Network Working Group
Internet-Draft
Intended status: Informational
Expires: 30 August 2026

C. Huitema
Private Octopus Inc.
S. Nandakumar
C. Jennings
Cisco

26 February 2026

Testing of Christian’s Congestion Control Code (C4)

draft-huitema-ccwg-c4-test-02

Abstract

Christian’s Congestion Control Code is a new congestion control algorithm designed to support Real-Time applications such as Media over QUIC. It is designed to drive towards low delays, with good support for the "application limited" behavior frequently found when using variable rate encoding, and with fast reaction to congestion to avoid the "priority inversion" happening when congestion control overestimates the available capacity. The design was validated by series of simulations, and also by initial deployments in control networks. We describe here these simulations and tests.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Christian’s Congestion Control Code (C4) is a new congestion control algorithm designed to support Real-Time multimedia applications, specifically multimedia applications using QUIC [RFC9000] and the Media over QUIC transport [I-D.ietf-moq-transport]. The design was validated by series of simulations, and also by initial deployments in control networks. We describe here these simulations (see Section 2) and tests (see Section 3)

2. Simulations

We tested the design by running a series of simulations, which covered:

* reaction to network events

* competition with other congestion control algorithms

* handling of high jitter environments

* handling of multimedia applications

2.1. Testing strategy

We are running the tests using the picoquic network simulator [Picoquic_ns]. The simulator embeds the picoquic implementation of QUIC [Picoquic]. Picoquic itself comes with support for a variety of congestion control protocols, including Cubic and BBR. We added an implementation of C4.

That implementation is designed so that the same code can be used in execution over the network and in simulations, the main difference being a replacement of the socket API by a simulation API. When running in simulation, the code runs in "virtual time", with a virtual clock driven by simulation events such as arrival and departure of packets from simulated queues. With the virtual clock mechanism, we can simulate in a fraction of a second a connection that would last 10 seconds in "real time".

Our basic test is to run a simulation, measure the simulated duration of a connection, and compare that to the expected nominal value. When we were developing the C4, we used that for detecting possible regressions, and progressively refine the specification of the algorithm.

Simulations include random events, such as network jitter or the precise timing of packet arrivals and departure. Minute changes in starting conditions can have cascading effects. When running the same test multiple times, we are likely to observe different values each time. When comparing two code versions, we may also observe different results of a given test, but it is hard to know whether these results. To get reliable results, we run each test 100 times, and we consider the test passing if all the worse result of these 100 times was withing the expected value.

2.2. Reaction to network events

The first series of simulation test how C4 behaves in simple scenarios when it is the sole user of a link. The list of test includes:

* a 20Mbps connection,
* a 200Mbps connection,
* a geostationary satellite connection,
* a sudden increase in path capacity, i.e. "low and up"
* a sudden decrese in path capacity followed by a return to normal, i.e. "drop and back"
* a sudden drop to 0 of path capacity for 2 seconds, i.e. a "black hole"
* a sudden increase in path latency, from "short" to "long"

2.2.1. Simulation of a simple 20Mbps connection

This scenario simulates a 10MB download over a 20 Mbps link, with an 80ms RTT, and a bottlneck buffer capacity corresponding to 1 BDP. The test verifies that 100 simulations all complete in less than 4.7 seconds.

In a typical simulation, we see a initial phase complete in less than 800ms, followed by a recovery phase in which the transmission rate stabilizes to the line rate. After that, the RTT remains very close to the path RTT, except for periodic small bumps during the "push" transitions.

2.2.2. Simulation of a simple 200Mbps connection

This scenario simulates a 20MB download over a 200 Mbps link, with a 40ms RTT, and a bottleneck buffer capacity corresponding to 1 BDP. The test verifies that 100 simulations all complete in less than 1.3 seconds.

This short test shows that the initial phase correctly discover the path capacity, and that the transmission operates at the expected rate after that.

