

Anhang 3

Graphische Darstellung der TC Messungen

```
setwd("/home/lisa/Darmstadt/05_Speicher und Datennetze IoT/Praktikum/Git/mqtt-qos-roundtrip/R_Analysis/01")
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("./latenzTC1mbps.Rda")
load("./latenzTc10kbps.Rda")
load("./latenzTc10mbps.Rda")
load("./latenzTc100kbps.Rda")
load("./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`),]

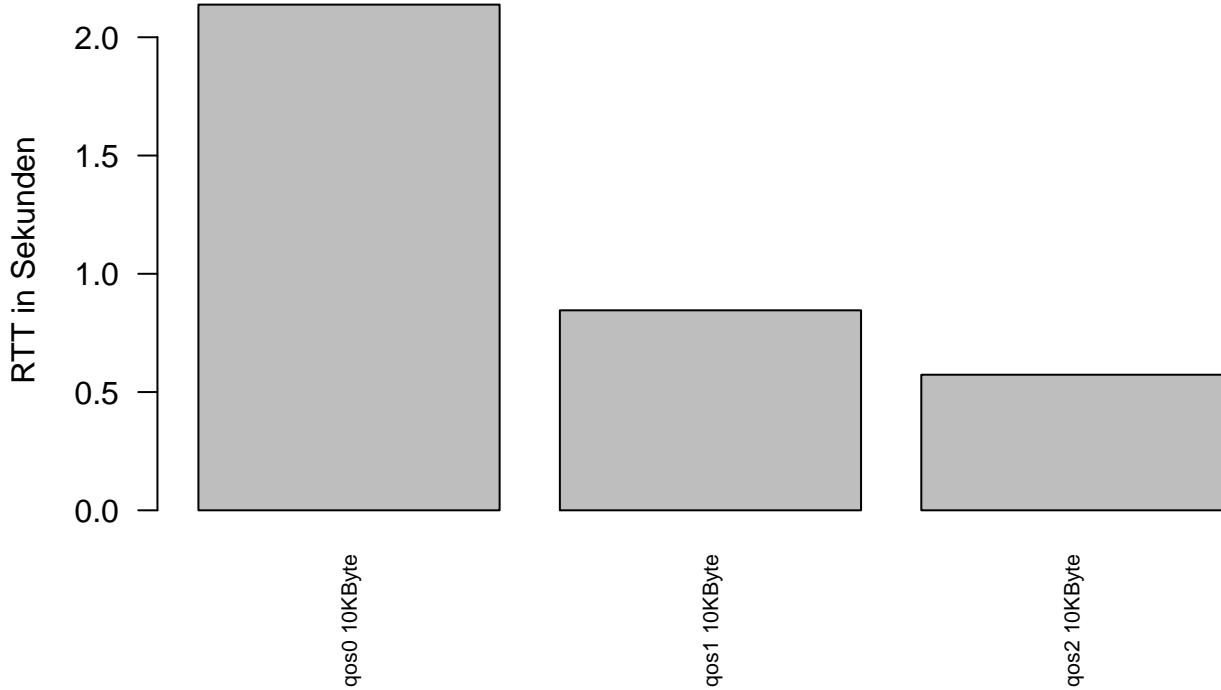
tcLogsAgg2 %>%
  kable() %>%
  kable_styling()

```

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

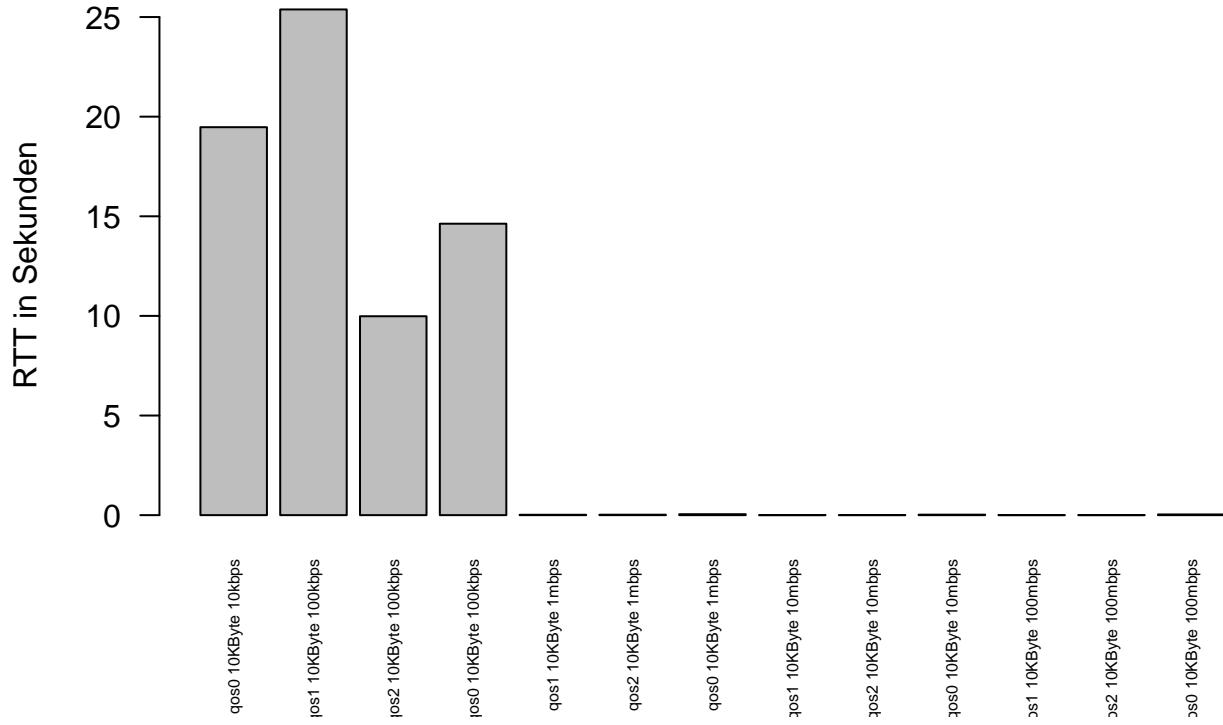
```
tcLogsAgg$Names <- paste(tcLogsAgg$`tcLogs$QoS`, tcLogsAgg$`tcLogs$Size`)
tcLogsAgg<-tcLogsAgg[order(tcLogsAgg$`tcLogs$Byte`),]
barplot(tcLogsAgg$`tcLogs$rtt`, main = "Latenz TC nach QoS und Paketgröße", ylab = "RTT in Sekunden", a
```

Latenz TC nach QoS und Paketgröße



```
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 :)
```

Latenz TC nach Max Durchsatz QoS und Paketgröße



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

```

reg_tcLogs <- lm(tcLogs$rtt ~ tcLogs$QoS + tcLogs$ByteD, data = tcLogs)
summary(reg_tcLogs)
#>
## Call:
## lm(formula = tcLogs$rtt ~ tcLogs$QoS + tcLogs$ByteD, data = tcLogs)
#>
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -3.689 -1.468 -0.926 -0.254 57.826 
#>
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 3.734e+00 6.867e-02 54.37 <2e-16 ***
## tcLogs$QoSqos1 -2.210e+00 8.977e-02 -24.62 <2e-16 ***
## tcLogs$QoSqos2 -2.448e+00 9.081e-02 -26.95 <2e-16 ***
## tcLogs$ByteD -3.477e-08 9.341e-10 -37.23 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
## Residual standard error: 6.697 on 34718 degrees of freedom
## Multiple R-squared:  0.04885, Adjusted R-squared:  0.04876 
## F-statistic: 594.3 on 3 and 34718 DF, p-value: < 2.2e-16
reg_tcLogs2 <- lm(tcLogs$rtt ~ tcLogs$QoS + tcLogs$MaxDurc, data = tcLogs)
summary(reg_tcLogs2)
#>
## Call:

```

```

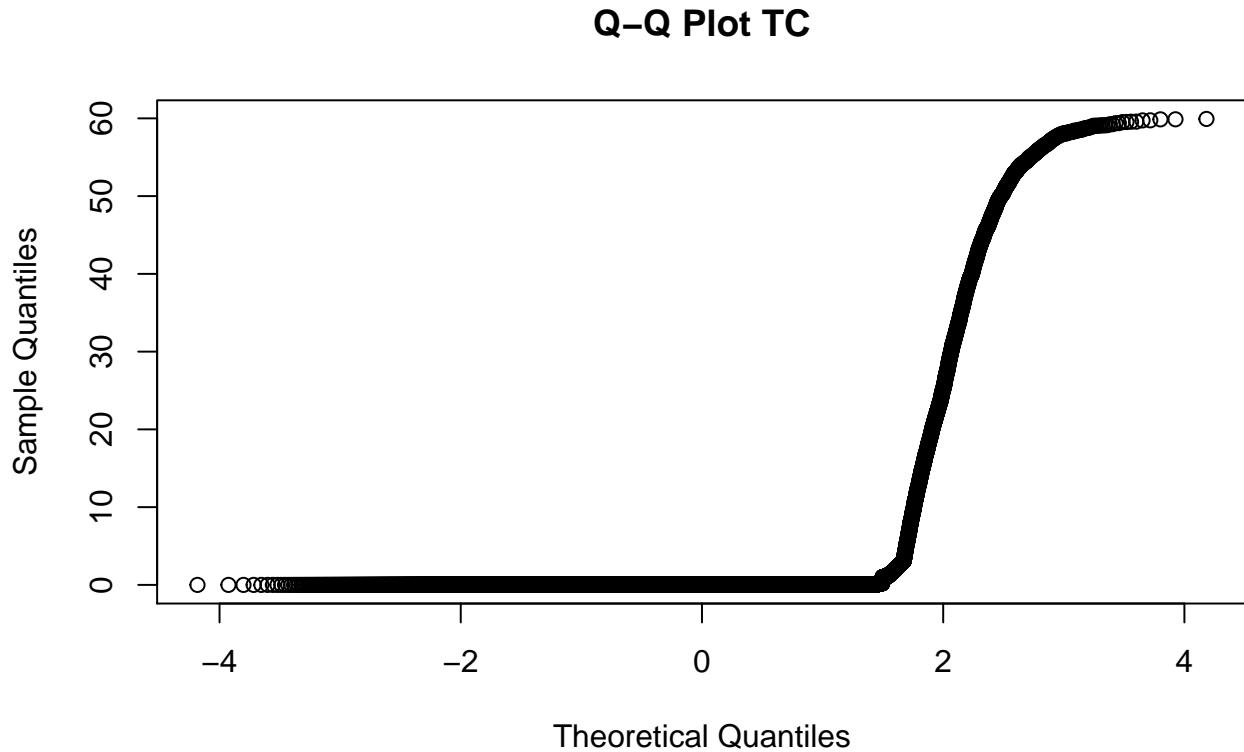
#> lm(formula = tcLogs$rtt ~ tcLogs$QoS + tcLogs$MaxDurc, data = tcLogs)
#>
#> Residuals:
#>      Min       1Q   Median      3Q      Max
#> -18.808  -0.721   0.077   0.643  41.494
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 18.99314   0.10474 181.340 <2e-16 ***
#> tcLogs$QoSqos1 -1.36387   0.06587 -20.705 <2e-16 ***
#> tcLogs$QoSqos2 -0.77823   0.06707 -11.604 <2e-16 ***
#> tcLogs$MaxDurc100mbps -18.68789   0.11402 -163.907 <2e-16 ***
#> tcLogs$MaxDurc10kbps  0.47799   0.39360   1.214   0.225
#> tcLogs$MaxDurc10mbps -18.26930   0.10731 -170.251 <2e-16 ***
#> tcLogs$MaxDurc1mbps  -18.24990   0.12901 -141.461 <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 4.888 on 34715 degrees of freedom
#> Multiple R-squared:  0.4933, Adjusted R-squared:  0.4932
#> F-statistic:  5633 on 6 and 34715 DF, p-value: < 2.2e-16

```

```

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

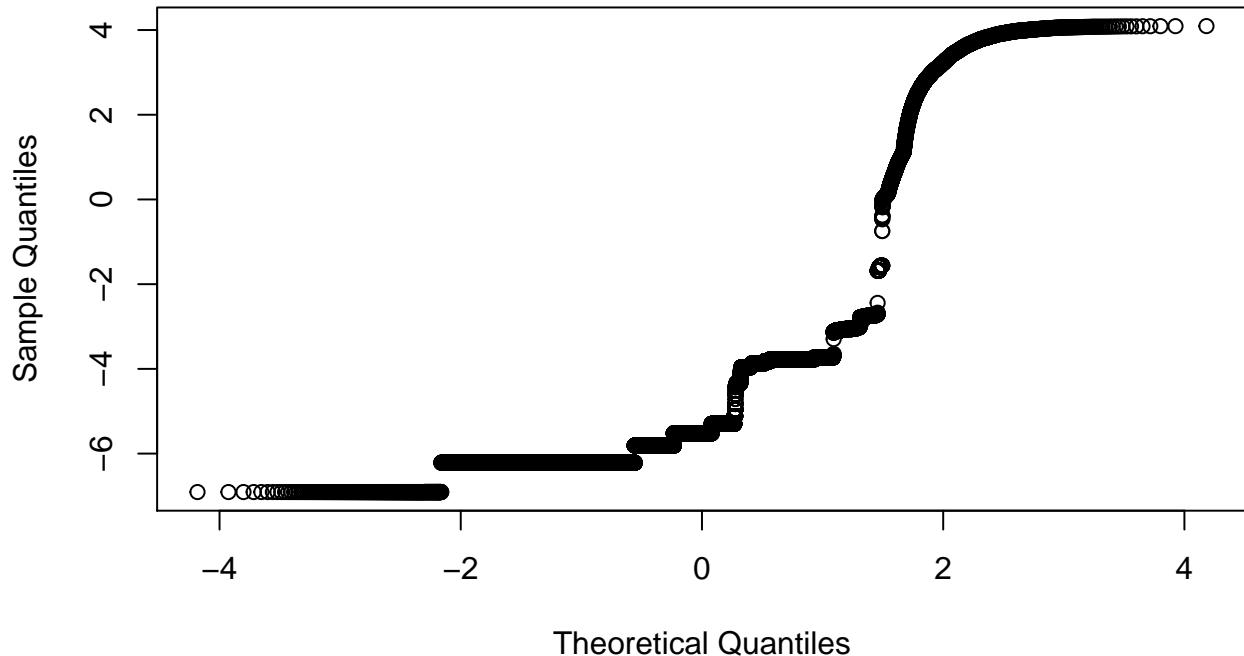
```



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 Beobachtungen also auf einer Geraden liegen)

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

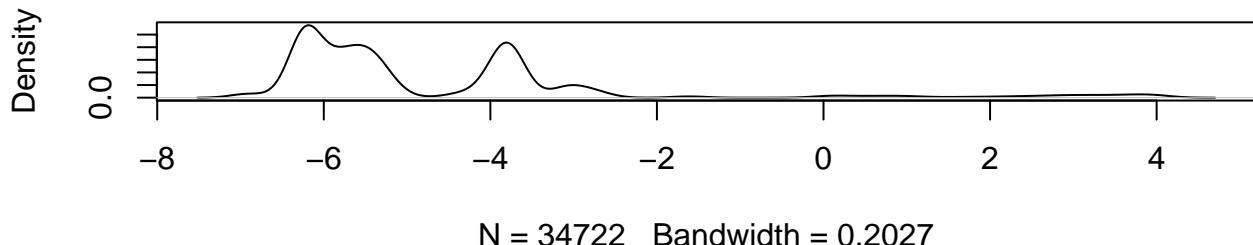
Q-Q Plot TC



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

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

density.default(x = log(tcLogs\$rtt))



N = 34722 Bandwidth = 0.2027

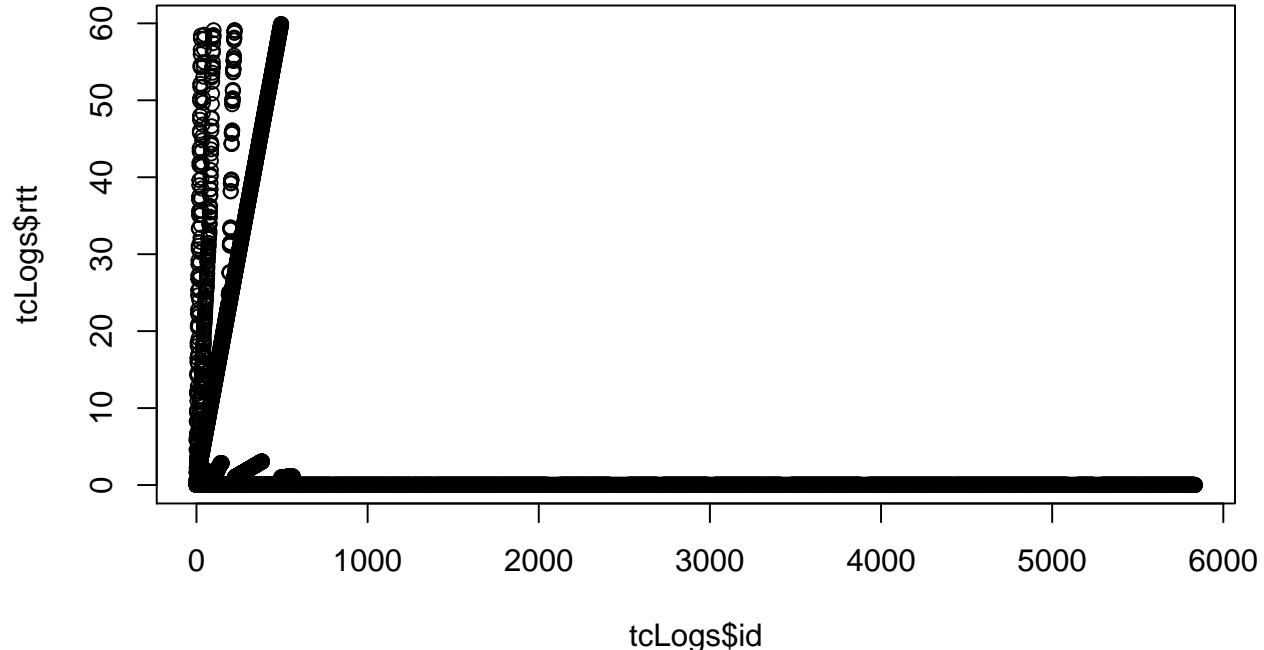
```
reg_tcLogs <- lm(log(tcLogs$rtt) ~ tcLogs$QoS, data = tcLogs)
summary(reg_tcLogs)
#>
#> Call:
#> lm(formula = log(tcLogs$rtt) ~ tcLogs$QoS, data = tcLogs)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -2.3199 -0.9913 -0.5987 -0.2830  9.3164
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
```

```

#> (Intercept) -5.22331    0.01665 -313.65 <2e-16 ***
#> tcLogs$QoSqos1 0.52375    0.02679   19.55 <2e-16 ***
#> tcLogs$QoSqos2 1.73401    0.02721   63.73 <2e-16 ***
#> ---
#> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 2.079 on 34719 degrees of freedom
#> Multiple R-squared: 0.1053, Adjusted R-squared: 0.1053
#> F-statistic: 2044 on 2 and 34719 DF, p-value: < 2.2e-16

```

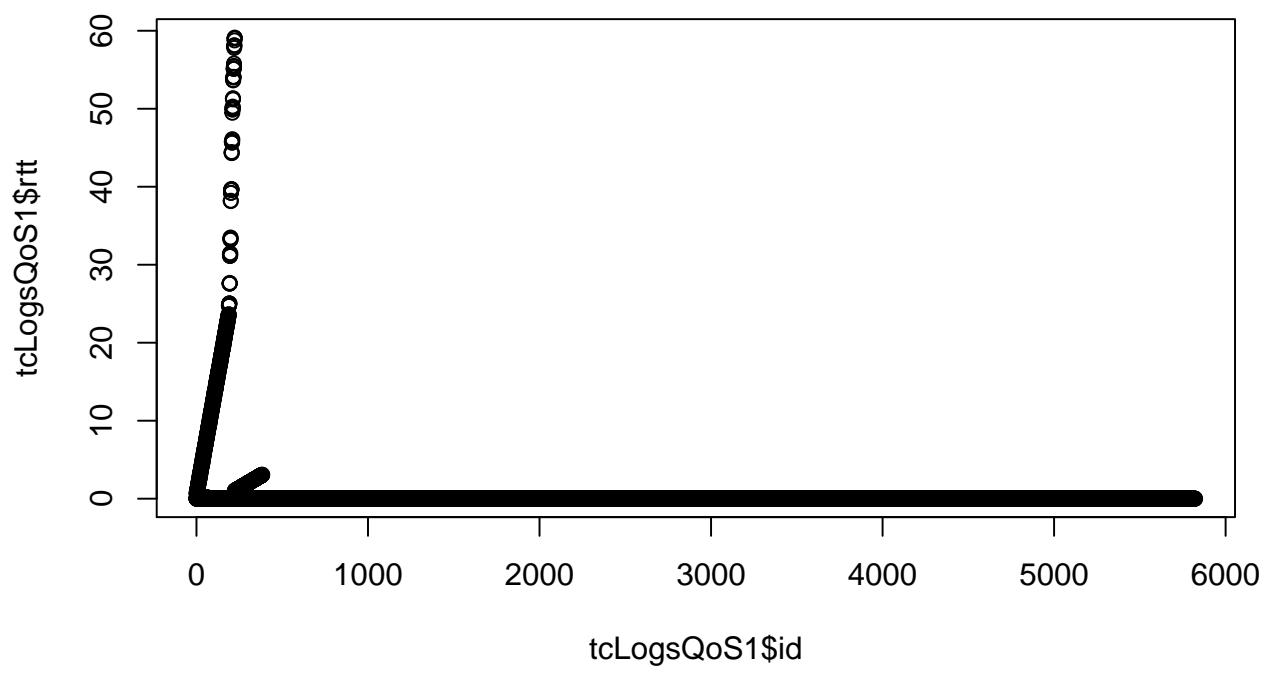
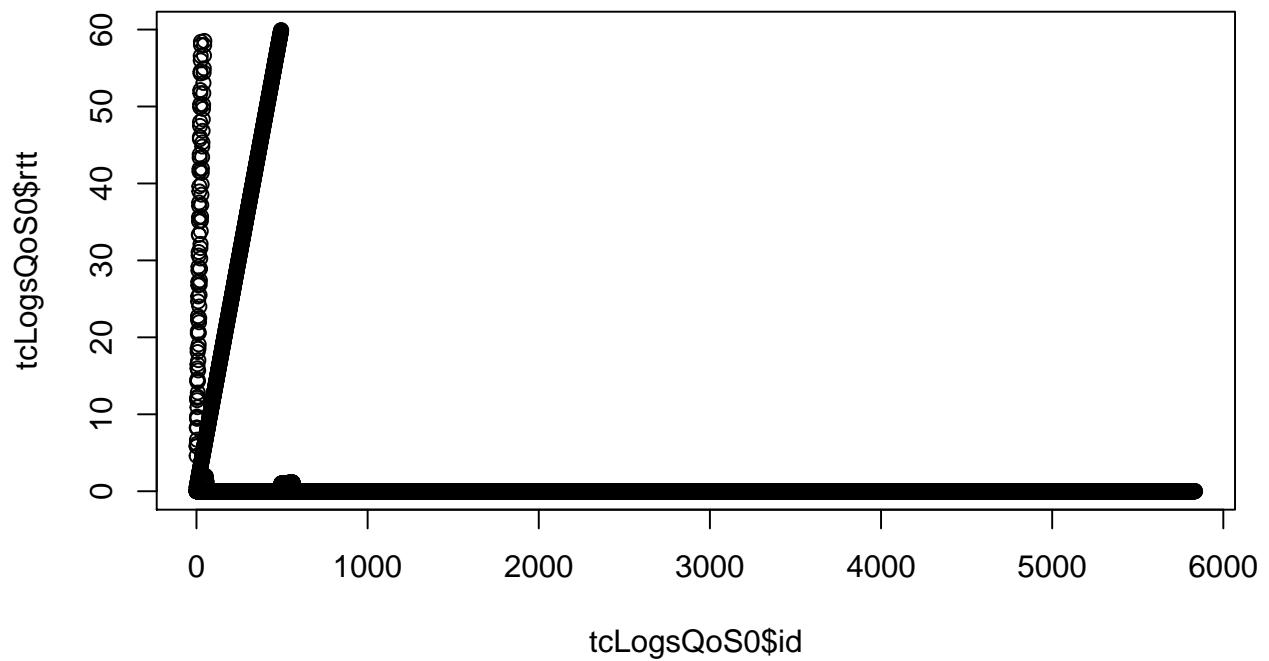
```
plot(tcLogs$id, tcLogs$rtt)
```

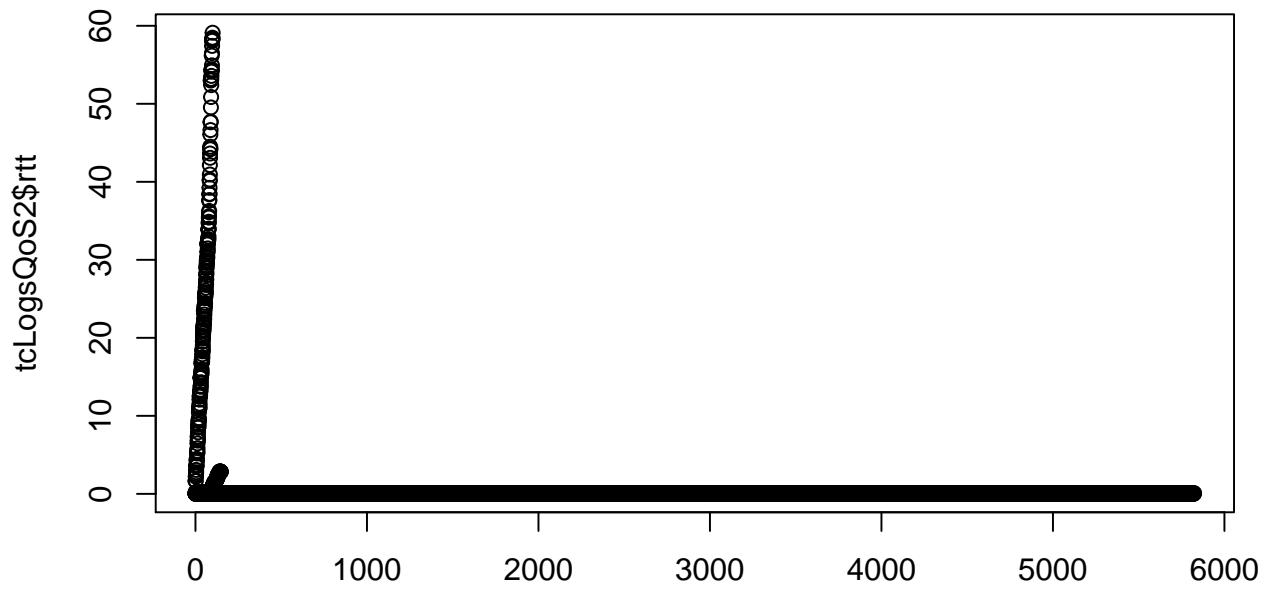


```
#####
# Aufsplittung nach QoS #
#####
```

