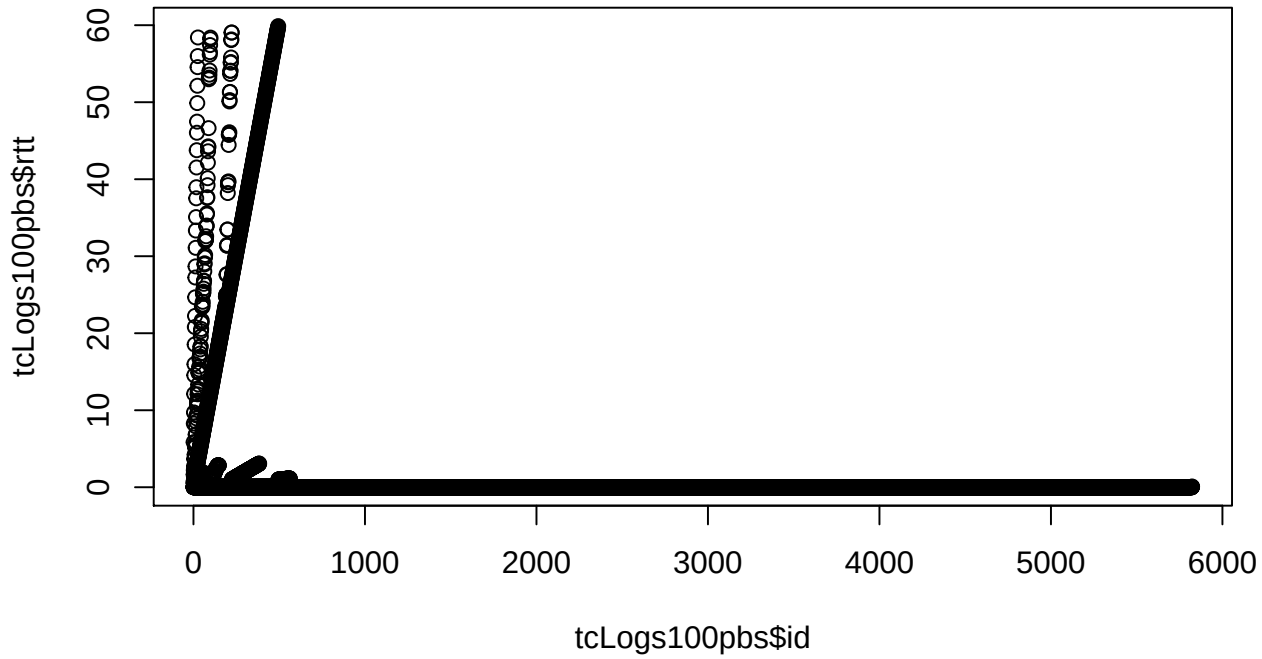

```
########################################
# Pakete pro Sekunde _ Aufsplittung Qos #
########################################

tcLogs1pbs<-tcLogs[tcLogs$PproSek == "1pbs",]
tcLogs10pbs<-tcLogs[tcLogs$PproSek == "10pbs",]
tcLogs100pbs<-tcLogs[tcLogs$PproSek == "100pbs",]

plot(tcLogs1pbs$id, tcLogs1pbs$rtt)
```



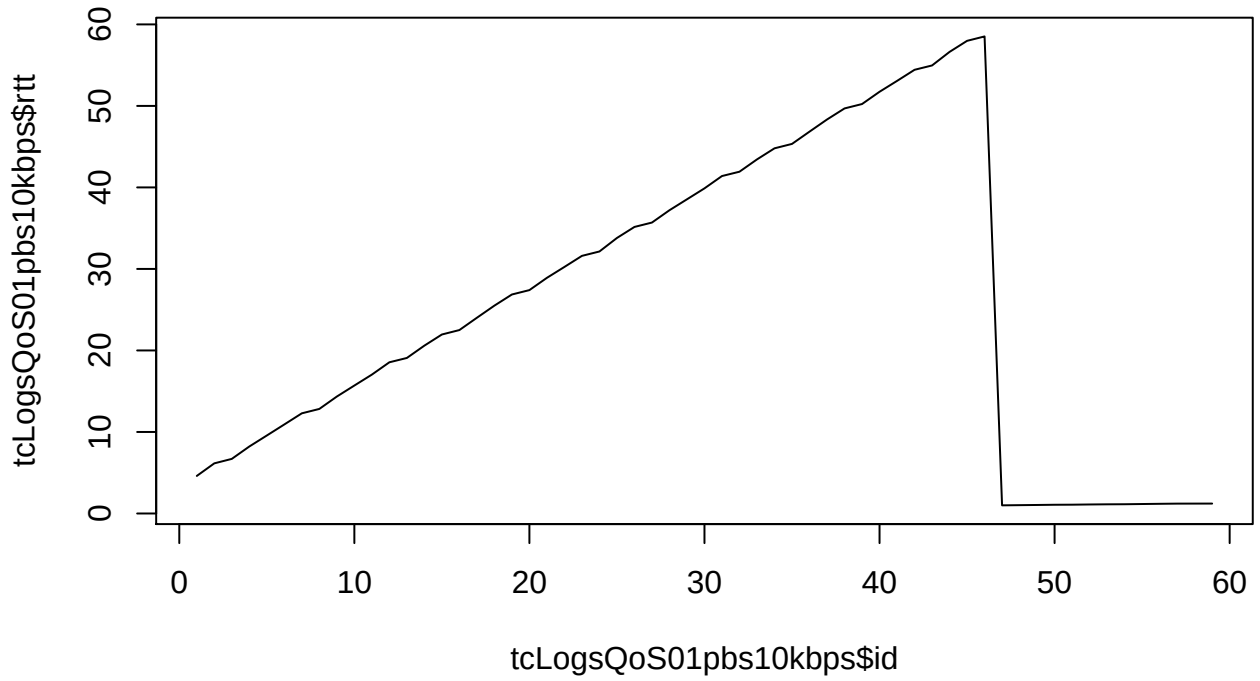

```
plot(tcLogs10pbs$id, tcLogs10pbs$rtt)
```



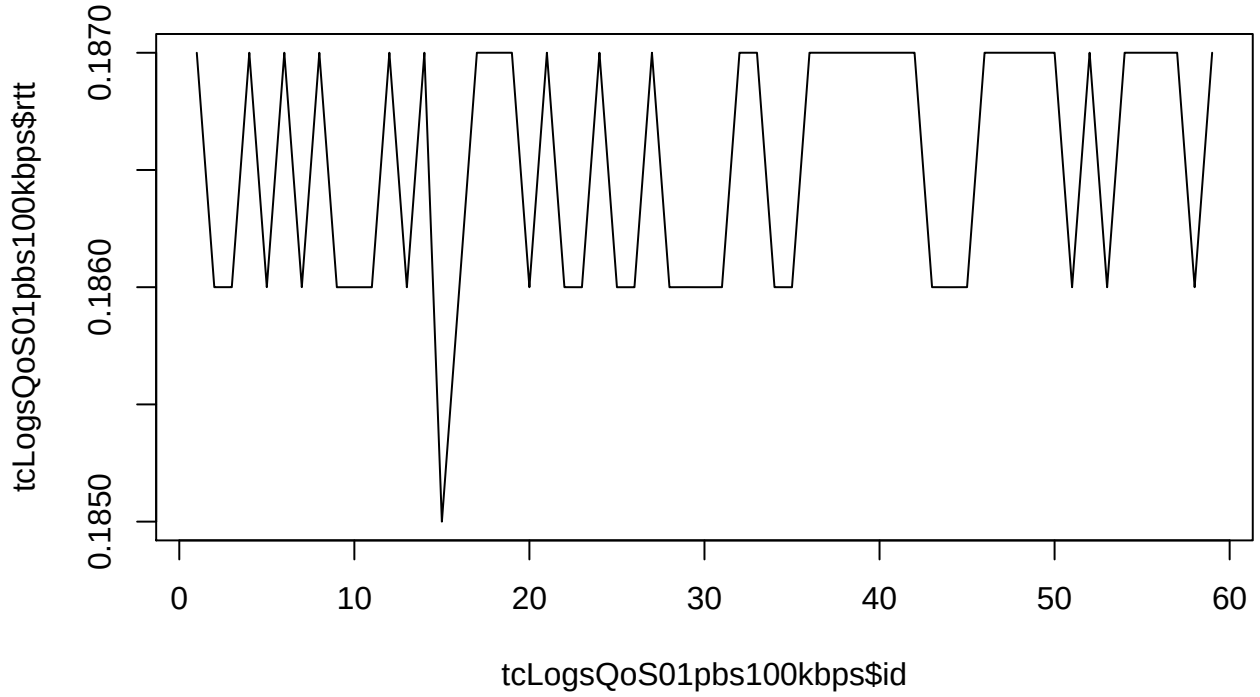

```
plot(tcLogs100pbs$id, tcLogs100pbs$rtt)
```



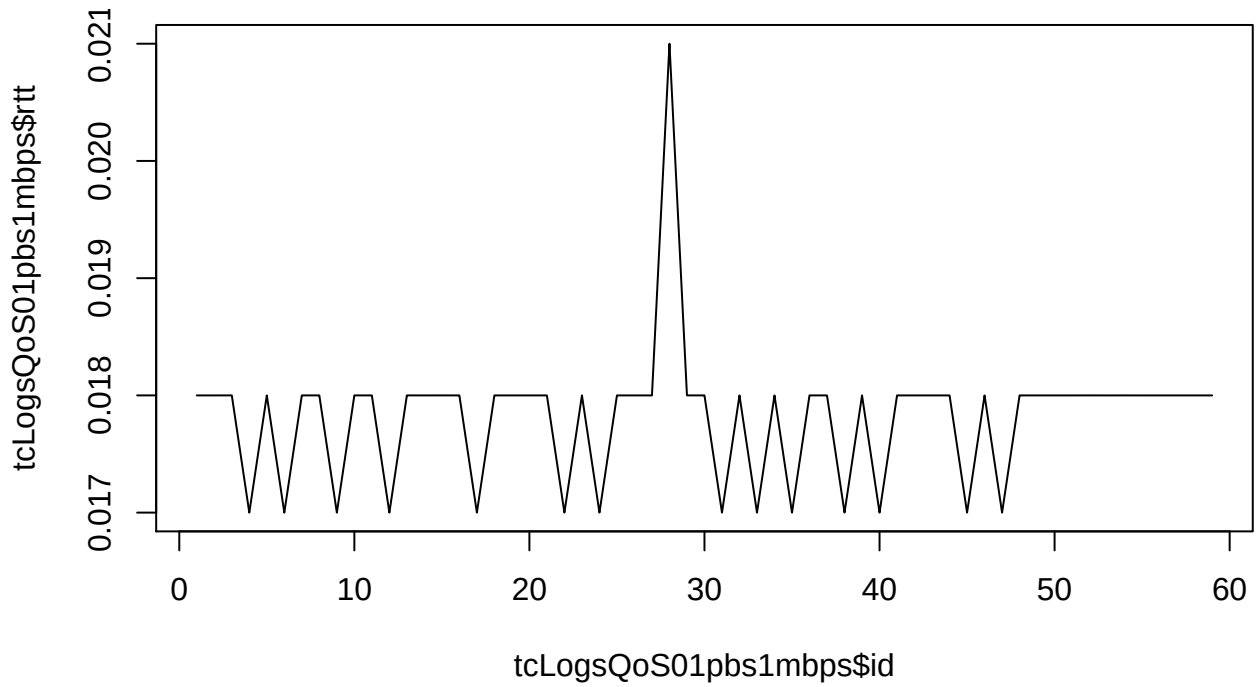

```
################################################################
# QoS Level _ Pakete pro Sekunde Qos _ Aufsplittung Grenze #
################################################################
## QoS0_1pbs - Aufsplittung MaxDurc

tcLogsQoS01pbs10kbps<-tcLogsQoS01pbs[tcLogsQoS01pbs$MaxDurc == "10kbps",]
tcLogsQoS01pbs100kbps<-tcLogsQoS01pbs[tcLogsQoS01pbs$MaxDurc == "100kbps",]
tcLogsQoS01pbs1mbps<-tcLogsQoS01pbs[tcLogsQoS01pbs$MaxDurc == "1mbps",]
tcLogsQoS01pbs10mbps<-tcLogsQoS01pbs[tcLogsQoS01pbs$MaxDurc == "10mbps",]
tcLogsQoS01pbs100mbps<-tcLogsQoS01pbs[tcLogsQoS01pbs$MaxDurc == "100mbps",]

plot(tcLogsQoS01pbs10kbps$id, tcLogsQoS01pbs10kbps$rtt, type = "l")
```



