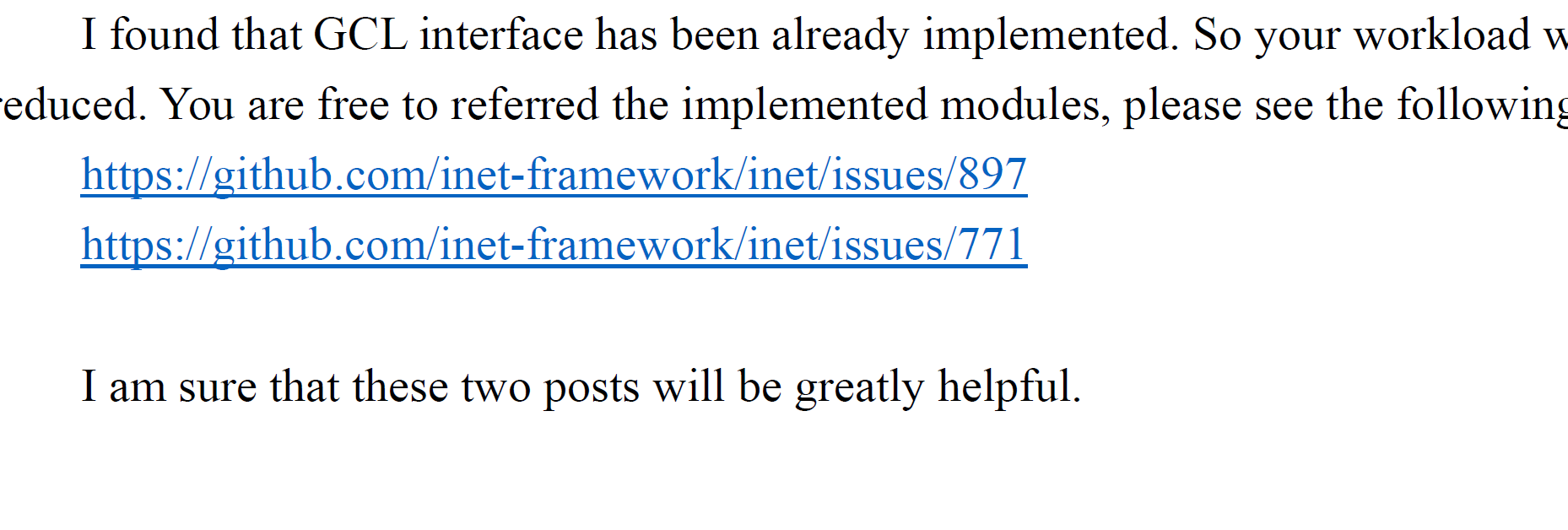
Please check the project requirement for task 4.



You are free to referred the implemented modules, please see the following posts:Please note that most parts of the configuration interface have already been implemented. So you can directly use it, please do not implement it from scratch (from the beginning).

<https://github.com/inet-framework/inet/issues/897>

**Problem with the GCL feature #897**

Closed

[ar8384](https://github.com/ar8384) opened this issue on Jul 3, 2023 · 18 comments

Closed

[**Problem with the GCL feature**](https://github.com/inet-framework/inet/issues/897#top)**#897**

[ar8384](https://github.com/ar8384) opened this issue on Jul 3, 2023 · 18 comments

**Comments**

[](https://github.com/ar8384)[**ar8384**](https://github.com/ar8384)**commented**[**on Jul 3, 2023**](https://github.com/inet-framework/inet/issues/897#issue-1786182936)**• edited**