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

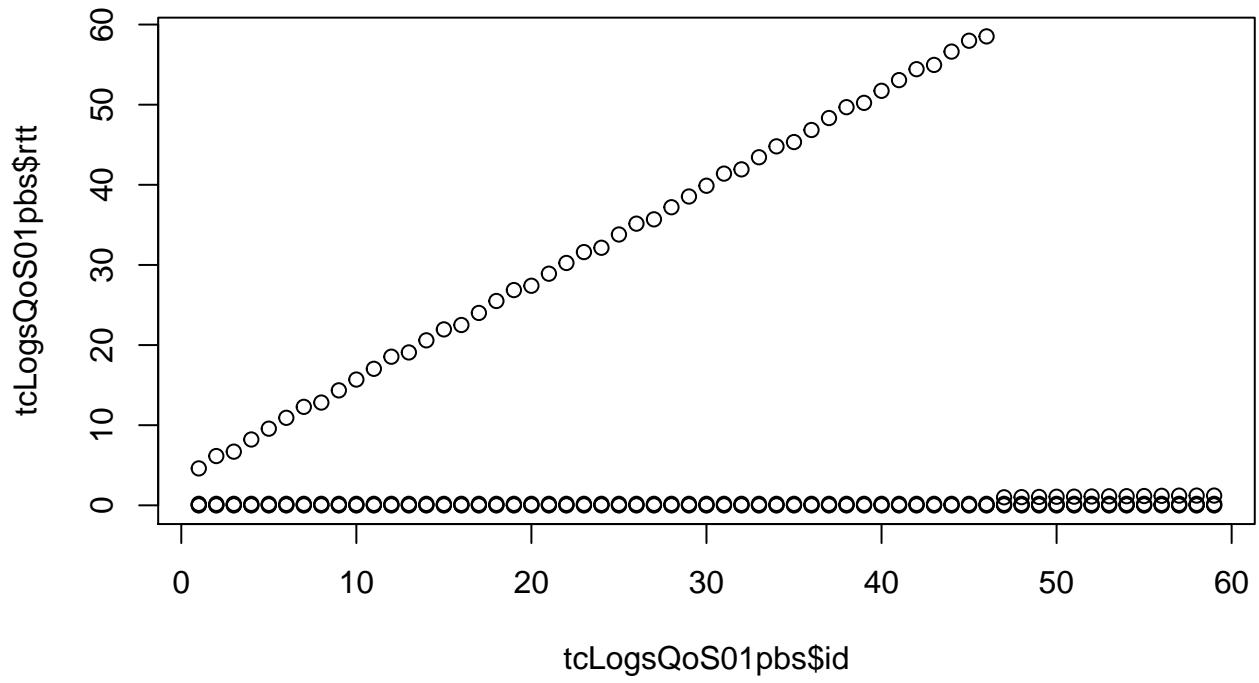
```
plot(tcLogsQoS0$id, tcLogsQoS0$rtt)
```



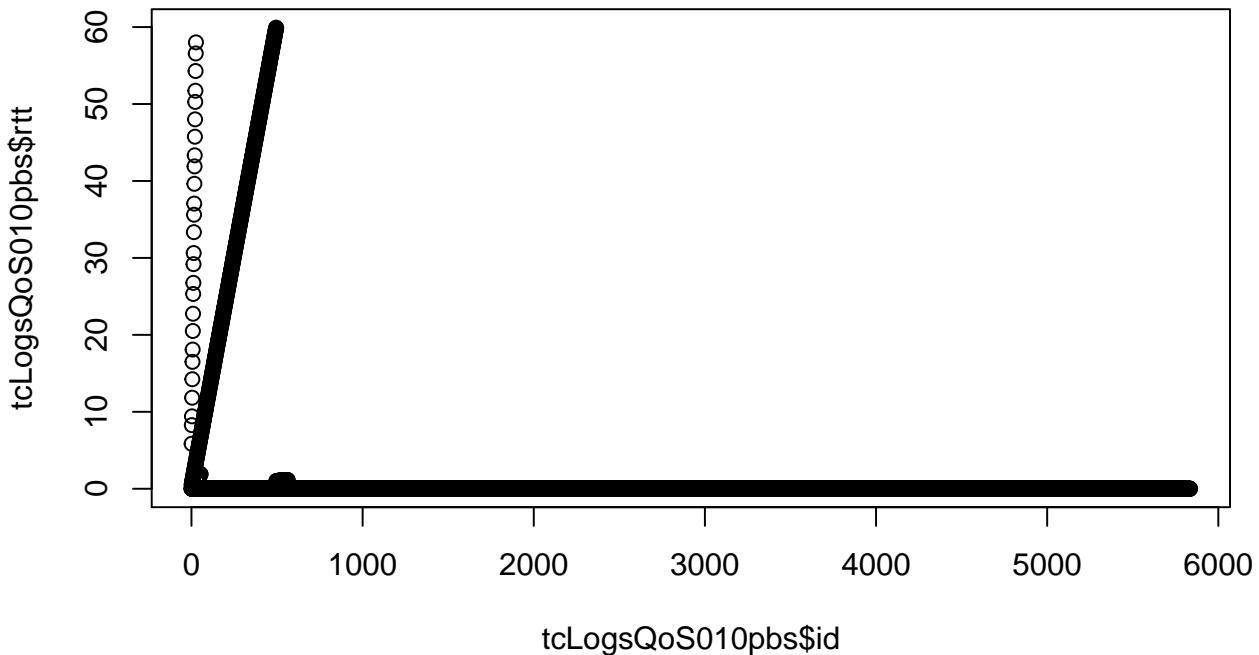


```
#####
# QoS Level - Aufsplittung nach Pakete pro Sekunde Qos #
#####
## QoS0 - Aufsplittung
tcLogsQoS01pbs<-tcLogsQoS0[tcLogsQoS0$PproSek == "1pbs",]
tcLogsQoS010pbs<-tcLogsQoS0[tcLogsQoS0$PproSek == "10pbs",]
tcLogsQoS0100pbs<-tcLogsQoS0[tcLogsQoS0$PproSek == "100pbs",]

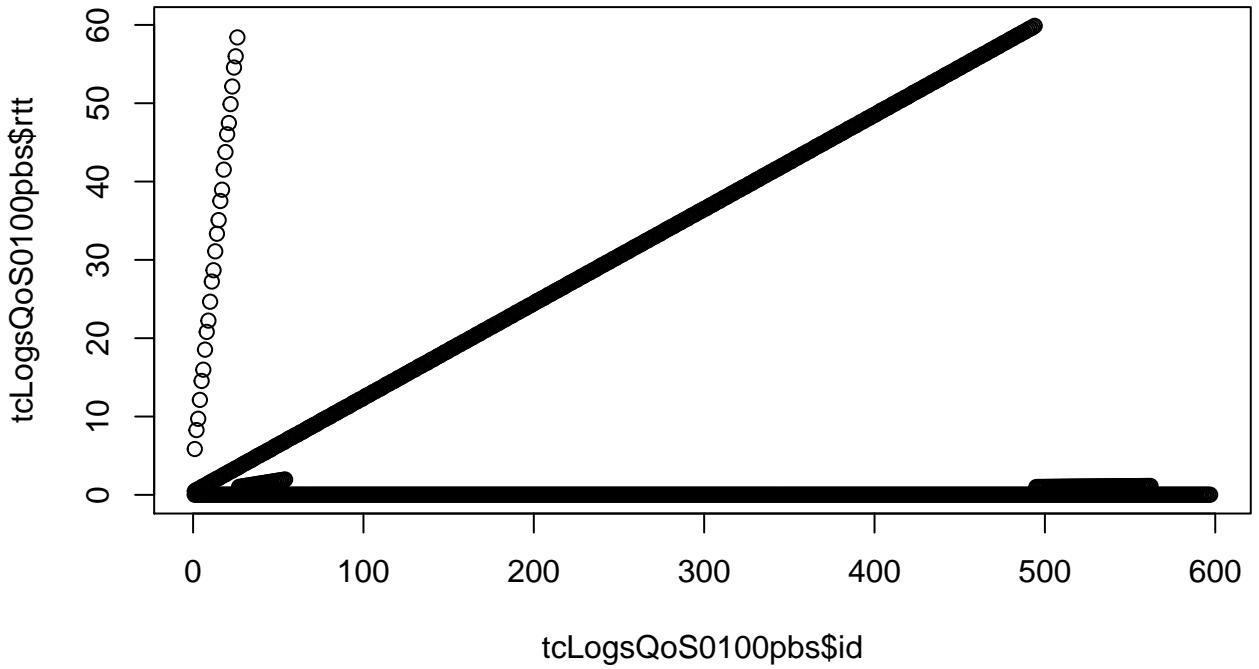
plot(tcLogsQoS01pbs$id, tcLogsQoS01pbs$rtt)
```



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



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



```
## QoS1 - Aufsplittung - Eine Grafik!
```

```
plot(tcLogsQoS01pbs$id, tcLogsQoS01pbs$rtt, type = "l", main = "Aufsplittung aller Messungen mit QoS_0")
points(tcLogsQoS01pbs$id, tcLogsQoS010pbs$rtt, col="red", type = "l")
points(tcLogsQoS0100pbs$id, tcLogsQoS0100pbs$rtt, col="blue", type = "l")
```

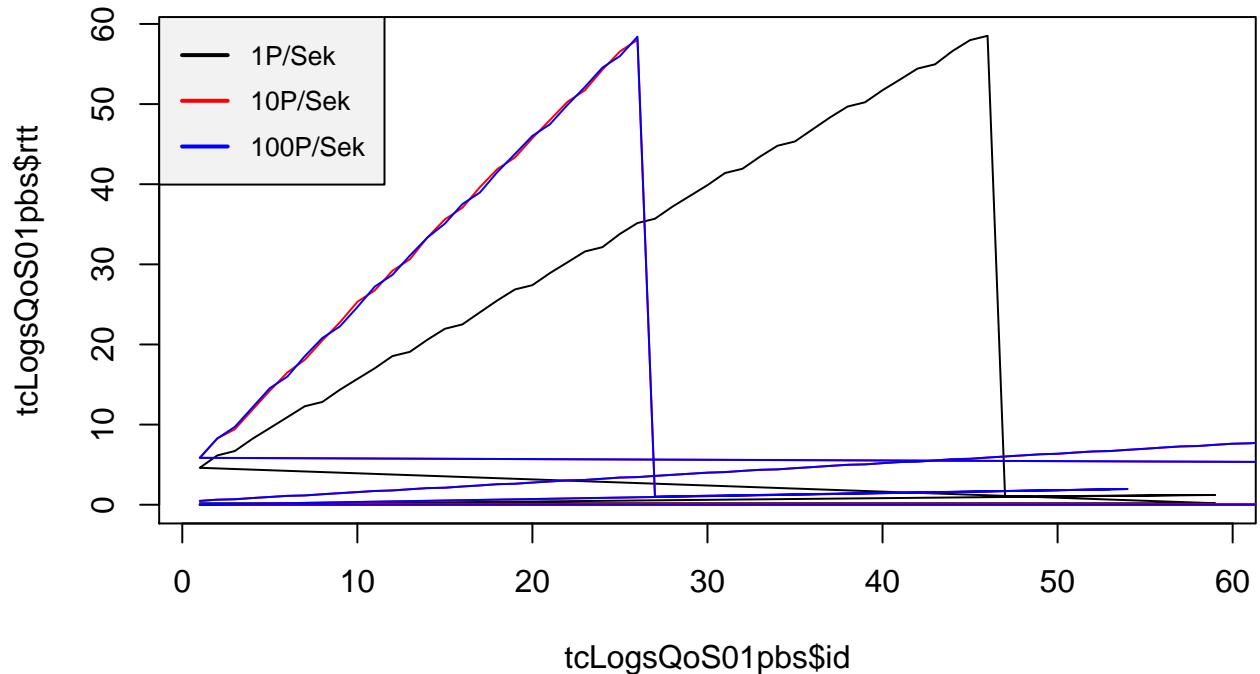
```
legend("topleft", c("1P/Sek", "10P/Sek", "100P/Sek"), text.width = 7, cex = 0.8,
       col = c("black", "red", "blue"))
```

```

text.col = "black" ,lwd = c(2, 2, 2),
y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

Aufsplittung aller Messungen mit QoS_0 nach Paketen/Sek

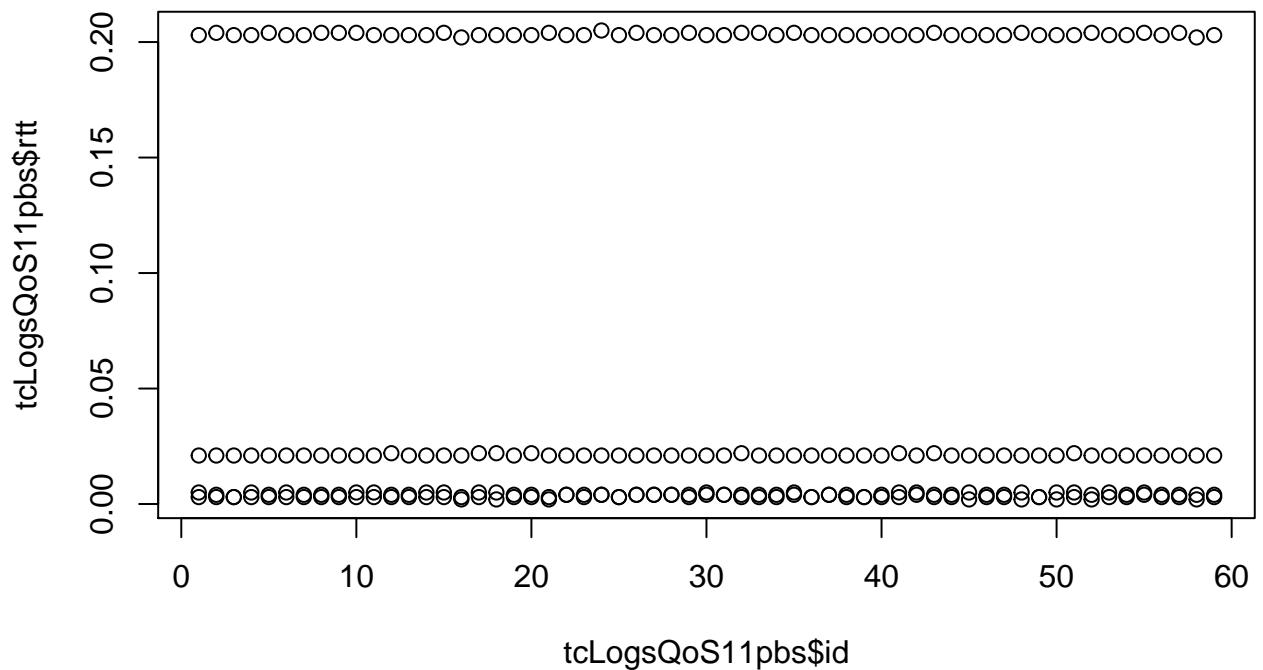


```

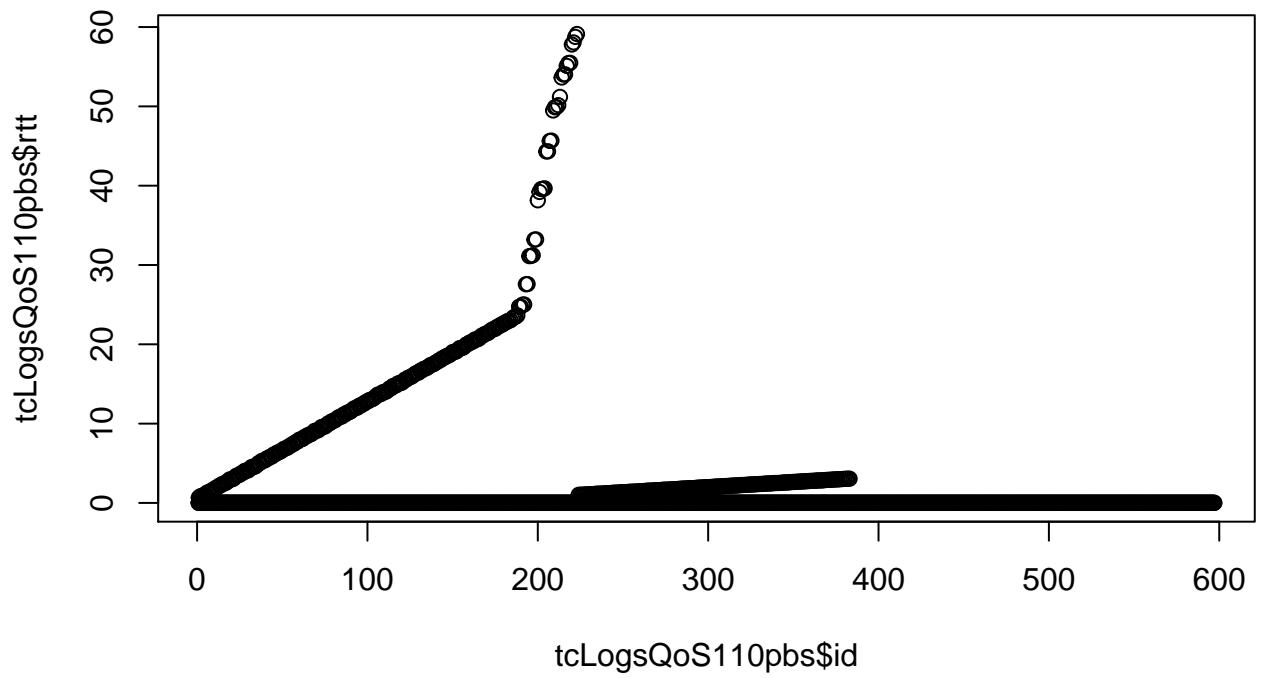
## QoS1 - Aufsplittung
tcLogsQoS11pbs<-tcLogsQoS1[tcLogsQoS1$PproSek == "1pbs",]
tcLogsQoS110pbs<-tcLogsQoS1[tcLogsQoS1$PproSek == "10pbs",]
tcLogsQoS1100pbs<-tcLogsQoS1[tcLogsQoS1$PproSek == "100pbs",]

plot(tcLogsQoS11pbs$id, tcLogsQoS11pbs$rtt)

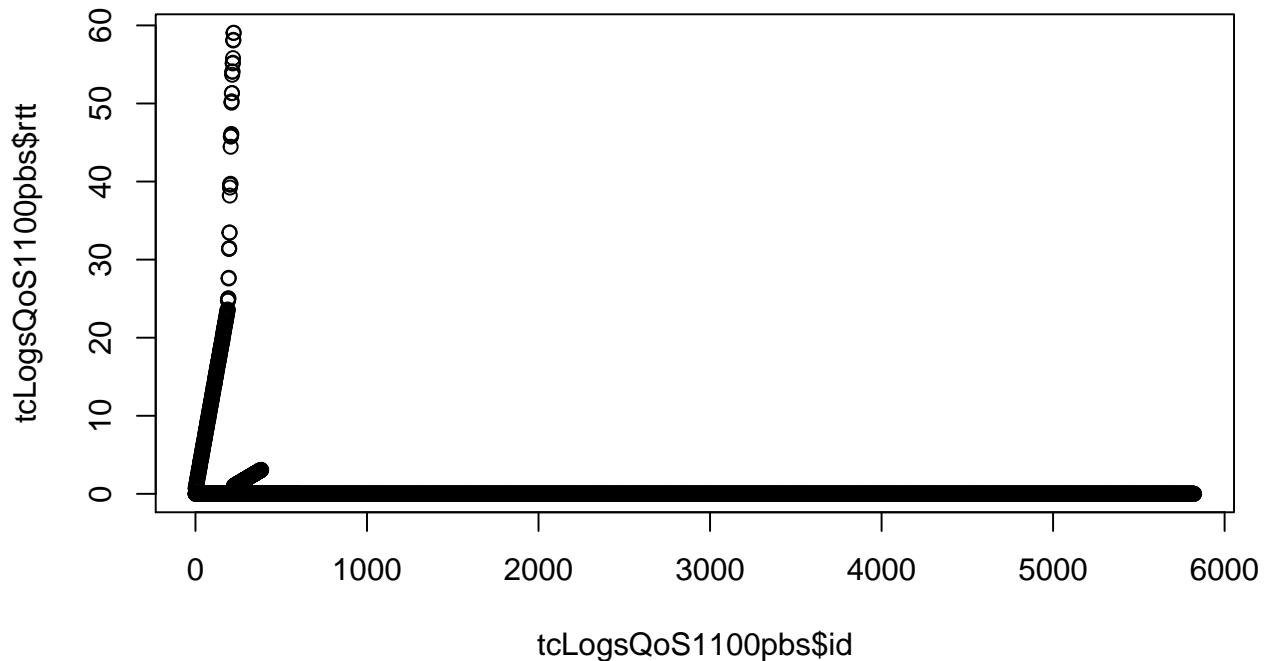
```



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



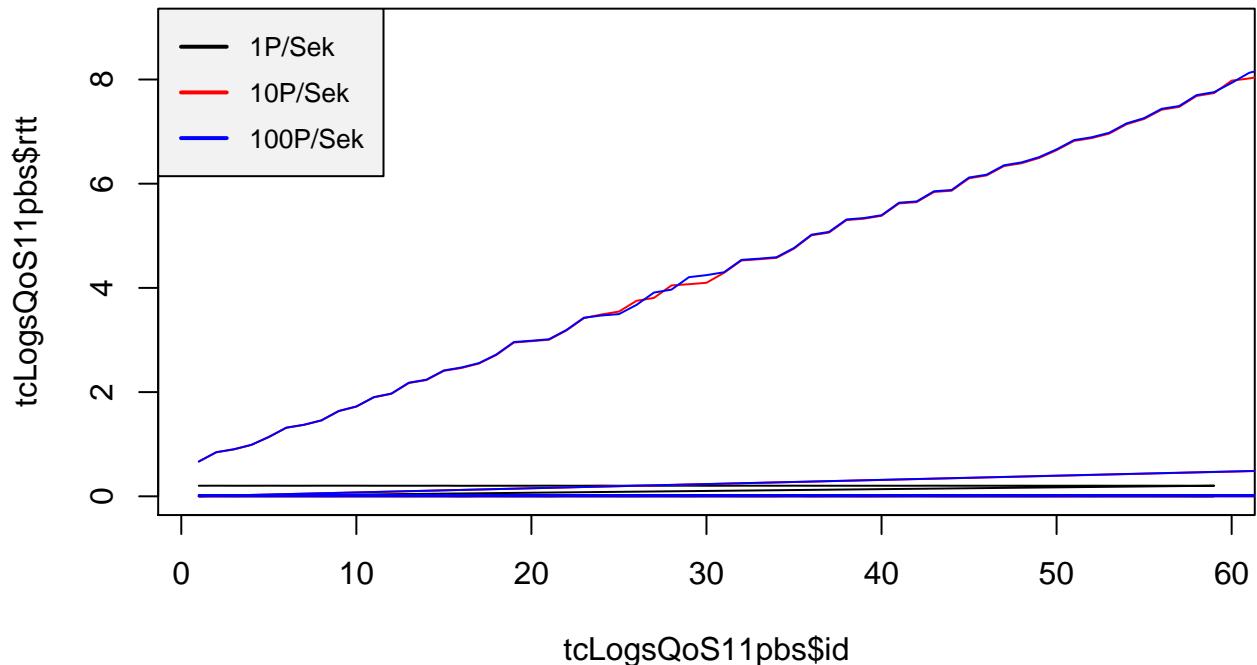
```
plot(tcLogsQoS1100pbs$id, tcLogsQoS1100pbs$rtt)
```



```
## QoS1 - Aufsplittung - eine Grafik!
plot(tcLogsQoS11pbs$id, tcLogsQoS11pbs$rtt, type = "l", ylim = c(0, 9),
     main = "Aufsplittung aller Messungen mit QoS_1 nach Paketen/Sek")
points(tcLogsQoS110pbs$id, tcLogsQoS110pbs$rtt, col="red", type = "l")
points(tcLogsQoS1100pbs$id, tcLogsQoS1100pbs$rtt, col="blue", type = "l")

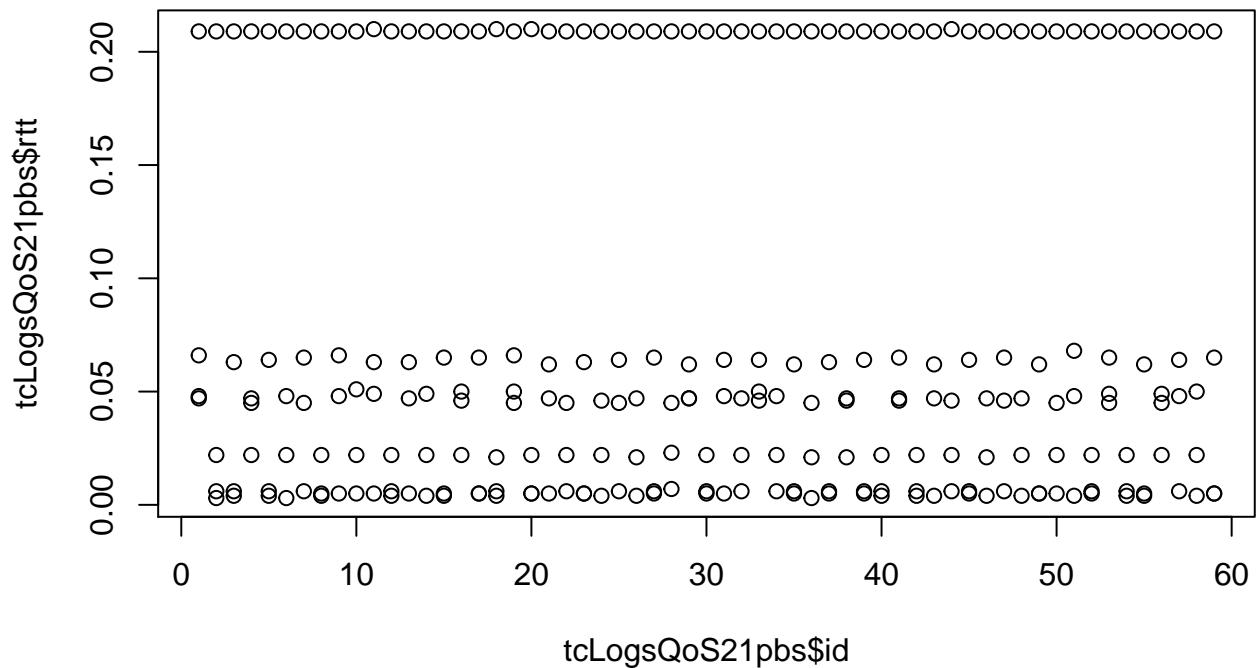
legend("topleft", c("1P/Sek", "10P/Sek", "100P/Sek"), text.width = 7, cex = 0.8,
       col = c("black", "red", "blue"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")
```

Aufsplittung aller Messungen mit QoS_1 nach Paketen/Sek

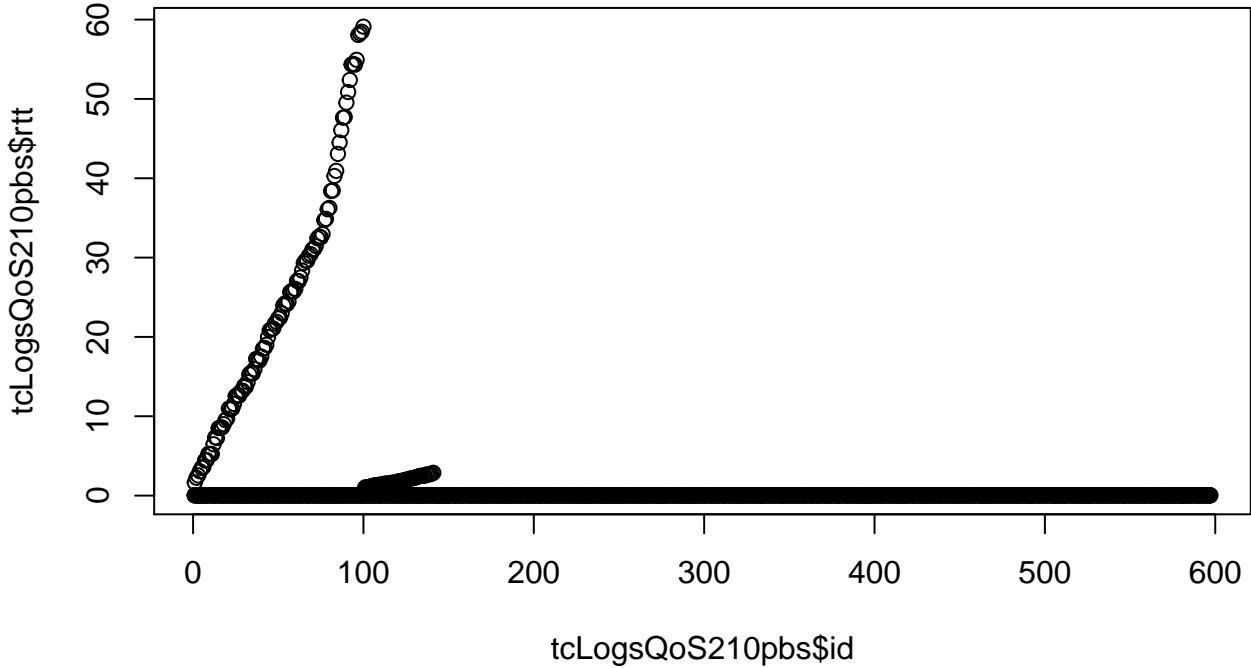


