The 4.2.1 release of [simmer](http://r-simmer.org/), the Discrete-Event Simulator for R, is on CRAN with quite interesting new features and fixes. As discussed in the mailing list, there is a way to handle the specific case in which an arrival is rejected because a queue is full:

library(simmer)

reject <- trajectory() %>%

log\_("kicked off...")

patient <- trajectory() %>%

seize("nurse", continue=FALSE, reject=reject) %>%

log\_("nurse seized") %>%

timeout(5) %>%

release("nurse") %>%

log\_("nurse released")

env <- simmer() %>%

add\_resource("nurse", 1, 0) %>%

add\_generator("patient", patient, at(0, 1)) %>%

run()

## 0: patient0: nurse seized

## 1: patient1: kicked off...

## 5: patient0: nurse released

But as [Tom Lawton pointed out](https://groups.google.com/d/msg/simmer-devel/-KWhDnPuBSQ/U2Ch8U9UCAAJ), until now, there was no way of handling any alternative path for an arrival that was preempted and “kicked off” from a resource. This mechanism has been implemented into the new handle\_unfinished() activity:

patient <- trajectory() %>%

handle\_unfinished(reject) %>%

seize("nurse") %>%

log\_("nurse seized") %>%

timeout(5) %>%

release("nurse") %>%

log\_("nurse released")

env <- simmer() %>%

add\_resource("nurse", 1, 0, preemptive=TRUE, queue\_size\_strict=TRUE) %>%

add\_generator("LowPrio\_patient", patient, at(0), priority=0) %>%

add\_generator("HighPrio\_patient", patient, at(1), priority=10) %>%

run()

## 0: LowPrio\_patient0: nurse seized

## 1: HighPrio\_patient0: nurse seized

## 1: LowPrio\_patient0: kicked off...

## 6: HighPrio\_patient0: nurse released

Note that such a mechanism is more general, because it also covers the first scenario:

env <- simmer() %>%

add\_resource("nurse", 1, 0) %>%

add\_generator("patient", patient, at(0, 1)) %>%

run()

## 0: patient0: nurse seized

## 1: patient1: kicked off...

## 5: patient0: nurse released

Whenever rejection (or preemption) happens and it is catched by the appropriate handler, the new getter get\_seized() may be useful to know which resource was abandoned.

Finally, the readership may find interesting the new section about the implementation of [state-dependent service rates](https://r-simmer.org/articles/simmer-06-queueing.html#state-dependent-service-rates) in the [Queueing Systems](https://r-simmer.org/articles/simmer-06-queueing.html) vignette. See below for a complete list of changes.

**New features:**

* New handle\_unfinished() activity sets a drop-out trajectory for unfinished arrivals, i.e., those dropped from a resource (due to preemption, resource shrinkage or a rejected seize) or those that leave a trajectory (#178 addressing #177).
* New release\_all() and release\_selected\_all() activities automatically retrieve the amount of resources seized and release it (#180 addressing #25).
* New get\_seized() and get\_seized\_selected() getters allow an arrival to retrieve the amount of resources seized (#180 addressing #179).
* New stop\_if() activity sets a conditional breakpoint (#181 addressing #100).

**Minor changes and fixes:**

* Fix performance issues in data sources (#176).
* Update CITATION.
* Fix monitored activity for preempted arrivals (as part of #178).
* Fix seizes/releases with a null amount (as part of #180).
* Rename internal status codes (as part of #181).
* Provide more context on error or warning (as part of #181).
* Extend the Queueing Systems vignette with a section about state-dependent service rates.
* Fix performance issues in getters (#183).

*Article originally published in Enchufa2.es:* [*simmer 4.2.1*](https://www.enchufa2.es/archives/simmer-4-2-1.html).