2.2.3. Simulation of a geostationary satellite connection

This scenario simulates a 100MB download over a 250 Mbps link, with a 600ms RTT, and a bottleneck buffer capacity corresponding to 1 BDP, i.e., simulating a geostationary satellite connection. The scenario also tests the support for careful resume [I-D.ietf-tsvwg-careful-resume] by setting the remembered CWND to 18750000 bytes and the remembered RTT to 600.123ms. The test verifies that 100 simulations all complete in less than 7.7 seconds.

2.2.4. Low and up

The "low and up" scenario simulates a sudden increase in the capacity of the path. At the beginning of the simulation, the simulated bandwidth is set at 5 Mbps. It increases to 10 Mbps after 2.5 seconds. The RTT remains constant at 100ms. The test verifies that 100 simulations of a 7MB download all complete in less than 8.6 seconds.

The goal of the test is to verify that C4 promptly discovers the increase in bandwidth, and increases the transmission rate.

2.2.5. Drop and back

The "drop and back" scenario simulates a sudden decrease in the capacity of the path, followed by return to normal. At the beginning of the simulation, the simulated bandwidth is set at 10 Mbps. It decreases to 5 Mbps after 1.5 second, then returns to 10 Mbps after 2 seconds. The RTT remains constant at 100ms. The test verifies that 100 simulations of a 7MB download all complete in less than 8.25 seconds.

The goal of the test is to verify that C4 adapts promptly to changes in the available bandwidth on a path.

2.2.6. Black Hole

The "black hole" scenario simulates a sudden decrease in the capacity of the path, followed by return to normal. At the beginning of the simulation, the simulated bandwidth is set at . After 2 seconds, the path capacity is set to 0, and is restored to normal 2 seconds later. The RTT remains constant at 70ms. The test verifies that 100 simulations of a 10MB download all complete in less than 6.1 seconds.

The goal of the test is to verify that C4 recovers promptly after a short suspension of the path.

2.2.7. Short to long

The "short and long" scenario simulates a sudden increase in the latency of the path. At the beginning of the simulation, the simulated RTT is set at 30ms. After 1 second, the latency increases to 100ms. The data rate remains constant at 100ms. The test verifies that 100 simulations of a 20MB download all complete in less than 22 seconds.

The goal of the test is to verify that C4 react properly exercises the "slow down" mechanism to discover the new RTT.

2.3. L4S and ECN

The "ECN" test simulates a 20 Mbps link, with an 80ms RTT, and a bottleneck buffer capacity corresponding to 1 BDP. The test verifies that 100 simulated downloads of 10 MB all complete in less than 4.5 seconds.

2.4. Handling of High Jitter Environments

In the design of C4, we have been paying special attention to "bad Wi-Fi" environments, in which the usual delays of a few milliseconds could spike to 50 or even 200ms. We spent a lot of time trying to understand what causes such spikes. Our main hypothesis is that this happens when multiple nearby Wi-Fi networks operate on the same frequency or "channel", which causes collisons due to the hidden node problem. This causes collisions and losses, to which Wi-Fi responses involves two leves of exponential back-off.

We built a model to simulate this jitter by combining two generators:

* A random value r between 0 and 1 ms to model collision avoidance,

* A Poisson arrival model with lambda=1 providing the number N1 of short scale 1ms intervals to account for collision defferal and retry,

* A Poisson arrival arrival model with lambda = 12, and an interval length of 7.5ms to account for Wi-Fi packet restransmission.

We combine these generators models by using a coefficient "x" that indicates the general degree of collisions and repetitions:

* For a fraction (1-x) of the packets, we set the number N2 to 0.

* For a fraction (x) of the packets, we compute N2 from the Poisson arrival model with lambda = 12, and an interval length of 7.5ms.

The latency for a single sample will be: ~~~ latency = N1_1ms + N2_7.5ms if N1 >= 1: latency -= r ~~~ The coefficient x is derived from the target average jitter value. If the target is 1ms or less, we set x to zero. If it is higher than 91ms, we set x to 1. If it is in between, we set: ~~~ x = (average_jitter - 1ms)/90ms ~~~ We have been using this simulation of jitter to test our implementation of multiple congestion control algorithms.