```
## QoS2 - Aufsplittung
tcLogsQoS21pbs<-tcLogsQoS2[tcLogsQoS2$PproSek == "1pbs",]
tcLogsQoS210pbs<-tcLogsQoS2[tcLogsQoS2$PproSek == "10pbs",]
tcLogsQoS2100pbs<-tcLogsQoS2[tcLogsQoS2$PproSek == "100pbs",]

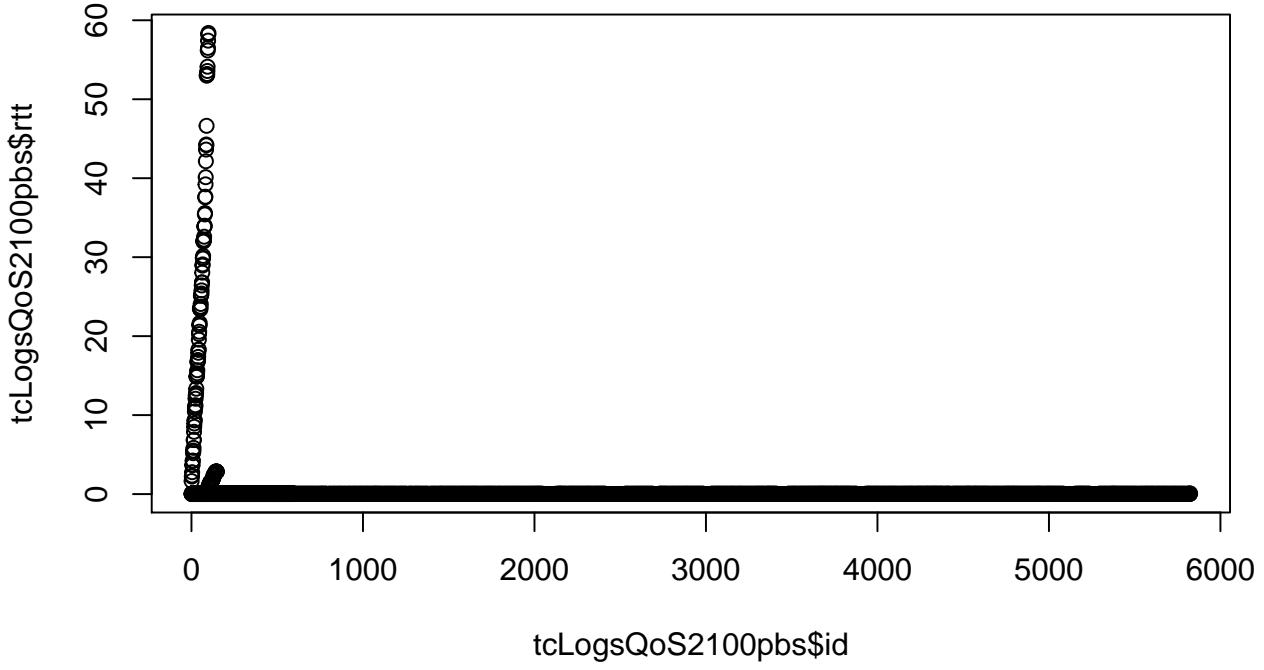
plot(tcLogsQoS21pbs$id, tcLogsQoS21pbs$rtt)
```



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



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



```
## QoS2 - Aufsplittung - eine Grafik!
plot(tcLogsQoS21pbs$id, tcLogsQoS21pbs$rtt, type = "l", ylim = c(0, 30),
     main = "Aufsplittung aller Messungen mit QoS_2 nach Paketen/Sek")
points(tcLogsQoS210pbs$id, tcLogsQoS210pbs$rtt, col="red", type = "l")
points(tcLogsQoS2100pbs$id, tcLogsQoS2100pbs$rtt, col="blue", type = "l")

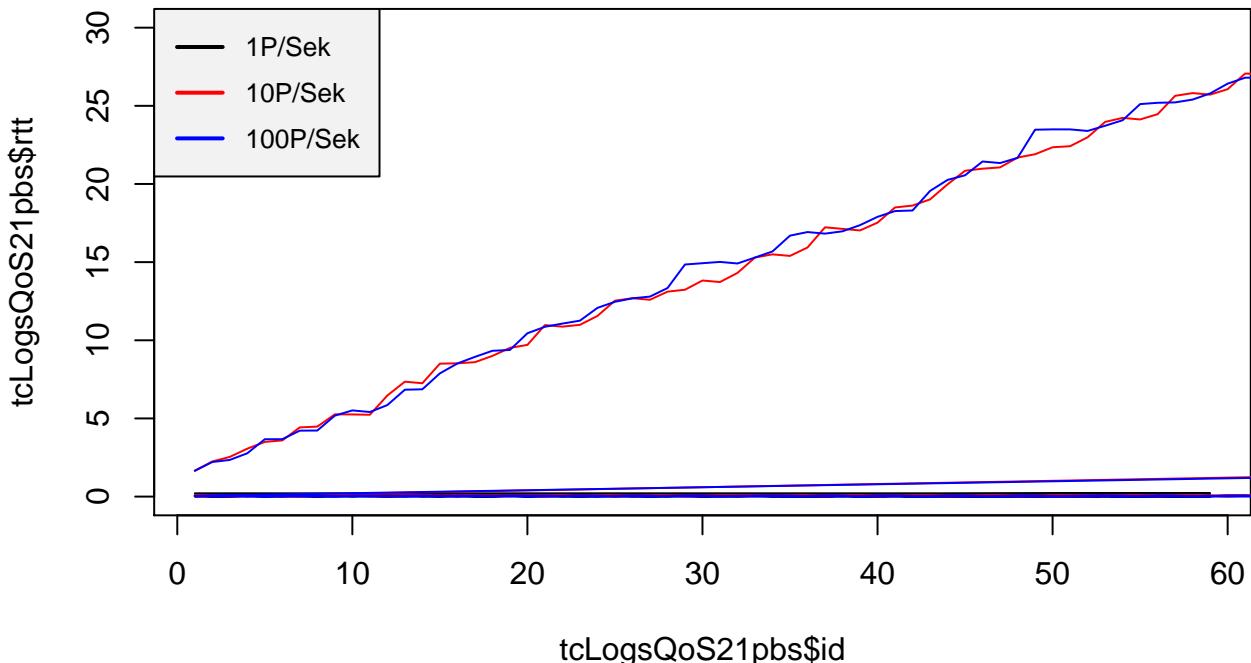
legend("topleft", c("1P/Sek", "10P/Sek", "100P/Sek"), text.width = 7, cex = 0.8,
```

```

col = c("black", "red", "blue"),
text.col = "black", lwd = c(2, 2, 2),
y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

Aufsplittung aller Messungen mit QoS_2 nach Paketen/Sek



```

#####
# 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 #
#####

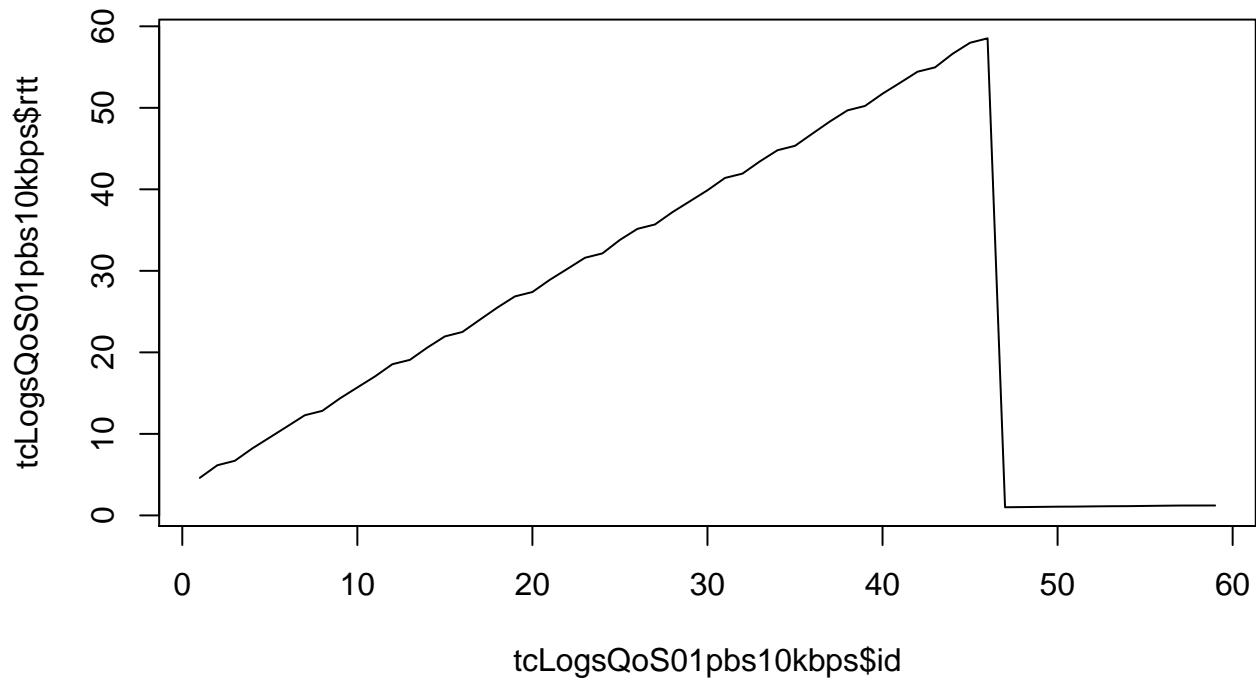
#####
# QoS 0 #
#####

## 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",]
```

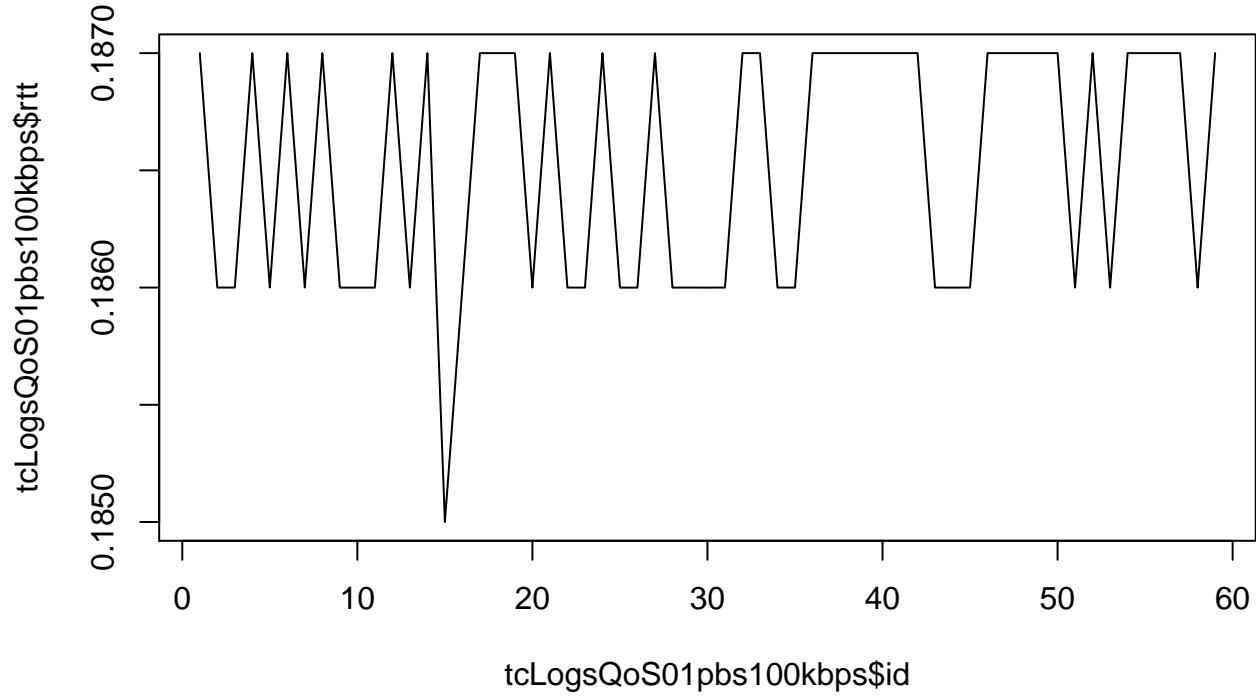
```
plot(tcLogsQoS01pbs10kbps$id, tcLogsQoS01pbs10kbps$rtt, type = "l", main = "QoS_0 1_Paket/Sek 10kb_Max")
```

QoS_0 1_Paket/Sek 10kb_Max



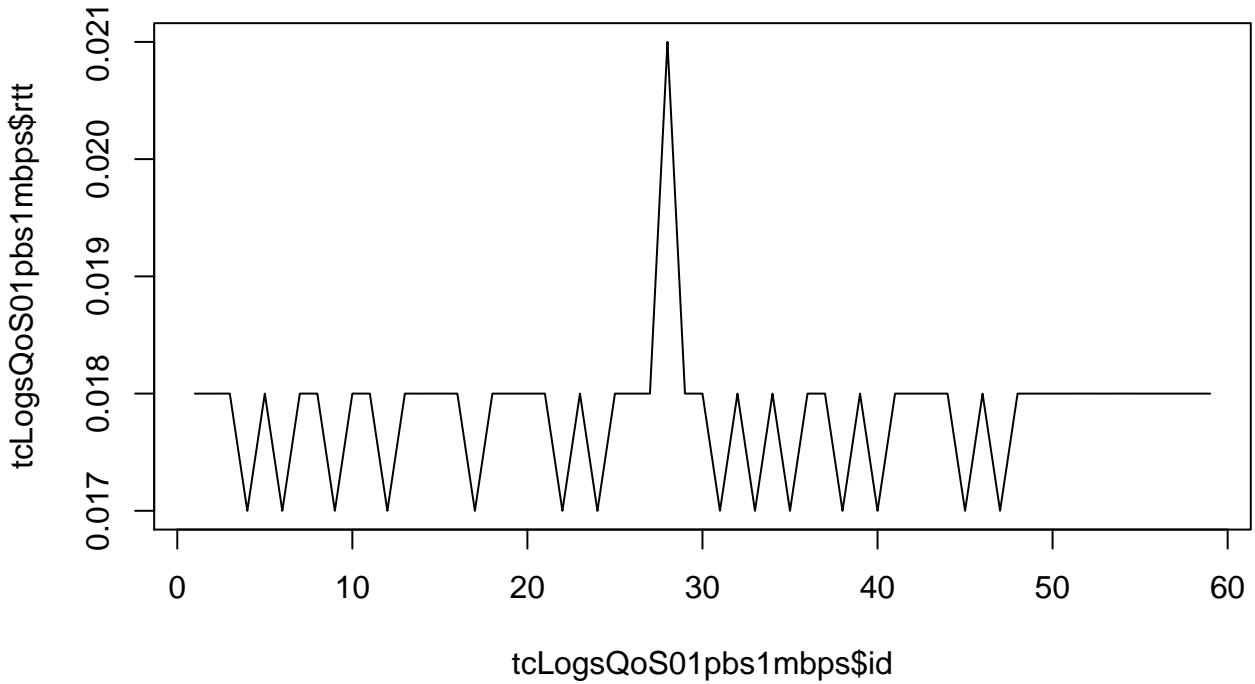
```
plot(tcLogsQoS01pbs100kbps$id, tcLogsQoS01pbs100kbps$rtt, type = "l", main = "QoS_0 1_Paket/Sek 100kb_Max")
```

QoS_0 1_Paket/Sek 100kb_Max



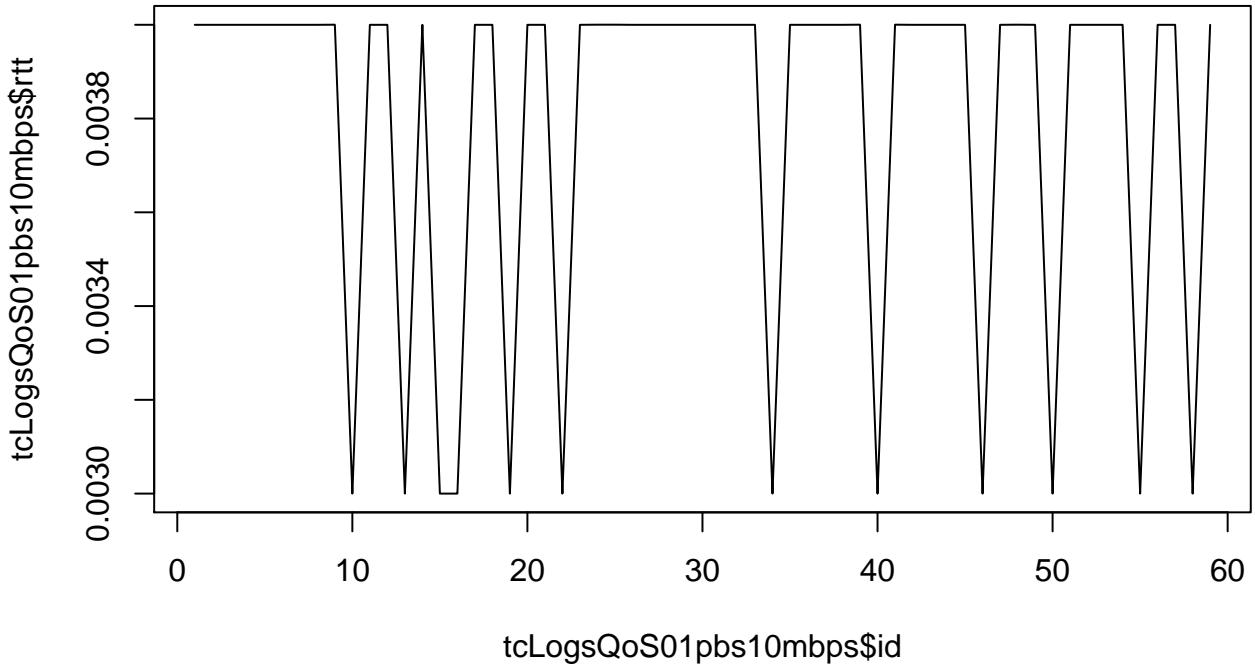
```
plot(tcLogsQoS01pbs1mbps$id, tcLogsQoS01pbs1mbps$rtt, type = "l", main = "QoS_0_1_Paket/Sek_1mb_Max")
```

QoS_0_1_Paket/Sek_1mb_Max



```
plot(tcLogsQoS01pbs10mbps$id, tcLogsQoS01pbs10mbps$rtt, type = "l", main = "QoS_0_1_Paket/Sek_10mb_Max")
```

QoS_0_1_Paket/Sek_10mb_Max



```
#plot(tcLogsQoS01pbs100mbps$id, tcLogsQoS01pbs100mbps$rtt, type = "l", main = "QoS_0_1_Paket/Sek_100mb_Max")
```

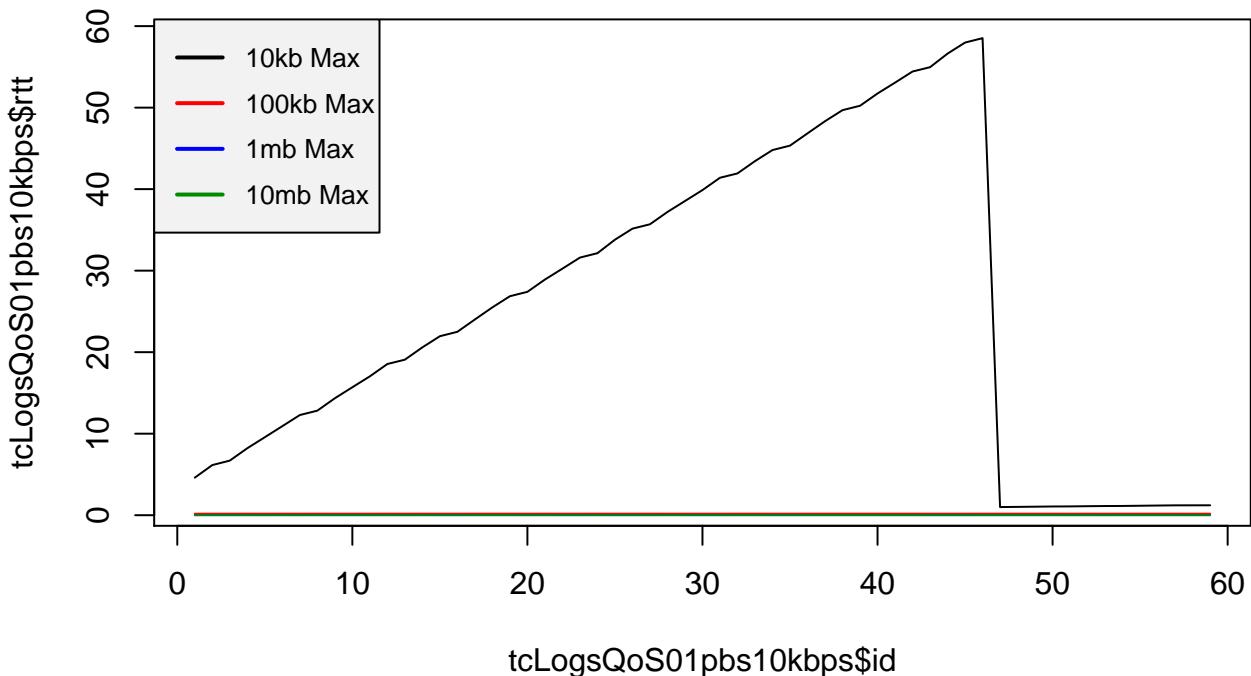
```

## Eine Grafik
plot(tcLogsQoS01pbs10kbps$id, tcLogsQoS01pbs10kbps$rtt, type = "l", main = "QoS0 1Paket/Sek aufgeteilt nach Max Durchsatz",
      points(tcLogsQoS01pbs100kbps$id, tcLogsQoS01pbs100kbps$rtt, type = "l", col = "red")
      points(tcLogsQoS01pbs1mbps$id, tcLogsQoS01pbs1mbps$rtt, type = "l", col = "blue")
      points(tcLogsQoS01pbs10mbps$id, tcLogsQoS01pbs10mbps$rtt, type = "l", col = "green4"))

legend("topleft", c("10kb Max", "100kb Max", "1mb Max", "10mb Max"),
       text.width = 7, cex = 0.8,
       col = c("black", "red", "blue", "green4"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

QoS0 1Paket/Sek aufgeteilt nach Max Durchsatz



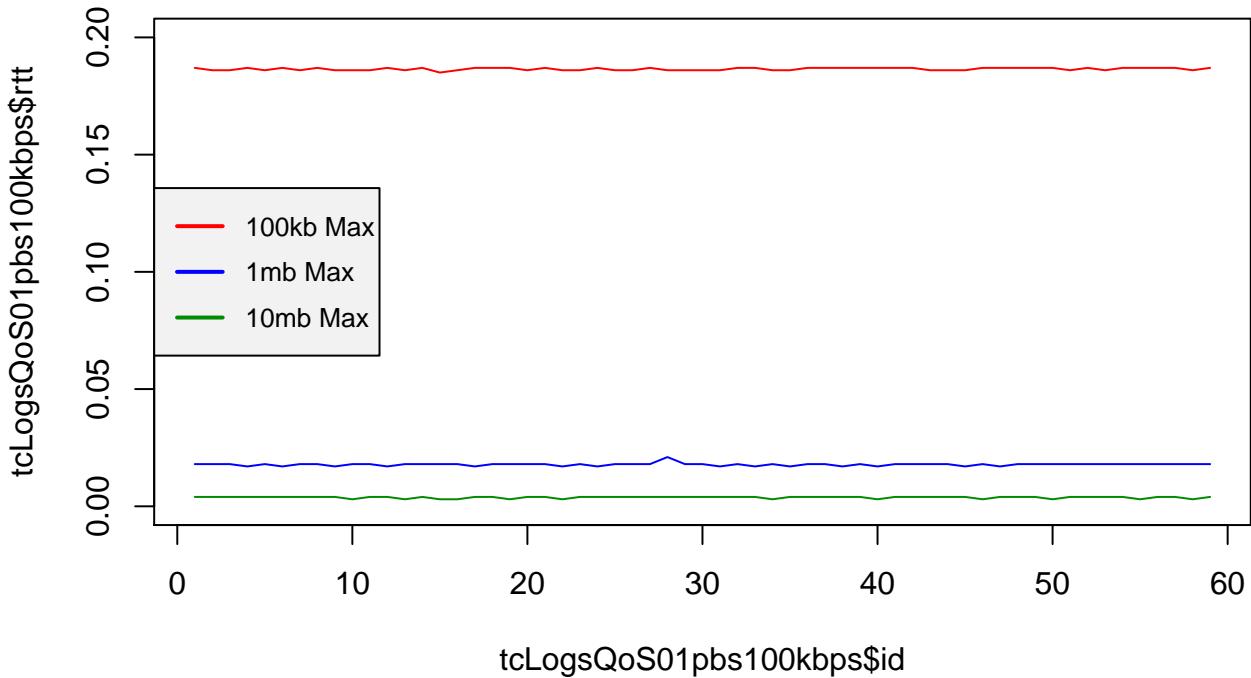
```

plot(tcLogsQoS01pbs100kbps$id, tcLogsQoS01pbs100kbps$rtt, ylim = c(0,0.2), type = "l", col = "red", main = "QoS0 1Paket/Sek aufgeteilt nach Max Durchsatz",
      points(tcLogsQoS01pbs1mbps$id, tcLogsQoS01pbs1mbps$rtt, type = "l", col = "blue")
      points(tcLogsQoS01pbs10mbps$id, tcLogsQoS01pbs10mbps$rtt, type = "l", col = "green4"))

legend("left", c("100kb Max", "1mb Max", "10mb Max"),
       text.width = 7, cex = 0.8,
       col = c("red", "blue", "green4"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