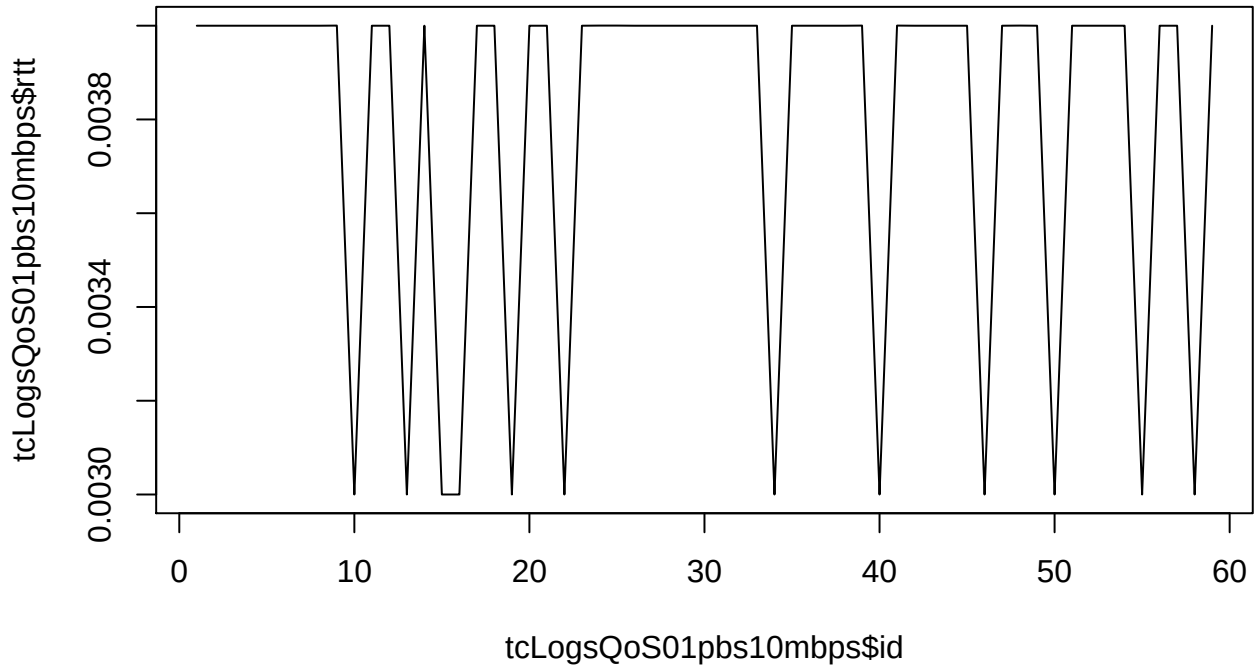

```
plot(tcLogsQoS01pbs100kbps$id, tcLogsQoS01pbs100kbps$rtt, type = "l")
```




```
plot(tcLogsQoS01pbs1mbps$id, tcLogsQoS01pbs1mbps$rtt, type = "l")
```



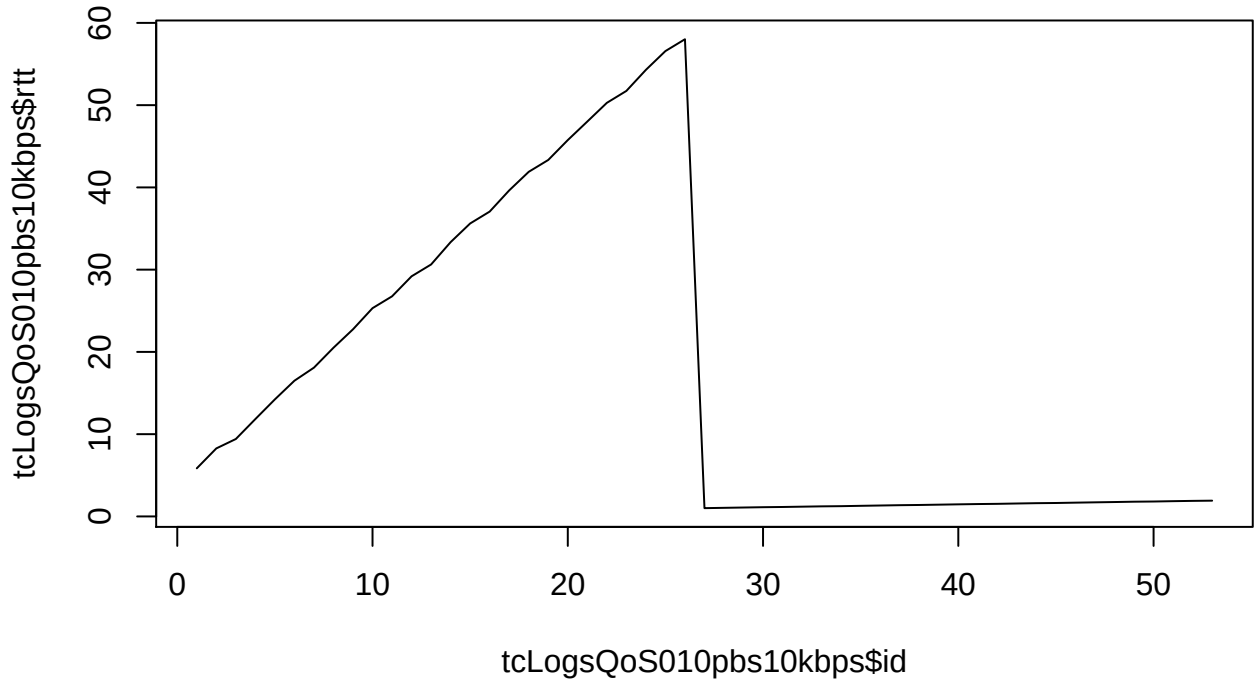

```
plot(tcLogsQoS01pbs10mbps$id, tcLogsQoS01pbs10mbps$rtt, type = "l")
```



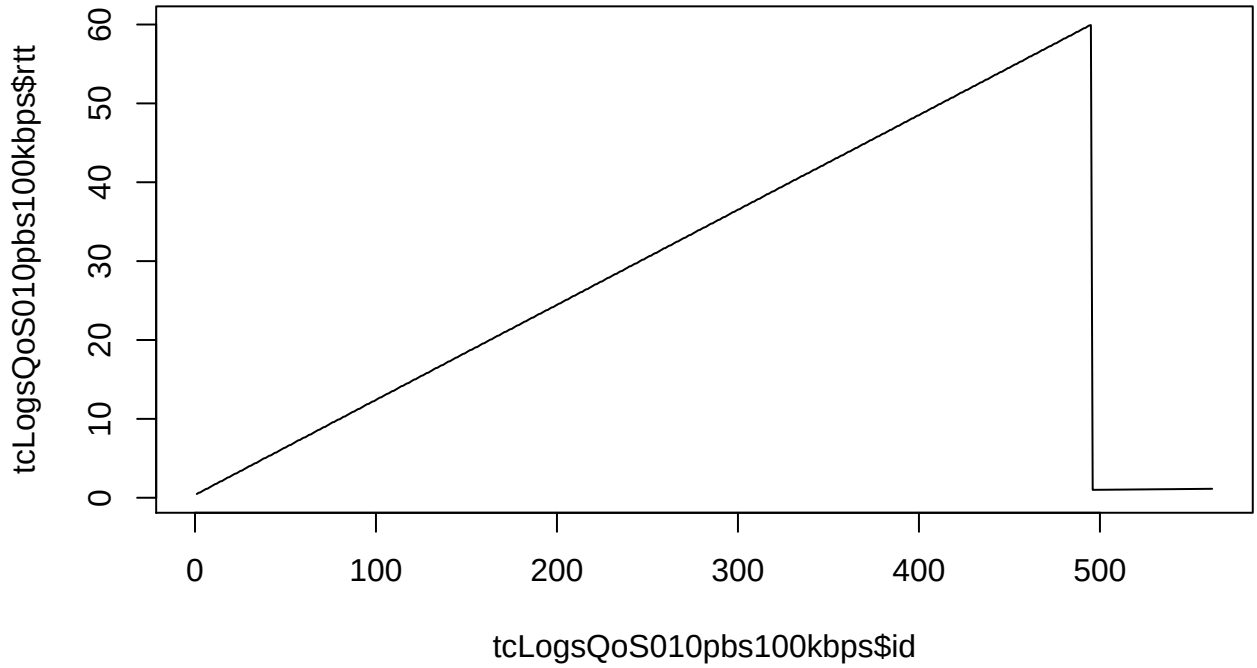

```
#plot(tcLogsQoS01pbs100mbps$id, tcLogsQoS01pbs100mbps$rtt, type = "l")
```

```
## QoS1_10pbs - Aufsplittung MaxDurc
tcLogsQoS010pbs10kbps<-tcLogsQoS010pbs[tcLogsQoS010pbs$MaxDurc == "10kbps",]
tcLogsQoS010pbs100kbps<-tcLogsQoS010pbs[tcLogsQoS010pbs$MaxDurc == "100kbps",]
tcLogsQoS010pbs1mbps<-tcLogsQoS010pbs[tcLogsQoS010pbs$MaxDurc == "1mbps",]
tcLogsQoS010pbs10mbps<-tcLogsQoS010pbs[tcLogsQoS010pbs$MaxDurc == "10mbps",]
tcLogsQoS010pbs100mbps<-tcLogsQoS010pbs[tcLogsQoS010pbs$MaxDurc == "100mbps",]
```

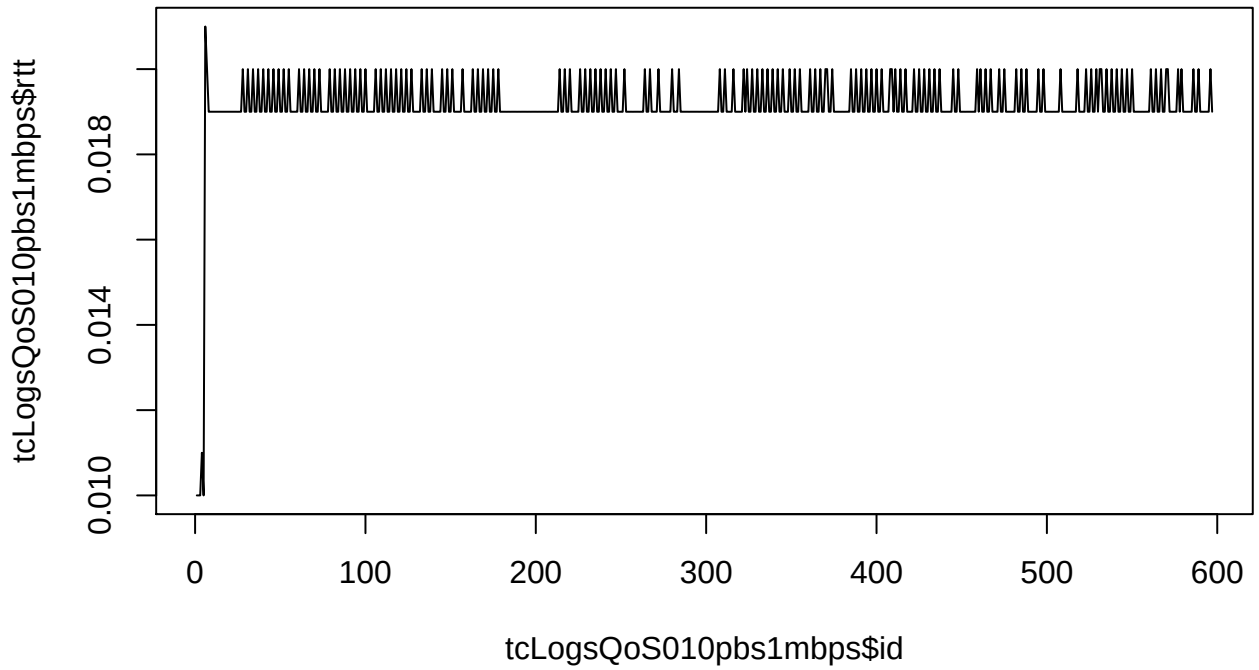
```
plot(tcLogsQoS010pbs10kbps$id, tcLogsQoS010pbs10kbps$rtt, type = "l")
```



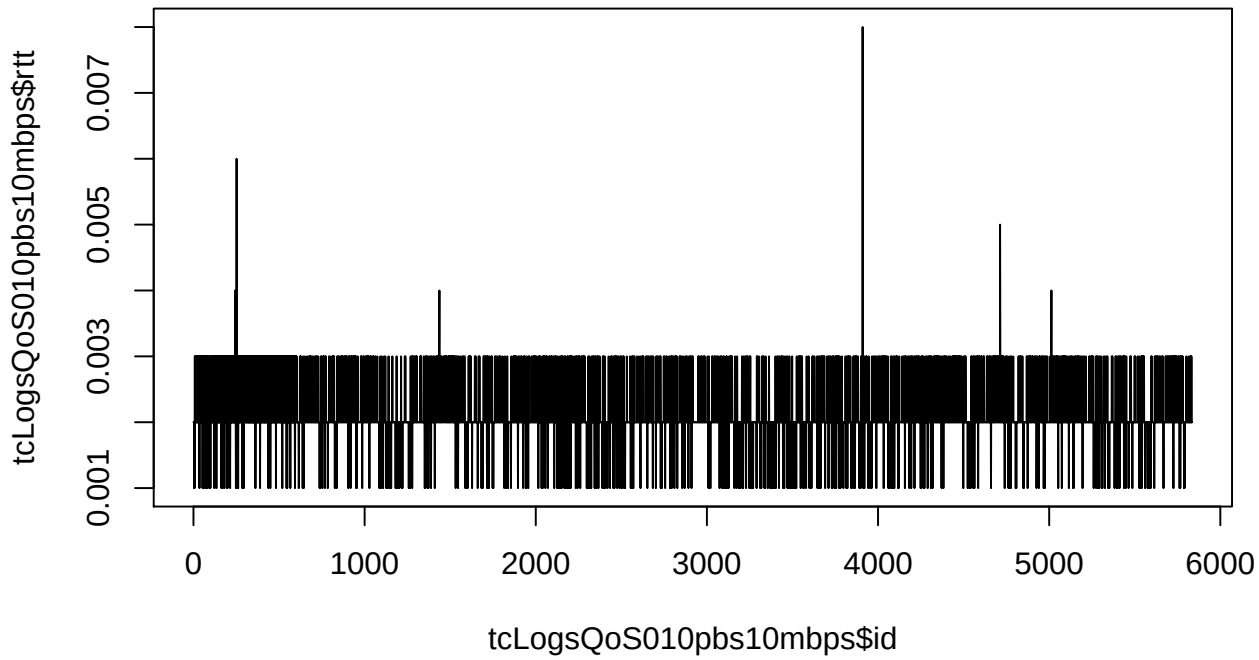

```
plot(tcLogsQoS010pbs100kbps$id, tcLogsQoS010pbs100kbps$rtt, type = "l")
```




```
plot(tcLogsQoS010pbs1mbps$id, tcLogsQoS010pbs1mbps$rtt, type = "l")
```



