

Simulation_mm1

2024-11-29

Functions

Vysochanskij–Petunin’s bound calculation function

Inputs:

delay_hops: Vector of delay at each hop a: Upper bound percentile from 0 to 1 Output:

Vysochanskij–Petunin’s upper bound

```
func_bounds_VP <- function(delay_hops, a)
{
  mu = sum(delay_hops)
  sigma = sqrt(sum(delay_hops^2))
  k <- sqrt(4/9/(1-a))
  Prop_VP <- 1 - 4/9/(k^2)
  upper_bound_VP <- k * sigma + mu
  return(upper_bound_VP)
}
```

Input data

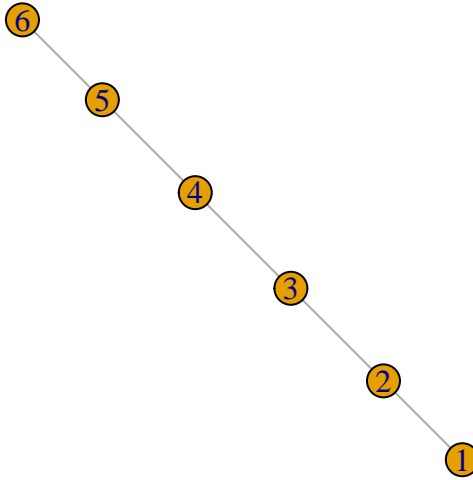
The average packet size is taken from: Amsterdam Internet Exchange Ethernet Frame Size Distribution, statistics available online at <https://stats.ams-ix.net/sflow/size.html>, accessed on July 2023

Packet sizes in Bytes: 64-127, 128-511, 512 - 1023, 1024 - 1513, 1514, more than 1515.

With their probabilities: 0.332, 0.054, 0.033, 0.037, 0.346, 0.146, 0.052

Calculating 5-hop e2e delay for 5 links and 6 nodes

```
edges <- c(1, 2, 2, 3, 3, 4, 4, 5, 5, 6)
graph <- graph(edges, directed = FALSE)
plot(graph, vertex.label = V(graph)$name)
```



```

PS_size=c((64+127)/2,(128+255)/2,(256+511)/2, (512+1023)/2, (1024+1513)/2, 1514, (1515+9100)/2)
PS_weights=c(33.2/100, 5.4/100, 3.3/100, 3.7/100, 34.6/100, 14.6/100, 5.2/100)
N = sum(PS_size*PS_weights)
N

```

```
## [1] 1019.035
```

```

var_N <- sum(PS_size^2*PS_weights) - N^2
Cs2 <- var_N/(N^2)

Capacity_Gbps = 10
Capacity_bps = 10*1e9
load = rep(0.9, 5)
distance = rep(4, 5)
set.seed(123)

```

10G Capacity

Theoretical calculations

The mean delay $E(D)$ is given by: $E(D) = E(X) \frac{1}{1-\rho} = \frac{1}{\mu-\lambda} = Sd(D)$

```

N = 1019.035
Capacity_ps = Capacity_Gbps * 1e9/(8*N)
mu = 1/Capacity_ps
traffic_ps = Capacity_ps*load
theor_delay_link_mm1 = 1/(Capacity_ps*(1-load))
theor_prop_dalay <- distance*5/1e6

cat("Theoretical M/M/1 delays:", sum(theor_delay_link_mm1)* 1e6 + sum(theor_prop_dalay)*1e6, "us \n")

## Theoretical M/M/1 delays: 140.7614 us

upper_bound_mm1_p99 <- func_bounds_VP(theor_delay_link_mm1, 0.99)
upper_bound_mm1_p90 <- func_bounds_VP(theor_delay_link_mm1, 0.9)

cat("VP D.90", upper_bound_mm1_p90*1e6, "us \n")

## VP D.90 79.19162 us

cat("VP D.99", upper_bound_mm1_p99*1e6, "us \n")

## VP D.99 162.2884 us

cat("VP D.90 with prop delay", upper_bound_mm1_p90*1e6 + sum(theor_prop_dalay)*1e6, "us \n")

## VP D.90 with prop delay 179.1916 us

cat("VP D.99 with prop delay", upper_bound_mm1_p99*1e6 + sum(theor_prop_dalay)*1e6, "us \n")

## VP D.99 with prop delay 262.2884 us