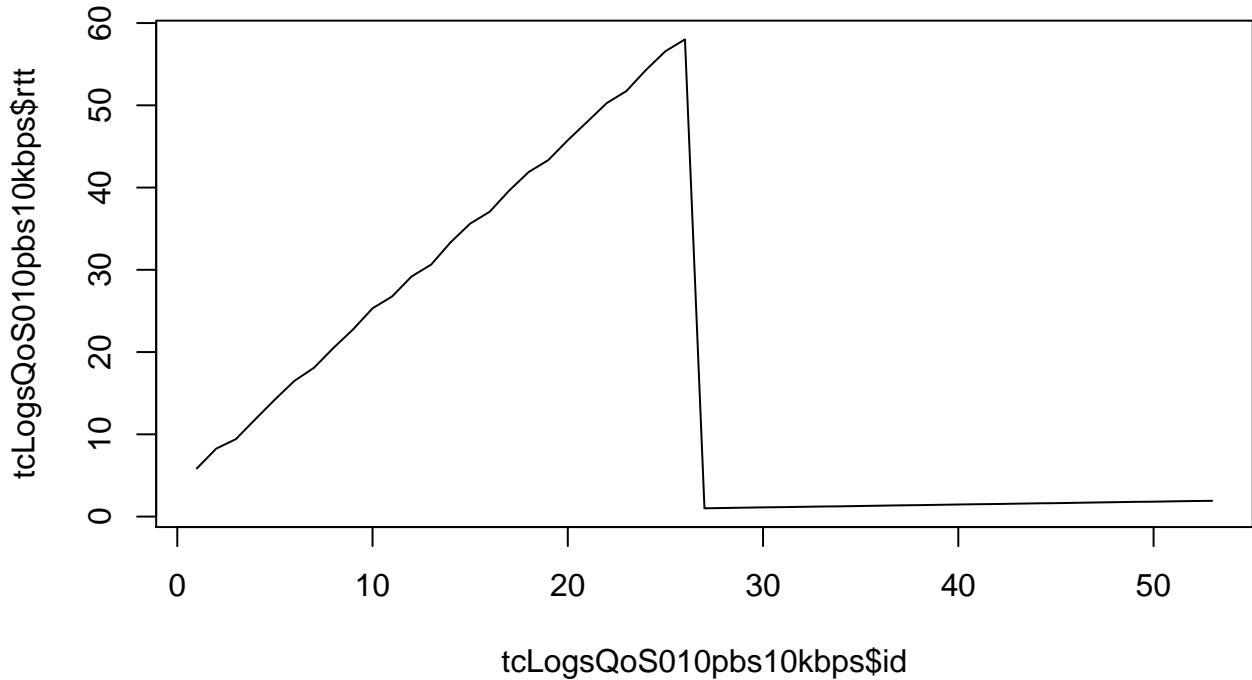
QoS0 1 Paket/Sek nach Max (ohne 10kb)



```
#####
# QoS 0 #
#####
## QoS0_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",]

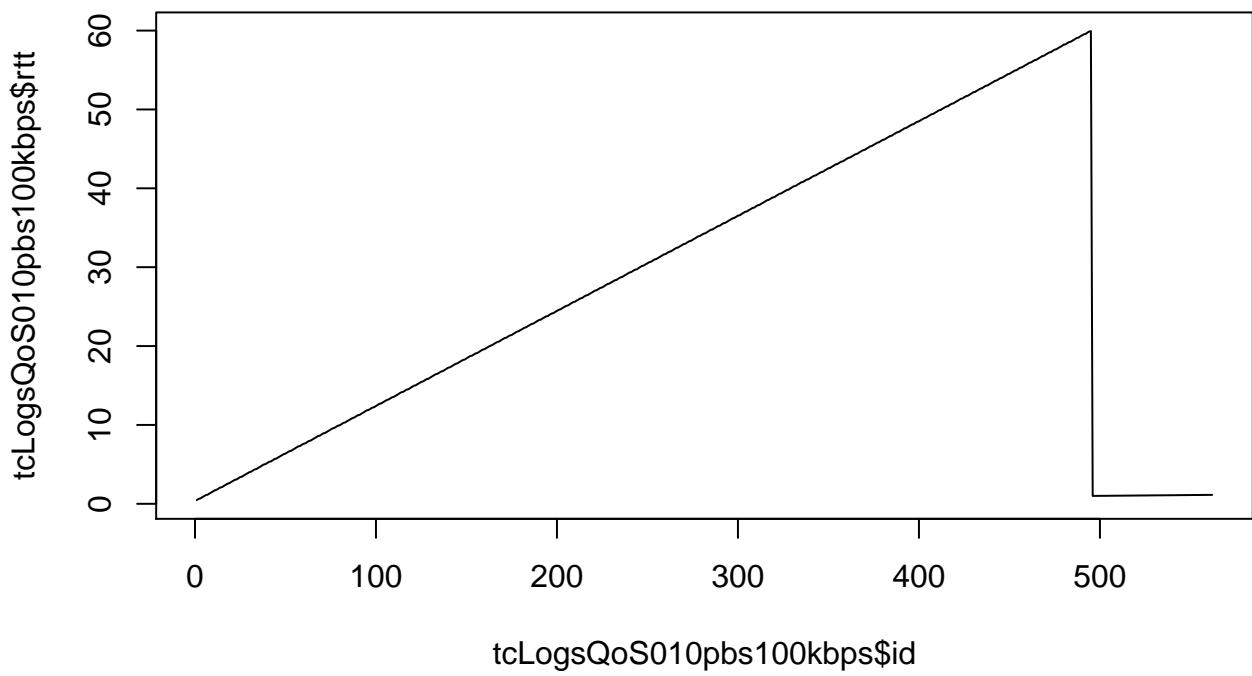
plot(tcLogsQoS010pbs10kbps$id, tcLogsQoS010pbs10kbps$rtt, type = "l", main = "QoS_0_10_Pakete/Sek 10kb_1")
```

QoS_0 10_Pakete/Sek 10kb_Max



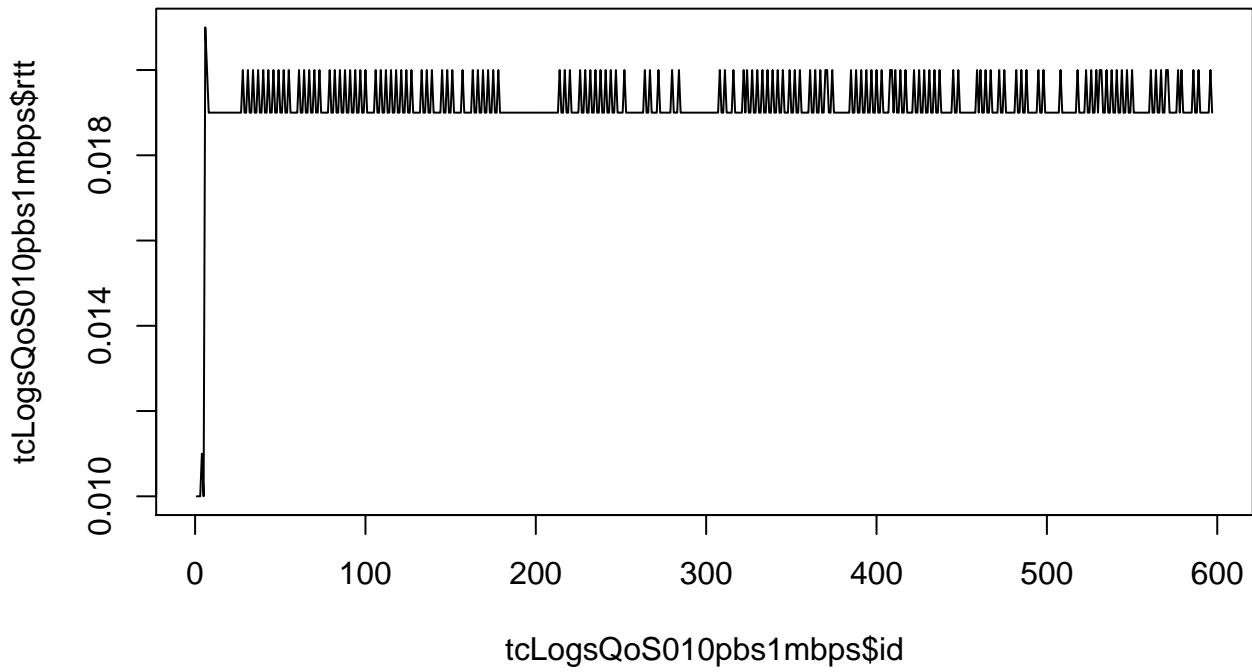
```
plot(tcLogsQoS010pbs100kbps$id, tcLogsQoS010pbs100kbps$rtt, type = "l", main = "QoS_0 10_Pakete/Sek 100kb_Max")
```

QoS_0 10_Pakete/Sek 100kb_Max



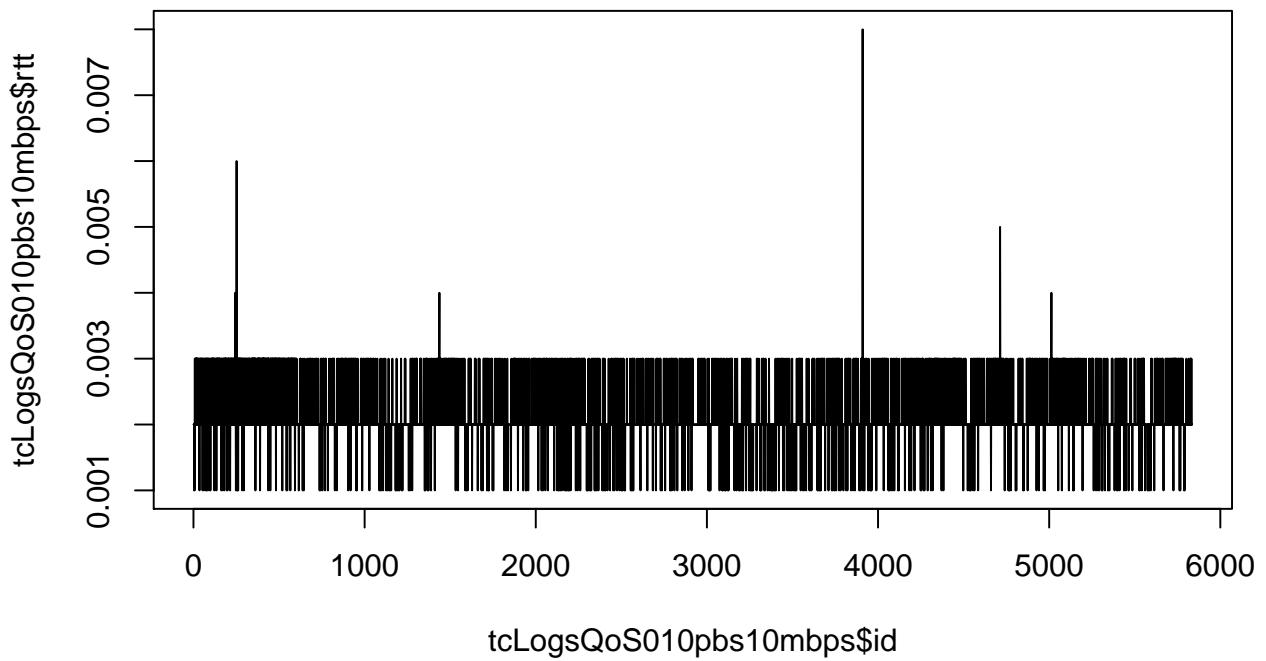
```
plot(tcLogsQoS010pbs100kbps$id, tcLogsQoS010pbs100kbps$rtt, type = "l", main = "QoS_0 10_Pakete/Sek 100kb_Max")
```

QoS_0 10_Pakete/Sek 1mb_Max



```
plot(tcLogsQoS010pbs1mbps$id, tcLogsQoS010pbs1mbps$rtt, type = "l", main = "QoS_0 10_Pakete/Sek 1mb_Max")
```

QoS_0 10_Pakete/Sek 10mb_Max



```
#plot(tcLogsQoS010pbs10mbps$id, tcLogsQoS010pbs10mbps$rtt, type = "l", main = "QoS_0 10_Pakete/Sek 10mb_Max")
```

```
## Eine Grafik
```

```
plot(tcLogsQoS010pbs10kbps$id, tcLogsQoS010pbs10kbps$rtt, type = "l", main = "QoS0 10Pakete/Sek aufgeteilt")
```

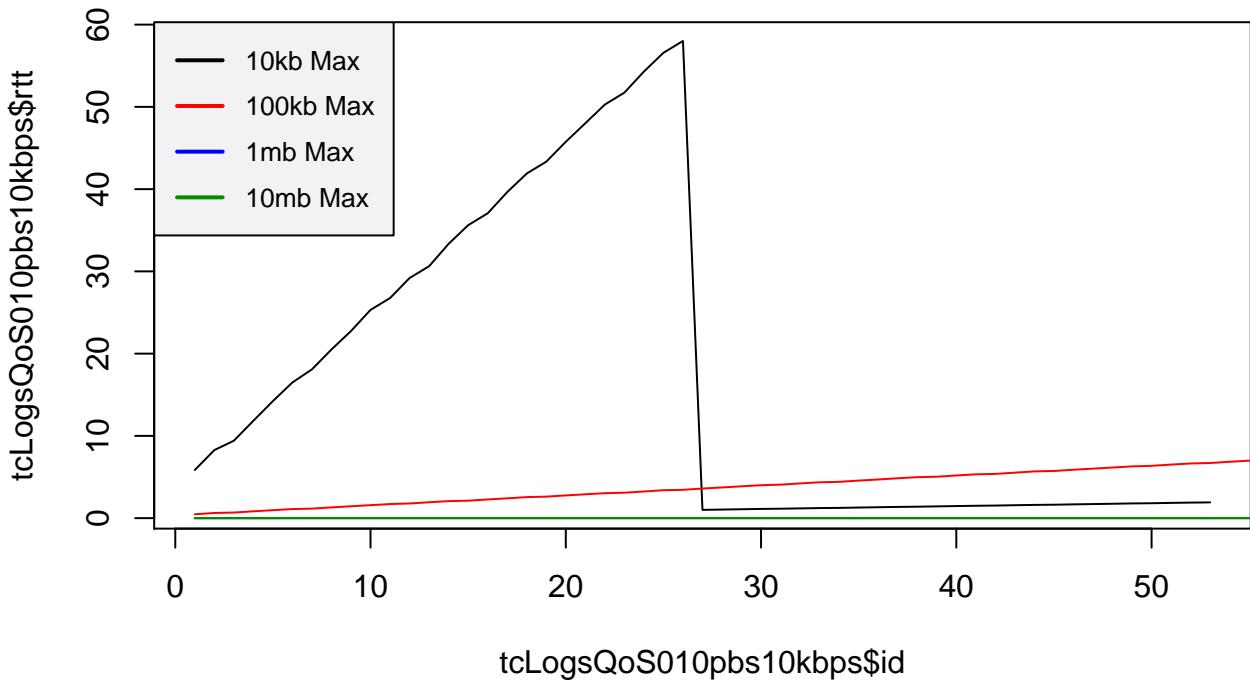
```

points(tcLogsQoS010pbs100kbps$id, tcLogsQoS010pbs100kbps$rtt, type = "l", col = "red")
points(tcLogsQoS010pbs1mbps$id, tcLogsQoS010pbs1mbps$rtt, type = "l", col = "blue")
points(tcLogsQoS010pbs10mbps$id, tcLogsQoS010pbs10mbps$rtt, type = "l", col = "green4")

legend("topleft", c("10kb Max", "100kb Max", "1mb Max", "10mb Max"),
       text.width = 7, cex = 0.8,
       col = c("black", "red", "blue", "green4"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

QoS0 10Pakete/Sek aufgeteilt nach Max Durchsatz



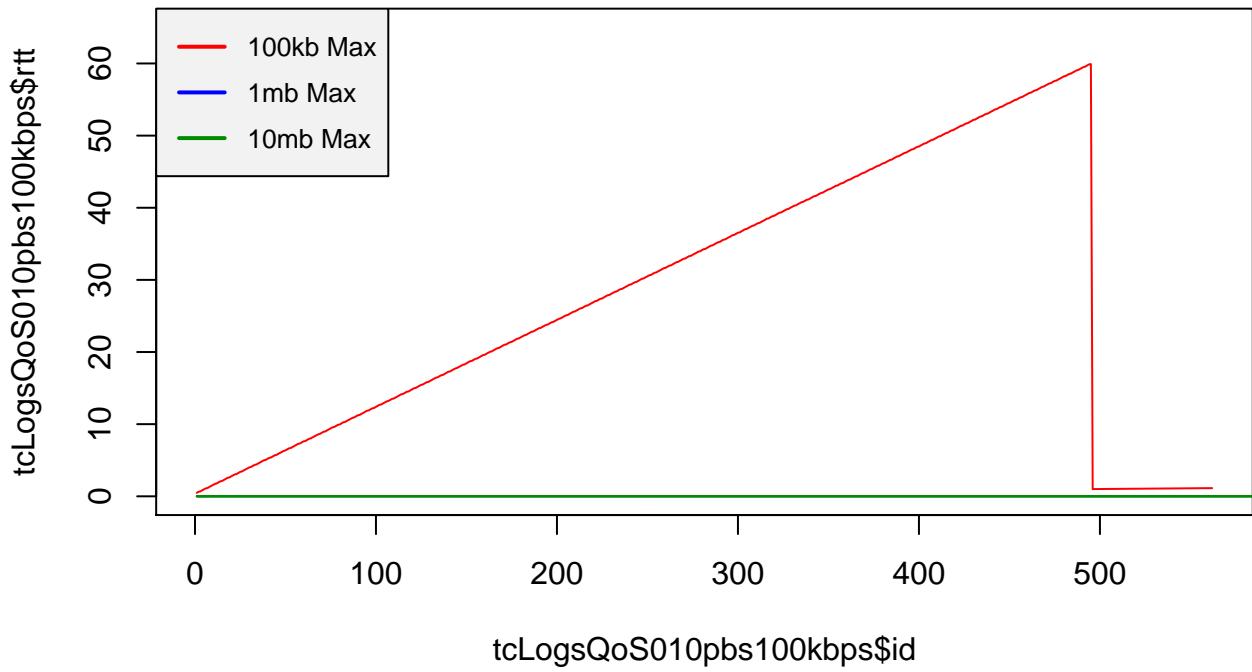
```

plot(tcLogsQoS010pbs100kbps$id, tcLogsQoS010pbs100kbps$rtt, ylim = c(0,65), type = "l", col = "red", ma
points(tcLogsQoS010pbs1mbps$id, tcLogsQoS010pbs1mbps$rtt, type = "l", col = "blue")
points(tcLogsQoS010pbs10mbps$id, tcLogsQoS010pbs10mbps$rtt, type = "l", col = "green4")

legend("topleft", c("100kb Max", "1mb Max", "10mb Max"), cex = 0.8,
       col = c("red", "blue", "green4"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

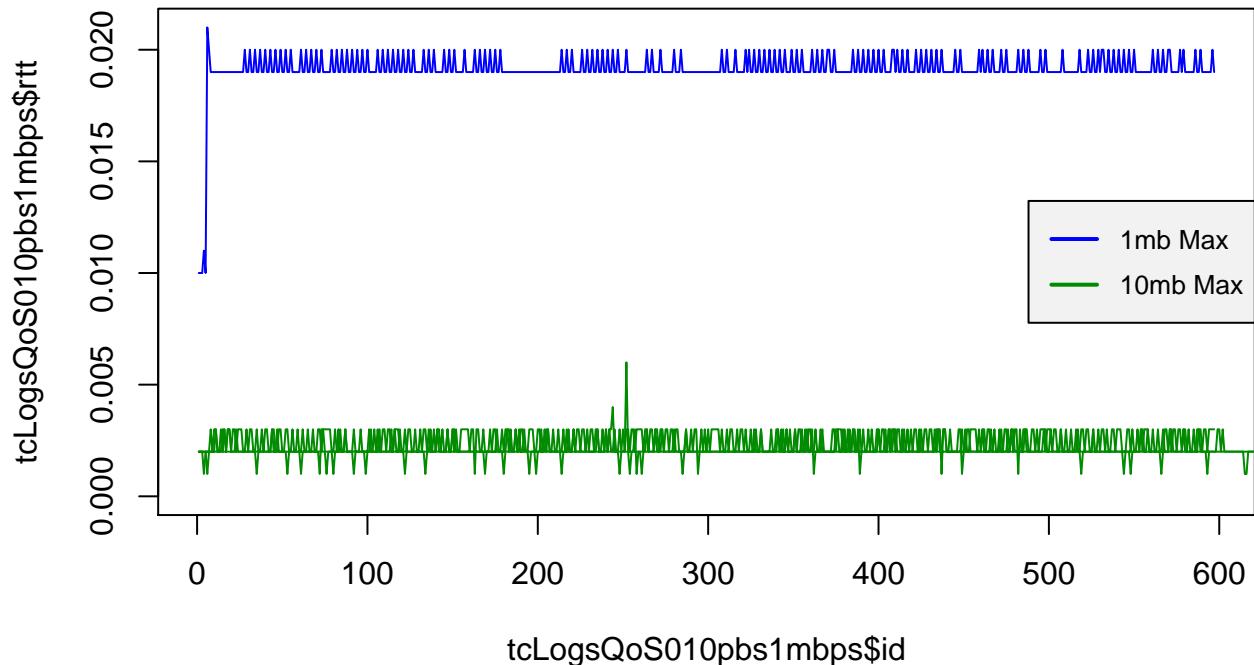
QoS 10Pakete/Sek nach Max (ohne 10kb)



```
plot(tcLogsQoS010pbs1mbps$id, tcLogsQoS010pbs1mbps$rtt, ylim = c(0,0.021), type = "l", col = "blue", mai = c(0,0,0,0))
points(tcLogsQoS010pbs10mbps$id, tcLogsQoS010pbs10mbps$rtt, type = "l", col = "green4")

legend("right", c("1mb Max", "10mb Max"), cex = 0.8,
       col = c("blue", "green4"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")
```

QoS0 10Pakete/Sek nach Max (ohne 10kb und 100kb)



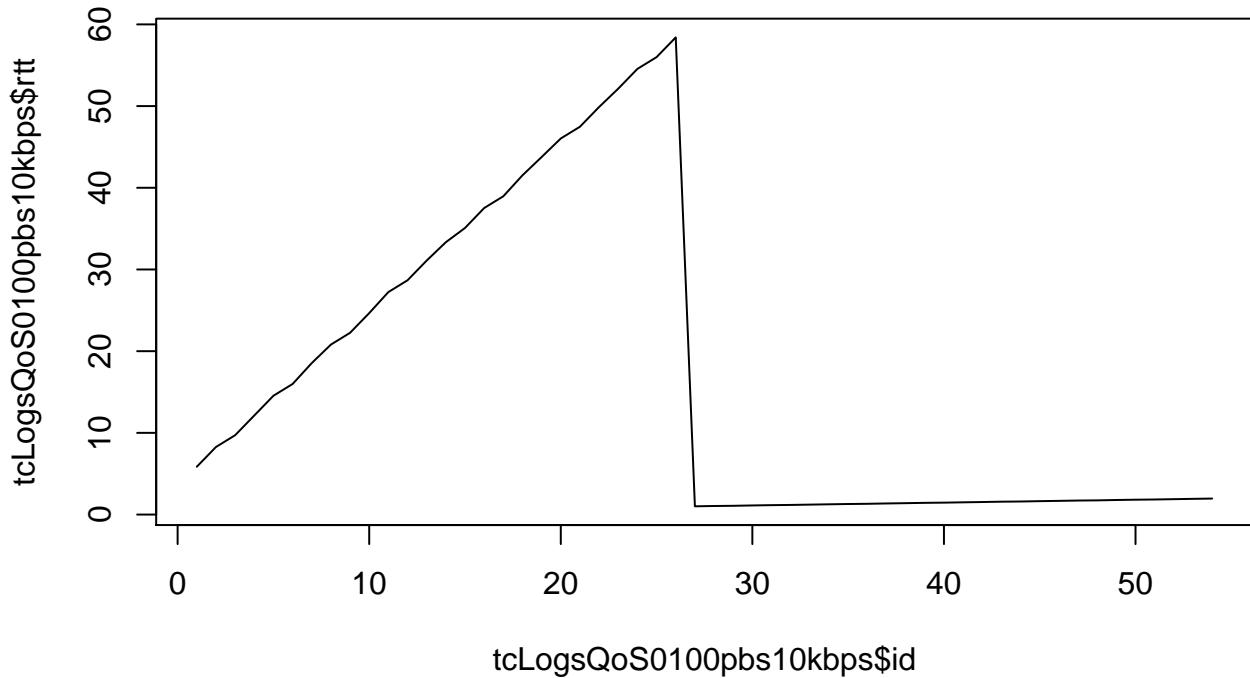
```

#####
# QoS 0 #
#####
## 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",]

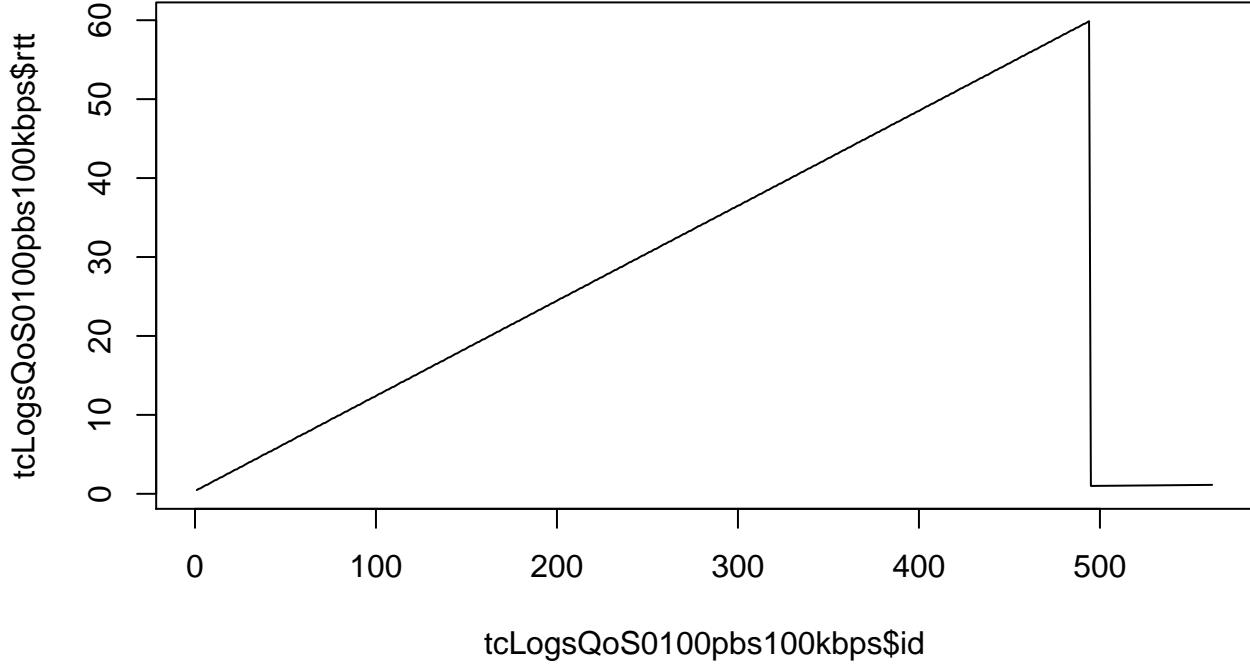
plot(tcLogsQoS0100pbs10kbps$id, tcLogsQoS0100pbs10kbps$rtt, type = "l", main = "QoS_0 100_Pakete/Sek 10")

