Fuentes de Tráfico

Natalia Clivio

2016

# Caracterización fuente de datos

Una vez, cargadas las trazas de cada suscriptor se calculan los parámetros de cada traza. Estos parámetros son: Capacidad del canal **C** que esta determinada por la velocidad contratada por el adonado, para este caso son 6 Mbps. \***Tamaño de buffer** B **Taza promedio de arribo Varianza de la taza de arribo parámetro Hurst** H **Parámetro temporal** t **Parámetro espacial** s\*\*

C<-6 #Capacidad del canal Mbps  
H<-0.8

Calculando la tasa promedio de arribo de cada traza:

for (i in 1:30) {  
 cat("u",i,"<-","mean(traza",i,"$Total.Incoming.bps/1000000, na.rm = TRUE)","\n",sep = "")  
}

## u1<-mean(traza1$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u2<-mean(traza2$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u3<-mean(traza3$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u4<-mean(traza4$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u5<-mean(traza5$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u6<-mean(traza6$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u7<-mean(traza7$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u8<-mean(traza8$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u9<-mean(traza9$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u10<-mean(traza10$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u11<-mean(traza11$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u12<-mean(traza12$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u13<-mean(traza13$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u14<-mean(traza14$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u15<-mean(traza15$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u16<-mean(traza16$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u17<-mean(traza17$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u18<-mean(traza18$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u19<-mean(traza19$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u20<-mean(traza20$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u21<-mean(traza21$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u22<-mean(traza22$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u23<-mean(traza23$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u24<-mean(traza24$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u25<-mean(traza25$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u26<-mean(traza26$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u27<-mean(traza27$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u28<-mean(traza28$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u29<-mean(traza29$Total.Incoming.bps/1000000, na.rm = TRUE)  
## u30<-mean(traza30$Total.Incoming.bps/1000000, na.rm = TRUE)

Calculando las varianzas de cada traza

for (i in 1:30) {  
 cat("V",i,"<-","var(traza",i,"$Total.Incoming.bps/1000000, na.rm = TRUE)","\n",sep = "")  
}

## V1<-var(traza1$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V2<-var(traza2$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V3<-var(traza3$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V4<-var(traza4$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V5<-var(traza5$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V6<-var(traza6$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V7<-var(traza7$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V8<-var(traza8$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V9<-var(traza9$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V10<-var(traza10$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V11<-var(traza11$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V12<-var(traza12$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V13<-var(traza13$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V14<-var(traza14$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V15<-var(traza15$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V16<-var(traza16$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V17<-var(traza17$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V18<-var(traza18$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V19<-var(traza19$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V20<-var(traza20$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V21<-var(traza21$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V22<-var(traza22$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V23<-var(traza23$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V24<-var(traza24$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V25<-var(traza25$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V26<-var(traza26$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V27<-var(traza27$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V28<-var(traza28$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V29<-var(traza29$Total.Incoming.bps/1000000, na.rm = TRUE)  
## V30<-var(traza30$Total.Incoming.bps/1000000, na.rm = TRUE)

Los datos de las trazas son:

for (i in 1:30) {  
 cat(i,"u",sep=",")  
 }

## 1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,u12,u13,u14,u15,u16,u17,u18,u19,u20,u21,u22,u23,u24,u25,u26,u27,u28,u29,u30,u

for (i in 1:30) {  
 cat(i,"V",sep=",")  
 }

## 1,V2,V3,V4,V5,V6,V7,V8,V9,V10,V11,V12,V13,V14,V15,V16,V17,V18,V19,V20,V21,V22,V23,V24,V25,V26,V27,V28,V29,V30,V

u<-c(u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,u12,u13,u14,u15,u16,u17,u18,u19,u20,u21,u22,u23,u24,u25,u26,u27,u28,u29,u30)  
v<-c(V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,V11,V12,V13,V14,V15,V16,V17,V18,V19,V20,V21,V22,V23,V24,V25,V26,V27,V28,V29,V30)  
   
tabla<-data.frame("Taza Arribo"=u,"Varianza Arribo"=v)  
tabla

## Taza.Arribo Varianza.Arribo  
## 1 1.171088 0.5341858  
## 2 1.184703 0.3234680  
## 3 1.195172 0.5590680  
## 4 1.379985 0.5294907  
## 5 1.212043 0.3569780  
## 6 1.213871 0.4550850  
## 7 1.474693 1.6559925  
## 8 1.291474 0.2230746  
## 9 1.269373 2.2713761  
## 10 1.288172 0.5183902  
## 11 1.308722 0.4661152  
## 12 1.318064 0.5807520  
## 13 1.319015 0.7146869  
## 14 1.326088 0.7190917  
## 15 1.532733 3.0379125  
## 16 1.336745 0.3998109  
## 17 1.363311 0.7713211  
## 18 1.377716 0.8275499  
## 19 1.403901 0.6928111  
## 20 1.692597 0.4344020  
## 21 1.634591 0.5657844  
## 22 1.458788 0.4870205  
## 23 1.476288 0.4441687  
## 24 1.518427 1.0600024  
## 25 1.589997 1.3818158  
## 26 2.047814 1.8927629  
## 27 1.897302 0.5320073  
## 28 2.149381 1.3263307  
## 29 2.185992 4.5779950  
## 30 2.378141 0.8458021

calculando los parámetros temporal y espacial de cada traza

B<-0.150   
  
t<-(B/(C-u))\*(H/(1-H)) #time parameter (us)  
s<-(B+(C+u)\*t)/(v\*t^(2\*H)) #space parameter (bitss^-1)  
  