2.4.1. Bad Wi-Fi test

The "bad Wi-Fi" test simulates a connection experiencing a high level of jitter. The average jitter is set to 7ms, which implies multiple spikes of 100 to 200ms every second. The data rate is set to 10Mbps, and the base RTT before jitter is set to 2ms, i.e., simulating a local server. The test pass if 100 different 10MB downloads each complete in less than 4.3 seconds.

2.4.2. Wifi fade trial

The "Wi-Fi fade" trial simulates varying conditions. The connection starts with a data rate of 20Mbps, an 80ms latency, and Wi-Fi jitter with average 1ms. After 1 second, the data rate drops to 2Mbps and the jitter average increases to 12ms. After another 2 seconds, data rate and jitter return to the original condition. The test pass if 100 different 10MB downloads each complete in less than 6.2 seconds.

2.4.3. Wifi suspension trial

The "Wi-Fi suspension" test simulates a connection experiencing multiple "suspensions". For every 1.8 second of a 2 second interval, the data rate is set to 20Mbps, and the base RTT before jitter is set to 10ms. For the last 200ms of these intervals, the data rate is set to 0. This model was developed before we got a better understanding of the Wi-Fi jitter. It is obsolete, but we kept it as a test case anyhow. The test pass if 100 different 10MB downloads each complete in less than 5.4 seconds.

2.5. Competition with itself

In accordance with [RFC9743], we design series of tests of multiple competing flows all using C4. We want to test different conditions, such as data rate and latency, and also different scenarios, such as testing whether the "background" connection starts at the same time, before or after the "main" connection.

We test that the bandwidth is shared reasonably by testing the completion time of a download, and setting the target value so it can only be achieved if the main connection gets "about half" of the bandwidth.

2.5.1. Two short C4 simultaneous connections

Our first test simulates two C4 connections starting at the same time and using the same path. The path has a 20Mbps data rate and 80ms RTT. The background connection tries to download 10MB, the main connection downloads 5MB. The test pass if in 100 trials the main connection completes in less than 6.7 seconds.

2.5.2. Short background C4 connection first

The "background first" test simulates two C4 connections using the same path with the background connection starting 0.5 seconds before the main connection. The path has a 20Mbps data rate and 80ms RTT. The background connection tries to download 10MB, the main connection downloads 5MB. The test pass if in 100 trials the main connection completes in less than 8.15 seconds after the beginning of the trial.

2.5.3. Short background C4 connection last

The "background last" simulates two C4 connections using the same path with the background connection starting at the 0.5 seconds after the main connection. The path has a 50Mbps data rate and 30ms RTT. The background connection tries to download 20MB, the main connection downloads 10MB. The test pass if in 100 trials the main connection completes in less than 4.1 seconds after the beginning of the trial.

2.5.4. Two long C4 connections

The long connection test simulates two C4 connections starting at the same time and using the same path. The path has a 20Mbps data rate and 80ms RTT. The background connection tries to download 30MB, the main connection downloads 20MB. The test pass if in 100 trials the main connection completes in less than 23 seconds.

2.5.5. Long background C4 connection last

The long "background last" test simulates two C4 connections using the same path with the background connection starting 1 second after the main connection. The path has a 10Mbps data rate and 70ms RTT. The background connection tries to download 15MB, the main connection downloads 10MB. The test pass if in 100 trials the main connection completes in less than 23 seconds after the beginning of the trial.

2.5.6. Compete with C4 over bad Wi-Fi

The "compete over bad Wi-Fi" test simulates two C4 connections using the same "bad Wi-Fi" path and starting, with the main connection starting 1 second after the background connection. The path has a 10Mbps data rate and 2ms RTT, plus Wi-Fi jitter set to 7ms average -- the same jitter characteristics as in the "bad Wi-Fi" test (see Section 2.4.1). The background connection tries to download 10MB, the main connection downloads 4MB. The test pass if in each of 100 trials the main connection completes in less than 12 seconds after the beginning of the trial.

2.6. Competition with Cubic

In accordance with [RFC9743], we design series of tests of multiple competing flows using C4 and Cubic. We want to test different conditions, such as data rate and latency, and also different scenarios, such as testing whether the "background" connection starts at the same time, before or after the "main" connection.