```

QoS_0 100_Pakete/Sek 10kb_Max

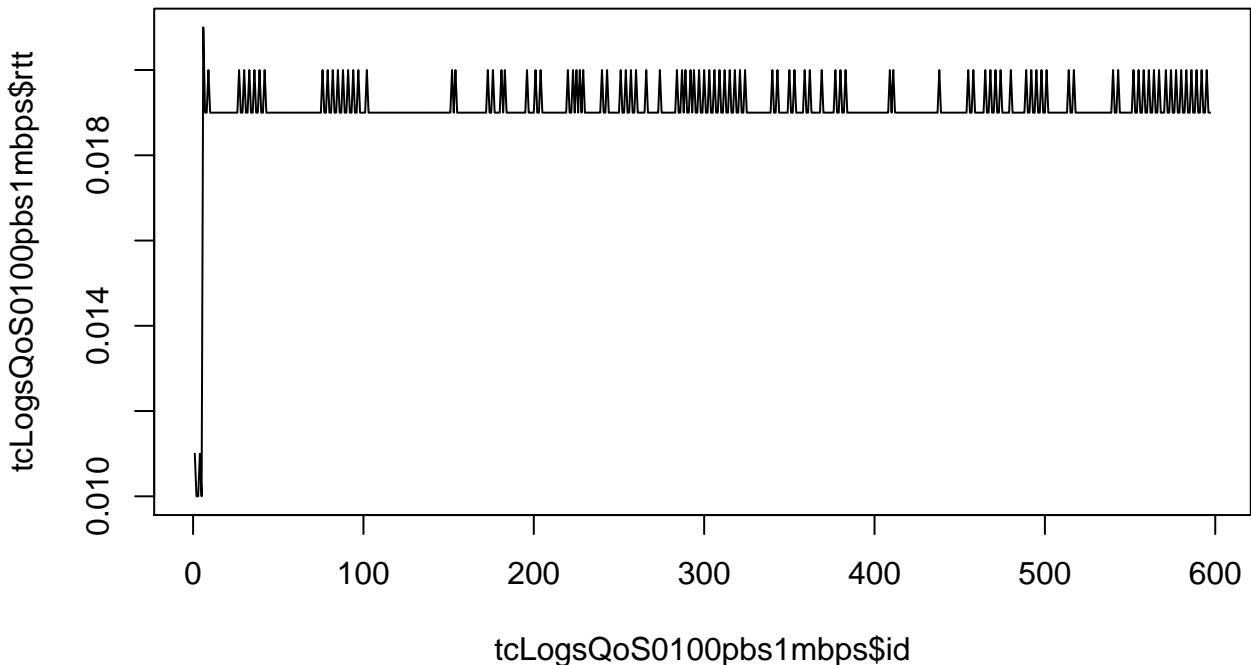


QoS_0 100_Pakete/Sek 100kb_Max



```
plot(tcLogsQoS0100pbs100kbps$id, tcLogsQoS0100pbs100kbps$rtt, type = "l", main = "QoS_0 100_Pakete/Sek 100kb_Max")
```

QoS_0 100_Pakete/Sek 1mb_Max

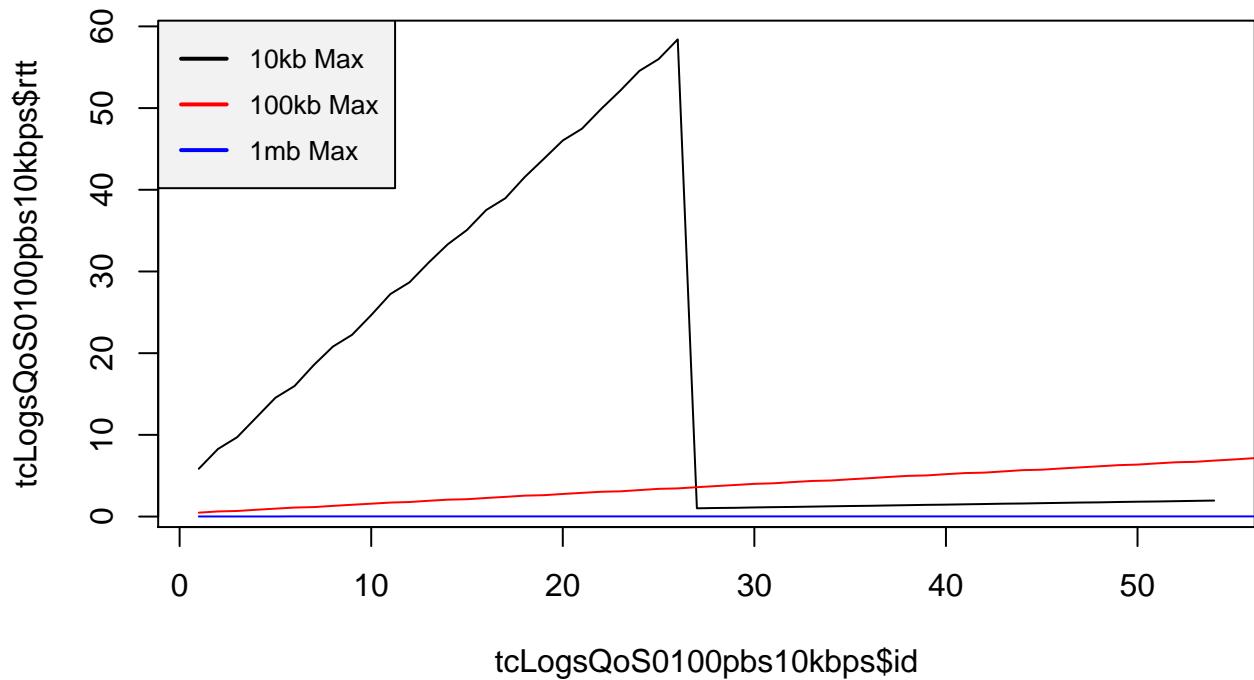


```
#plot(tcLogsQoS0100pbs10mbps$id, tcLogsQoS0100pbs10mbps$rtt, type = "l", main = "QoS_0 100_Pakete/Sek 10mbps")
#plot(tcLogsQoS0100pbs100mbps$id, tcLogsQoS0100pbs100mbps$rtt, type = "l", main = "QoS_0 100_Pakete/Sek 100mbps")

## Eine Grafik
plot(tcLogsQoS0100pbs10kbps$id, tcLogsQoS0100pbs10kbps$rtt, type = "l", main = "QoS_0 100Pakete/Sek nach 10kb Max"),
points(tcLogsQoS0100pbs100kbps$id, tcLogsQoS0100pbs100kbps$rtt, type = "l", col = "red")
points(tcLogsQoS0100pbs1mbps$id, tcLogsQoS0100pbs1mbps$rtt, type = "l", col = "blue")

legend("topleft", c("10kb Max", "100kb Max", "1mb Max"),
       text.width = 7, cex = 0.8,
       col = c("black", "red", "blue", "green4"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")
```

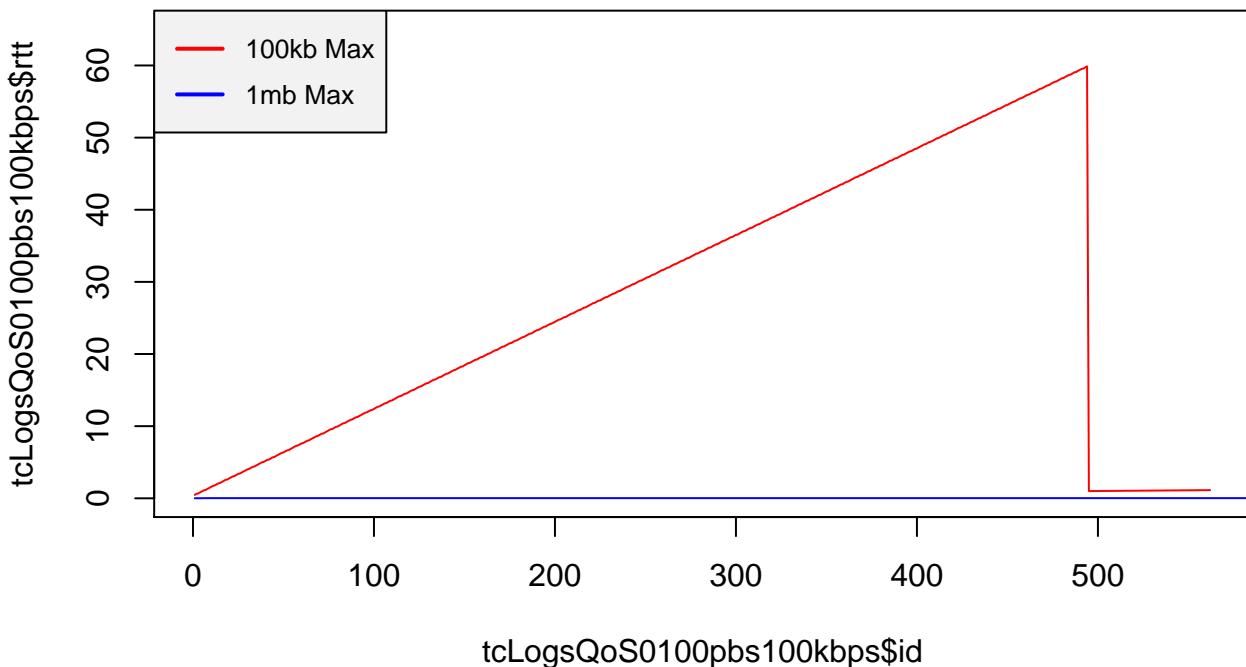
QoS 100Pakete/Sek nach Max (ohne 10mb und 100mb)



```
plot(tcLogsQoS0100pbs100kbps$id, tcLogsQoS0100pbs100kbps$rtt, ylim = c(0,65), type = "l", col = "red", n=1)
points(tcLogsQoS0100pbs1mbps$id, tcLogsQoS0100pbs1mbps$rtt, type = "l", col = "blue")

legend("topleft", c("100kb Max", "1mb Max"), cex = 0.8,
       col = c("red", "blue", "green4"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")
```

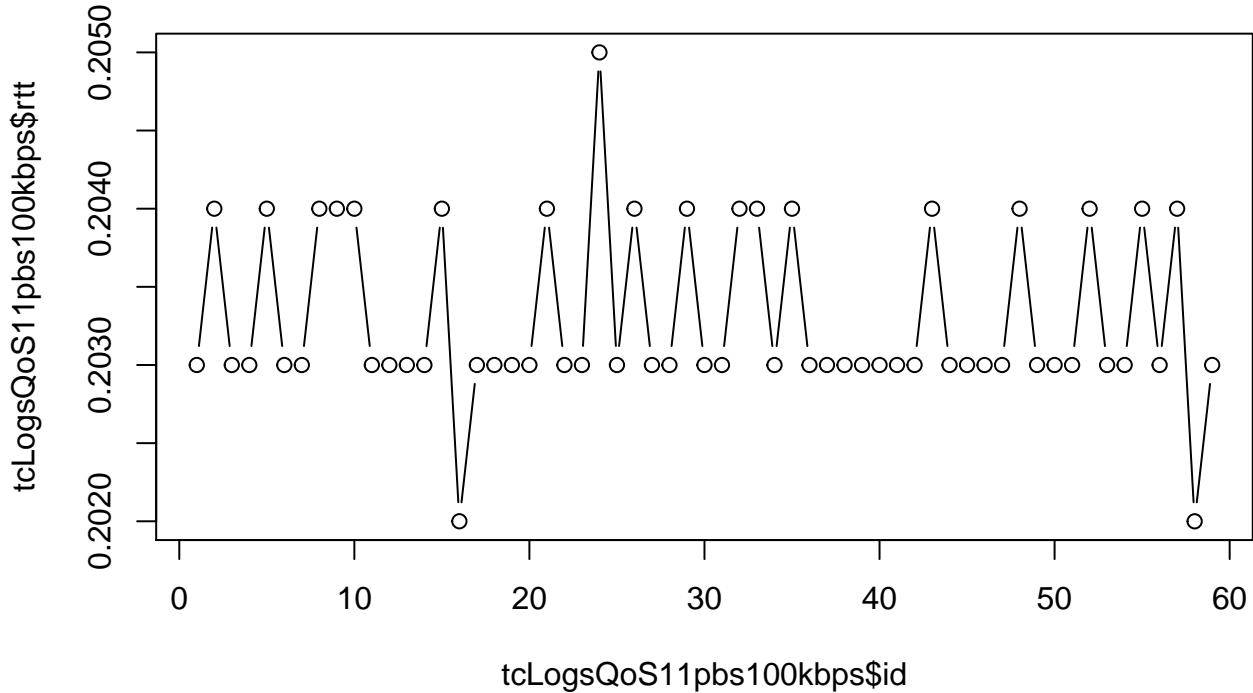
QoS 100Pakete/Sek nach Max (ohne 10kb, 10mb, 100mb)



```
#####
# QoS 1 #
#####
## 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",]

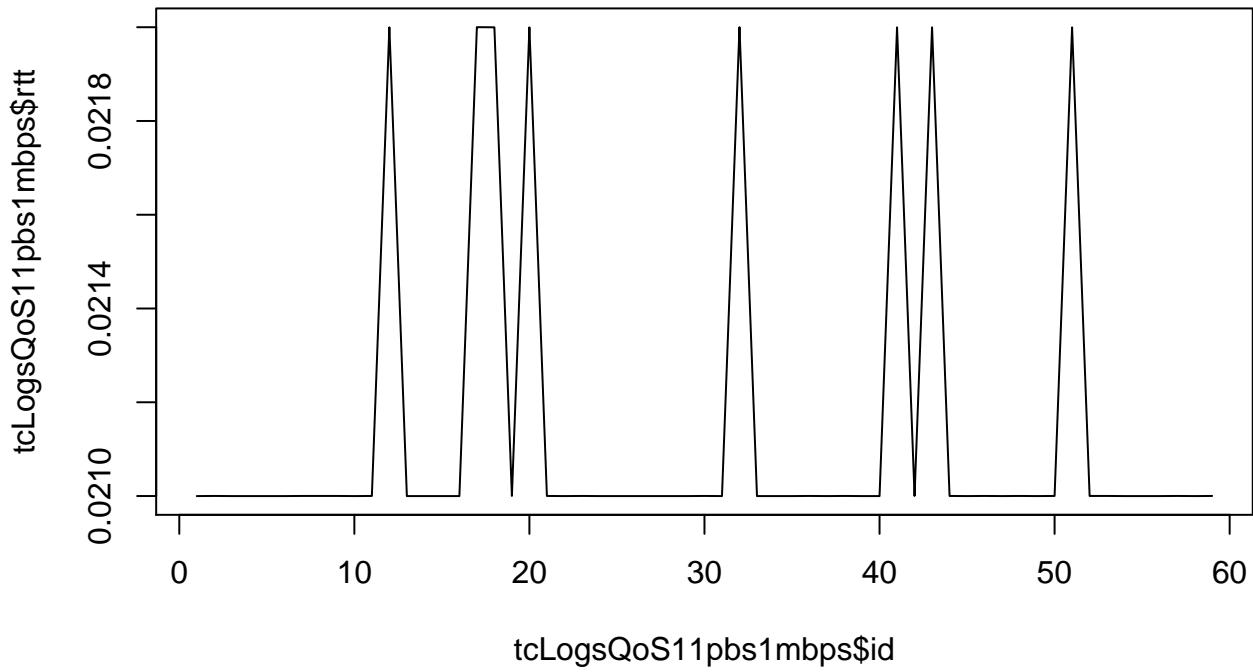
#plot(tcLogsQoS11pbs10kbps$id, tcLogsQoS11pbs10kbps$rtt, type = "l", main = "QoS_1_1_Paket/Sek 10kb_Max")
plot(tcLogsQoS11pbs100kbps$id, tcLogsQoS11pbs100kbps$rtt, type = "b", main = "QoS_1_1_Paket/Sek 100kb_Max")
```

QoS_1 1_Paket/Sek 100kb_Max



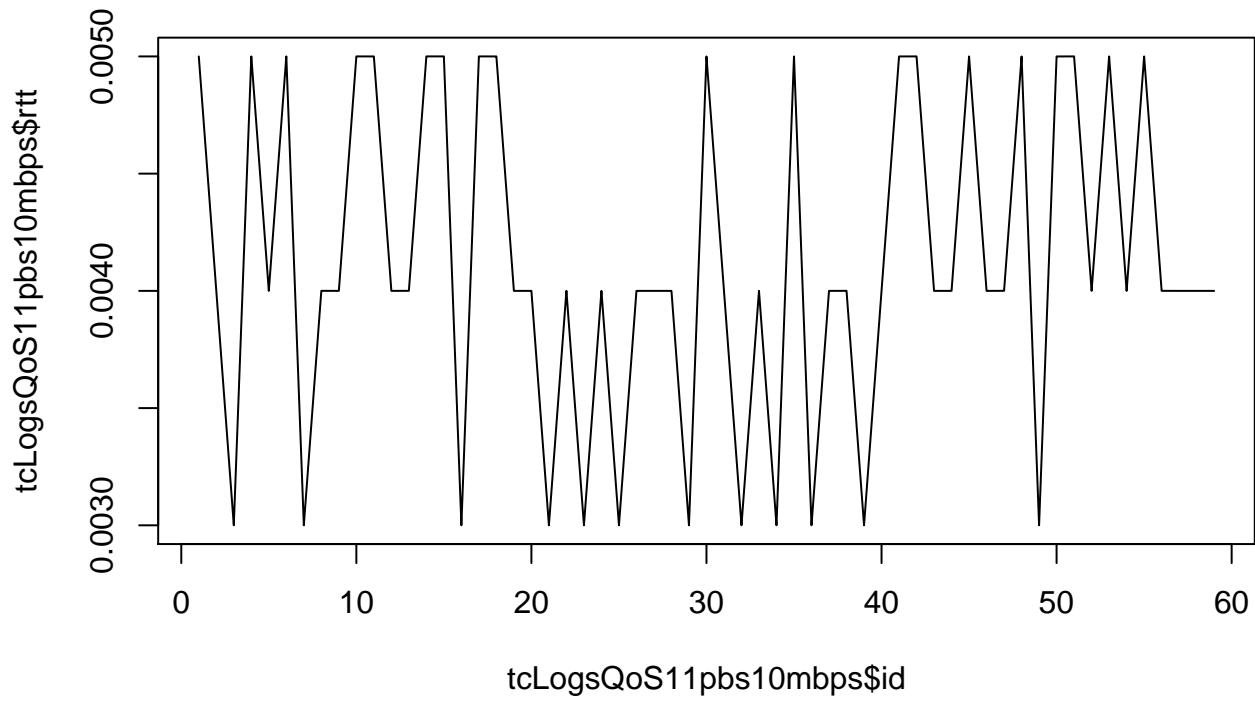
```
plot(tcLogsQoS11pbs1mbps$id, tcLogsQoS11pbs1mbps$rtt, type = "l", main = "QoS_1 1_Paket/Sek 1mb_Max")
```

QoS_1 1_Paket/Sek 1mb_Max



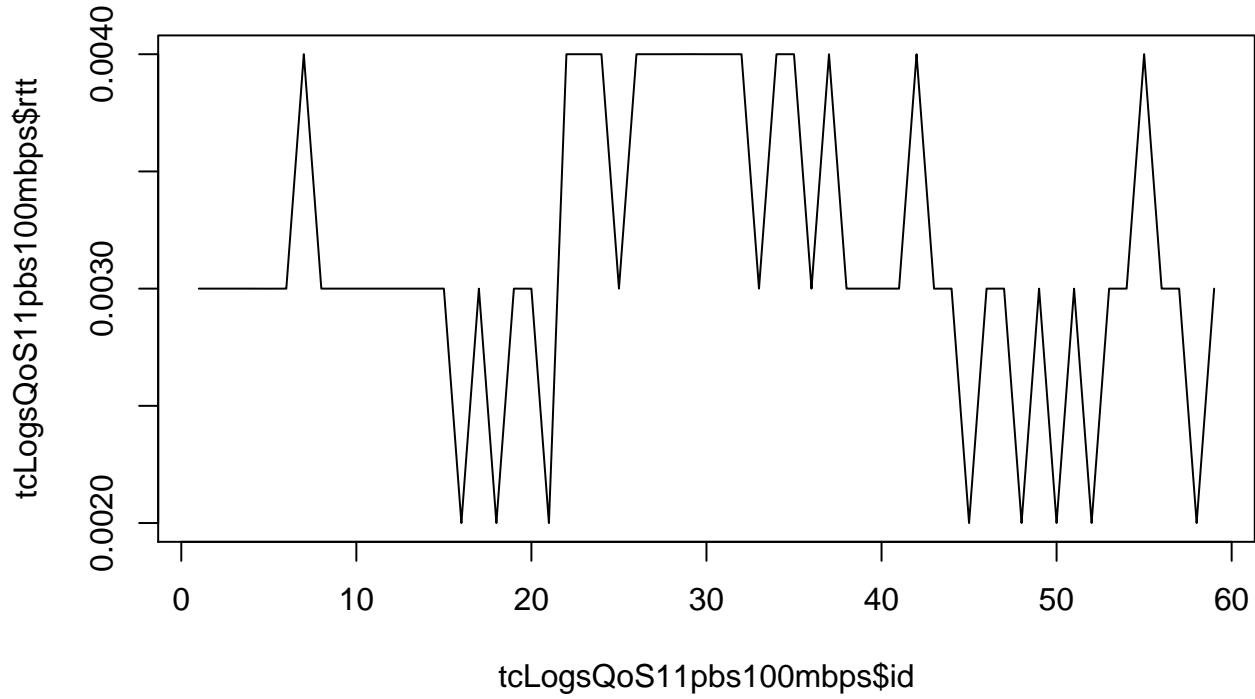
```
plot(tcLogsQoS11pbs10mbps$id, tcLogsQoS11pbs10mbps$rtt, type = "l", main = "QoS_1 1_Paket/Sek 10mb_Max")
```

QoS_1 1_Paket/Sek 10mb_Max



```
plot(tcLogsQoS11pbs100mbps$id, tcLogsQoS11pbs100mbps$rtt, type = "l", main = "QoS_1 1_Paket/Sek 100mb_Max")
```

QoS_1 1_Paket/Sek 100mb_Max



```
## Eine Grafik
plot(tcLogsQoS11pbs100kbps$id, tcLogsQoS11pbs100kbps$rtt, ylim = c(0, 0.25), type = "l", col = "red", main = "QoS_1 1_Paket/Sek 100kbps_Max")
points(tcLogsQoS11pbs1mbps$id, tcLogsQoS11pbs1mbps$rtt, type = "l", col = "blue")
```

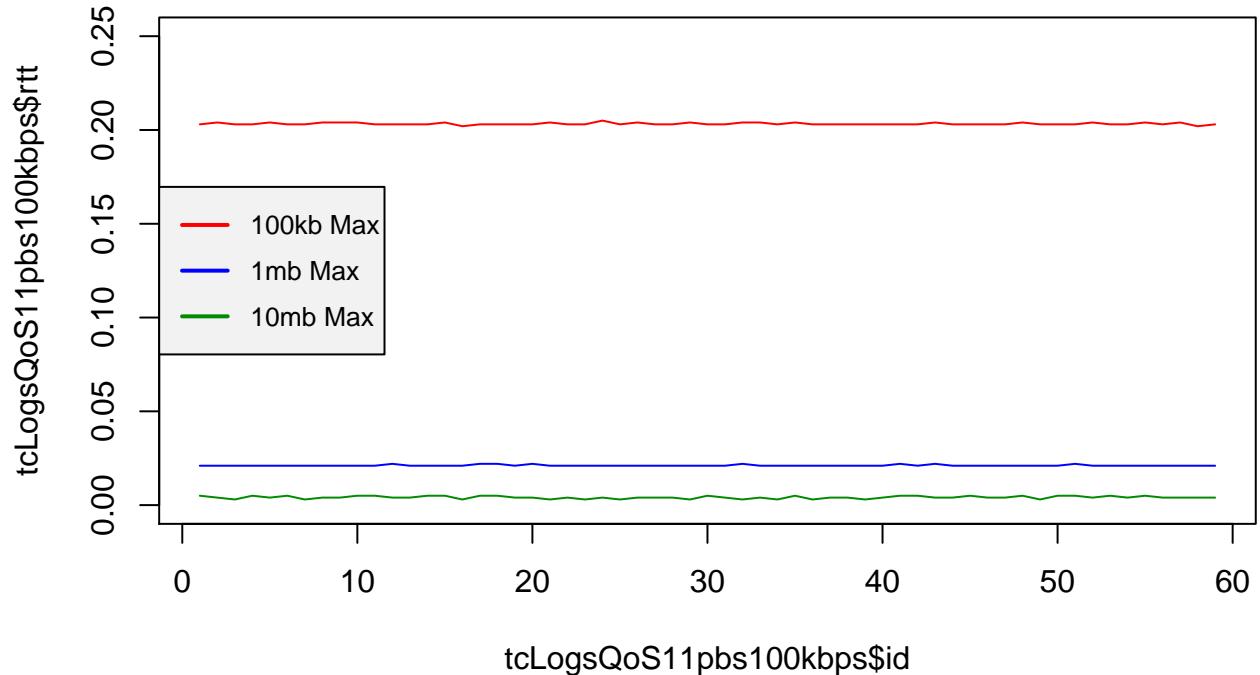
```

points(tcLogsQoS11pbs10mbps$id, tcLogsQoS11pbs10mbps$rtt, type = "l", col = "green4")

legend("left", c("100kb Max", "1mb Max", "10mb Max"),
       text.width = 7, cex = 0.8,
       col = c("red", "blue", "green4"),
       text.col = "black", lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

QoS1 1Paket/Sek aufgeteilt nach Max Durchsatz (ohne 10kb)



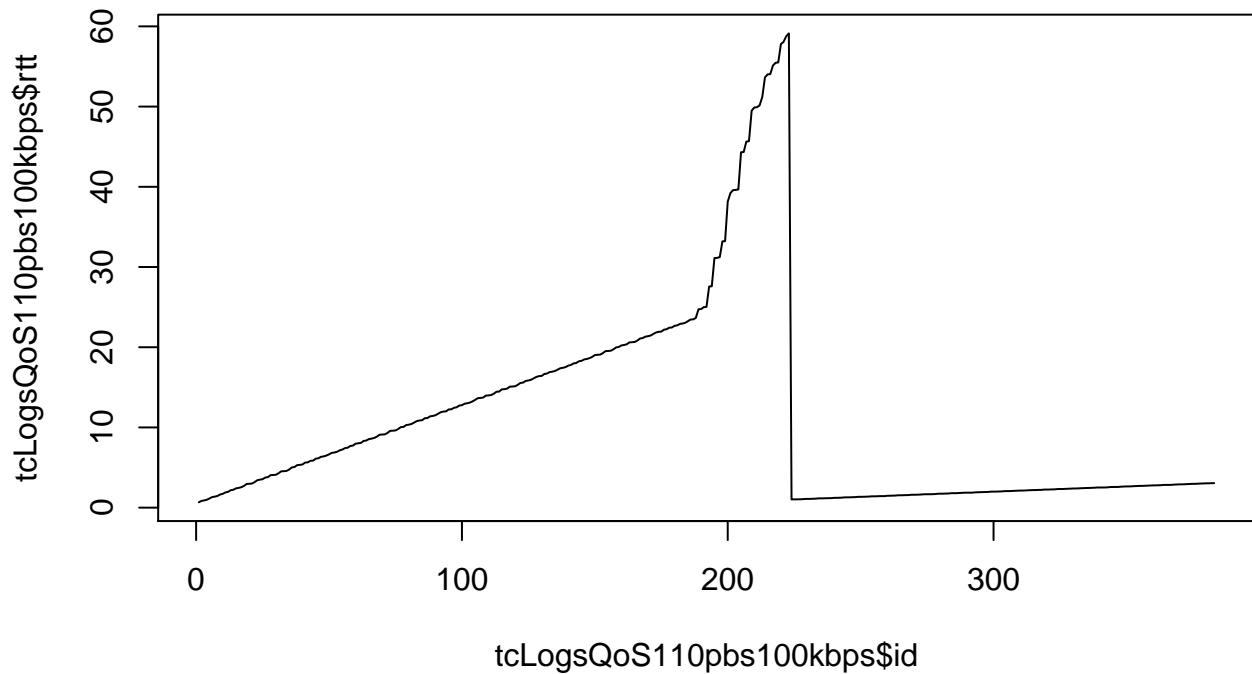
```