|  |
| --- |
| Hello everyone,  I wanted to try the new GCL feature that came with INET 4.5. I ran the following simulation by adjusting the time-aware shaping showcase:  [General]  network = inet.networks.tsn.TsnLinearNetwork  sim-time-limit = 1s  description = "Traffic shaping using time-aware shapers"  outputvectormanager-class="omnetpp::envir::SqliteOutputVectorManager"  output-vector-file = results/resvec.sqlite  #\*\*.displayGateSchedules = true  #\*\*.gateFilter = "\*\*.eth[1].\*\*"  #\*\*.gateScheduleVisualizer.height = 16  #\*\*.gateScheduleVisualizer.placementHint = "top"  ###########################################  # Client configuration #  ###########################################  # client applications  \*.client.numApps = 2  \*.client.app[\*].typename = "UdpSourceApp"  \*.client.app[0].display-name = "best effort"  \*.client.app[1].display-name = "video"  \*.client.app[\*].io.destAddress = "server"  \*.client.app[0].io.destPort = 1000  \*.client.app[1].io.destPort = 1001  \*.client.app[0].source.packetLength = 1500B - 54B # 54B = 8B (UDP) + 20B (IP) + 14B (ETH MAC) + 4B (ETH FCS) + 8B (ETH PHY)  \*.client.app[1].source.packetLength = 1500B - 54B # 54B = 8B (UDP) + 20B (IP) + 14B (ETH MAC) + 4B (ETH FCS) + 8B (ETH PHY)  \*.client.app[0].source.productionInterval = uniform(0us, 400us)  \*.client.app[1].source.productionInterval = exponential(400us)  # enable outgoing streams  \*.client.hasOutgoingStreams = true  # client stream identification  \*.client.bridging.streamIdentifier.identifier.mapping = [{stream: "best effort", packetFilter: expr(udp.destPort == 1000)}, {stream: "video", packetFilter: expr(udp.destPort == 1001)}]  # client stream encoding  \*.client.bridging.streamCoder.encoder.mapping = [{stream: "best effort", pcp: 0}, {stream: "video", pcp: 4}]  ###########################################  # Server configuration #  ###########################################  # server applications  \*.server.numApps = 2  \*.server.app[\*].typename = "UdpSinkApp"  \*.server.app[0].io.localPort = 1000  \*.server.app[1].io.localPort = 1001  ##########################################  # Switch configuration #  ##########################################  # enable egress traffic shaping  \*.switch.hasEgressTrafficShaping = true  # disable forwarding IEEE 802.1Q C-Tag  \*.switch.bridging.directionReverser.reverser.excludeEncapsulationProtocols = ["ieee8021qctag"]  # time-aware traffic shaping  \*.switch.eth[\*].macLayer.queue.numTrafficClasses = 2  \*.switch.eth[\*].macLayer.queue.\*[0].display-name = "best effort"  \*.switch.eth[\*].macLayer.queue.\*[1].display-name = "video"  \*.switch.eth[\*].macLayer.queue.gateControlList.hasGcl = true  \*.switch.eth[\*].macLayer.queue.gateControlList.numGates = 2  \*.switch.eth[\*].macLayer.queue.gateControlList.offset = 0ms  \*.switch.eth[\*].macLayer.queue.gateControlList.durations = [4ms, 2ms, 2ms, 2ms]  \*.switch.eth[\*].macLayer.queue.gateControlList.gateStates = ["10", "00", "01", "00"]  # The following configuration is the same as the one above.  #\*.switch.eth[\*].macLayer.queue.transmissionGate[0].offset = 0ms  #\*.switch.eth[\*].macLayer.queue.transmissionGate[0].durations = [4ms, 6ms] # period is 10  #\*.switch.eth[\*].macLayer.queue.transmissionGate[1].offset = 6ms  #\*.switch.eth[\*].macLayer.queue.transmissionGate[1].durations = [2ms, 8ms]  At the end, it seemed to me that using switch.eth[\*].macLayer.queue.gateControlList is not equivalent to using switch.eth[\*].macLayer.queue.transmissionGate as I got very different results... It seems that switch.eth[\*].macLayer.queue.gateControlList is more equivalent to no TSN scheduling...  Did I miss something in the configuration?  Thank you in advance ;) |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Oct 9, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1752546591)

|  |
| --- |
| The configuration line \*.switch.eth[\*].macLayer.queue.gateControlList.hasGcl = true is wrong. The hasGcl parameter is in the queue, so it should read \*.switch.eth[\*].macLayer.queue.hasGcl = true instead. |