We test that the bandwidth is shared reasonably by testing the completion time of a download, and setting the target value so it can only be achieved if the main connection gets "about half" of the bandwidth.

2.6.1. Two short C4 and Cubic connections

Our first test simulates two C4 and Cubic connections starting at the same time and using the same path. The path has a 20Mbps data rate and 80ms RTT. The background Cubic connection tries to download 10MB, the main connection downloads 5MB. The test pass if in 100 trials the main connection completes in less than 6.8 seconds.

2.6.2. Two long C4 and Cubic connections

The long connection test simulates two C4 and Cubic connections starting at the same time and using the same path. The path has a 20Mbps data rate and 80ms RTT. The background connection tries to download 30MB, the main connection downloads 20MB. The test pass if in 100 trials the main connection completes in less than 23 seconds.

2.6.3. Long Cubic background connection last

The long "background last" test simulates two C4 and Cubic connections using the same path with the background Cubic connection starting 1 second after the main connection. The path has a 10Mbps data rate and 70ms RTT. The background connection tries to download 15MB, the main connection downloads 10MB. The test pass if in 100 trials the main connection completes in less than 23 seconds after the beginning of the trial.

2.6.4. Compete with Cubic over bad Wi-Fi

The "compete over bad Wi-Fi" test simulates two C4 and Cubic connections using the same "bad Wi-Fi" path, with the main connection starting 1 second after the background connection. The path has a 10Mbps data rate and 2ms RTT, plus Wi-Fi jitter set to 7ms average -- the same jitter characteristics as in the "bad Wi-Fi" test (see Section 2.4.1). The background connection tries to download 10MB, the main connection downloads 4MB. The test pass if in each of 100 trials the main connection completes in less than 12.5 seconds after the beginning of the trial.

2.7. Competition with BBR

In accordance with [RFC9743], we design series of tests of multiple competing flows using C4 and BBR. We want to test different conditions, such as data rate and latency, and also different scenarios, such as testing whether the "background" connection starts at the same time, before or after the "main" connection.

We test that the bandwidth is shared reasonably by testing the completion time of a download, and setting the target value so it can only be achieved if the main connection gets "about half" of the bandwidth.

2.7.1. Two short C4 and BBR connections

Our first test simulates two C4 and BBR connections starting at the same time and using the same path. The path has a 20Mbps data rate and 80ms RTT. The background BBR connection tries to download 10MB, the main connection downloads 5MB. The test pass if in 100 trials the main connection completes in less than 6.7 seconds.

2.7.2. Two long C4 and BBR connections

The long connection test simulates two C4 and BBR connections starting at the same time and using the same path. The path has a 20Mbps data rate and 80ms RTT. The background connection tries to download 30MB, the main connection downloads 20MB. The test pass if in 100 trials the main connection completes in less than 23 seconds.

2.7.3. Long BBR background connection last

The long "background last" test simulates two C4 and BBR connections using the same path with the background BBR connection starting 1 second after the main connection. The path has a 10Mbps data rate and 70ms RTT. The background connection tries to download 15MB, the main connection downloads 10MB. The test pass if in 100 trials the main connection completes in less than 23 seconds after the beginning of the trial.

2.7.4. Compete with BBR over bad Wi-Fi

The "compete over bad Wi-Fi" test simulates two C4 and BBR connections using the same "bad Wi-Fi" path, with the main connection starting 1 second after the background BBR connection. The path has a 10Mbps data rate and 2ms RTT, plus Wi-Fi jitter set to 7ms average -- the same jitter characteristics as in the "bad Wi-Fi" test (see Section 2.4.1). The background connection tries to download 10MB, the main connection downloads 4MB. The test pass if in each of 100 trials the main connection completes in less than 14.5 seconds after the beginning of the trial.

2.8. Handling of Multimedia Applications

C4 is specifically designed to properly handle multimedia applications. We test that function by running simulations of a call including:

* a simulated audio stream sending 80 bytes simulated audio segments every 20 ms.

* a simulated compressed video stream, sending 30 frames per second, organized as groups of 30 frames each starting with a 37500 bytes simulated I-Frame followed by 149 3750 bytes P-frames.