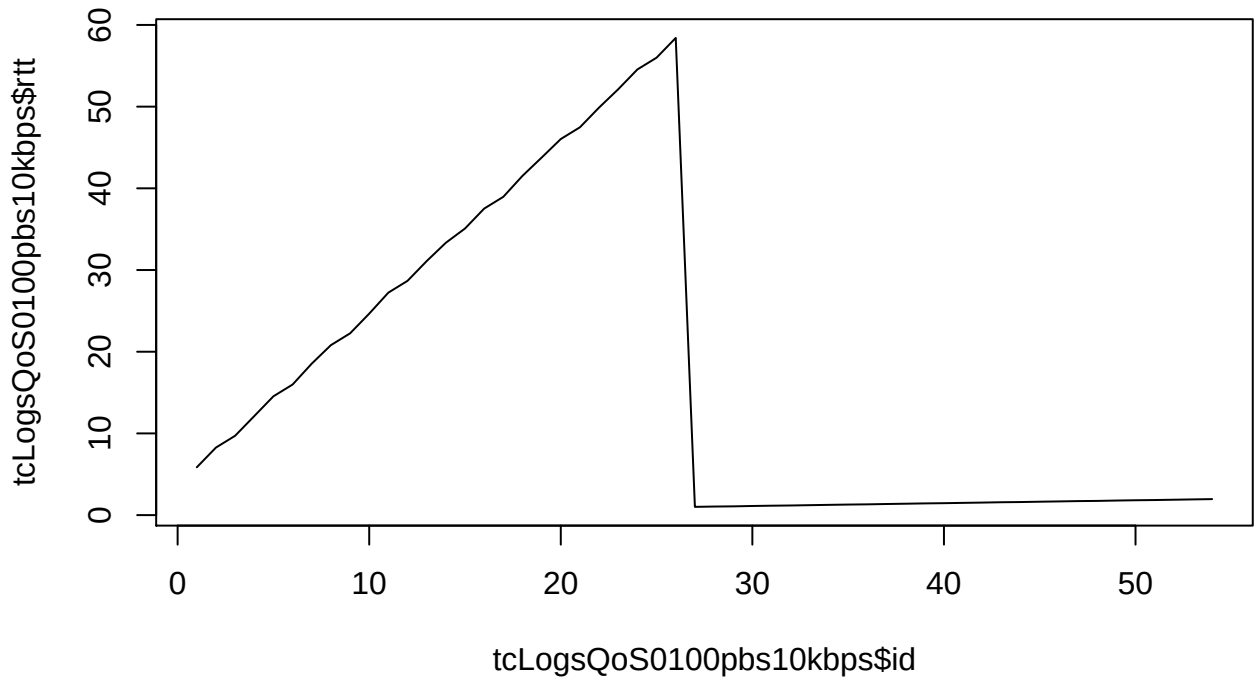

```
plot(tcLogsQoS010pbs10mbps$id, tcLogsQoS010pbs10mbps$rtt, type = "l")
```



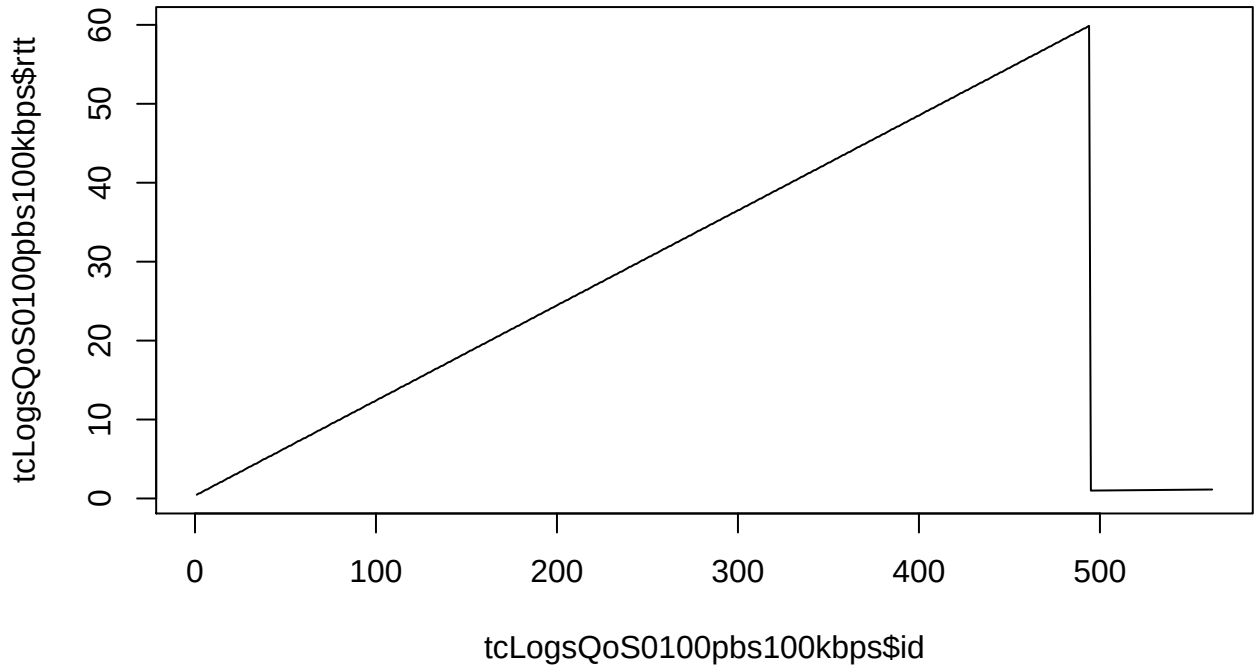

```
#plot(tcLogsQoS010pbs100mbps$id, tcLogsQoS010pbs100mbps$rtt, type = "l")
```

```
## QoS0_100pbs - Aufsplittung MaxDurc
tcLogsQoS0100pbs10kbps<-tcLogsQoS0100pbs[tcLogsQoS0100pbs$MaxDurc == "10kbps",]
tcLogsQoS0100pbs100kbps<-tcLogsQoS0100pbs[tcLogsQoS0100pbs$MaxDurc == "100kbps",]
tcLogsQoS0100pbs1mbps<-tcLogsQoS0100pbs[tcLogsQoS0100pbs$MaxDurc == "1mbps",]
#tcLogsQoS0100pbs10mbps<-tcLogsQoS0100pbs[tcLogsQoS0100pbs$MaxDurc == "10mbps",]
#tcLogsQoS0100pbs100mbps<-tcLogsQoS0100pbs[tcLogsQoS0100pbs$MaxDurc == "100mbps",]

#par(mfrow=c(1,3))
```

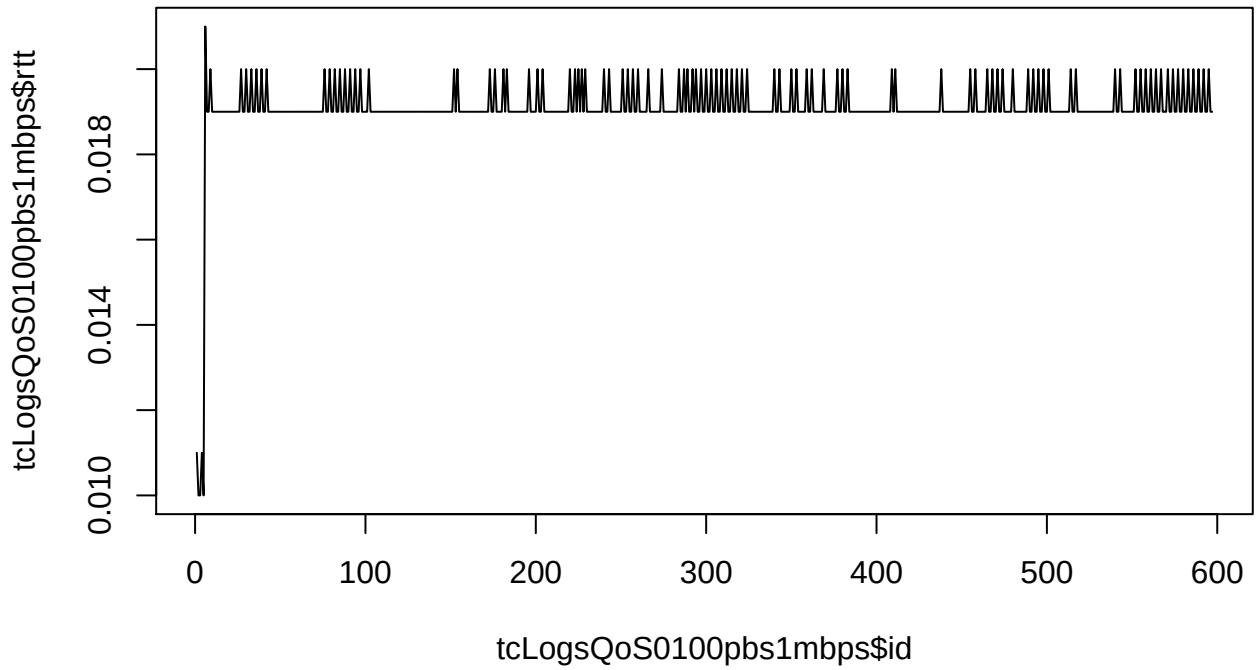
```r
plot(tcLogsQoS0100pbs10kbps$id, tcLogsQoS0100pbs10kbps$rtt, type = "l")
```




```r
plot(tcLogsQoS0100pbs100kbps$id, tcLogsQoS0100pbs100kbps$rtt, type = "l")
```

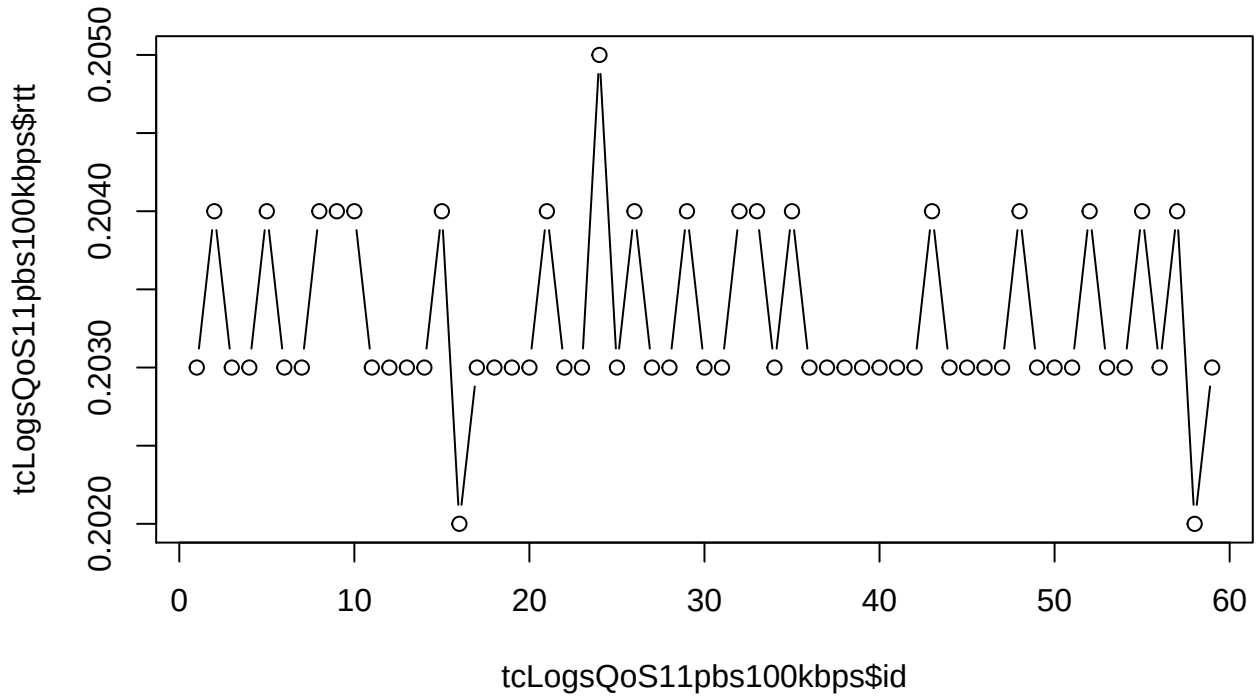



```r
plot(tcLogsQoS0100pbs1mbps$id, tcLogsQoS0100pbs1mbps$rtt, type = "l")
```