#t<-log10(seq(length=100, from=1, to=200)) #time parameter (us)  
#s<-log10(seq(length=100, from=1, to=5)) #space parameter (bits^-1)

Por lo tanto el ancho de banda efectivo para cada traza es:

BWE<-u+(((s\*v)/2)\*(t^(2\*H-1)))  
BWE

## [1] 5.360245 5.378967 5.393362 5.647479 5.416559 5.419073 5.777703  
## [8] 5.525777 5.495388 5.521237 5.549492 5.562338 5.563646 5.573371  
## [15] 5.857508 5.588025 5.624552 5.644359 5.680364 6.077320 5.997562  
## [22] 5.755833 5.779896 5.837837 5.936246 6.565744 6.358791 6.705399  
## [29] 6.755739 7.019944

datos<-data.frame("Taza\_Arribo"=u,"Varianza\_Arribo"=v,"time"=t,"space"=s,BWE)  
datos

## Taza\_Arribo Varianza\_Arribo time space BWE  
## 1 1.171088 0.5341858 0.1242516 54.812960 5.360245  
## 2 1.184703 0.3234680 0.1246029 90.476852 5.378967  
## 3 1.195172 0.5590680 0.1248744 52.329107 5.393362  
## 4 1.379985 0.5294907 0.1298697 54.857995 5.647479  
## 5 1.212043 0.3569780 0.1253144 81.903794 5.416559  
## 6 1.213871 0.4550850 0.1253623 64.242758 5.419073  
## 7 1.474693 1.6559925 0.1325877 17.467954 5.777703  
## 8 1.291474 0.2230746 0.1274284 130.677842 5.525777  
## 9 1.269373 2.2713761 0.1268331 12.844950 5.495388  
## 10 1.288172 0.5183902 0.1273391 56.240736 5.521237  
## 11 1.308722 0.4661152 0.1278969 62.497906 5.549492  
## 12 1.318064 0.5807520 0.1281521 50.142640 5.562338  
## 13 1.319015 0.7146869 0.1281782 40.744190 5.563646  
## 14 1.326088 0.7190917 0.1283721 40.483168 5.573371  
## 15 1.532733 3.0379125 0.1343103 9.496261 5.857508  
## 16 1.336745 0.3998109 0.1286655 72.780948 5.588025  
## 17 1.363311 0.7713211 0.1294027 37.684692 5.624552  
## 18 1.377716 0.8275499 0.1298060 35.103093 5.644359  
## 19 1.403901 0.6928111 0.1305455 41.883481 5.680364  
## 20 1.692597 0.4344020 0.1392951 65.874759 6.077320  
## 21 1.634591 0.5657844 0.1374442 50.732422 5.997562  
## 22 1.458788 0.4870205 0.1321233 59.438116 5.755833  
## 23 1.476288 0.4441687 0.1326344 65.120988 5.779896  
## 24 1.518427 1.0600024 0.1338816 27.234232 5.837837  
## 25 1.589997 1.3818158 0.1360543 20.819343 5.936246  
## 26 2.047814 1.8927629 0.1518147 14.793979 6.565744  
## 27 1.897302 0.5320073 0.1462452 53.154874 6.358791  
## 28 2.149381 1.3263307 0.1558191 20.959994 6.705399  
## 29 2.185992 4.5779950 0.1573148 6.055989 6.755739  
## 30 2.378141 0.8458021 0.1656608 32.278703 7.019944

El ancho de banda efectivo promedio para la fuente de datos es:

BWEmean<-mean(BWE)  
BWEmean

## [1] 5.812325

El intervalo de confianza, calculando el percentil 95 para el ancho de banda efectivo promedio para una fuente de datos es:

n<<-30  
sd<-sqrt((1/(n-1)\*(sum((BWE-BWEmean)^2))))  
  
normalci<-BWEmean+c(-1,1)\*qnorm(0.95)\*sd/sqrt(n)  
normalci

## [1] 5.67924 5.94541

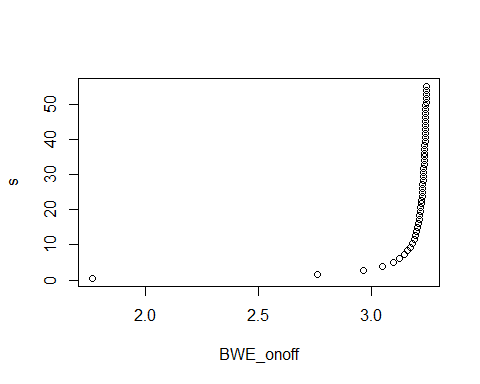
# Caracterización fuente de video

Preparando los datos

data\_VoD<-read.csv("C:/Users/NataliaA/Documents/Maestria/Tráfico L3/VoD/Subscribers\_detail.csv",  
 header=TRUE,sep=";",na.strings="NA",dec=",")