#####
# QoS 1 #
#####
## 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", main = "QoS_1_10_Pakete/Sek 10kb")
plot(tcLogsQoS110pbs100kbps$id, tcLogsQoS110pbs100kbps$rtt, type = "l", main = "QoS_1_10_Pakete/Sek 100kb")

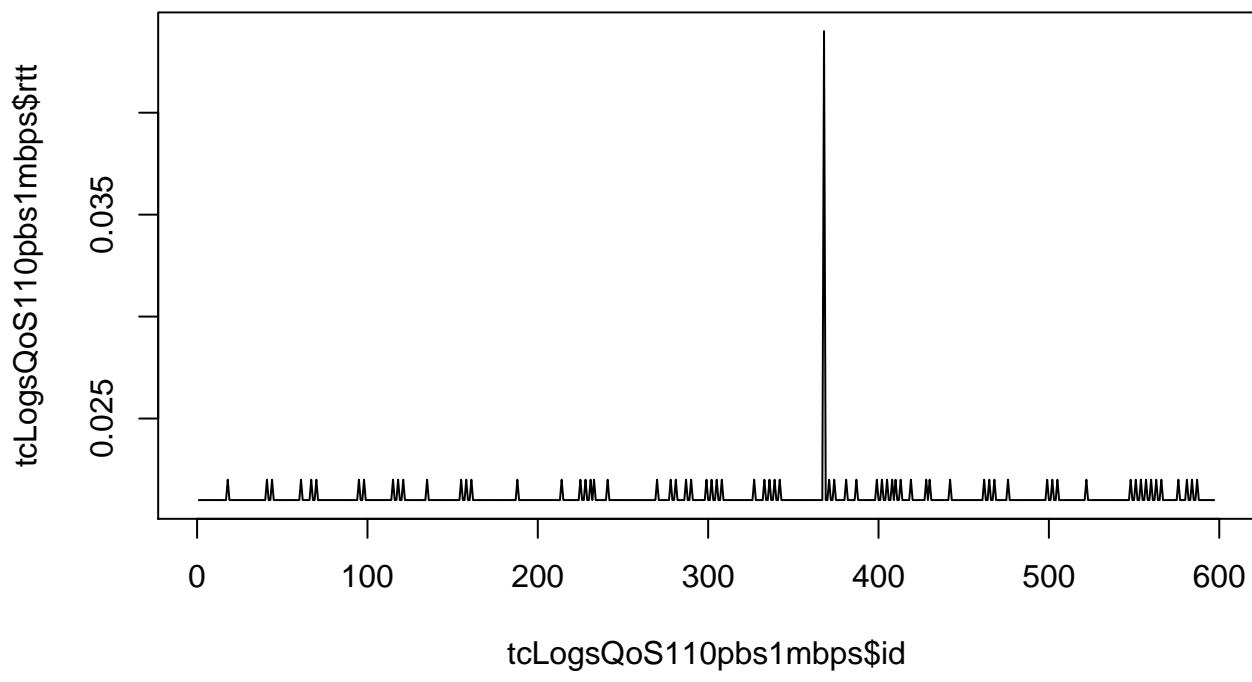
```

QoS_1 10_Pakete/Sek 100kb_Max



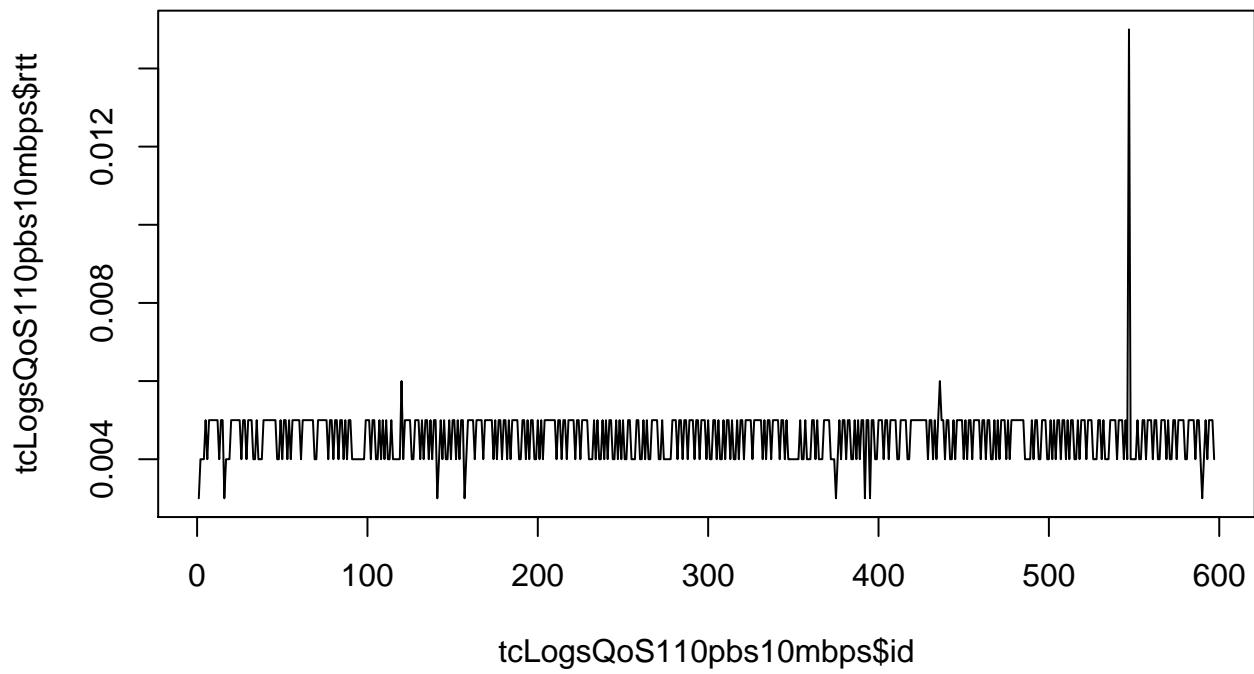
```
plot(tcLogsQoS110pbs1mbps$id, tcLogsQoS110pbs1mbps$rtt, type = "l", main = "QoS_1 10_Pakete/Sek 1mb_Max")
```

QoS_1 10_Pakete/Sek 1mb_Max



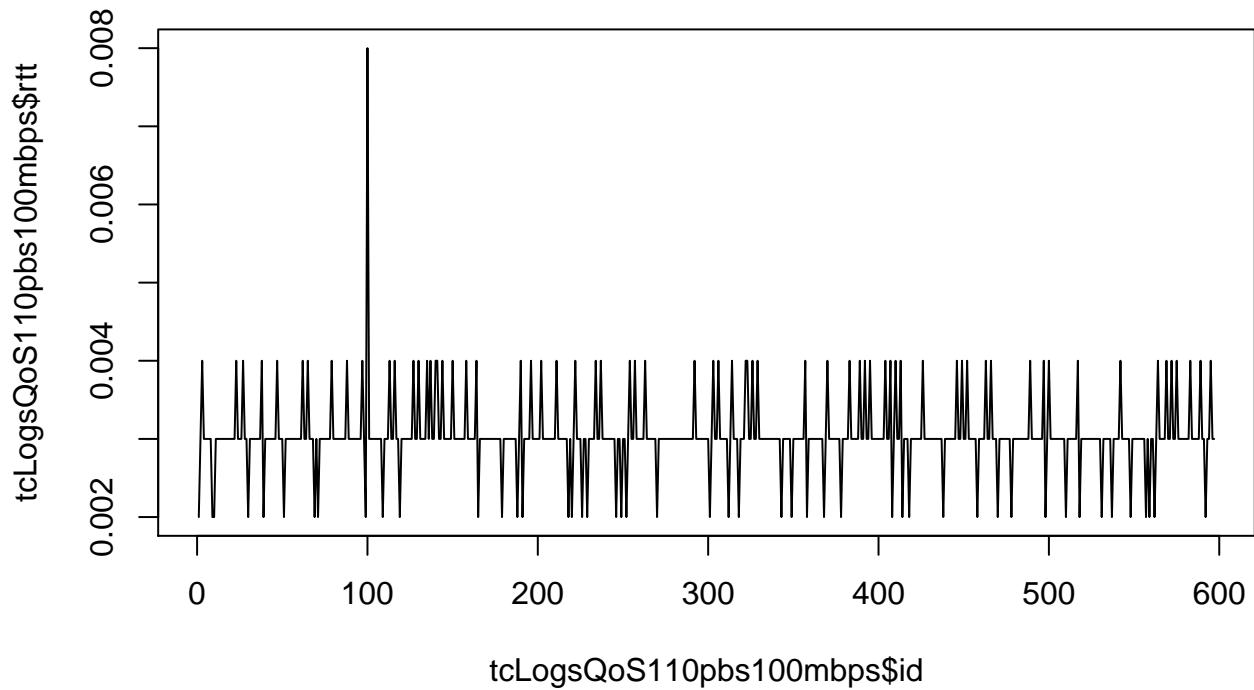
```
plot(tcLogsQoS110pbs10mbps$id, tcLogsQoS110pbs10mbps$rtt, type = "l", main = "QoS_1 10_Pakete/Sek 10mb_Max")
```

QoS_1 10_Pakete/Sek 10mb_Max



```
plot(tcLogsQoS110pbs100mbps$id, tcLogsQoS110pbs100mbps$rtt, type = "l", main = "QoS_1 10_Pakete/Sek 100mb_Max")
```

QoS_1 10_Pakete/Sek 100mb_Max



```
## Eine Grafik  
plot(tcLogsQoS110pbs100kbps$id, tcLogsQoS110pbs100kbps$rtt, ylim = c(0, 60), type = "l", col = "red", main = "QoS_1 10_Pakete/Sek 100kbps_Max")
```

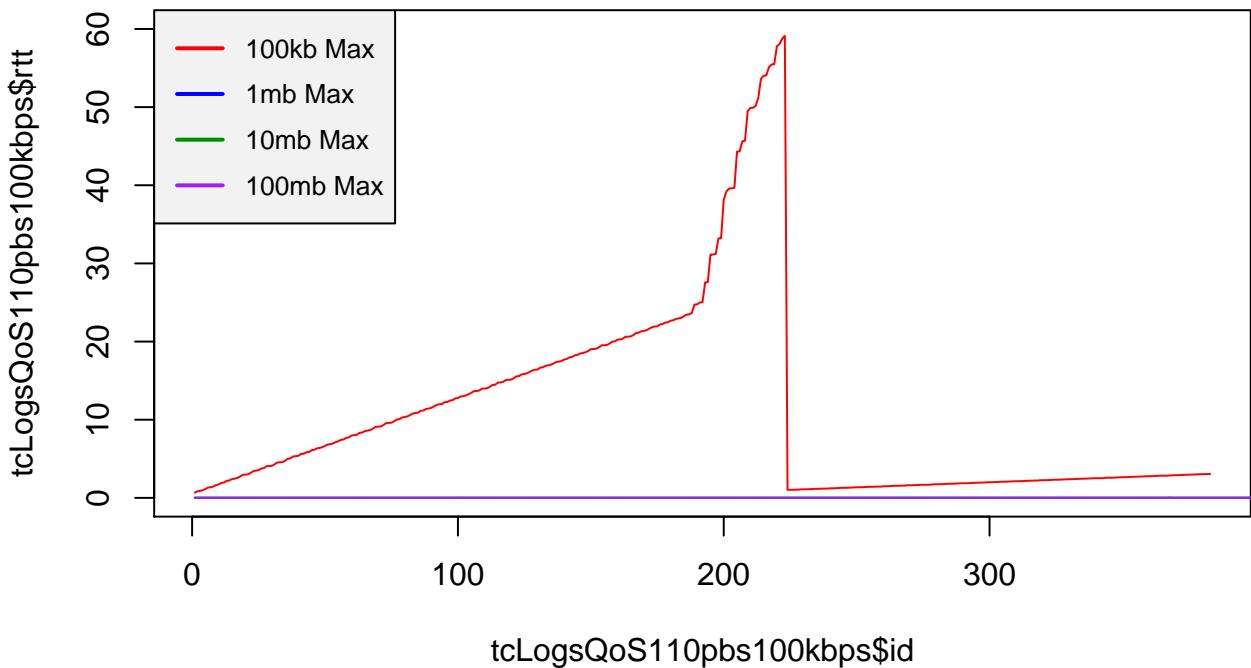
```

points(tcLogsQoS110pbs1mbps$id, tcLogsQoS110pbs1mbps$rtt, type = "l", col = "blue")
points(tcLogsQoS110pbs10mbps$id, tcLogsQoS110pbs10mbps$rtt, type = "l", col = "green4")
points(tcLogsQoS110pbs100mbps$id, tcLogsQoS110pbs100mbps$rtt, type = "l", col = "purple")

legend("topleft", c("100kb Max", "1mb Max", "10mb Max", "100mb Max"),
       cex = 0.8,
       col = c("red", "blue", "green4", "purple"),
       text.col = "black", lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

QoS1 10Pakete/Sek aufgeteilt nach Max Durchsatz (ohne 10kb)



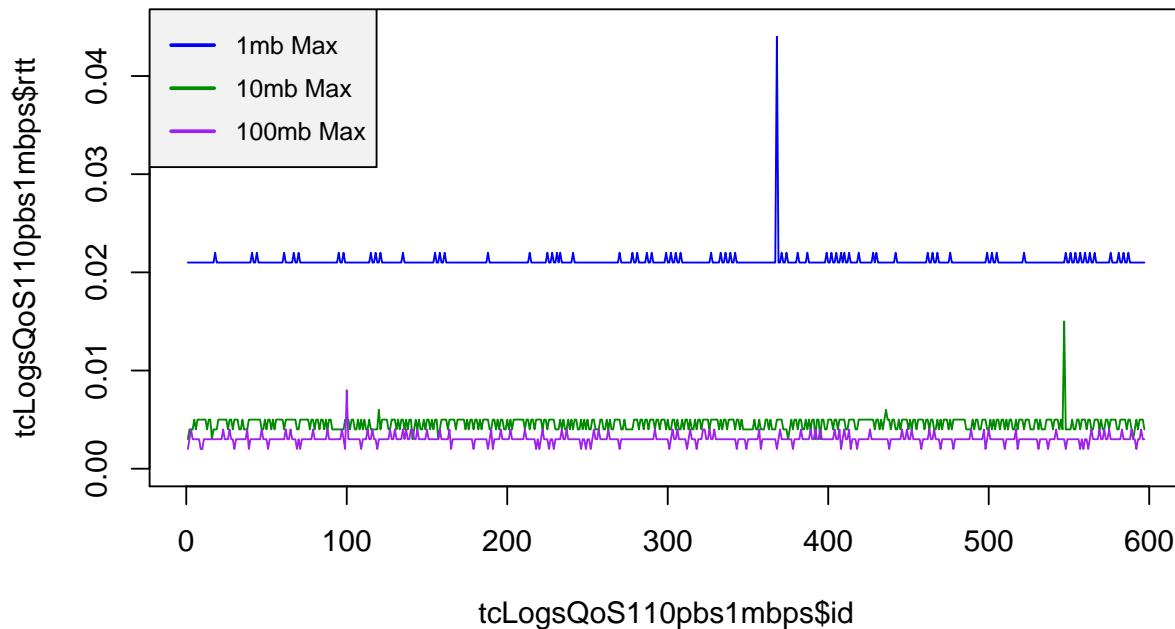
```

plot(tcLogsQoS110pbs1mbps$id, tcLogsQoS110pbs1mbps$rtt, ylim = c(0,0.045), type = "l", col = "blue", max)
points(tcLogsQoS110pbs10mbps$id, tcLogsQoS110pbs10mbps$rtt, type = "l", col = "green4")
points(tcLogsQoS110pbs100mbps$id, tcLogsQoS110pbs100mbps$rtt, type = "l", col = "purple")

legend("topleft", c("1mb Max", "10mb Max", "100mb Max"),
       cex = 0.8,
       col = c("blue", "green4", "purple"),
       text.col = "black", lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

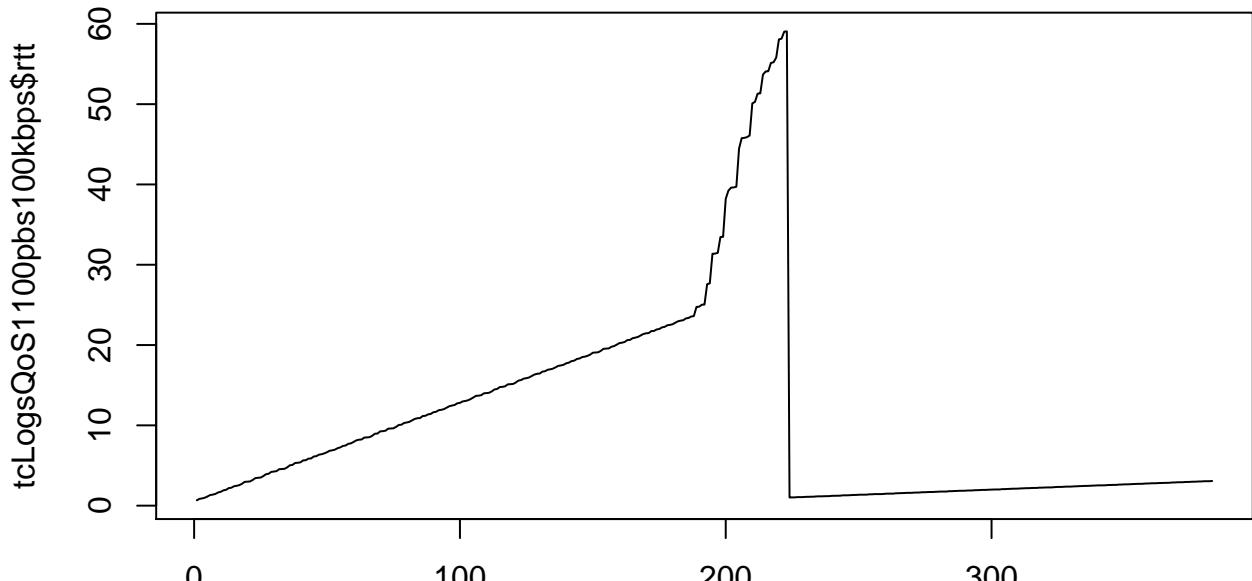
QoS1 10Pakete/Sek aufgeteilt nach Max Durchsatz (ohne 10kb und 100kk)



```
#####
# QoS 1 #
#####
## 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", main = "QoS_1 100_Pakete/Sek 1")
plot(tcLogsQoS1100pbs100kbps$id, tcLogsQoS1100pbs100kbps$rtt, type = "l", main = "QoS_1 100_Pakete/Sek 1")
```

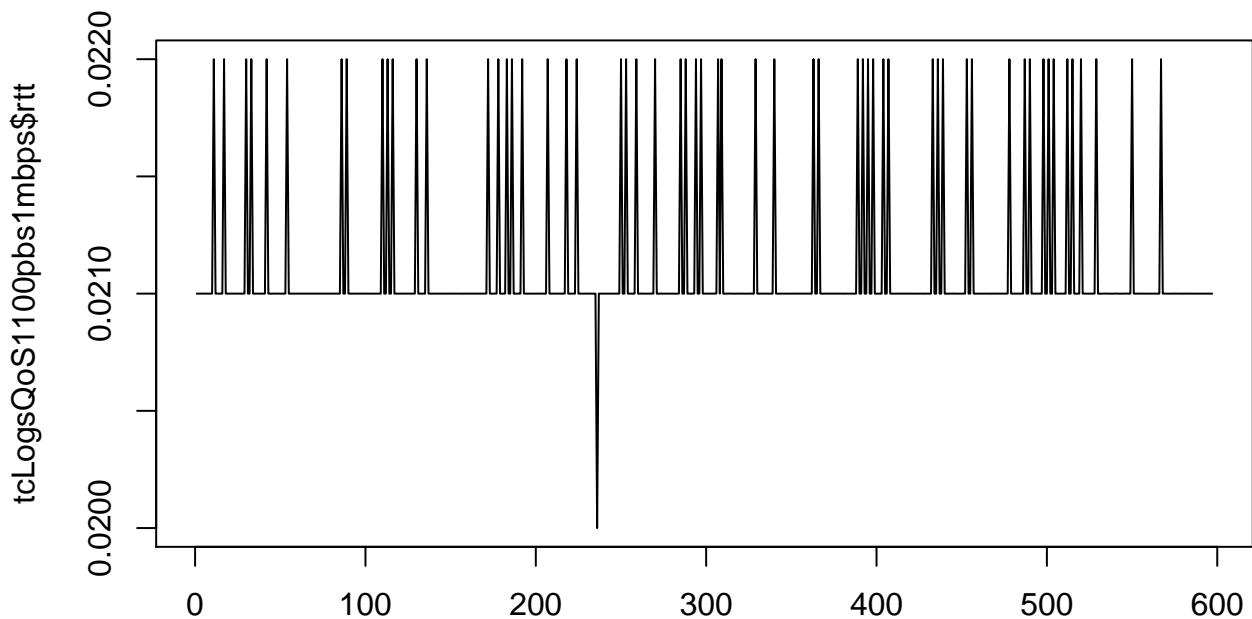
QoS_1 100_Pakete/Sek 100kb_Max



tcLogsQoS1100pbs100kbps\$id

```
plot(tcLogsQoS1100pbs1mbps$id, tcLogsQoS1100pbs1mbps$rtt, type = "l", main = "QoS_1 100_Pakete/Sek 1mb_Max")
```

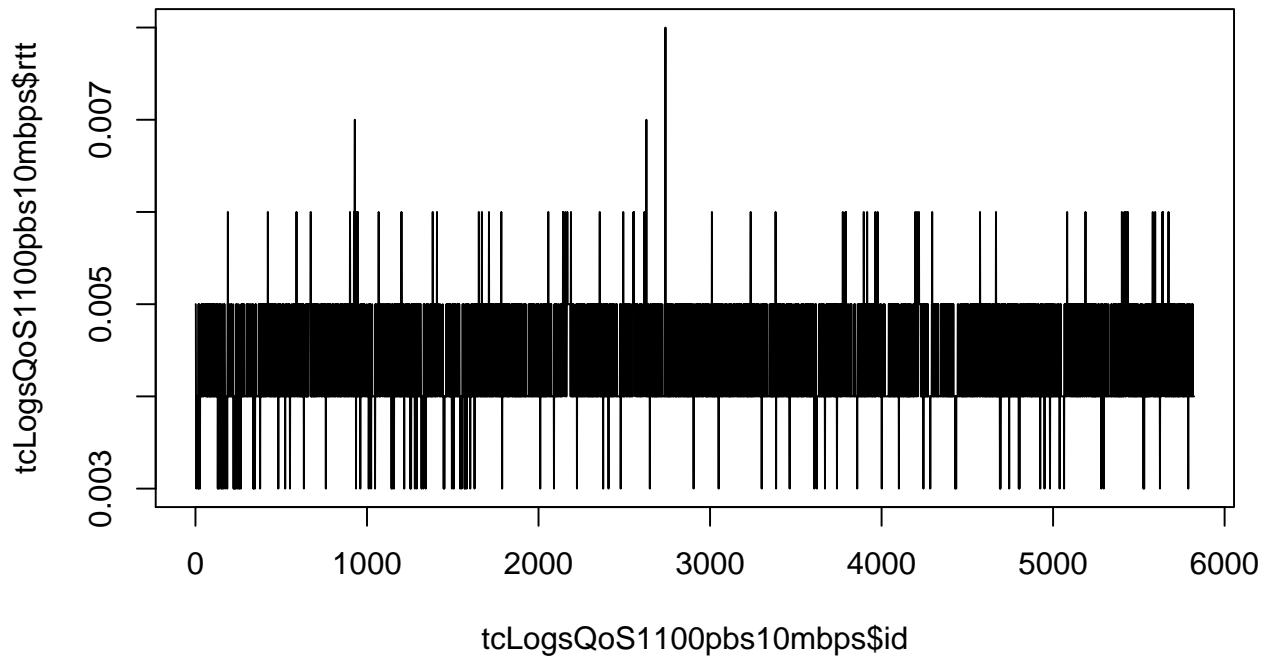
QoS_1 100_Pakete/Sek 1mb_Max



tcLogsQoS1100pbs1mbps\$id

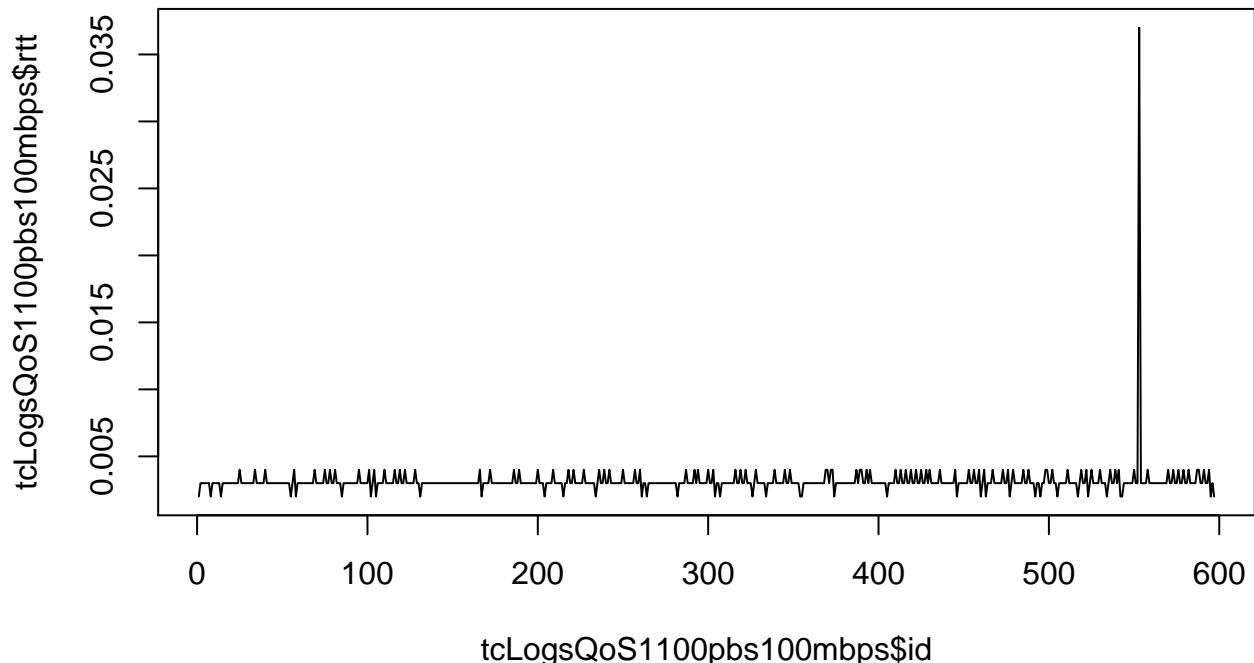
```
plot(tcLogsQoS1100pbs10mbps$id, tcLogsQoS1100pbs10mbps$rtt, type = "l", main = "QoS_1 100_Pakete/Sek 10mb_Max")
```

QoS_1 100_Pakete/Sek 10mb_Max



```
plot(tcLogsQoS1100pbs100mbps$id, tcLogsQoS1100pbs100mbps$rtt, type = "l", main = "QoS_1 100_Pakete/Sek 10mb_Max")
```

QoS_1 100_Pakete/Sek 100mb_Max



