1. queue::Vector{Int}

serving::Vector{Int}

N::Int

NW::Vector{Int}

WQ::Vector{Time}

WS::Vector{Time}

TS::Vector{Vector{Time}}

Tmax::Time

IAT::Time

iatime::Time

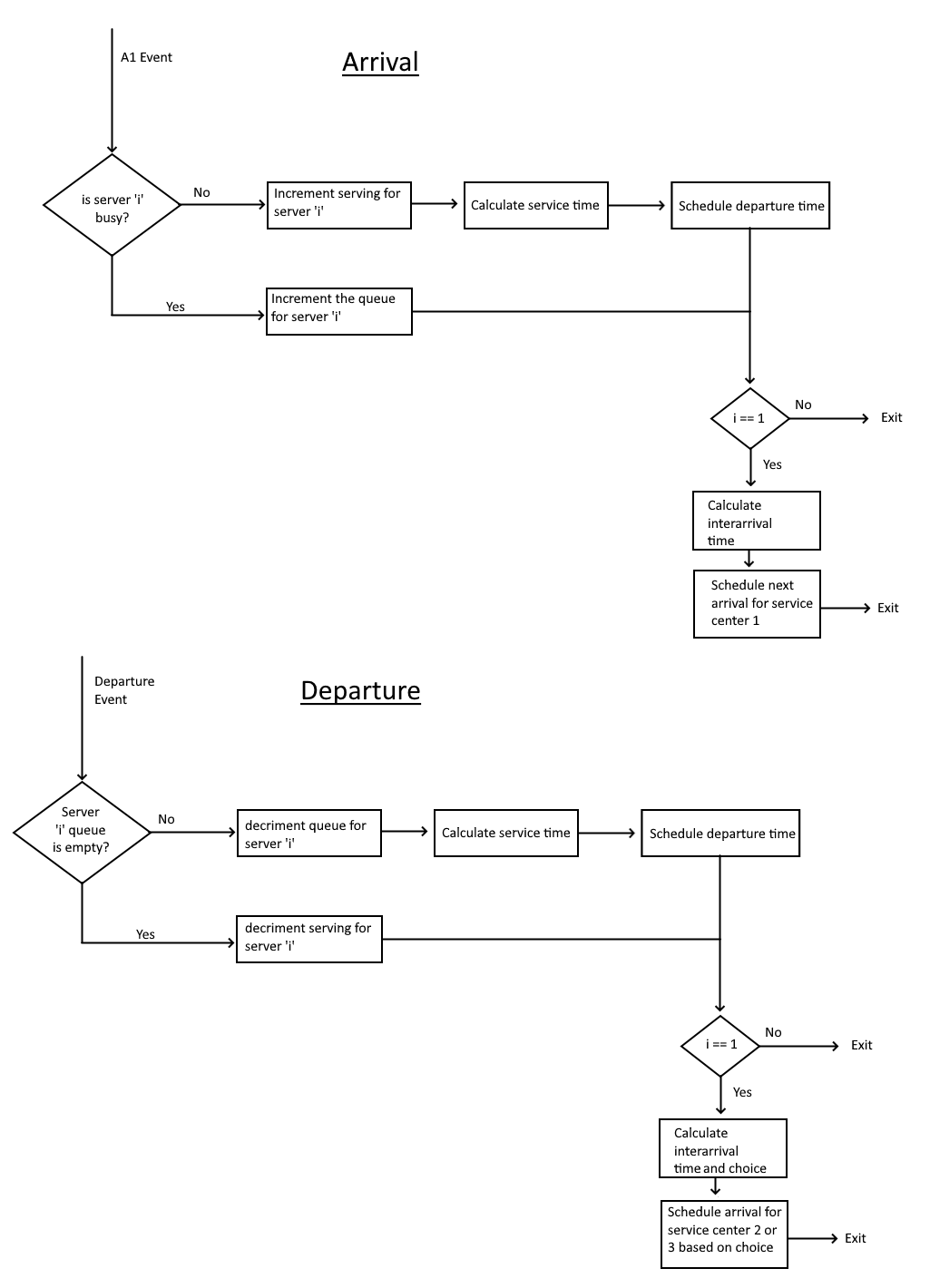
stime::Vector{Time}

1. struct A1 <: ArrivalEvent end

struct D1 <: DepartureEvent end

struct D2 <: DepartureEvent end

struct D3 <: DepartureEvent end

1. 
2. Example output from “[ThreeCenterSystem] my\_tests.jl”:

With an equal chance to go to center 2 and 3, they should have very similar total wait times:

Mean total wait time from service center 1: 9.946

Mean total wait time from service center 2: 13.544

Mean total wait time from service center 3: 13.59

With different chances to go to center 2 and 3, they should have very different total wait times:

Mean total wait time from service center 1: 10.032

Mean total wait time from service center 2: 3.07

Mean total wait time from service center 3: 41.738

1. queue::Vector{Int}

serving::Vector{Int}

departures::Vector{Int} # added to [ThreeCenterSystem] for Q2

N::Int

NW::Vector{Int}

WQ::Vector{Time}

WS::Vector{Time}

TS::Vector{Vector{Time}}

Tmax::Time

IAT::Time

iatime::Time

stime::Vector{Time}

1. struct FBA1 <: ArrivalEvent end

struct FBD1 <: DepartureEvent end

struct FBD2 <: DepartureEvent end

struct FBD3 <: DepartureEvent end

1. Diagram, schematic

   Description automatically generated
2. e) f) g) Example output:

for FeedBackSystem{Int64}([2, 1, 1], 100) with an end time of: 100

Mean percent idle time from service center 1: 0.31298

Mean percent idle time from service center 2: 0.32284

Mean percent idle time from service center 3: 0.52486

Mean percent idle time from all service centers: 0.38689333333333337

Mean percent busy time from service center 1: 0.3615

Mean percent busy time from service center 2: 0.67716

Mean percent busy time from service center 3: 0.47514

Mean percent busy time from all service centers: 0.5045999999999999

Mean rate of departure from service center 1: 0.2847

Mean rate of departure from service center 2: 0.19072

Mean rate of departure from service center 3: 0.13346

Mean rate of departure from all service centers: 0.20296

for ThreeCentersSystem{Int64}([2, 1, 1], 100) with an end time of: 100

Mean percent idle time from service center 1: 0.42064

Mean percent idle time from service center 2: 0.61604

Mean percent idle time from service center 3: 0.6308400000000001

Mean percent idle time from all service centers: 0.55584

Mean percent busy time from service center 1: 0.36738

Mean percent busy time from service center 2: 0.38396

Mean percent busy time from service center 3: 0.36916

Mean percent busy time from all service centers: 0.3735

Mean rate of departure from service center 1: 0.22264

Mean rate of departure from service center 2: 0.10728

Mean rate of departure from service center 3: 0.10490000000000001

Mean rate of departure from all service centers: 0.14493999999999999

for FeedBackSystem{Int64}([2, 1, 1], 200) with an end time of: 200

Mean percent idle time from service center 1: 0.30829

Mean percent idle time from service center 2: 0.27465

Mean percent idle time from service center 3: 0.49774999999999997

Mean percent idle time from all service centers: 0.36023

Mean percent busy time from service center 1: 0.34729999999999994

Mean percent busy time from service center 2: 0.7253499999999999

Mean percent busy time from service center 3: 0.50225

Mean percent busy time from all service centers: 0.5249666666666667

Mean rate of departure from service center 1: 0.29311

Mean rate of departure from service center 2: 0.20535

Mean rate of departure from service center 3: 0.14237

Mean rate of departure from all service centers: 0.21361

for ThreeCentersSystem{Int64}([2, 1, 1], 200) with an end time of: 200

Mean percent idle time from service center 1: 0.42475999999999997

Mean percent idle time from service center 2: 0.6122099999999999

Mean percent idle time from service center 3: 0.61498

Mean percent idle time from all service centers: 0.55065

Mean percent busy time from service center 1: 0.35606000000000004

Mean percent busy time from service center 2: 0.38779

Mean percent busy time from service center 3: 0.38502000000000003

Mean percent busy time from all service centers: 0.37629

Mean rate of departure from service center 1: 0.22418