Top of Form

Bottom of Form

[](https://github.com/ar8384)Author [**ar8384**](https://github.com/ar8384)**commented**[**on Oct 10, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1755518488)

|  |
| --- |
| Thank you for your answer ;)  I'm not sure that this is the problem, as when I change \*.switch.eth[\*].macLayer.queue.hasGcl = true, the simulation finishes with an error: Simulation terminated with exit code: 8b Whereas the simulation is running prefectly fine with \*.switch.eth[\*].macLayer.queue.gateControlList.hasGcl = true.  Thank you for your help |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Oct 11, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1757931371)

|  |
| --- |
| That may be caused by an entirely different issue. It's impossible to decide what happens from an exit code. |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Oct 12, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1759574514)

|  |
| --- |
| This has been fixed in the master branch. Could you please verify? |

Top of Form

Bottom of Form

[](https://github.com/ar8384)Author [**ar8384**](https://github.com/ar8384)**commented**[**on Oct 12, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1759781619)

|  |
| --- |
| Hello,  I verified and the simulations are running without errors.  I however ran into another problem: with this configuration:  \*.switch.eth[\*].macLayer.queue.hasGcl = true  \*.switch.eth[\*].macLayer.queue.gateControlList.numGates = 2  \*.switch.eth[\*].macLayer.queue.gateControlList.offset = 0ms  \*.switch.eth[\*].macLayer.queue.gateControlList.durations = [4ms, 2ms, 2ms, 2ms]  \*.switch.eth[\*].macLayer.queue.gateControlList.gateStates = ["10", "00", "01", "00"]  I obtain the following gate state:    whereas it should look more to something like this, obtained with (which is I think the equivalent):  \*.switch.eth[\*].macLayer.queue.transmissionGate[0].offset = 0ms  \*.switch.eth[\*].macLayer.queue.transmissionGate[0].durations = [4ms, 6ms] # period is 10  \*.switch.eth[\*].macLayer.queue.transmissionGate[1].offset = 4ms # real offset = duration - offset  \*.switch.eth[\*].macLayer.queue.transmissionGate[1].durations = [2ms, 8ms]    If I understood correctly, GateControlList is a translator toward PeriodicGate. If then, I should retrieve the same results from the sqlite result file.  Thank you very much and sorry to bother |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Oct 12, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1759912243)

|  |
| --- |
| No problem, it's valuable to get bug reports. This component was contributed by another INET user. I'll check what's going on. |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Oct 13, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1761221563)

|  |
| --- |
| I have pushed a fix on the master branch. Could you please verify if this solves the problem? |

Top of Form

Bottom of Form

[](https://github.com/ar8384)Author [**ar8384**](https://github.com/ar8384)**commented**[**on Oct 15, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1763335577)

|  |
| --- |
| It works.  I may have, however, found another issue (last one I hope). When I change  \*.switch.eth[\*].macLayer.queue.numTrafficClasses = 2  \*.switch.eth[\*].macLayer.queue.hasGcl = true  \*.switch.eth[\*].macLayer.queue.gateControlList.numGates = 2  \*.switch.eth[\*].macLayer.queue.gateControlList.offset = 0ms  \*.switch.eth[\*].macLayer.queue.gateControlList.durations = [4ms, 2ms, 2ms, 2ms]  \*.switch.eth[\*].macLayer.queue.gateControlList.gateStates = ["10", "00", "01", "00"]  to  \*.switch.eth[\*].macLayer.queue.numTrafficClasses = 8  \*.switch.eth[\*].macLayer.queue.hasGcl = true  \*.switch.eth[\*].macLayer.queue.gateControlList.numGates = 8  \*.switch.eth[\*].macLayer.queue.gateControlList.offset = 0ms  \*.switch.eth[\*].macLayer.queue.gateControlList.durations = [4ms, 3ms, 2ms, 1ms]  \*.switch.eth[\*].macLayer.queue.gateControlList.gateStates = ["10000000", "01111111", "00000010", "00000001"]  I get the following error: (omnetpp::cValueArray): get(): index 2 is out of bounds -- in module (inet::queueing::GateControlList) TsnLinearNetwork.switch.eth[0].macLayer.queue.gateControlList (id=242), during network initialization  In fact, the GCL seems to work when numGates = 1 or 2, and not when numGates = 3, 4, 5, 6, 7, 8 (and that the gateStates is set accordingly).  Is there another parameter that I missed? |