```
## Eine Grafik  
plot(tcLogsQoS110pbs100kbps$id, tcLogsQoS110pbs100kbps$rtt, ylim = c(0, 60), type = "l", col = "red", main = "QoS_1 100_Pakete/Sek 100mb_Max")  
points(tcLogsQoS1100pbs1mbps$id, tcLogsQoS1100pbs1mbps$rtt, type = "l", col = "blue")
```

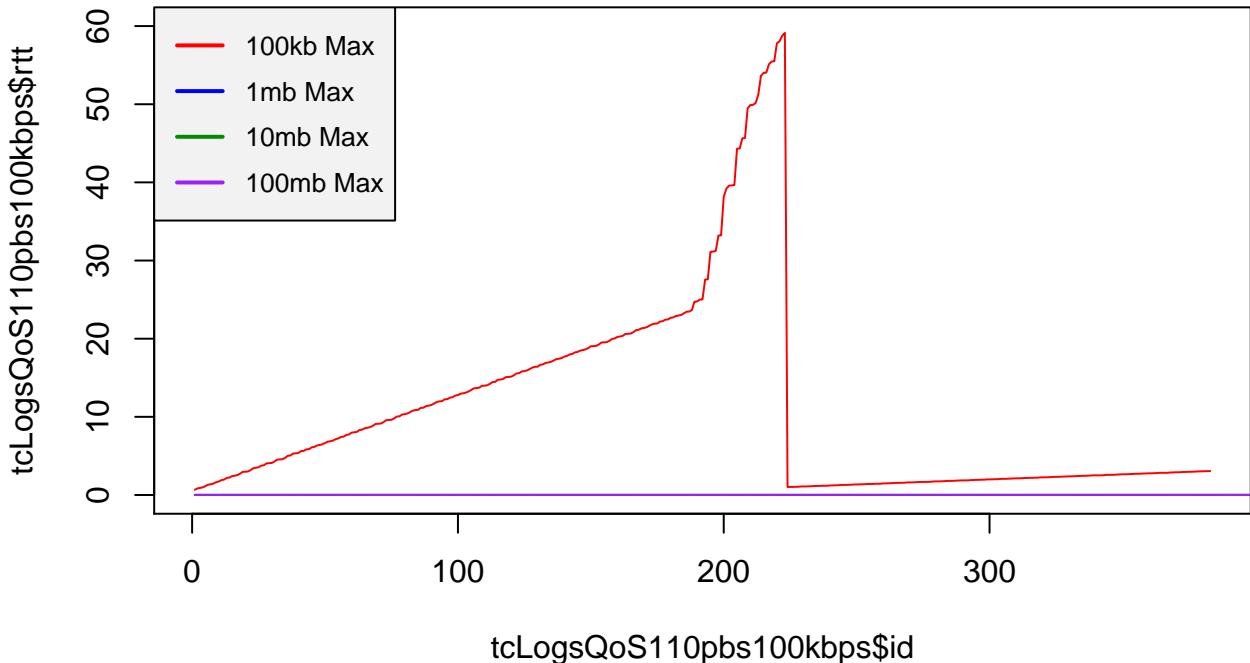
```

points(tcLogsQoS1100pbs10mbps$id, tcLogsQoS1100pbs10mbps$rtt, type = "l", col = "green4")
points(tcLogsQoS1100pbs100mbps$id, tcLogsQoS1100pbs100mbps$rtt, type = "l", col = "purple")

legend("topleft", c("100kb Max", "1mb Max", "10mb Max", "100mb Max"),
       cex = 0.8,
       col = c("red", "blue", "green4", "purple"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

QoS1 100Pakete/Sek aufgeteilt nach Max Durchsatz



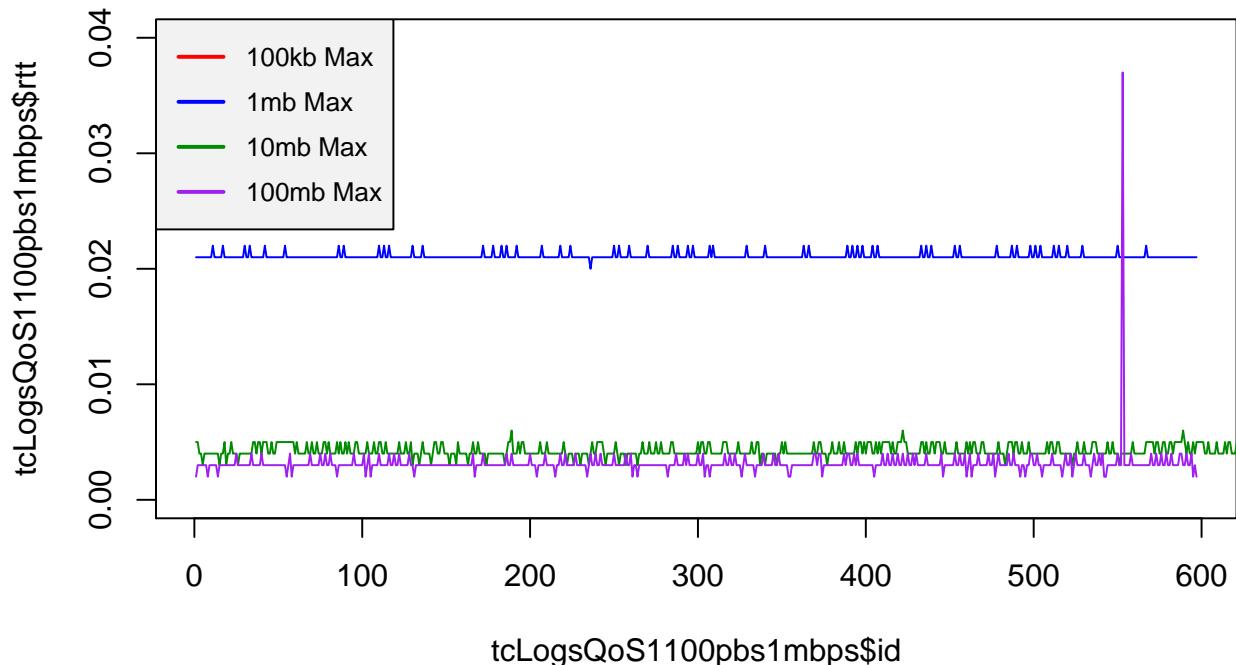
```

plot(tcLogsQoS1100pbs1mbps$id, tcLogsQoS1100pbs1mbps$rtt, ylim = c(0, 0.04), type = "l", col = "blue", n=1)
points(tcLogsQoS1100pbs10mbps$id, tcLogsQoS1100pbs10mbps$rtt, type = "l", col = "green4")
points(tcLogsQoS1100pbs100mbps$id, tcLogsQoS1100pbs100mbps$rtt, type = "l", col = "purple")

legend("topleft", c("100kb Max", "1mb Max", "10mb Max", "100mb Max"),
       cex = 0.8,
       col = c("red", "blue", "green4", "purple"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

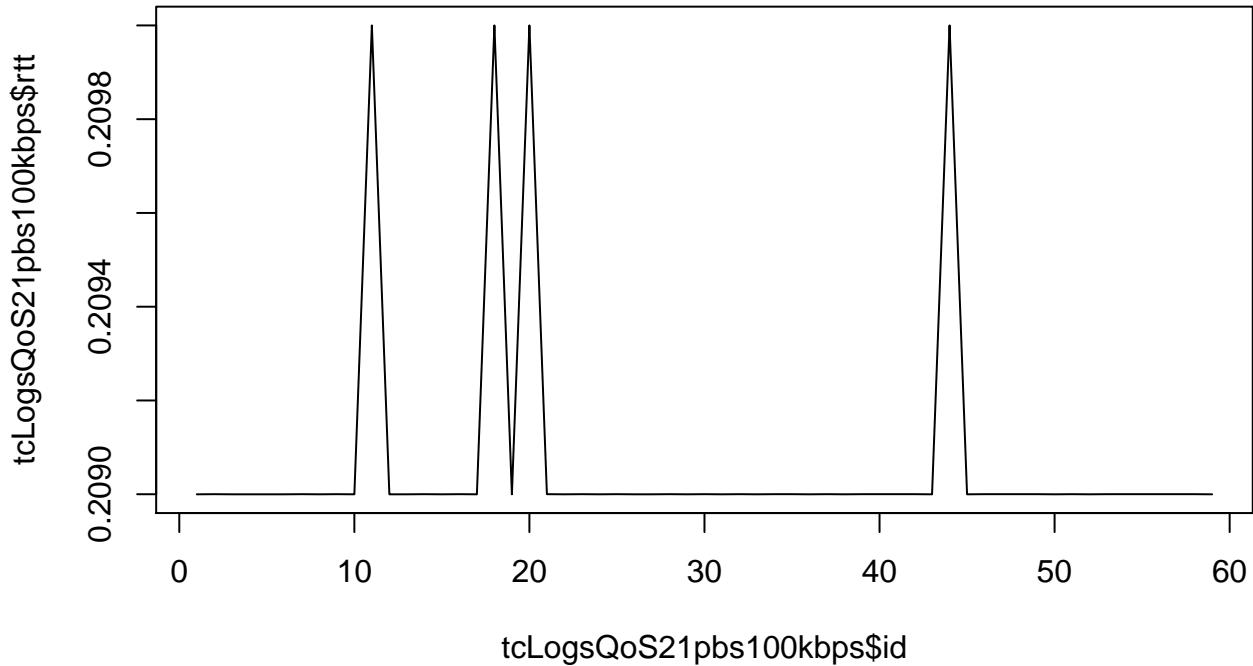
QoS1 100Pakete/Sek aufgeteilt nach Max Durchsatz (ohne 100kb)



```
#####
# QoS 2 #
#####
## 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", ]

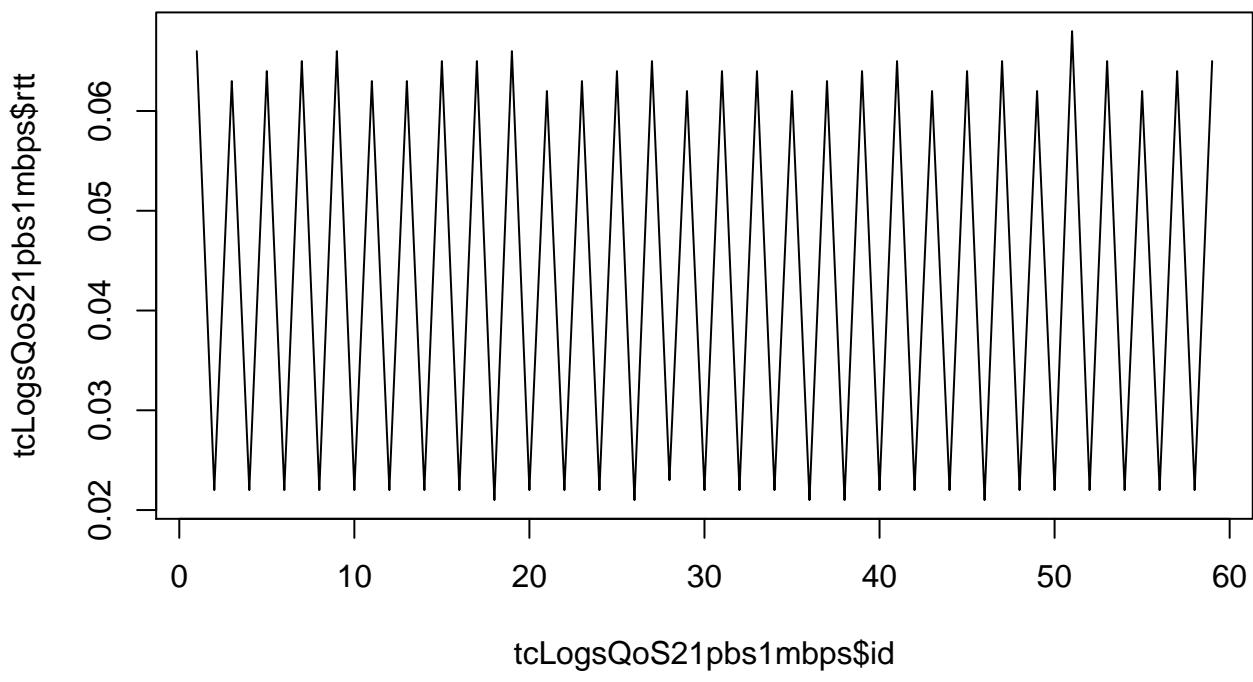
#plot(tcLogsQoS21pbs10kbps$id, tcLogsQoS21pbs10kbps$rtt, type = "l", main = "QoS_2_1_Pakete/Sek 10kb_Max")
plot(tcLogsQoS21pbs100kbps$id, tcLogsQoS21pbs100kbps$rtt, type = "l", main = "QoS_2_1_Pakete/Sek 100kb_Max")
```

QoS_2 1_Pakete/Sek 100kb_Max



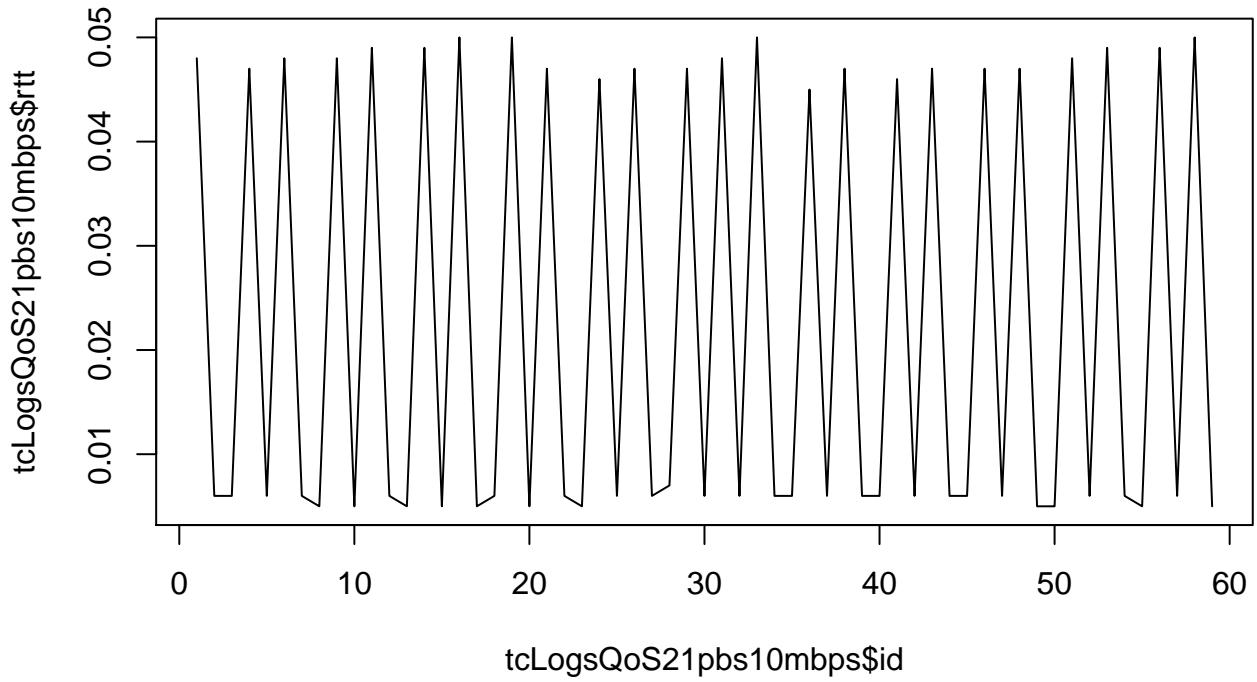
```
plot(tcLogsQoS21pbs1mbps$id, tcLogsQoS21pbs1mbps$rtt, type = "l", main = "QoS_2 1_Pakete/Sek 1mb_Max")
```

QoS_2 1_Pakete/Sek 1mb_Max

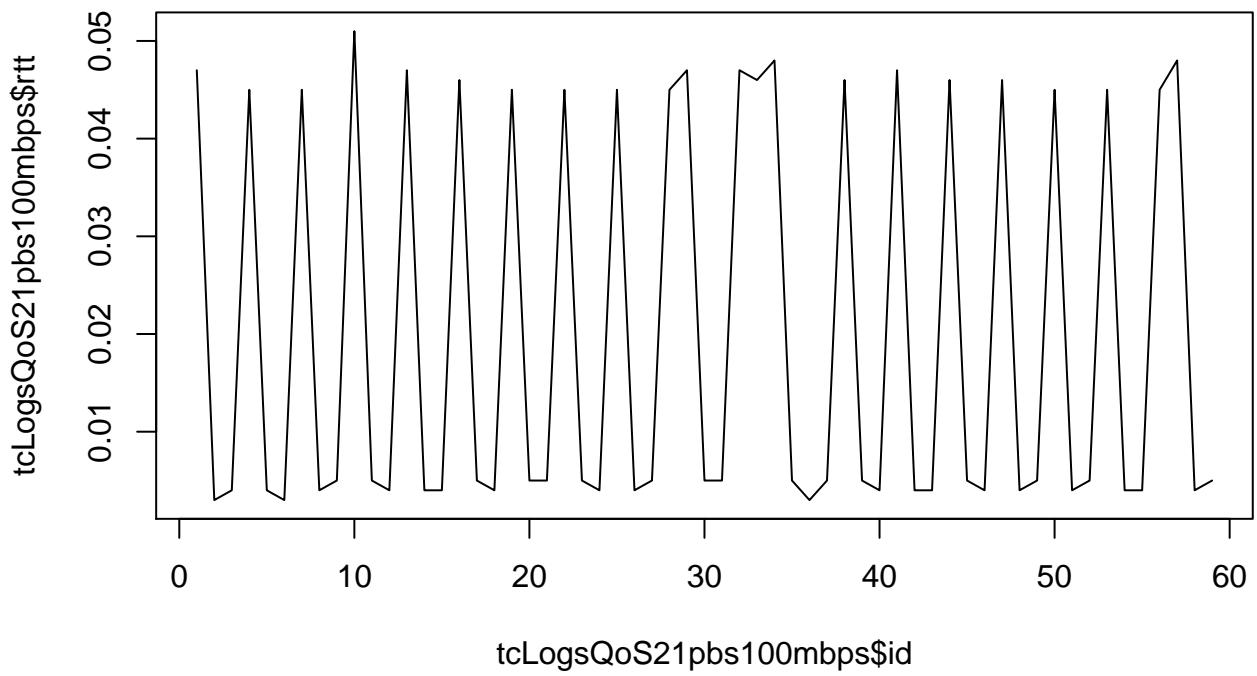


```
plot(tcLogsQoS21pbs10mbps$id, tcLogsQoS21pbs10mbps$rtt, type = "l", main = "QoS_2 1_Pakete/Sek 10mb_Max")
```

QoS_2 1_Pakete/Sek 10mb_Max



QoS_2 1_Pakete/Sek 100mb_Max



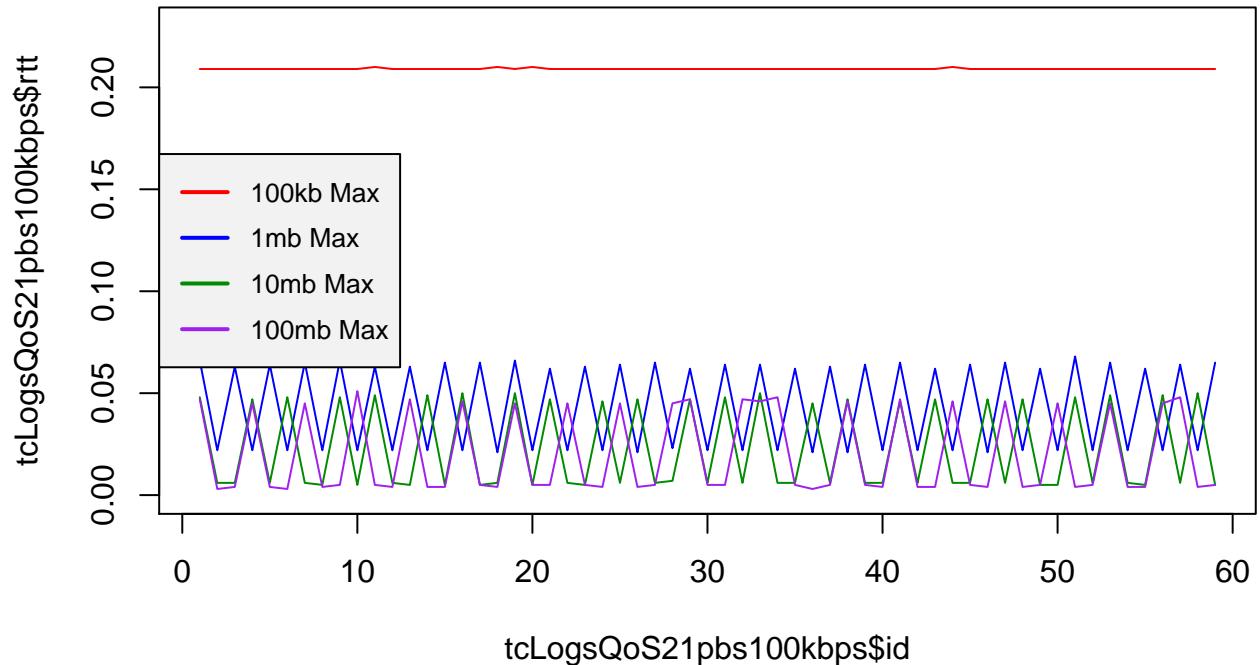
```

points(tcLogsQoS21pbs100mbps$id, tcLogsQoS21pbs100mbps$rtt, type = "l", col = "purple")

legend("left", c("100kb Max", "1mb Max", "10mb Max", "100mb Max"),
       cex = 0.8,
       col = c("red", "blue", "green4", "purple"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

QoS2 1Paket/Sek aufgeteilt nach Max Durchsatz (ohne 10kb)



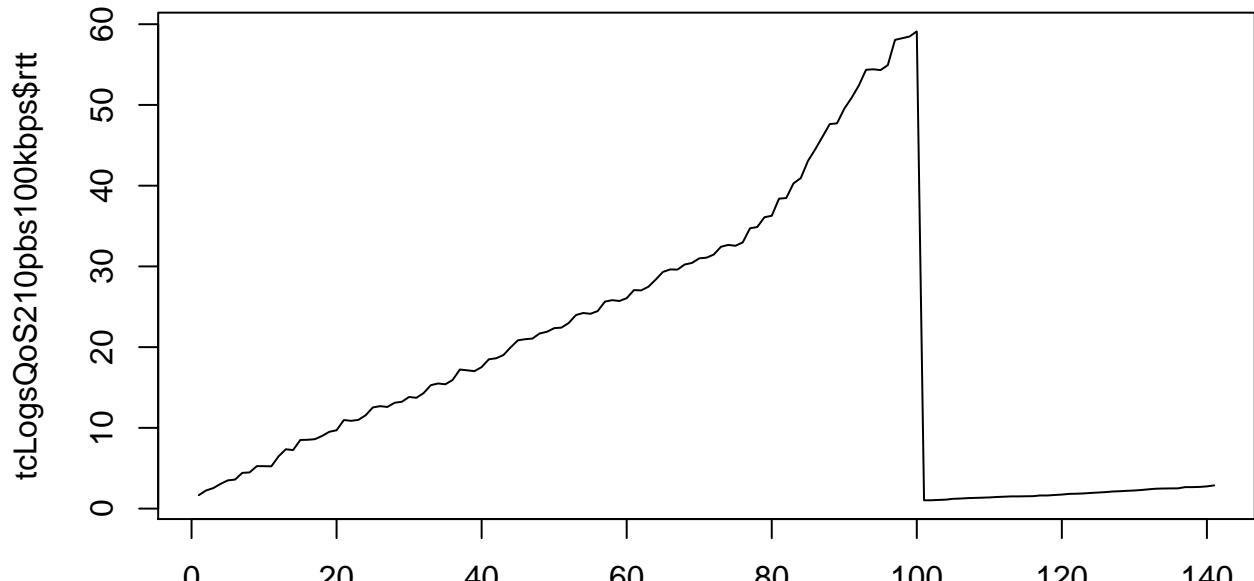
```

#####
# QoS 2 #
#####
## 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", main = "QoS_2 10_Pakete/Sek 10kb")
plot(tcLogsQoS210pbs100kbps$id, tcLogsQoS210pbs100kbps$rtt, type = "l", main = "QoS_2 10_Pakete/Sek 100kb")

```

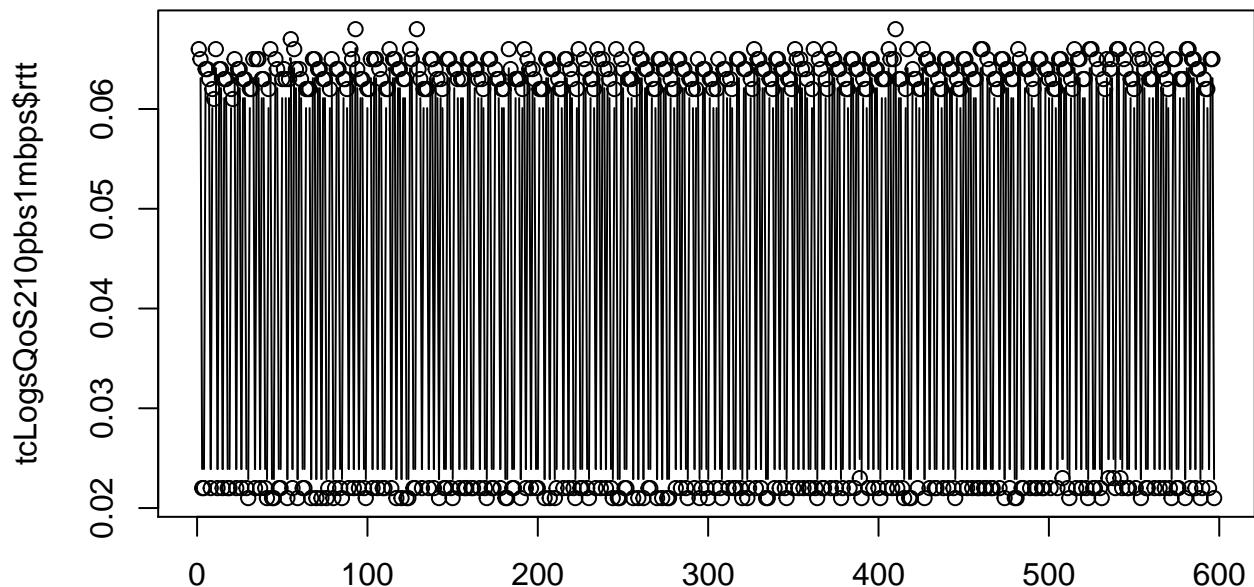
QoS_2 10_Pakete/Sek 100kb_Max



`tcLogsQoS210pbs100kbps$id`

```
plot(tcLogsQoS210pbs1mbps$id, tcLogsQoS210pbs1mbps$rtt, type = "b", main = "QoS_2 10_Pakete/Sek 1mb_Max")
```