Mean rate of departure from service center 2: 0.11034000000000001

Mean rate of departure from service center 3: 0.10880000000000001

Mean rate of departure from all service centers: 0.14777333333333334

for FeedBackSystem{Int64}([2, 1, 1], 1000) with an end time of: 1000

Mean percent idle time from service center 1: 0.311504

Mean percent idle time from service center 2: 0.23480199999999998

Mean percent idle time from service center 3: 0.485572

Mean percent idle time from all service centers: 0.3439593333333333

Mean percent busy time from service center 1: 0.34195600000000004

Mean percent busy time from service center 2: 0.7651979999999999

Mean percent busy time from service center 3: 0.514428

Mean percent busy time from all service centers: 0.5405273333333334

Mean rate of departure from service center 1: 0.295294

Mean rate of departure from service center 2: 0.218338

Mean rate of departure from service center 3: 0.14694

Mean rate of departure from all service centers: 0.22019066666666665

for ThreeCentersSystem{Int64}([2, 1, 1], 1000) with an end time of: 1000

Mean percent idle time from service center 1: 0.436242

Mean percent idle time from service center 2: 0.612072

Mean percent idle time from service center 3: 0.6122340000000001

Mean percent idle time from all service centers: 0.553516

Mean percent busy time from service center 1: 0.349088

Mean percent busy time from service center 2: 0.387928

Mean percent busy time from service center 3: 0.387766

Mean percent busy time from all service centers: 0.37492733333333333

Mean rate of departure from service center 1: 0.22226400000000002

Mean rate of departure from service center 2: 0.110584

Mean rate of departure from service center 3: 0.11070999999999999

Mean rate of departure from all service centers: 0.14785266666666666

h) both systems have the same interarrival times, service times, and choice chance.

Interarrivaltime = rand(Exponential(4.5))

Servicetime = rand(Poisson(3.5))

choice = rand(Bernoulli(0.50)) # for simplicity

feedback\_choice = rand(Bernoulli(0.50)) # for simplicity

Fsim = FSystem(c1 = 2, c2 = 1, c3 = 1, end\_time = 100)

ThCsim = ThCSystem(c1 = 2, c2 = 1, c3 = 1, end\_time = 100)

In the feedback system, service centers 1 and 2 had very similar percentages of time for being idle, while the three center system showed that service centers 2 and 3 had very similar idle percentages. Overall, the feedback system had less percentage of time being idle.

The percentage of time that service center 1 was busy in the feedback system was very similar to what it was in the three center system. The busy percentages of all the service centers in the three center system were similar. The feedback system had higher busy percentages overall, with service center 2 having the highest percent time being busy – probably because it is the bottleneck to exit the system.

As the end time increased, the similar values got more similar.

1. See Code in “MyDistributions.jl”

Chart, line chart

Description automatically generated

* This is generated from:

qqnorm(rand(MyExp(30), 200), qqline = :fit, title = "MyExp")

Chart, line chart

Description automatically generated

* This is generated from:

qqnorm(rand(Exponential(30), 200), qqline = :fit, title = " Exponential ")

Chart, scatter chart

Description automatically generated

* This is generated from:

qqplot(rand(MyExp(30), 200), rand(Exponential(30), 200), qqline = :fit, title = "Compare")

The graphs: “MyExp”, and “Exponential” look quite similar, and the “Compare” graph follows the regression line which suggests that MyExp and Exponential produce similar random values

1. Using benchmarking tools I found that MyExp took 4.414 μs to generate 200 values, while it took Exponential 788.350 ns (0.788350 μs) to do the same.

MyExp also took 5 allocations while Exponential only took 1.