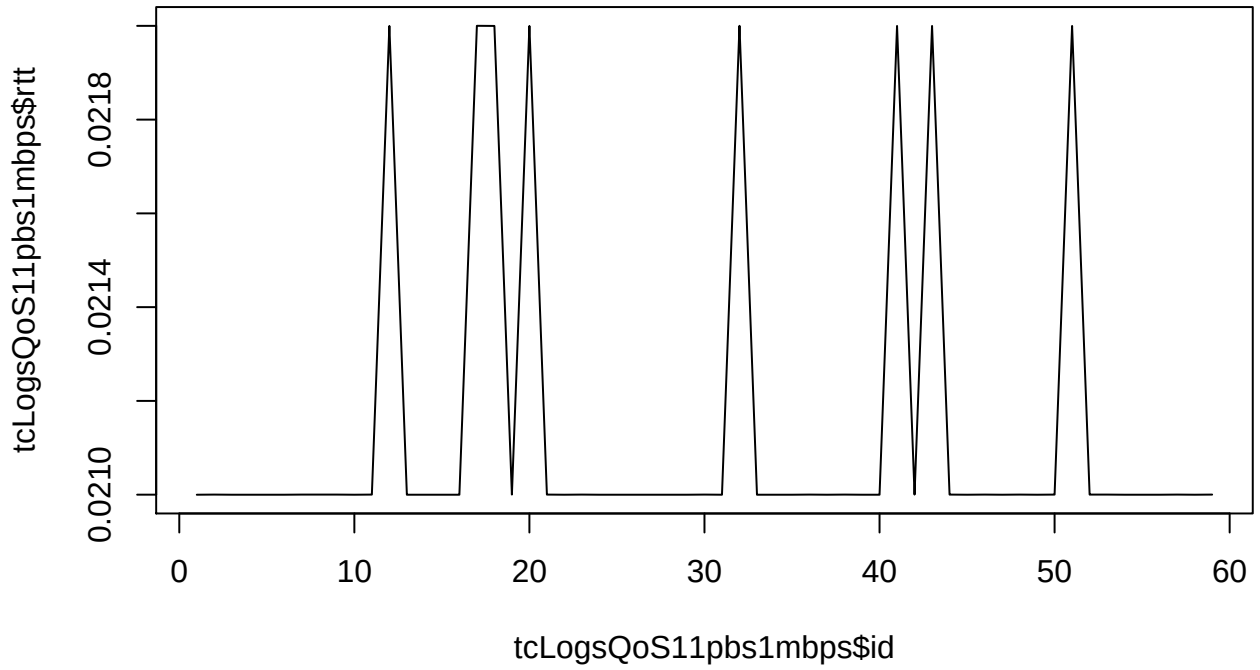



```
#plot(tcLogsQoS0100pbs10mbps$id, tcLogsQoS0100pbs10mbps$rtt, type = "l")
#plot(tcLogsQoS0100pbs100mbps$id, tcLogsQoS0100pbs100mbps$rtt, type = "l")
```

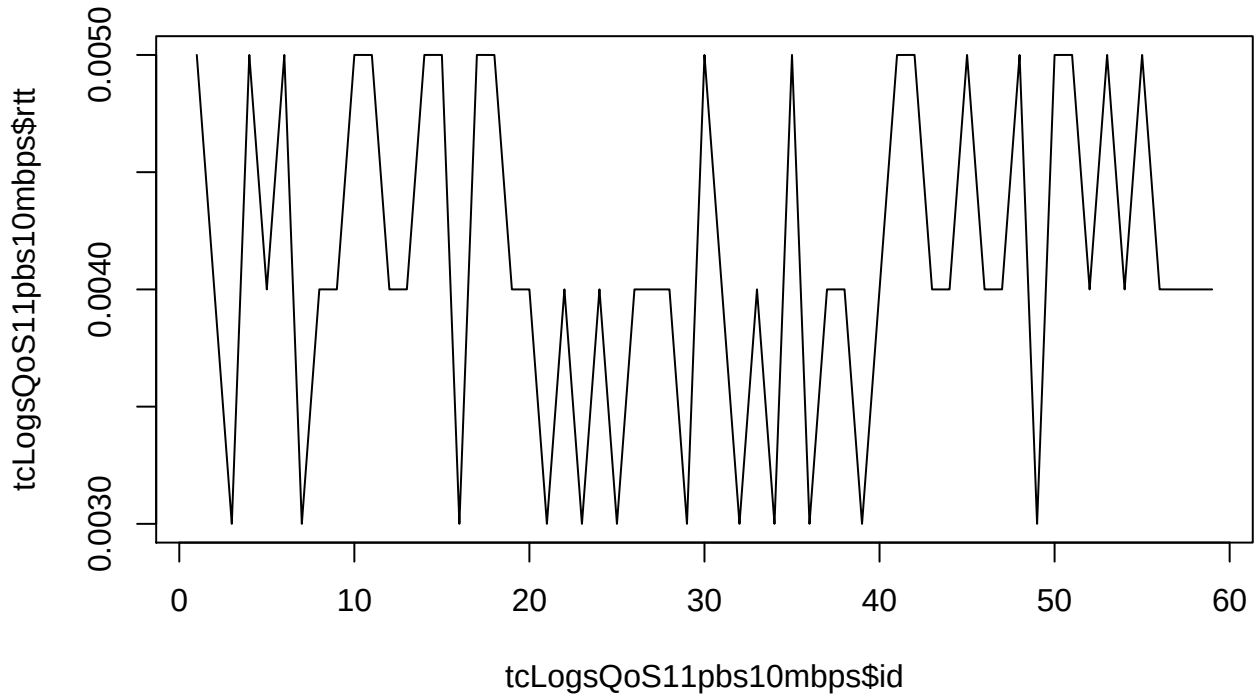
```
## QoS1_1pbs - Aufsplittung MaxDurc
tcLogsQoS11pbs10kbps<-tcLogsQoS11pbs[tcLogsQoS11pbs$MaxDurc == "10kbps",]
tcLogsQoS11pbs100kbps<-tcLogsQoS11pbs[tcLogsQoS11pbs$MaxDurc == "100kbps",]
tcLogsQoS11pbs1mbps<-tcLogsQoS11pbs[tcLogsQoS11pbs$MaxDurc == "1mbps",]
tcLogsQoS11pbs10mbps<-tcLogsQoS11pbs[tcLogsQoS11pbs$MaxDurc == "10mbps",]
tcLogsQoS11pbs100mbps<-tcLogsQoS11pbs[tcLogsQoS11pbs$MaxDurc == "100mbps",]

#plot(tcLogsQoS11pbs10kbps$id, tcLogsQoS11pbs10kbps$rtt, type = "l")
plot(tcLogsQoS11pbs100kbps$id, tcLogsQoS11pbs100kbps$rtt, type = "b")
```



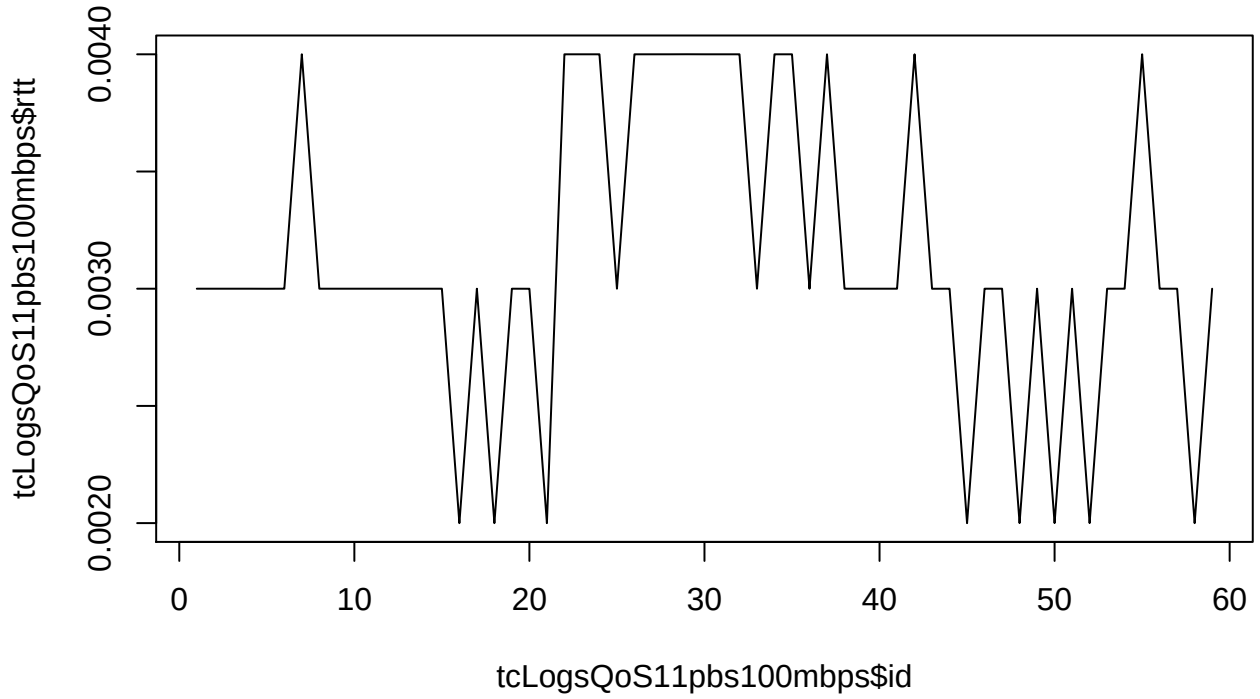

```
plot(tcLogsQoS11pbs1mbps$id, tcLogsQoS11pbs1mbps$rtt, type = "l")
```




```
plot(tcLogsQoS11pbs10mbps$id, tcLogsQoS11pbs10mbps$rtt, type = "l")
```

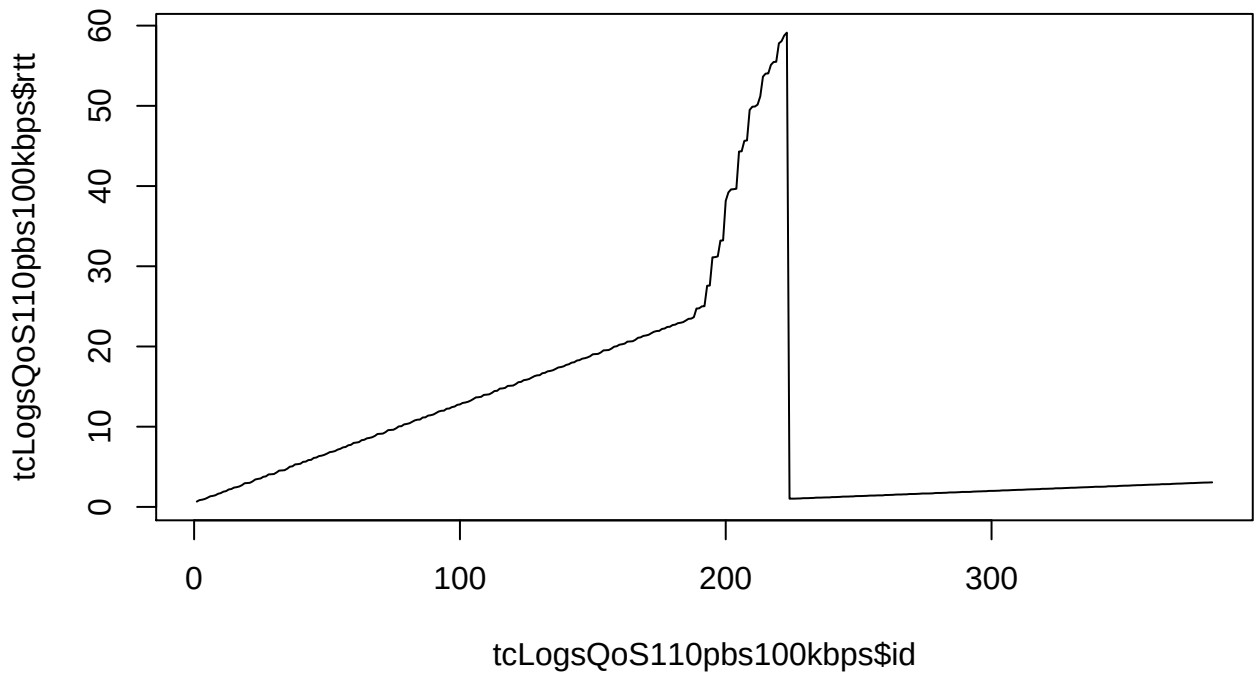



```
plot(tcLogsQoS11pbs100mbps$id, tcLogsQoS11pbs100mbps$rtt, type = "l")
```