Top of Form

👍1

Bottom of Form

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Oct 15, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1763383795)**via email**

|  |
| --- |
| I'm just guessing because I'm not in front of my computer right now. I believe that the number of PeriodicGate submodules is still 2, you should change that also to 8. IIRC it's the numTrafficCategories parametet. I hope this helps, levy  […](https://github.com/inet-framework/inet/issues/897) |

Top of Form

Bottom of Form

[](https://github.com/ar8384)Author [**ar8384**](https://github.com/ar8384)**commented**[**on Oct 16, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1764114992)

|  |
| --- |
| Hmm... there doesn't seem to be such a parameter... except numTrafficClasses, which I set to 8.  The parameters of PeriodicGates;  parameters:  string clockModule = default(""); // relative path of a module that implements IClock; optional  bool initiallyOpen @mutable = default(true); // specifies if the gate is initially open or closed  double offset @mutable @unit(s) = default(0s); // specifies where the period starts in the list of gate state change durations; where it is in the period at the start  object durations @mutable @unit(s) = default([]); // list of time durations within the period that determine if the gate is open or closed; period is the sum of the durations  bool scheduleForAbsoluteTime = default(true); // when a clock is used relative means that setting the clock will not affect the simulation time of the event  int openSchedulingPriority = default(0); // FES scheduling priority for the next gate open event  int closeSchedulingPriority = default(0); // FES scheduling priority for the next gate close event  bool enableImplicitGuardBand = default(true); // implicit guard band means that a packet is not allowed to be forwarded if it cannot finish transmission before the end of window.  @class(PeriodicGate);  @signal[guardBandStateChanged](type=bool);  @statistic[guardBandState](title="guard band state"; type=enum; enum=ON, OFF; source=guardBandStateChanged; record=vector; interpolationmode=sample-hold);  The parameters of GateControlList:  parameters:  int numGates = default(8);  object durations @mutable @unit(s) = default([]); // list of entry durations, e.g. [4ms, 2ms, 2ms, 2ms]  object gateStates @mutable = default([]); // list of gateStates, e.g. ["10000000", "01111111", "10000000", "01111111"]  @class(GateControlList);  @display("i=block/table");  As far as I remember, in PeriodicGates, there is no need to define the number of gates, as it is done like:  \*.switch.eth[\*].macLayer.queue.transmissionGate[0].offset = 0ms  \*.switch.eth[\*].macLayer.queue.transmissionGate[0].durations = [4ms, 6ms] # period is 10  \*.switch.eth[\*].macLayer.queue.transmissionGate[1].offset = 6ms  \*.switch.eth[\*].macLayer.queue.transmissionGate[1].durations = [2ms, 8ms] |

Top of Form

Bottom of Form

[](https://github.com/Musteblume)Contributor [**Musteblume**](https://github.com/Musteblume)**commented**[**on Oct 17, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1766623338)