Aplicando modelo ON-OFF

Pon<-data\_VoD$Pon #Propabilidad de estado ON  
Poff<-data\_VoD$Poff #Propabilidad de estado OFF  
  
h<-7.5 #Mbps tasa pico  
  
Pon<-mean(Pon)  
Poff<-mean(Poff)  
  
space<-seq(length=50, from=0.5, to=55)  
s<-(space)  
BWE\_onoff<-(1/s)\*log10(Poff+(exp((h\*s))\*Pon))   
plot(BWE\_onoff,s)



De lo anterior se observa que s tiene un conportamiento asintótico, para lo cual considera un valor máximo de s antes de volverse indeterminado en 45

Pon<-data\_VoD$Pon #Propabilidad de estado ON  
Poff<-data\_VoD$Poff #Propabilidad de estado OFF  
  
s<-45  
BWE\_onoff<-(1/s)\*log10(Poff+(exp((h\*s))\*Pon))   
BWE\_onoff

## [1] 3.237578 3.242252 3.237876 3.241145 3.241451 3.239395 3.239947  
## [8] 3.237925 3.236842 3.242622 3.239110 3.240204 3.236866 3.237904  
## [15] 3.237047 3.238632 3.240315 3.238586 3.239084 3.238770 3.239930  
## [22] 3.242679 3.238579 3.237202 3.238552 3.240022 3.240343 3.238718  
## [29] 3.239154 3.241227

El ancho de banda efectivo promedio para la fuente de datos es:

BWEmean<-mean(BWE\_onoff)  
BWEmean

## [1] 3.239332

El intervalo de confianza, calculando el percentil 95 para el ancho de banda efectivo promedio para una fuente de datos es:

n<<-30 #Cantidad de trazas muestreadas  
sd<-sqrt((1/(n-1)\*(sum((BWE\_onoff-BWEmean)^2))))  
  
normalci<-BWEmean+c(-1,1)\*qnorm(0.95)\*sd/sqrt(n)  
normalci

## [1] 3.238832 3.239831

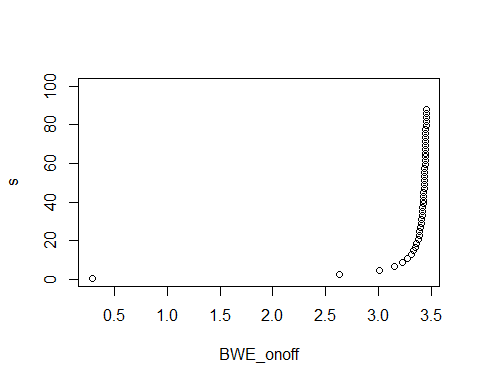
# Caracterización fuente de voz IP

Preparando los datos

data\_VoIP<-read.csv("C:/Users/NataliaA/Documents/Maestria/Tráfico L3/VoIP/Consumo\_Voip.csv",  
 header=TRUE,sep=";",na.strings="NA",dec=",")

Aplicando modelo ON-OFF

Pon<-data\_VoIP$Pon #Propabilidad de estado ON  
Poff<-data\_VoIP$Poff #Propabilidad de estado OFF  
  
h<-8 #Kbps tasa pico  
  
Pon<-mean(Pon)  
Poff<-mean(Poff)  
  
space<-seq(length=50, from=0.5, to=100)  
s<-(space)  
BWE\_onoff<-(1/s)\*log10(Poff+(exp((h\*s))\*Pon))   
plot(BWE\_onoff,s)



De lo anterior se observa que s tiene un conportamiento asintótico

Pon<-data\_VoD$Pon #Propabilidad de estado ON  
Poff<-data\_VoD$Poff #Propabilidad de estado OFF  
  
s<-88 #Antes de que el BWE se vuelva indeterminado  
BWE\_onoff<-(1/s)\*log10(Poff+(exp((h\*s))\*Pon))   
BWE\_onoff

## [1] 3.464317 3.466708 3.464470 3.466142 3.466298 3.465247 3.465529  
## [8] 3.464495 3.463941 3.466897 3.465101 3.465660 3.463953 3.464484  
## [15] 3.464046 3.464856 3.465717 3.464833 3.465088 3.464927 3.465520  
## [22] 3.466926 3.464829 3.464125 3.464816 3.465567 3.465731 3.464900  
## [29] 3.465123 3.466183

El ancho de banda efectivo promedio para la fuente de datos es:

BWEmean<-mean(BWE\_onoff)  
BWEmean

## [1] 3.465214

El intervalo de confianza, calculando el percentil 95 para el ancho de banda efectivo promedio para una fuente de datos es:

n<<-30 #Cantidad de trazas muestreadas  
sd<-sqrt((1/(n-1)\*(sum((BWE\_onoff-BWEmean)^2))))  
  
normalci<-BWEmean+c(-1,1)\*qnorm(0.95)\*sd/sqrt(n)  
normalci #[Kbps]

## [1] 3.464959 3.465470