```

Simmer simulation

```

env <- simmer("SuperDuperSim") #creating the simmer environment

env %>% add_resource('node_1') %>%#, queue_size = 100) %>%
  add_resource('node_2') %>%
  add_resource('node_3') %>%
  add_resource('node_4') %>%
  add_resource('node_5')

## simmer environment: SuperDuperSim | now: 0 | next:
## { Monitor: in memory }
## { Resource: node_1 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_2 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_3 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_4 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_5 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }

```

```

traffic_1_6_ <- trajectory() %>%
  seize("node_1") %>%
  timeout( function() rexp(1, Capacity_ps)) %>%
  release("node_1") %>%
  timeout(theor_prop_dalay[1]) %>%
  seize("node_2") %>%
  timeout( function() rexp(1, Capacity_ps)) %>%
  release("node_2") %>%
  timeout(theor_prop_dalay[2]) %>%
  seize("node_3") %>%
  timeout(function() rexp(1, Capacity_ps)) %>%
  release("node_3") %>%
  timeout(theor_prop_dalay[3]) %>%
  seize("node_4") %>%
  timeout(function() rexp(1, Capacity_ps)) %>%
  release("node_4") %>%
  timeout(theor_prop_dalay[4]) %>%
  seize("node_5") %>%
  timeout(function() rexp(1, Capacity_ps)) %>%
  release("node_5") %>%
  timeout(theor_prop_dalay[5])

```

```

env %>%
  add_generator("traffic_1_6_", traffic_1_6_, function() rexp(1, traffic_ps[1]))

```

```

## simmer environment: SuperDuperSim | now: 0 | next: 0
## { Monitor: in memory }
## { Resource: node_1 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_2 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_3 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_4 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_5 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Source: traffic_1_6_ | monitored: 1 | n_generated: 0 }

```

```

env %>% #env execution
  run(1000000/min(traffic_ps))

```

```

## simmer environment: SuperDuperSim | now: 0.9058088888888889 | next: 0.905808901334588
## { Monitor: in memory }
## { Resource: node_1 | monitored: TRUE | server status: 1(1) | queue status: 4(Inf) }
## { Resource: node_2 | monitored: TRUE | server status: 1(1) | queue status: 0(Inf) }
## { Resource: node_3 | monitored: TRUE | server status: 1(1) | queue status: 23(Inf) }
## { Resource: node_4 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_5 | monitored: TRUE | server status: 1(1) | queue status: 11(Inf) }
## { Source: traffic_1_6_ | monitored: 1 | n_generated: 1000451 }

```

```

all_arrivals_res <- data.frame(env %>%
  get_mon_arrivals(per_resource = FALSE) %>%
  transform(waiting_time_in_queue = round(end_time - start_time - activi
  transform(spending_time = end_time - start_time))

```

```
traffic <- dplyr::filter(all_arrivals_res, grepl("traffic_1_6_", all_arrivals_res$name))
print(mean(traffic$spending_time))
```

```
## [1] 0.0001410164
```

```
delay_sim_mm1 <- traffic$spending_time
```

```
cat("D.90 quantile", quantile(delay_sim_mm1, 0.9)*1e6, "us \n")
```

```
## D.90 quantile 165.8287 us
```

```
cat("D.99 quantile", quantile(delay_sim_mm1, 0.99)*1e6, "us \n")
```

```
## D.99 quantile 195.5612 us
```

100G Capacity

```
PS_size=c((64+127)/2,(128+255)/2,(256+511)/2, (512+1023)/2, (1024+1513)/2, 1514, (1515+9100)/2)
PS_weights=c(33.2/100, 5.4/100, 3.3/100, 3.7/100, 34.6/100, 14.6/100, 5.2/100)
N = sum(PS_size*PS_weights)
N
```

```
## [1] 1019.035
```

```
var_N <- sum(PS_size^2*PS_weights) - N^2
Cs2 <- var_N/(N^2)

Capacity_Gbps = 100
Capacity_bps = 100*1e9
load = rep(0.9, 5)
distance = rep(4, 5)
set.seed(123)
```