|  |
| --- |
| It works.  I may have, however, found another issue (last one I hope). When I change  \*.switch.eth[\*].macLayer.queue.numTrafficClasses = 2  \*.switch.eth[\*].macLayer.queue.hasGcl = true  \*.switch.eth[\*].macLayer.queue.gateControlList.numGates = 2  \*.switch.eth[\*].macLayer.queue.gateControlList.offset = 0ms  \*.switch.eth[\*].macLayer.queue.gateControlList.durations = [4ms, 2ms, 2ms, 2ms]  \*.switch.eth[\*].macLayer.queue.gateControlList.gateStates = ["10", "00", "01", "00"]  to  \*.switch.eth[\*].macLayer.queue.numTrafficClasses = 8  \*.switch.eth[\*].macLayer.queue.hasGcl = true  \*.switch.eth[\*].macLayer.queue.gateControlList.numGates = 8  \*.switch.eth[\*].macLayer.queue.gateControlList.offset = 0ms  \*.switch.eth[\*].macLayer.queue.gateControlList.durations = [4ms, 3ms, 2ms, 1ms]  \*.switch.eth[\*].macLayer.queue.gateControlList.gateStates = ["10000000", "01111111", "00000010", "00000001"]  I get the following error: (omnetpp::cValueArray): get(): index 2 is out of bounds -- in module (inet::queueing::GateControlList) TsnLinearNetwork.switch.eth[0].macLayer.queue.gateControlList (id=242), during network initialization  In fact, the GCL seems to work when numGates = 1 or 2, and not when numGates = 3, 4, 5, 6, 7, 8 (and that the gateStates is set accordingly).  Is there another parameter that I missed?  Having exactly the same error with my config:  \*.\*.eth[\*].macLayer.queue.numTrafficClasses = 8  ...  \*.switch\*.eth[\*].macLayer.queue.hasGcl = true  \*.switch\*.eth[\*].macLayer.queue.gateControlList.numGates = 8  ...  \*.switchFrontLeft.eth[0].macLayer.queue.gateControlList.offset = 0us  \*.switchFrontLeft.eth[0].macLayer.queue.gateControlList.durations = [3us, 6us, 991us]  \*.switchFrontLeft.eth[0].macLayer.queue.gateControlList.gateStates = ["11111101", "00000010", "11111101"]  Best regards |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Oct 18, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1767946065)

|  |
| --- |
| This bug has been fixed in the master branch. I've noticed something though. The order of 1s and 0s in the GCL strings is applied to PeriodicGate submodules in reverse order. I don't know why it's implemented like that. For example, ["10" ,"10"] results in transmissionGate[0] to be closed and transmissionGate[1] to be open. For me, it would be more intuitive if the first digit would correspond to the first transmissionGate submodule, and the last digit would correspond to the last transmissionGate submodule. Do you think we should use the same order instead? |

Top of Form

Bottom of Form

[](https://github.com/Musteblume)Contributor [**Musteblume**](https://github.com/Musteblume)**commented**[**on Oct 18, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1767958960)

|  |
| --- |
| This would be also more intuitive for me, but this screenshot from 802.1Q-2018 shows it in the currently implemented order: |

Top of Form

Bottom of Form

[](https://github.com/ar8384)Author [**ar8384**](https://github.com/ar8384)**commented**[**on Oct 18, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1767974949)

|  |
| --- |
| This bug has been fixed in the master branch. I've noticed something though. The order of 1s and 0s in the GCL strings is applied to PeriodicGate submodules in reverse order. I don't know why it's implemented like that. For example, ["10" ,"10"] results in transmissionGate[0] to be closed and transmissionGate[1] to be open. For me, it would be more intuitive if the first digit would correspond to the first transmissionGate submodule, and the last digit would correspond to the last transmissionGate submodule. Do you think we should use the same order instead?  To be honest, this is the reason why i did this test with 8 queues: I wanted to understand how the mapping was done. As [@Musteblume](https://github.com/Musteblume) said, I guess the order is 76543201 (like in NeSTiNg) or 76543210. But I agree with you [@levy](https://github.com/levy) , it would be more intuitive if the first digit would correspond to the first transmissionGate submodule. |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Oct 18, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1768065862)

|  |
| --- |
| While I understand [@Musteblume](https://github.com/Musteblume)'s reasoning related to the standard I still have concerns. The traffic class to transmission gate index mapping is determined by the classifier submodule of the time-aware shaper compound queue (e.g. PcpTrafficClassClassifier), because that's the one that classifies and pushes packets into the subqueues. The gateControlList submodule has no way of knowing what this mapping is, and the classifier can be replaced by the user. So it means that the gateControlList submodule cannot reliably map gate states (1s and 0s) to traffic categories. It can only map gate states to transmission gate indices.  Maybe we should have a separate mapping parameter that specifies the gate indices? For example, mapping = "76543210" would mean the current behavior, that is mapping gate states to gate indices in reverse order. Unfortunately, this parameter would be required because there's no good default for a variable number of gates.  I'm not sure what to do, we can also leave it as it is right now. What's your opinion? |