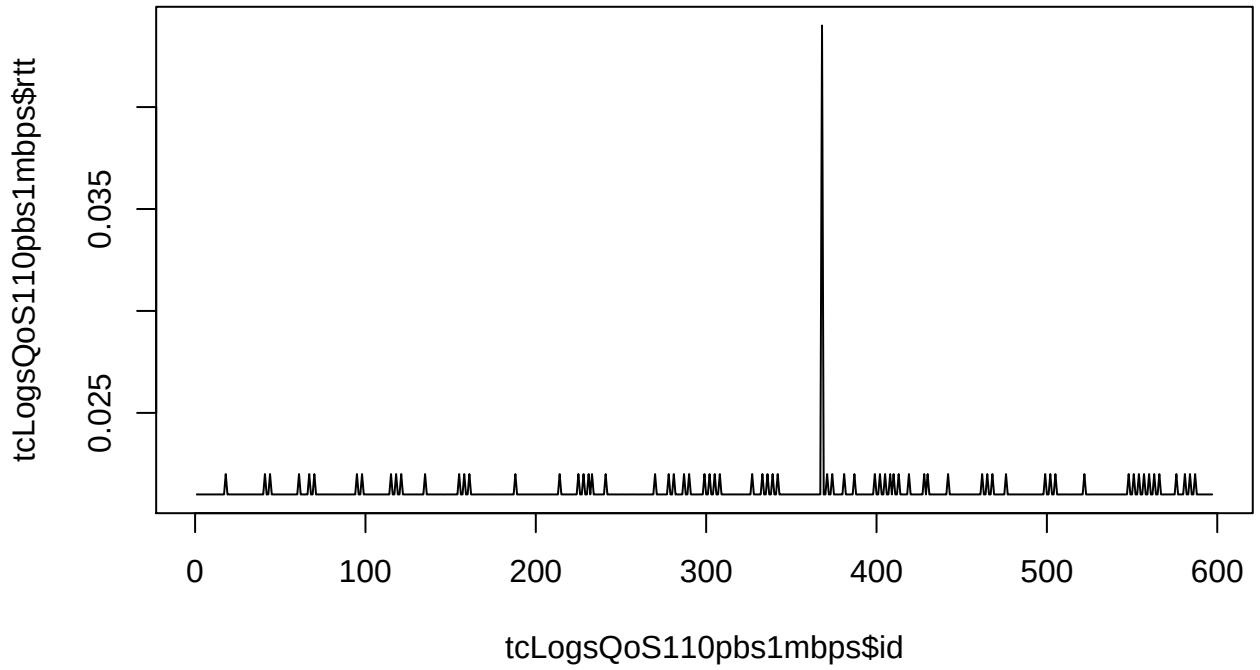



```
## QoS1_10pbs - Aufsplittung MaxDurc
tcLogsQoS110pbs10kbps<-tcLogsQoS110pbs[tcLogsQoS110pbs$MaxDurc == "10kbps",]
tcLogsQoS110pbs100kbps<-tcLogsQoS110pbs[tcLogsQoS110pbs$MaxDurc == "100kbps",]
tcLogsQoS110pbs1mbps<-tcLogsQoS110pbs[tcLogsQoS110pbs$MaxDurc == "1mbps",]
tcLogsQoS110pbs10mbps<-tcLogsQoS110pbs[tcLogsQoS110pbs$MaxDurc == "10mbps",]
tcLogsQoS110pbs100mbps<-tcLogsQoS110pbs[tcLogsQoS110pbs$MaxDurc == "100mbps",]

#plot(tcLogsQoS110pbs10kbps$id, tcLogsQoS110pbs10kbps$rtt, type = "l")
```

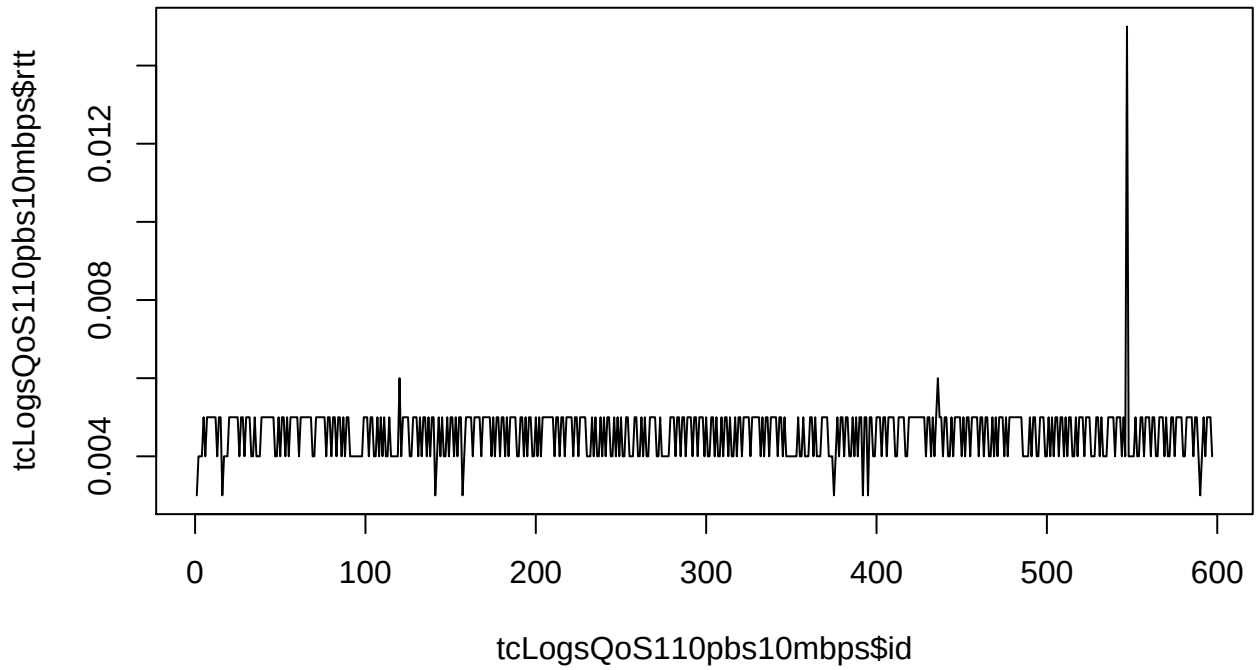
```
plot(tcLogsQoS110pbs100kbps$id, tcLogsQoS110pbs100kbps$rtt, type = "l")
```



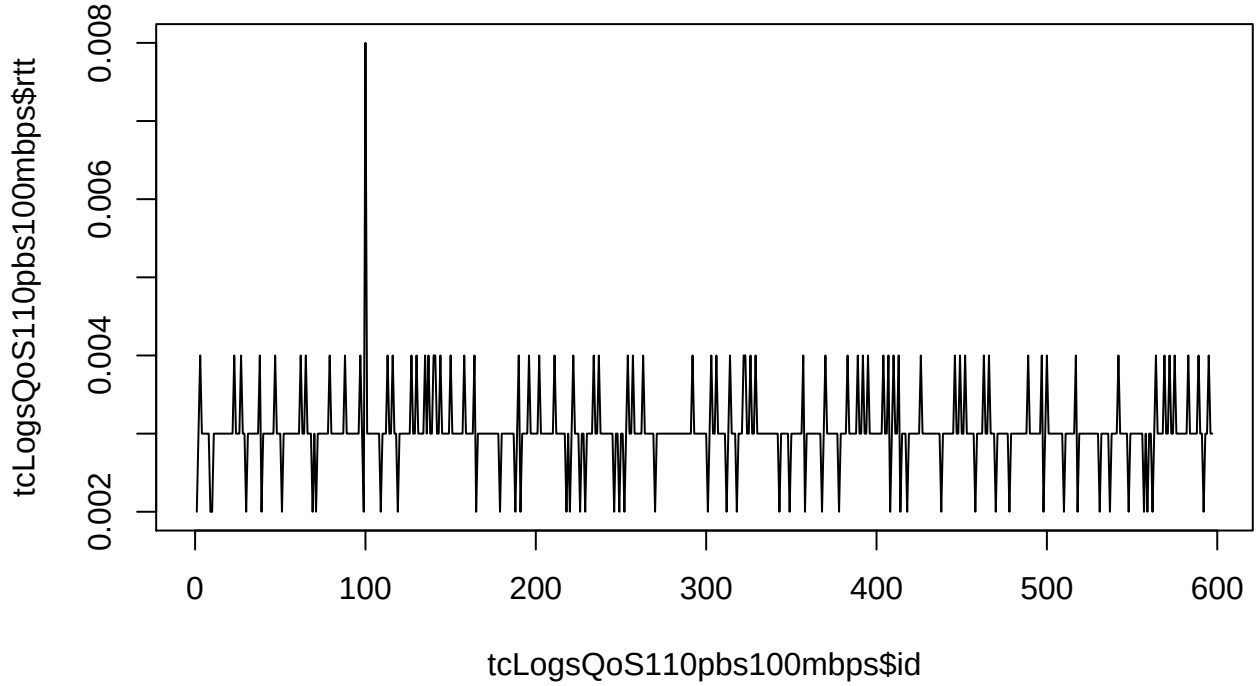

```
plot(tcLogsQoS110pbs1mbps$id, tcLogsQoS110pbs1mbps$rtt, type = "l")
```



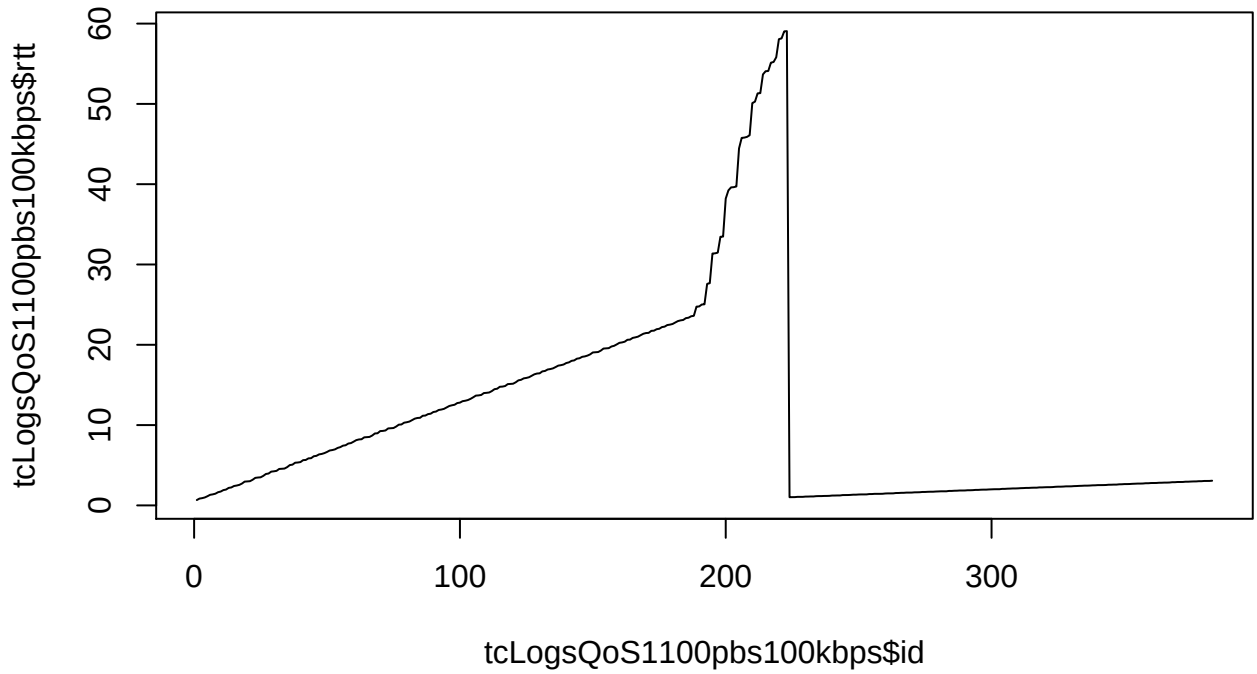

```
plot(tcLogsQoS110pbs10mbps$id, tcLogsQoS110pbs10mbps$rtt, type = "l")
```



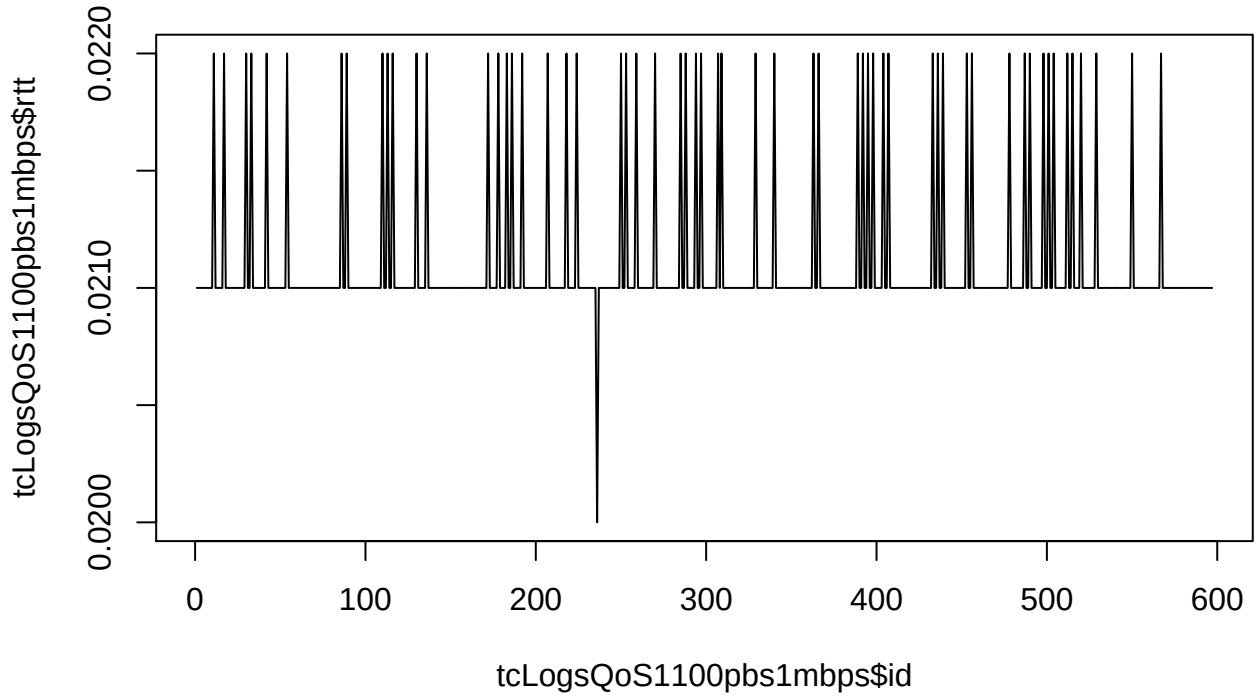

```
plot(tcLogsQoS110pbs100mbps$id, tcLogsQoS110pbs100mbps$rtt, type = "l")
```