Theoretical calculations

The mean delay $E(D)$ is given by: $E(D) = E(X) \frac{1}{1-\rho} = \frac{1}{\mu-\lambda} = Sd(D)$

```
N = 1019.035
Capacity_ps = Capacity_Gbps * 1e9/(8*N)
mu = 1/Capacity_ps
traffic_ps = Capacity_ps*load
theor_delay_link_mm1 = 1/(Capacity_ps*(1-load))
theor_prop_dalay <- distance*5/1e6

cat("Theoretical M/M/1 delays:", sum(theor_delay_link_mm1)* 1e6 + sum(theor_prop_dalay)*1e6, "us \n")
```

```
## Theoretical M/M/1 delays: 104.0761 us
```

```
upper_bound_mm1_p99 <- func_bounds_VP(theor_delay_link_mm1, 0.99)
upper_bound_mm1_p90 <- func_bounds_VP(theor_delay_link_mm1, 0.9)

cat("VP D.90", upper_bound_mm1_p90*1e6, "us \n")
```

```
## VP D.90 7.919162 us
```

```
cat("VP D.99", upper_bound_mm1_p99*1e6, "us \n")
```

```
## VP D.99 16.22884 us
```

```
cat("VP D.90 with prop delay", upper_bound_mm1_p90*1e6 + sum(theor_prop_delay)*1e6, "us \n")
```

```
## VP D.90 with prop delay 107.9192 us
```

```
cat("VP D.99 with prop delay", upper_bound_mm1_p99*1e6 + sum(theor_prop_delay)*1e6, "us \n")
```

```
## VP D.99 with prop delay 116.2288 us
```

Simmer simulation

```
env <- simmer("SuperDuperSim") #creating the simmer environment
```

```
env %>% add_resource('node_1') %>%#, queue_size = 100) %>%
  add_resource('node_2') %>%
  add_resource('node_3') %>%
  add_resource('node_4') %>%
  add_resource('node_5')
```

```
## simmer environment: SuperDuperSim | now: 0 | next:
```

```
## { Monitor: in memory }
```

```
## { Resource: node_1 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
```

```
## { Resource: node_2 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
```

```
## { Resource: node_3 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
```

```
## { Resource: node_4 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
```

```
## { Resource: node_5 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
```

```
traffic_1_6_ <- trajectory() %>%
  seize("node_1") %>%
  timeout( function() rexp(1, Capacity_ps)) %>%
  release("node_1") %>%
  timeout(theor_prop_delay[1]) %>%
  seize("node_2") %>%
  timeout( function() rexp(1, Capacity_ps)) %>%
  release("node_2") %>%
  timeout(theor_prop_delay[2]) %>%
```

```

seize("node_3") %>%
timeout(function() rexp(1, Capacity_ps)) %>%
release("node_3") %>%
timeout(theor_prop_dalay[3]) %>%
seize("node_4") %>%
timeout(function() rexp(1, Capacity_ps)) %>%
release("node_4") %>%
timeout(theor_prop_dalay[4]) %>%
seize("node_5") %>%
timeout(function() rexp(1, Capacity_ps)) %>%
release("node_5") %>%
timeout(theor_prop_dalay[5])

env %>%
  add_generator("traffic_1_6_", traffic_1_6_, function() rexp(1, traffic_ps[1]))

```

```

## simmer environment: SuperDuperSim | now: 0 | next: 0
## { Monitor: in memory }
## { Resource: node_1 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_2 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_3 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_4 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Resource: node_5 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Source: traffic_1_6_ | monitored: 1 | n_generated: 0 }

```

```

env %>% #env execution
  run(1000000/min(traffic_ps))

```

```

## simmer environment: SuperDuperSim | now: 0.0905808888888889 | next: 0.0905808920427916
## { Monitor: in memory }
## { Resource: node_1 | monitored: TRUE | server status: 1(1) | queue status: 0(Inf) }
## { Resource: node_2 | monitored: TRUE | server status: 1(1) | queue status: 17(Inf) }
## { Resource: node_3 | monitored: TRUE | server status: 1(1) | queue status: 6(Inf) }
## { Resource: node_4 | monitored: TRUE | server status: 1(1) | queue status: 9(Inf) }
## { Resource: node_5 | monitored: TRUE | server status: 0(1) | queue status: 0(Inf) }
## { Source: traffic_1_6_ | monitored: 1 | n_generated: 1001838 }

```

```

all_arrivals_res <- data.frame(env %>%
  get_mon_arrivals(per_resource = FALSE) %>%
  transform(waiting_time_in_queue = round(end_time - start_time - activi
  transform(spending_time = end_time - start_time))

traffic <- dplyr::filter(all_arrivals_res, grepl("traffic_1_6_", all_arrivals_res$name))

print(mean(traffic$spending_time))

```

```
## [1] 0.0001040776
```

```
delay_sim_mm1 <- traffic$spending_time
```

```
cat("D.90 quantile", quantile(delay_sim_mm1, 0.9)*1e6, "us \n")
```

```
## D.90 quantile 106.4486 us
```

```
cat("D.99 quantile", quantile(delay_sim_mm1, 0.99)*1e6, "us \n")
```

```
## D.99 quantile 109.3247 us
```