Top of Form

Bottom of Form

[](https://github.com/Musteblume)Contributor [**Musteblume**](https://github.com/Musteblume)**commented**[**on Oct 18, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1768460055)

|  |
| --- |
| I would lean towards index order (01234567) because this is what i would expect when configuring. |

Top of Form

Bottom of Form

[](https://github.com/ar8384)Author [**ar8384**](https://github.com/ar8384)**commented**[**on Oct 19, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1770419968)

|  |
| --- |
| I agree that having a separate mapping parameter that specifies the gate indices is a good idea. This way, everyone is responsible for his own mapping and it will be clear which gate correpond to which indice. |

Top of Form

👍2

Bottom of Form

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Oct 19, 2023**](https://github.com/inet-framework/inet/issues/897#issuecomment-1770985187)

|  |
| --- |
| I've pushed the mapping parameter change into master. |

Top of Form

❤️2

Bottom of Form

Top of Form

Bottom of Form

[Sign up for free](https://github.com/join?source=comment-repo)

<https://github.com/inet-framework/inet/issues/771>

**[TSN feature] GCL capacity in GateScheduleConfiguratorBase or PeriodicGate #771**

Closed

[yue2388253](https://github.com/yue2388253) opened this issue on Jun 20, 2022 · 10 comments

Closed

[**[TSN feature] GCL capacity in GateScheduleConfiguratorBase or PeriodicGate**](https://github.com/inet-framework/inet/issues/771#top)**#771**

[yue2388253](https://github.com/yue2388253) opened this issue on Jun 20, 2022 · 10 comments

**Comments**

[](https://github.com/yue2388253)[**yue2388253**](https://github.com/yue2388253)**commented**[**on Jun 20, 2022**](https://github.com/inet-framework/inet/issues/771#issue-1276247590)

|  |
| --- |
| Hi all,  The number of a gate control list is limited in a real TSN switch (e.g., 1024 gate control entries). Current implementation does not take this limitation into consideration. Is it possible to implement this feature?  Best regards,  Lam |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Jun 20, 2022**](https://github.com/inet-framework/inet/issues/771#issuecomment-1160057903)

|  |
| --- |
| The gate schedule configurator modules configure all existing gates in the network. Obviously, there's no way to configure a gate that doesn't exist, so all it could do perhaps is not to configure some of the gates which are there. I don't know if that makes sense, because then how could the configurator make sure that the jitter and latency limits are met.  This feature doesn't make too much sense to me. Can you describe the use case you are trying to solve? |

Top of Form

Bottom of Form

[](https://github.com/yue2388253)Author [**yue2388253**](https://github.com/yue2388253)**commented**[**on Jun 20, 2022**](https://github.com/inet-framework/inet/issues/771#issuecomment-1160369593)

|  |
| --- |
| The gate state is controlled by a gate control list (GCL) according to IEEE 802.1Qbv standard[1], as shown below. [image](https://user-images.githubusercontent.com/33801130/174596646-09e7cb3f-f735-4f7e-b3b8-d847b51c03bd.png)  The length of a GCL is not inifinte, due to the limited hardware capacity. In fact, a typical TSN switch can only support 1024 gate control entries. Therefore, a GateConfigurator should take this GCL limitation into consideration, otherwise the calculated schedule would be impractical.  I think it would be great to set a upper bound for the GCL length and verify whether the output of a GateConfigurator is feasible.  Reference: [1] "IEEE Standard for Local and metropolitan area networks -- Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic," in IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015) , vol., no., pp.1-57, 18 March 2016, doi: 10.1109/IEEESTD.2016.8613095. |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Jun 20, 2022**](https://github.com/inet-framework/inet/issues/771#issuecomment-1160388939)**via email**

|  |
| --- |
| Ah, I see. I misunderstood what kind of limit you are talking about. Yes, this makes sense, although it's not trivial to implement.  […](https://github.com/inet-framework/inet/issues/771) |