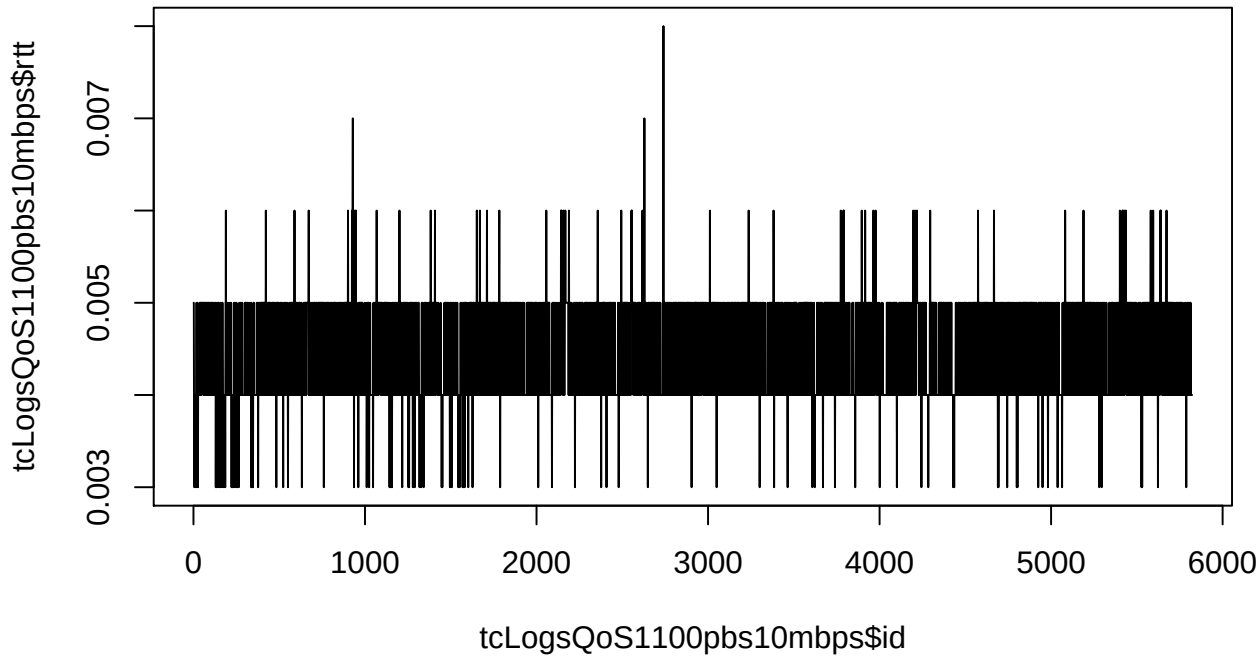

```
## QoS1_100pbs - Aufsplittung MaxDurc
tcLogsQoS1100pbs10kbps<-tcLogsQoS1100pbs[tcLogsQoS1100pbs$MaxDurc == "10kbps",]
tcLogsQoS1100pbs100kbps<-tcLogsQoS1100pbs[tcLogsQoS1100pbs$MaxDurc == "100kbps",]
tcLogsQoS1100pbs1mbps<-tcLogsQoS1100pbs[tcLogsQoS1100pbs$MaxDurc == "1mbps",]
tcLogsQoS1100pbs10mbps<-tcLogsQoS1100pbs[tcLogsQoS1100pbs$MaxDurc == "10mbps",]
tcLogsQoS1100pbs100mbps<-tcLogsQoS1100pbs[tcLogsQoS1100pbs$MaxDurc == "100mbps",]

#plot(tcLogsQoS1100pbs10kbps$id, tcLogsQoS1100pbs10kbps$rtt, type = "l")
plot(tcLogsQoS1100pbs100kbps$id, tcLogsQoS1100pbs100kbps$rtt, type = "l")
```



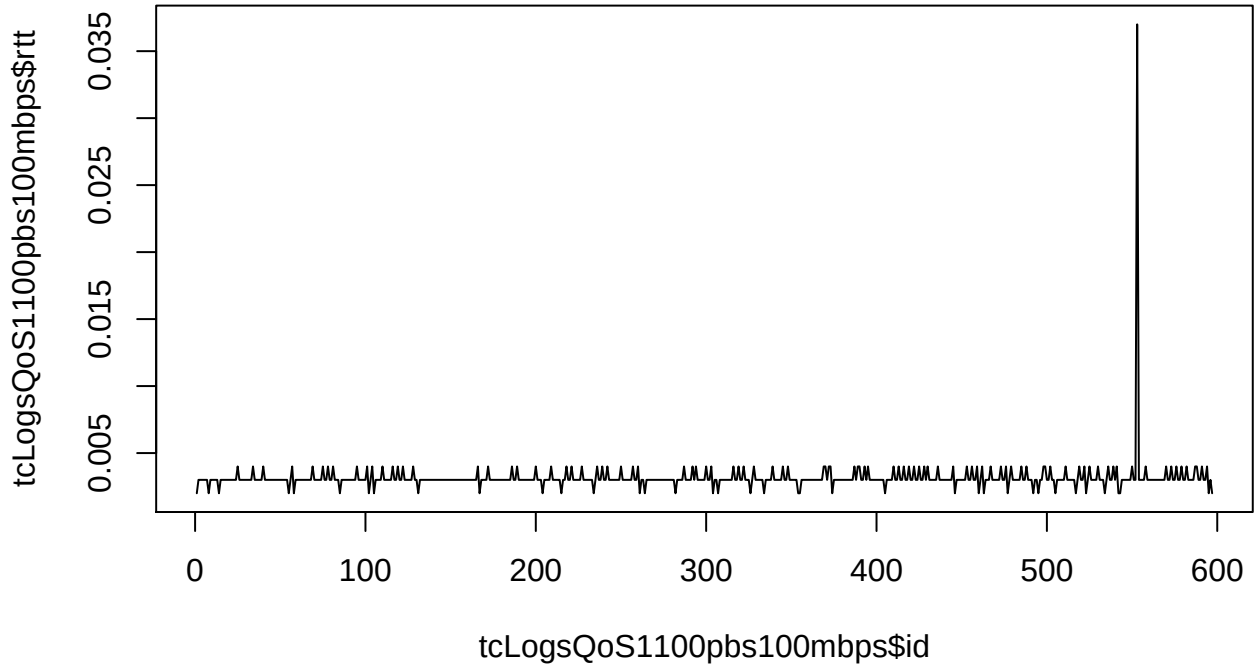

```
plot(tcLogsQoS1100pbs1mbps$id, tcLogsQoS1100pbs1mbps$rtt, type = "l")
```



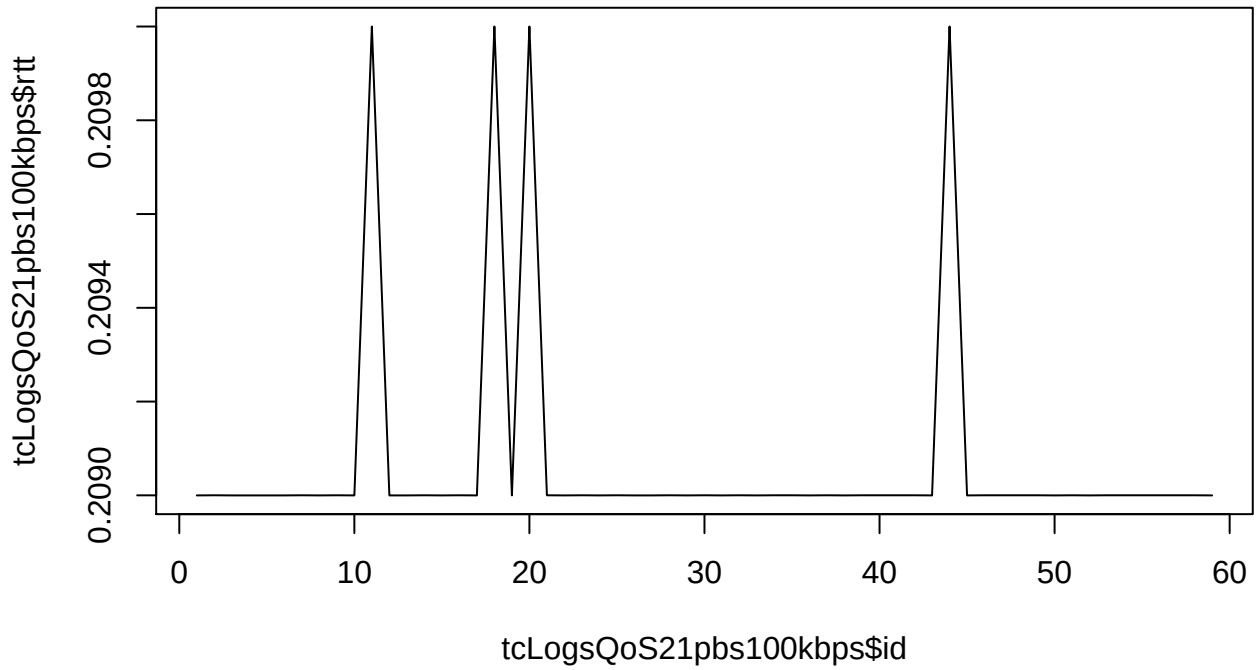

```
plot(tcLogsQoS1100pbs10mbps$id, tcLogsQoS1100pbs10mbps$rtt, type = "l")
```



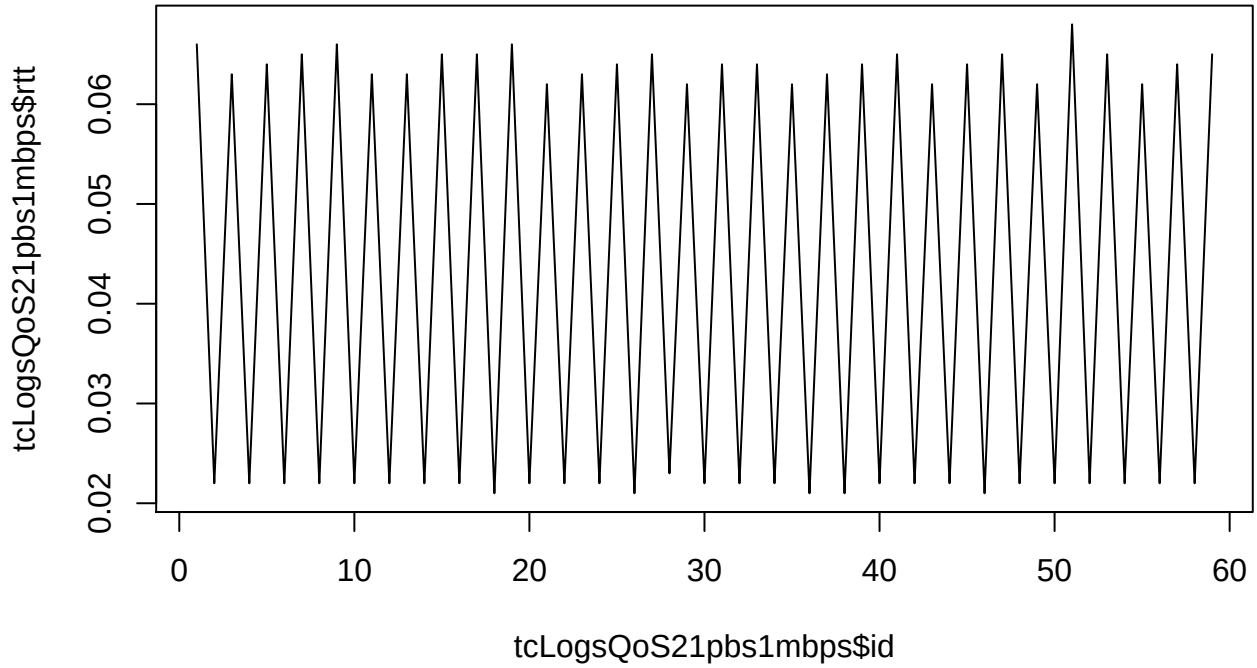

```
plot(tcLogsQoS1100pbs100mbps$id, tcLogsQoS1100pbs100mbps$rtt, type = "l")
```