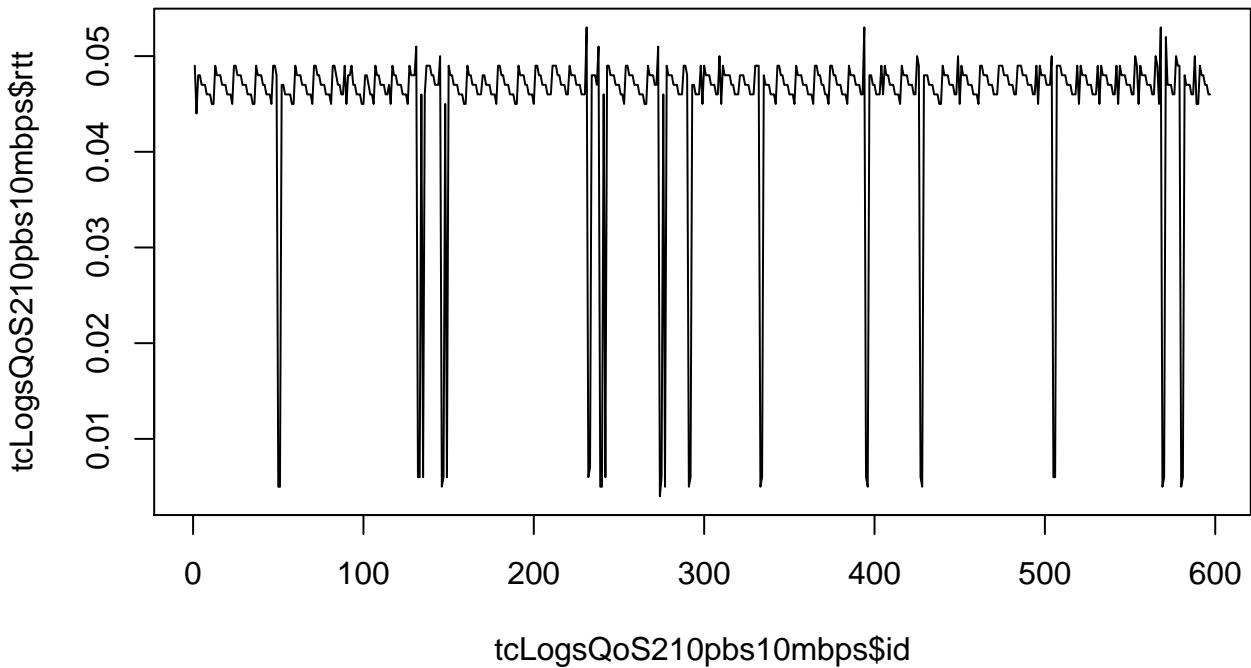
QoS_2 10_Pakete/Sek 1mb_Max



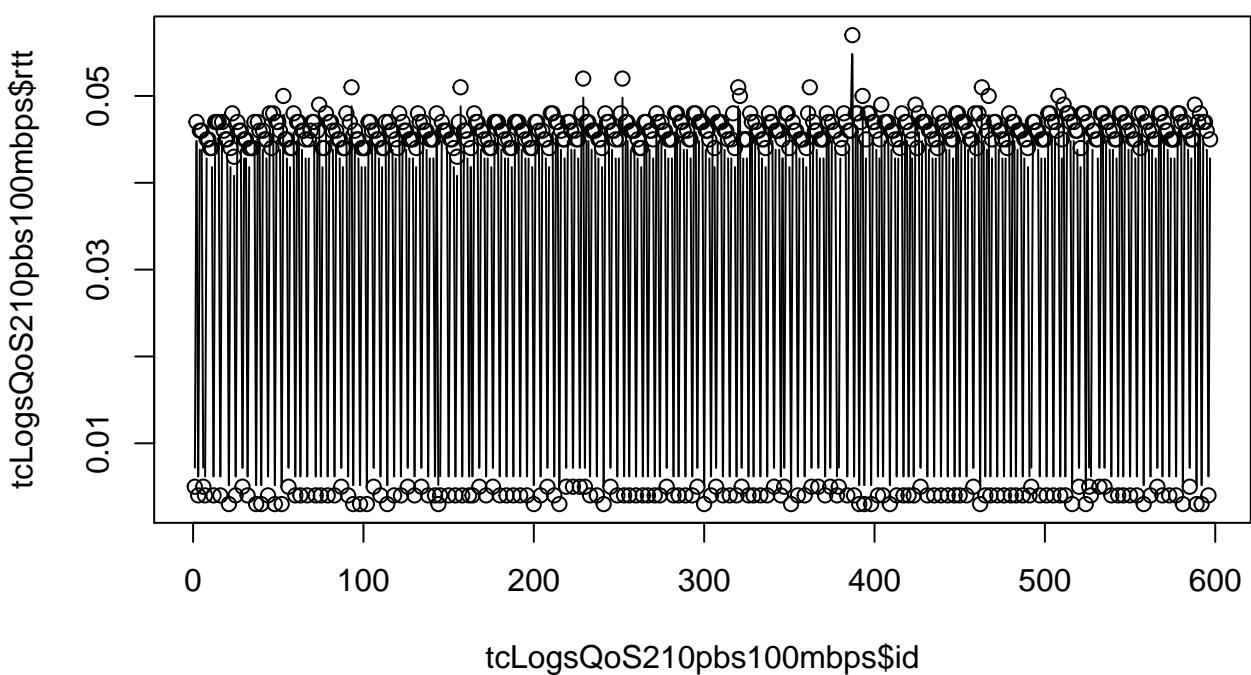
`tcLogsQoS210pbs1mbps$id`

```
plot(tcLogsQoS210pbs10mbps$id, tcLogsQoS210pbs10mbps$rtt, type = "l", main = "QoS_2 10_Pakete/Sek 10mb_Max")
```

QoS_2 10_Pakete/Sek 10mb_Max



QoS_2 10_Pakete/Sek 100mb_Max



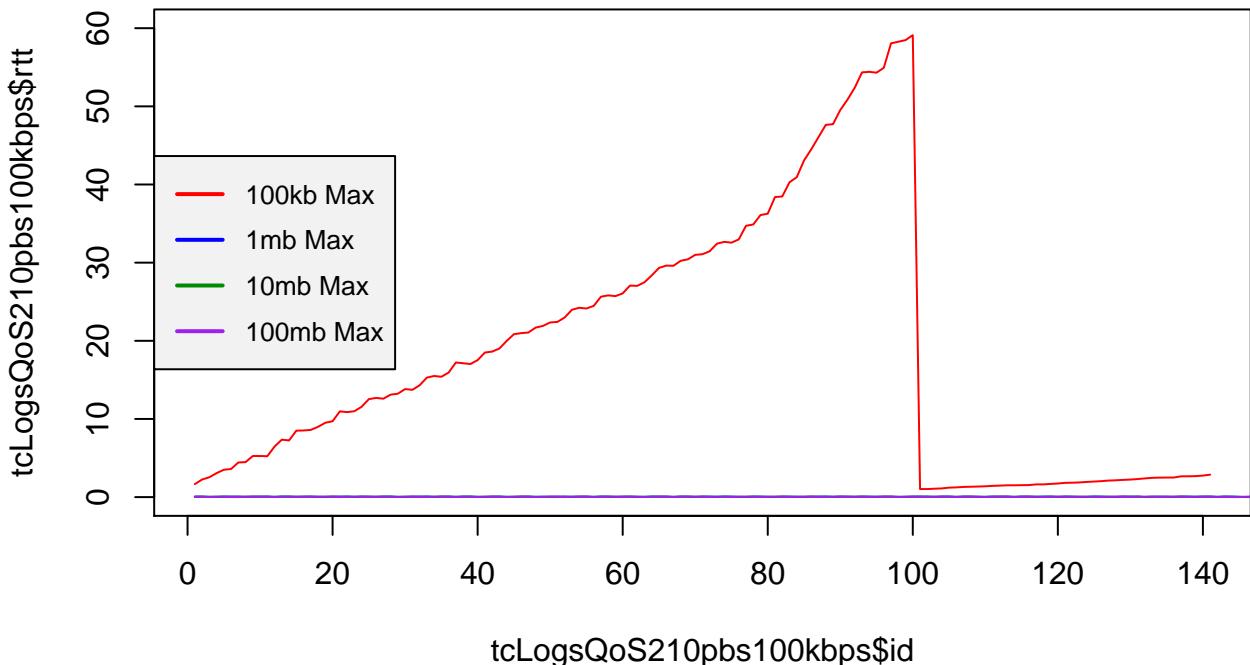
```

points(tcLogsQoS210pbs100mbps$id, tcLogsQoS210pbs100mbps$rtt, type = "l", col = "purple")

legend("left", c("100kb Max", "1mb Max", "10mb Max", "100mb Max"),
       cex = 0.8,
       col = c("red", "blue", "green4", "purple"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

QoS2 10Pakete/Sek aufgeteilt nach Max Durchsatz (ohne 10kb)



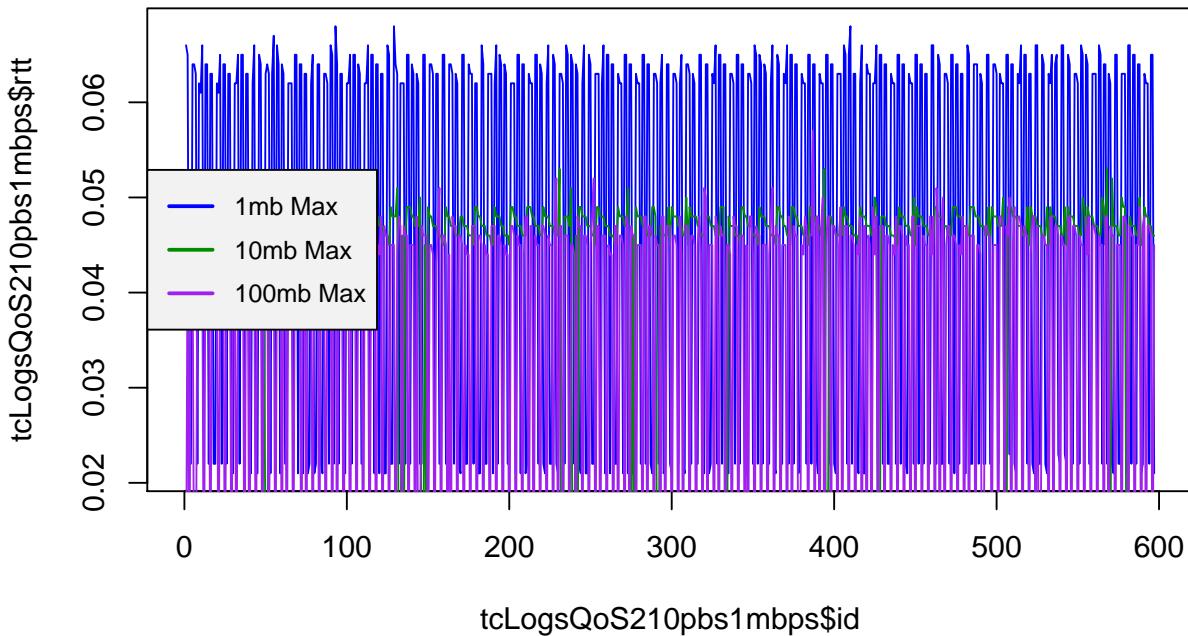
```

plot(tcLogsQoS210pbs1mbps$id, tcLogsQoS210pbs1mbps$rtt, type = "l", col = "blue", main = "QoS2 10Pakete",
      points(tcLogsQoS210pbs10mbps$id, tcLogsQoS210pbs10mbps$rtt, type = "l", col = "green4")
      points(tcLogsQoS210pbs100mbps$id, tcLogsQoS210pbs100mbps$rtt, type = "l", col = "purple")

legend("left", c("1mb Max", "10mb Max", "100mb Max"),
       cex = 0.8,
       col = c("blue", "green4", "purple"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

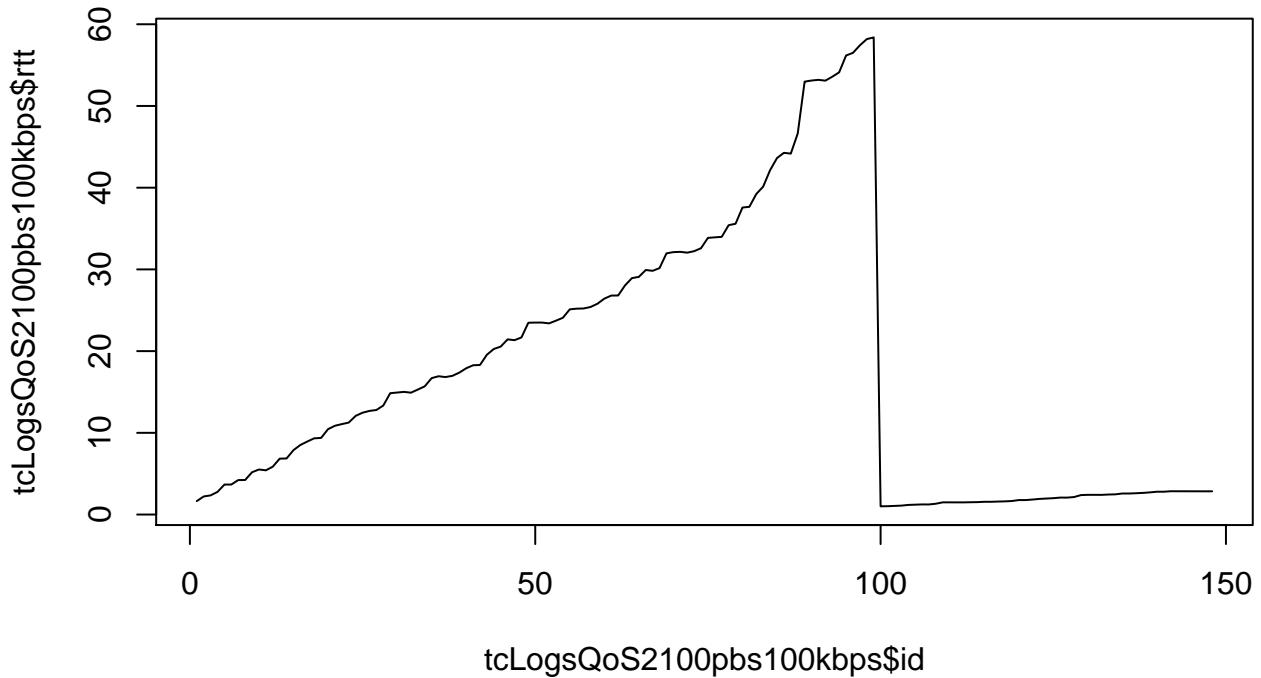
QoS 2 10Pakete/Sek aufgeteilt nach Max Durchsatz (ohne 10kb, 100kb)



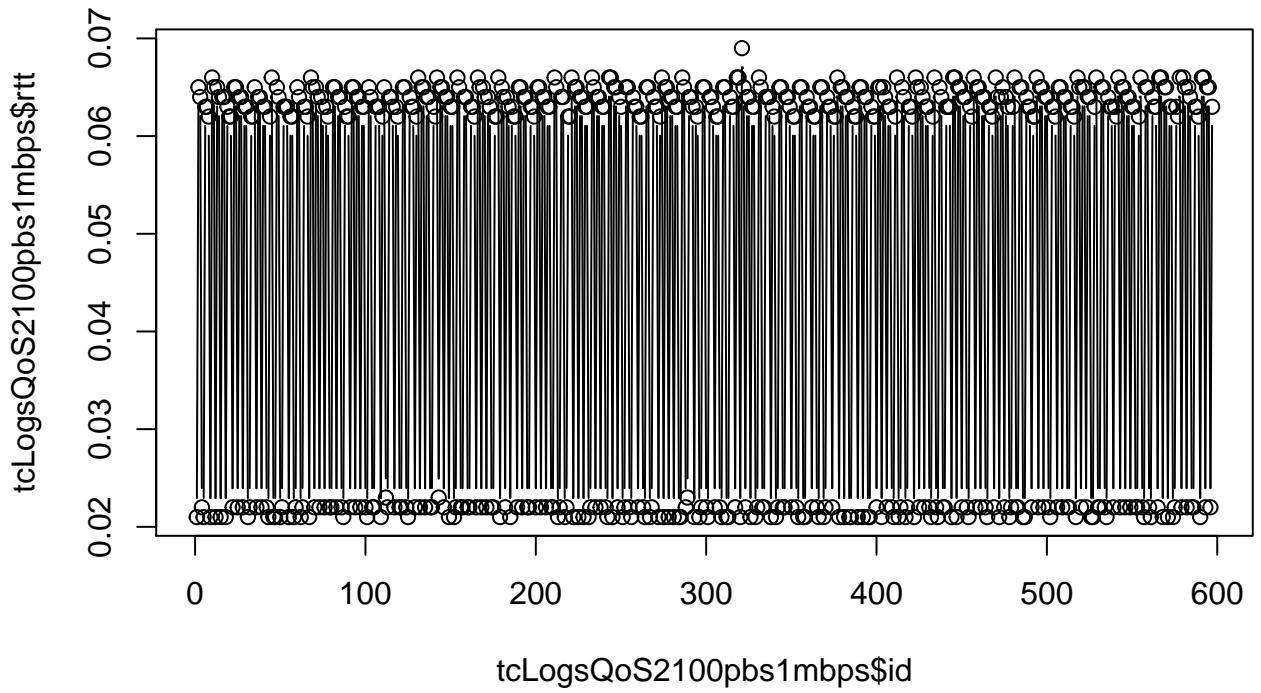
```
#####
# QoS 2 #
#####
## QoS2_100pbs - Aufsplittung MaxDurc
tcLogsQoS2100pbs10kbps<-tcLogsQoS2100pbs [tcLogsQoS2100pbs$MaxDurc == "10kbps",]
tcLogsQoS2100pbs100kbps<-tcLogsQoS2100pbs [tcLogsQoS2100pbs$MaxDurc == "100kbps",]
tcLogsQoS2100pbs1mbps<-tcLogsQoS2100pbs [tcLogsQoS2100pbs$MaxDurc == "1mbps",]
tcLogsQoS2100pbs10mbps<-tcLogsQoS2100pbs [tcLogsQoS2100pbs$MaxDurc == "10mbps",]
tcLogsQoS2100pbs100mbps<-tcLogsQoS2100pbs [tcLogsQoS2100pbs$MaxDurc == "100mbps",]

#plot(tcLogsQoS2100pbs10kbps$id, tcLogsQoS2100pbs10kbps$rtt, type = "l", main = "QoS_2 100_Pakete/Sek 10kbps")
plot(tcLogsQoS2100pbs100kbps$id, tcLogsQoS2100pbs100kbps$rtt, type = "l", main = "QoS_2 100_Pakete/Sek 100kbps")
```

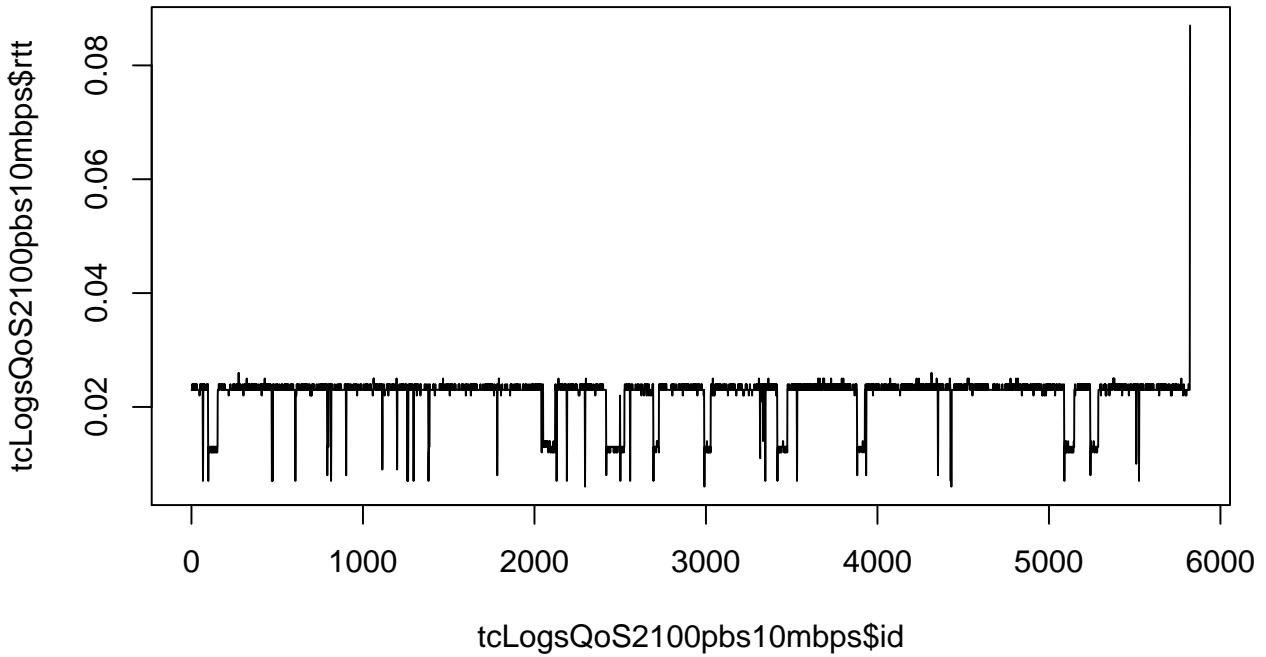
QoS_2 100_Pakete/Sek 100kb_Max



QoS_2 100_Pakete/Sek 1mb_Max

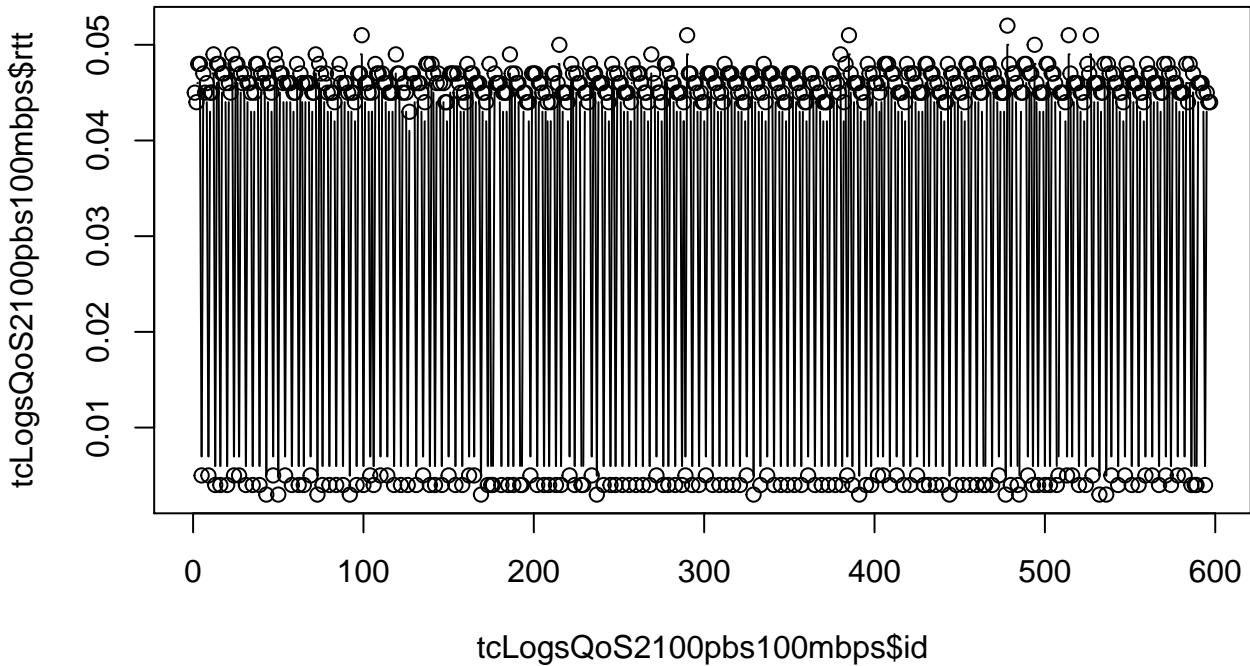


QoS_2 100_Pakete/Sek 10mb_Max



```
plot(tcLogsQoS2100pbs100mbps$id, tcLogsQoS2100pbs100mbps$rtt, type = "b", main = "QoS_2 100_Pakete/Sek 10mb_Max")
```

QoS_2 100_Pakete/Sek 100mb_Max



```
plot(tcLogsQoS2100pbs100kbps$id, tcLogsQoS2100pbs100kbps$rtt, ylim = c(0, 60), type = "l", col = "red",  
points(tcLogsQoS2100pbs1mbps$id, tcLogsQoS2100pbs1mbps$rtt, type = "l", col = "blue")  
points(tcLogsQoS2100pbs10mbps$id, tcLogsQoS2100pbs10mbps$rtt, type = "l", col = "green4")
```

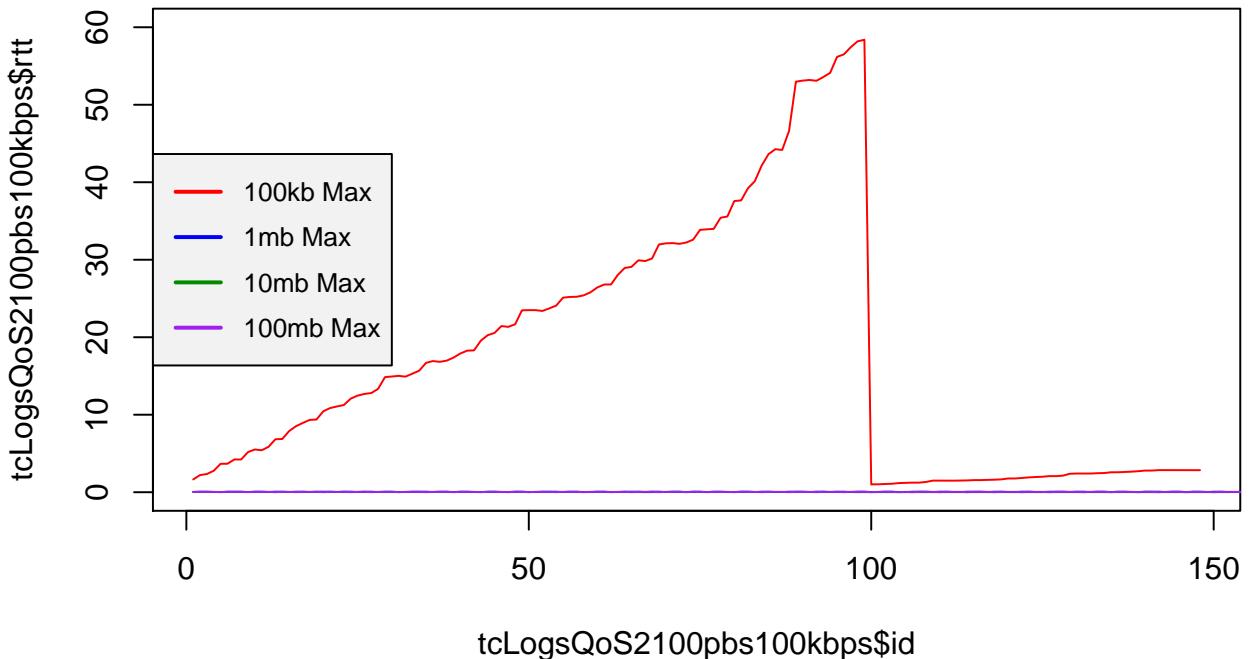
```

points(tcLogsQoS2100pbs100mbps$id, tcLogsQoS2100pbs100mbps$rtt, type = "l", col = "purple")

legend("left", c("100kb Max", "1mb Max", "10mb Max", "100mb Max"),
       cex = 0.8,
       col = c("red", "blue", "green4", "purple"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

QoS2 100Pakete/Sek aufgeteilt nach Max Durchsatz (ohne 10kb)



```

plot(tcLogsQoS2100pbs1mbps$id, tcLogsQoS2100pbs1mbps$rtt, type = "l", col = "blue", main = "QoS2 100Pakete/Sek")
points(tcLogsQoS2100pbs10mbps$id, tcLogsQoS2100pbs10mbps$rtt, type = "l", col = "green4")
points(tcLogsQoS2100pbs100mbps$id, tcLogsQoS2100pbs100mbps$rtt, type = "l", col = "purple")

legend("left", c("1mb Max", "10mb Max", "100mb Max"),
       cex = 0.8,
       col = c("blue", "green4", "purple"),
       text.col = "black" ,lwd = c(2, 2, 2),
       y.intersp = 1.5, merge = FALSE, bg = "gray95")

```

QoS 2 100Pakete/Sek aufgeteilt nach Max Durchsatz (ohne 10kb, 100kb)