Top of Form

Bottom of Form

[](https://github.com/yue2388253)Author [**yue2388253**](https://github.com/yue2388253)**commented**[**on Jun 22, 2022**](https://github.com/inet-framework/inet/issues/771#issuecomment-1163087367)**• edited**

|  |
| --- |
| Currently the time aware shaper is implemented via PeriodicGate, which is configured individually, and is different from the IEEE 802.1Qbv standard where gates are configured by a GCL. Is there any plan to re-implement it? I am currently doing some research on the GCL synthesis problem, and maybe I can be of some help. |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Jun 22, 2022**](https://github.com/inet-framework/inet/issues/771#issuecomment-1163191693)**via email**

|  |
| --- |
| I don't think that really matters and no there are no plans to change this. In principle, you could convert the data from one representation to the other in both directions. It is done this way because it is easier with the modular omnetpp architecture. Basically, you could have a GCL module which extracts this data from the underlying PeriodicGate modules. The other direction is also possible but it's even less appealing because the current modular architecture allows to create designs that go beyond what's in the standard. Best regards, levy  […](https://github.com/inet-framework/inet/issues/771) |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Jun 22, 2022**](https://github.com/inet-framework/inet/issues/771#issuecomment-1163209784)**via email**

|  |
| --- |
| So I thought about this problem a little bit more, and I think this could be solved the following way. The constraint model of the SAT solver has to be extended with separate variables for each slot that is available in the hardware GCL (e.g. 1024 variables per gate). Each variable would represent a moment in time when the gate opens or closes. Additional constraints have to be added to ensure that the values of these variables monotonically increase in time, and also that no transmission start/end happens outside a gate open period (defined by the values of two subsequent variables). The final result should be extracted using these new variables instead of the current approach where the transmission start/end variables are used. This change guarantees that the queues still keep the packet ordering, the traffic class priorities are still taken into consideration, etc. To further optimize the results, all gate open periods can be omitted which don't contain at least one transmission start/end pair. I don't think I'm going to do this in the near future but you could give it a try by changing the Z3GateScheduleConfigurator::computeGateScheduling method. Best regards, levy  […](https://github.com/inet-framework/inet/issues/771) |

Top of Form

Bottom of Form

[](https://github.com/yue2388253) Author [**yue2388253**](https://github.com/yue2388253)**commented**[**on Jun 23, 2022**](https://github.com/inet-framework/inet/issues/771#issuecomment-1164044299)**via email**