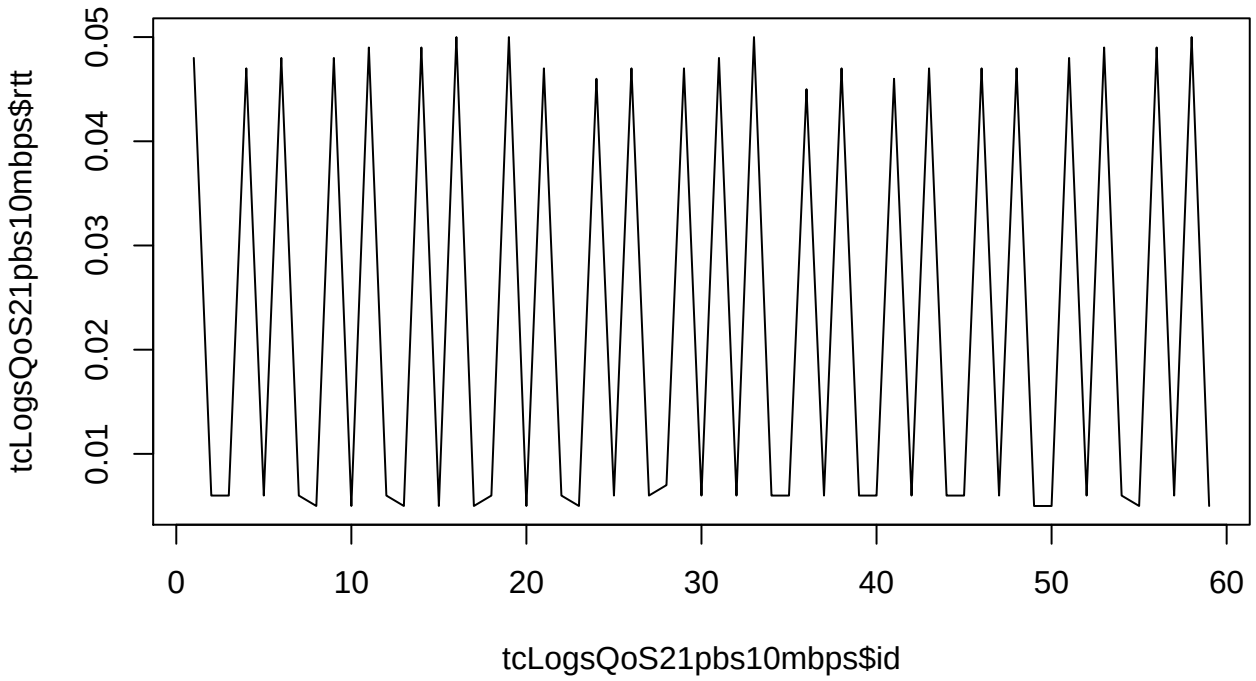

```
## QoS2_1pbs - Aufsplittung MaxDurc
tcLogsQoS21pbs10kbps<-tcLogsQoS21pbs[tcLogsQoS21pbs$MaxDurc == "10kbps",]
tcLogsQoS21pbs100kbps<-tcLogsQoS21pbs[tcLogsQoS21pbs$MaxDurc == "100kbps",]
tcLogsQoS21pbs1mbps<-tcLogsQoS21pbs[tcLogsQoS21pbs$MaxDurc == "1mbps",]
tcLogsQoS21pbs10mbps<-tcLogsQoS21pbs[tcLogsQoS21pbs$MaxDurc == "10mbps",]
tcLogsQoS21pbs100mbps<-tcLogsQoS21pbs[tcLogsQoS21pbs$MaxDurc == "100mbps",]

#plot(tcLogsQoS21pbs10kbps$id, tcLogsQoS21pbs10kbps$rtt, type = "l")
plot(tcLogsQoS21pbs100kbps$id, tcLogsQoS21pbs100kbps$rtt, type = "l")
```



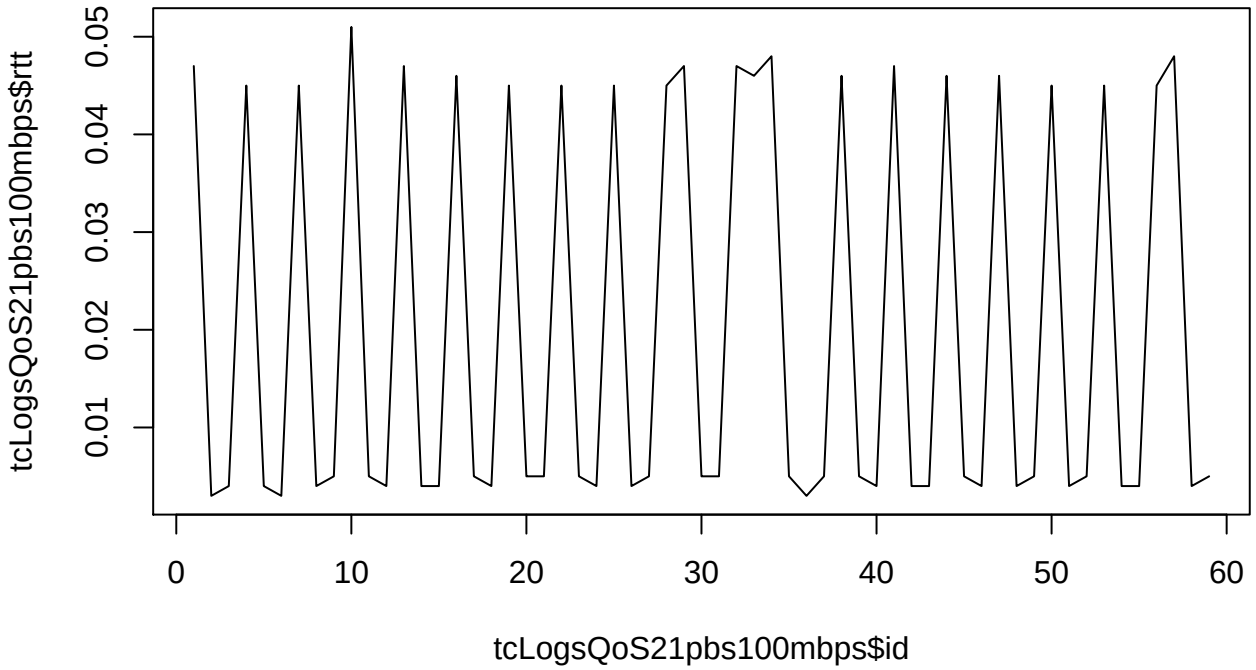

```
plot(tcLogsQoS21pbs1mbps$id, tcLogsQoS21pbs1mbps$rtt, type = "l")
```



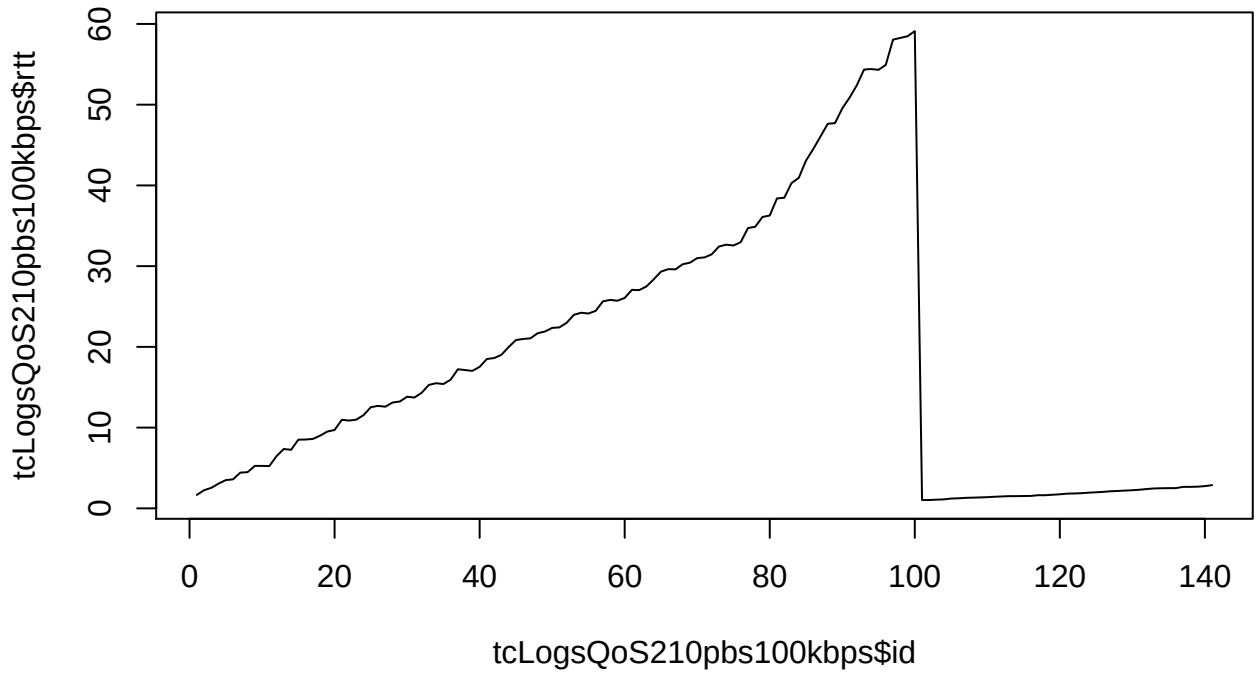

```
plot(tcLogsQoS21pbs10mbps$id, tcLogsQoS21pbs10mbps$rtt, type = "l")
```



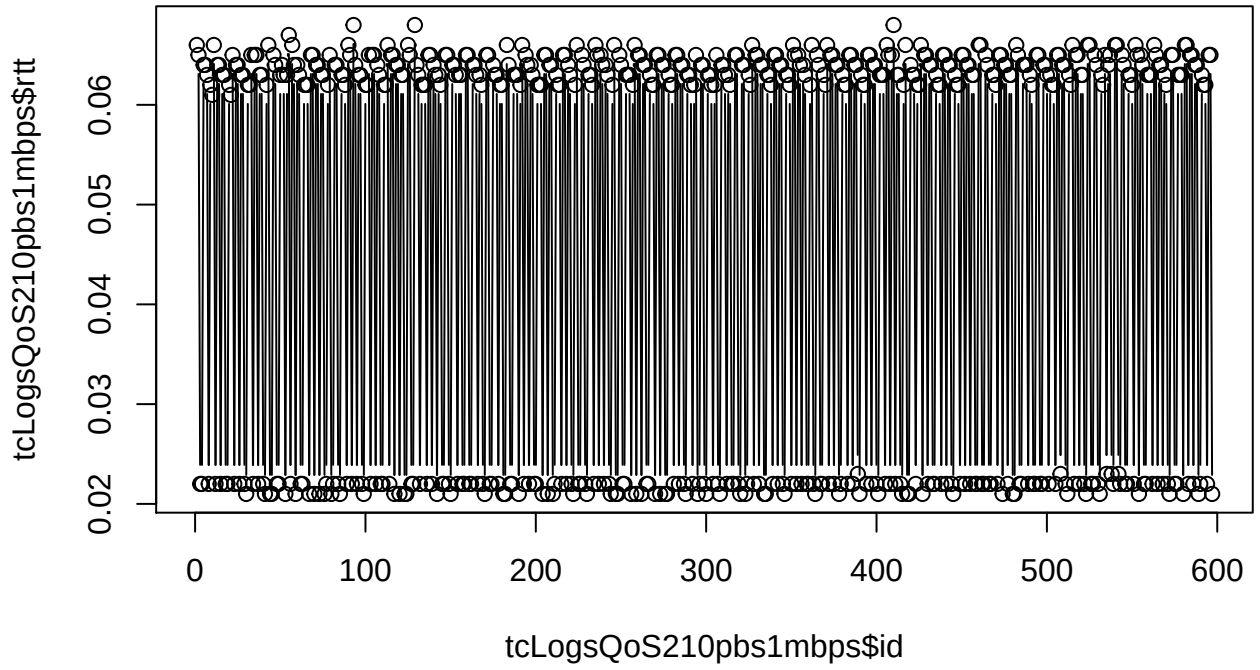

```
plot(tcLogsQoS21pbs100mbps$id, tcLogsQoS21pbs100mbps$rtt, type = "l")
```



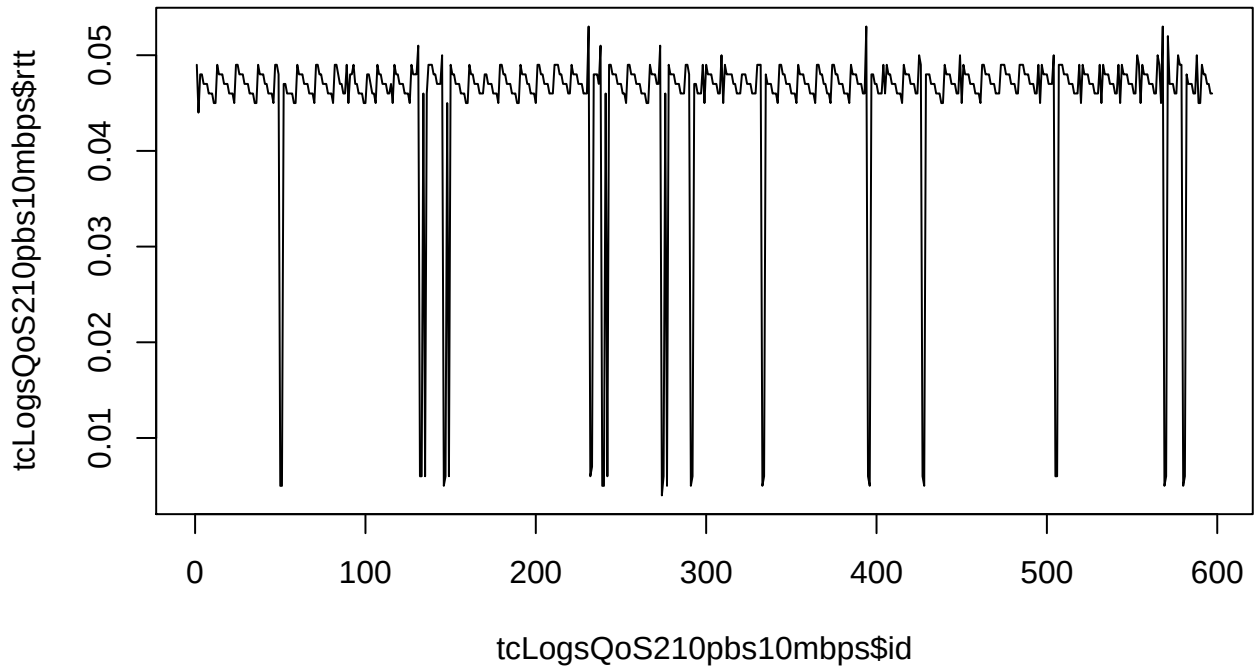

```
## QoS2_10pbs - Aufsplittung MaxDurc
tcLogsQoS210pbs10kbps<-tcLogsQoS210pbs[tcLogsQoS210pbs$MaxDurc == "10kbps",]
tcLogsQoS210pbs100kbps<-tcLogsQoS210pbs[tcLogsQoS210pbs$MaxDurc == "100kbps",]
tcLogsQoS210pbs1mbps<-tcLogsQoS210pbs[tcLogsQoS210pbs$MaxDurc == "1mbps",]
tcLogsQoS210pbs10mbps<-tcLogsQoS210pbs[tcLogsQoS210pbs$MaxDurc == "10mbps",]
tcLogsQoS210pbs100mbps<-tcLogsQoS210pbs[tcLogsQoS210pbs$MaxDurc == "100mbps",]

#plot(tcLogsQoS210pbs10kbps$id, tcLogsQoS210pbs10kbps$rtt, type = "l")
plot(tcLogsQoS210pbs100kbps$id, tcLogsQoS210pbs100kbps$rtt, type = "l")
```