|  |
| --- |
| Hi levy, I think having a GCL module to convert data from PeriodicGate to GCL format and vice versa is appealing, since it would be helpful for a user who has some GCL format data and wants to verify the result via Omnet. These GCL data may be generated by the user's own scheduling algorithm which may be written in other languages and cannot be changed to a GateScheduleConfigurator easily. Best regards, Lin  ---- Replied Message ---- | From | Levente \*\*\*@\*\*\*.\*\*\*> | | Date | 06/22/2022 22:52 | | To | \*\*\*@\*\*\*.\*\*\*> | | Cc | Jiashuo \*\*\*@\*\*\*.\*\*\*\*\*\*@\*\*\*.\*\*\*> | | Subject | Re: [inet-framework/inet] [TSN feature] GCL capacity in GateScheduleConfiguratorBase or PeriodicGate (Issue [#771](https://github.com/inet-framework/inet/issues/771)) | So I thought about this problem a little bit more, and I think this could be solved the following way. The constraint model of the SAT solver has to be extended with separate variables for each slot that is available in the hardware GCL (e.g. 1024 variables per gate). Each variable would represent a moment in time when the gate opens or closes. Additional constraints have to be added to ensure that the values of these variables monotonically increase in time, and also that no transmission start/end happens outside a gate open period (defined by the values of two subsequent variables). The final result should be extracted using these new variables instead of the current approach where the transmission start/end variables are used. This change guarantees that the queues still keep the packet ordering, the traffic class priorities are still taken into consideration, etc. To further optimize the results, all gate open periods can be omitted which don't contain at least one transmission start/end pair. I don't think I'm going to do this in the near future but you could give it a try by changing the Z3GateScheduleConfigurator::computeGateScheduling method. Best regards, levy  On Mon, Jun 20, 2022 at 2:10 PM Jiashuo Lin \*\*\*@\*\*\*.\*\*\*> wrote: The gate state is controlled by a gate control list (GCL) according to IEEE 802.1Qbv standard[1], as shown below. [image: image] <<https://user-images.githubusercontent.com/33801130/174596646-09e7cb3f-f735-4f7e-b3b8-d847b51c03bd.png>> The length of a GCL is not inifinte, due to the limited hardware capacity. In fact, a typical TSN switch can only support 1024 gate control entries. Therefore, a GateConfigurator should take this GCL limitation into consideration, otherwise the calculated schedule would be impractical. I think it would be great to set a upper bound for the GCL length and verify whether the output of a GateConfigurator is feasible. Reference: [1] "IEEE Standard for Local and metropolitan area networks -- Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic," in IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015) , vol., no., pp.1-57, 18 March 2016, doi: 10.1109/IEEESTD.2016.8613095. — Reply to this email directly, view it on GitHub <[#771 (comment)](https://github.com/inet-framework/inet/issues/771#issuecomment-1160369593)>, or unsubscribe <<https://github.com/notifications/unsubscribe-auth/AAA3OTJLHEWEUE43QQSDDY3VQBNURANCNFSM5ZHI7TYA>> . You are receiving this because you commented.Message ID: \*\*\*@\*\*\*.\*\*\*>  — Reply to this email directly, view it on GitHub, or unsubscribe. You are receiving this because you authored the thread.Message ID: \*\*\*@\*\*\*.\*\*\*> |

Top of Form

Bottom of Form

[](https://github.com/yue2388253)Author [**yue2388253**](https://github.com/yue2388253)**commented**[**on Jun 24, 2022**](https://github.com/inet-framework/inet/issues/771#issuecomment-1165261682)**• edited**

|  |
| --- |
| Hi Levy,  In the TSN TAS showcase ("showcases/tsn/trafficshaping/timeawareshaper/omnetpp.ini"), the gates are configured as follows:  \*.switch.eth[\*].macLayer.queue.transmissionGate[0].offset = 0ms  \*.switch.eth[\*].macLayer.queue.transmissionGate[0].durations = [4ms, 6ms] # period is 10 # length of periods  \*.switch.eth[\*].macLayer.queue.transmissionGate[1].offset = 6ms  \*.switch.eth[\*].macLayer.queue.transmissionGate[1].durations = [2ms, 8ms]  I'd like to implement a GCL module so that one can configure the gate states as follows, which is totally the same as the configuration above.  \*.switch.eth[\*].macLayer.queue.gateControlList.offset = 0ms  \*.switch.eth[\*].macLayer.queue.gateControlList.entries = ["01", "00", "10", "00"] // Or may be ["11111101", "11111100", "11111110", "11111100"]  \*.switch.eth[\*].macLayer.queue.gateControlList.durations = [4ms, 2ms, 2ms, 2ms]  Could you give me some advices that how to implement such a module? Thanks.  Best regards, Lin |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on Jun 24, 2022**](https://github.com/inet-framework/inet/issues/771#issuecomment-1165741282)**via email**

|  |
| --- |
| You just need to create a module which processes its parameters in the initialize method and looks up the individual gate modules and sets their parameters. The auto configurators essentially do the same, they start from a different set of parameters and process then differently.  […](https://github.com/inet-framework/inet/issues/771) |

Top of Form

Bottom of Form

[](https://github.com/levy)Contributor [**levy**](https://github.com/levy)**commented**[**on May 18, 2023**](https://github.com/inet-framework/inet/issues/771#issuecomment-1553184481)

|  |
| --- |
| The GateControlList module that was submitted in a pull request has already been merged into INET and it is part of INET 4.5. So I guess I can close this issue. |