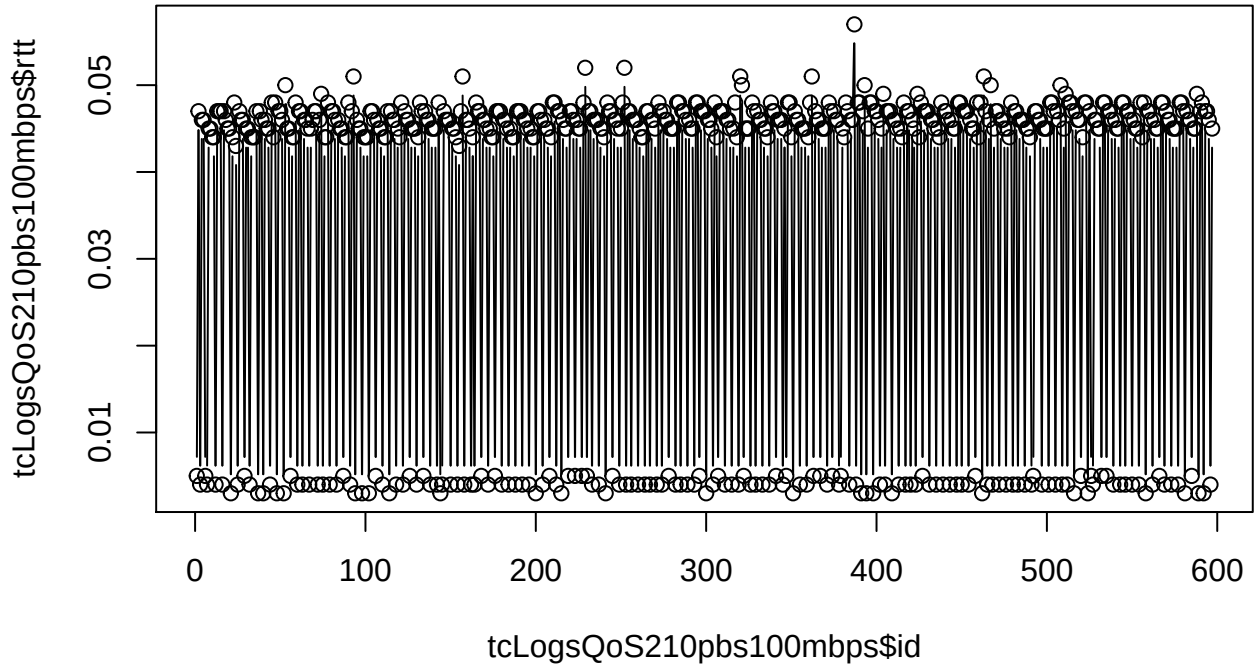

```
plot(tcLogsQoS210pbs1mbps$id, tcLogsQoS210pbs1mbps$rtt, type = "b")
```



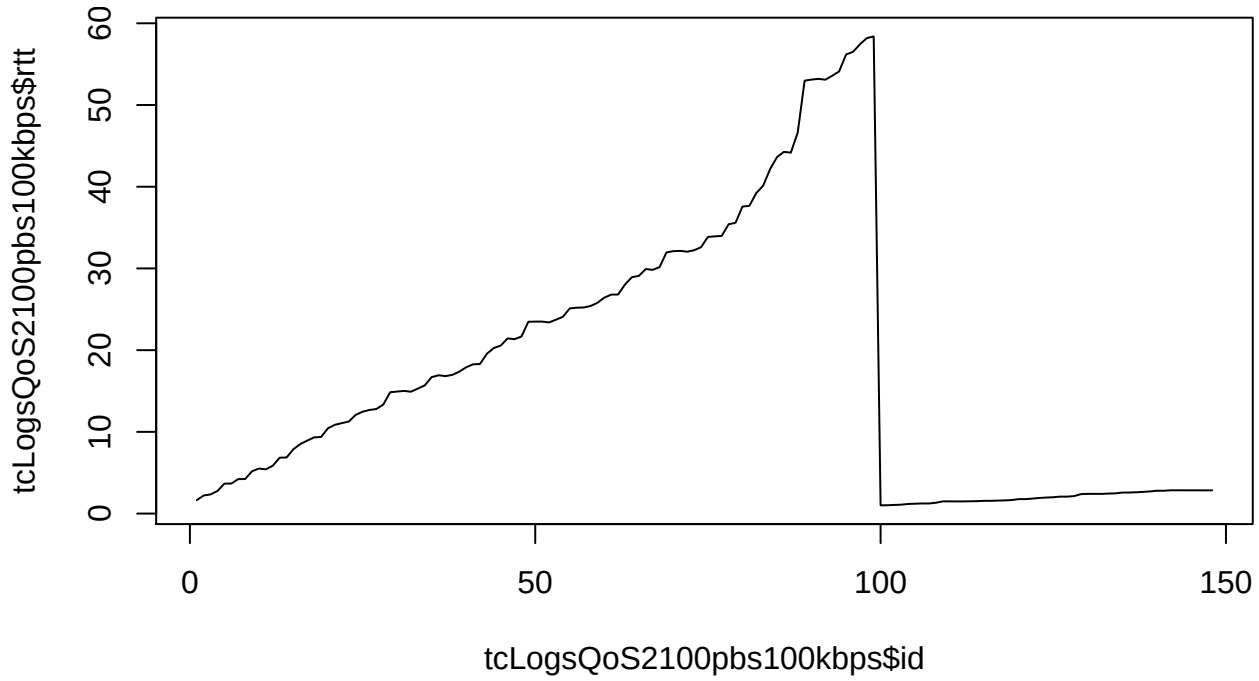

```
plot(tcLogsQoS210pbs10mbps$id, tcLogsQoS210pbs10mbps$rtt, type = "l")
```



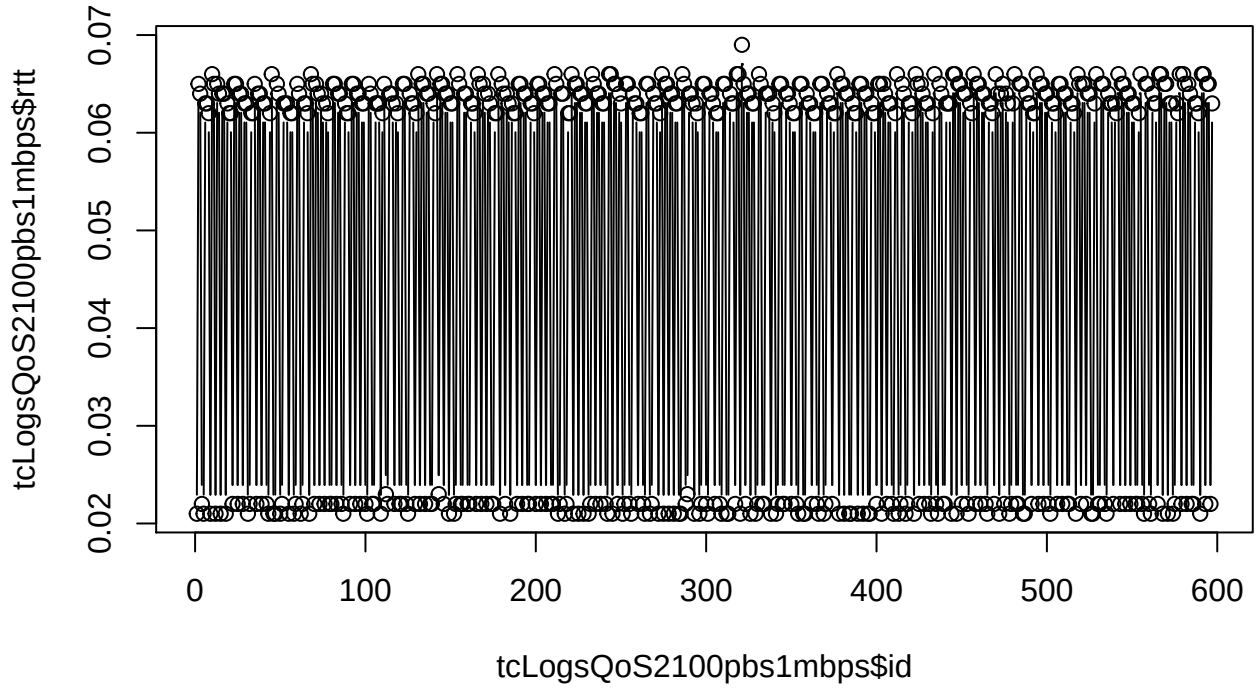

```
plot(tcLogsQoS210pbs100mbps$id, tcLogsQoS210pbs100mbps$rtt, type = "b")
```



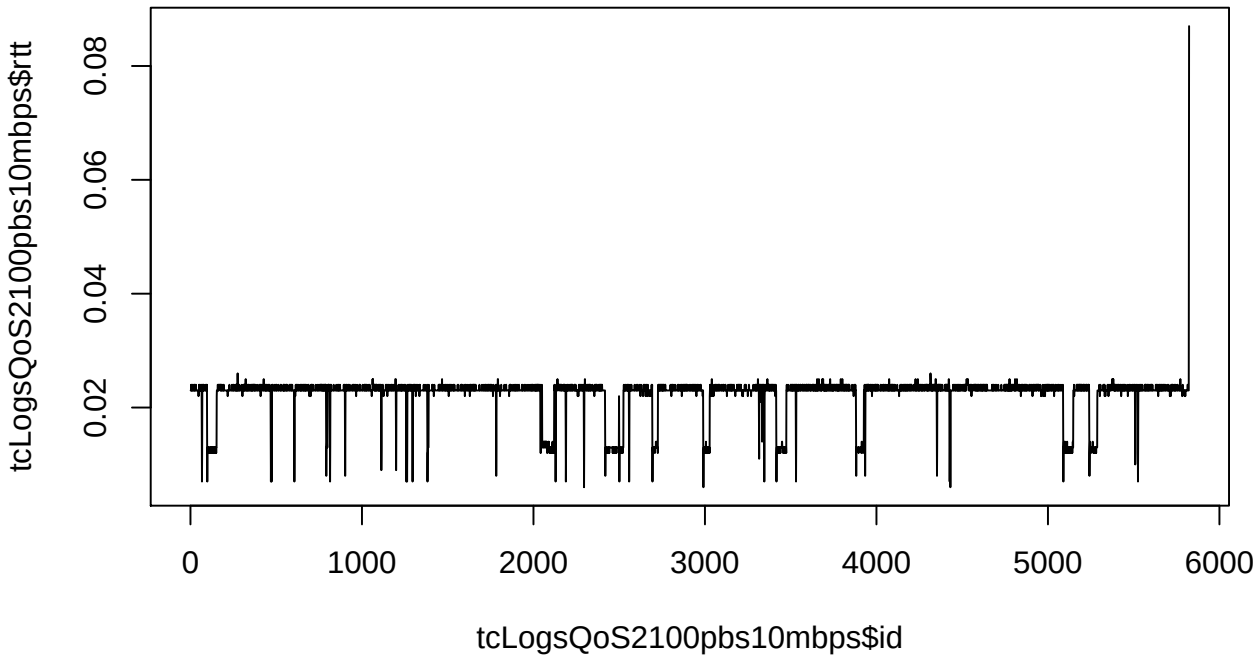

```
## QoS2_100pbs - Aufsplittung MaxDurc
tcLogsQoS2100pbs10kbps<-tcLogsQoS2100pbs[tcLogsQoS2100pbs$MaxDurc == "10kbps",]
tcLogsQoS2100pbs100kbps<-tcLogsQoS2100pbs[tcLogsQoS2100pbs$MaxDurc == "100kbps",]
tcLogsQoS2100pbs1mbps<-tcLogsQoS2100pbs[tcLogsQoS2100pbs$MaxDurc == "1mbps",]
tcLogsQoS2100pbs10mbps<-tcLogsQoS2100pbs[tcLogsQoS2100pbs$MaxDurc == "10mbps",]
tcLogsQoS2100pbs100mbps<-tcLogsQoS2100pbs[tcLogsQoS2100pbs$MaxDurc == "100mbps",]

#plot(tcLogsQoS2100pbs10kbps$id, tcLogsQoS2100pbs10kbps$rtt, type = "l")
plot(tcLogsQoS2100pbs100kbps$id, tcLogsQoS2100pbs100kbps$rtt, type = "l")
```




```
plot(tcLogsQoS2100pbs1mbps$id, tcLogsQoS2100pbs1mbps$rtt, type = "b")
```




```
plot(tcLogsQoS2100pbs10mbps$id, tcLogsQoS2100pbs10mbps$rtt, type = "l")
```




```
plot(tcLogsQoS2100pbs100mbps$id, tcLogsQoS2100pbs100mbps$rtt, type = "b")
```