* a simulated less compressed video stream, sending 30 frames per second, organized as groups of 30 frames each starting with a 62500 bytes simulated I-Frame followed by 149 6250 bytes P-frames.

The simulation sends each simulated audio segment as QUIC datagram, with QUIC priority 2, and each group of frames as a separate QUIC stream with priority 4 for the compressed stream, and a priority 6 for the less compressed stream.

If the frames delivered on the less compressed stream fall are delivered more than 250ms later than the expected time, the receiver sends a "STOP SENDING" request on the QUIC stream to cancel it; transmission will restart with the next group of frame, simulating a plausible "simulcast" behavior.

The simulator collects statistics on the delivery of media frame, which are summarized as average and maximum frame delivery delay. For each test, the simulation specifies an expected average and an expected maximum delay, as well as a "start measurement" time, typically set long enough to start after the initial "startup" phase. The test passes if the average and max value for the simulated audio and for the simulated compressed video measured after the start time are below the specified values.

2.8.1. Media on High Speed Connection

The "high speed" media test verifies how C4 handles media on a 100 Mbps connection with a 30ms RTT. The test lasts for 5 video groups of frames, i.e. 5 seconds. The measurements start 200ms after the start of the connection. The expected average delay is set to 31ms, and the maximum delay is set to 79ms. The test is successful if 100 trials are all successful.

2.8.2. Media on 10 Mbps Connection

The "high speed" media test verifies how C4 handles media on a 10 Mbps connection with a 40ms RTT. The test lasts for 5 video groups of frames, i.e. 5 seconds. The measurements start 200ms after the start of the connection. The expected average delay is set to 47ms, and the maximum delay is set to 160ms. The test is successful if 100 trials are all successful.

2.8.3. Media for 20 seconds

The "20 seconds" media test verifies that media performance does not degrade over time, simulating a 100Mbps connection with a 30ms RTT. The test lasts for 20 video groups of frames, i.e. 20 seconds. The measurements start 200ms after the start of the connection. The expected average delay is set to 33ms, and the maximum delay is set to 80ms. The test is successful if 100 trials are all successful.

2.8.4. Media over varying RTT

The "varying RTT" media test verifies that media performance does not degrade over time, simulating a 100Mbps connection with a 30ms RTT, that changes to a 100ms RTT after 1 second. The test lasts for 10 video groups of frames, i.e. 10 seconds. The measurements start 5 seconds after the start of the connection. The expected average delay is set to 110ms, and the maximum delay is set to 127ms. The test is successful if 100 trials are all successful.

2.8.5. Media over varying Wi-Fi

The "varying Wi-Fi" media test verifies that media performance does not degrade too much on a connection that has the kind of jitter discussed in Section 2.4. The connection has the characteristics similar to the "fading Wi-Fi" scenario described in Section 2.4.2. The connection starts with a data rate of 20Mbps, 40ms RTT, and Wi-Fi jitter with average 1ms. After 1 second, the data rate drops to 2Mbps and the jitter average increases to 12ms. The test lasts for 5 video groups of frames, i.e. 5 seconds. The measurements start 200ms after the start of the connection. The expected average delay is set to 110ms, and the maximum delay is set to 350ms. The test is successful if 100 trials are all successful.

2.8.6. Media with Wi-Fi suspensions

The "varying Wi-Fi" media test verifies that media performance does not degrade too much on a connection experiences suspensions as discussed in Section 2.4.3. For every 1.8 second of a 2 second interval, the data rate is set to 20Mbps, and the base RTT before jitter is set to 10ms. For the last 200ms of these intervals, the data rate is set to 0. The test lasts for 5 video groups of frames, i.e. 5 seconds. The measurements start 200ms after the start of the connection. The expected average delay is set to 23.6ms, and the maximum delay is set to 202ms. The test is successful if 100 trials are all successful.

2.8.7. Media over bad Wi-Fi

The "bad Wi-Fi" media test verifies that media performance does not degrade too much on a connection that has the kind of jitter discussed in Section 2.4. The connection has the characteristics similar to the "bad Wi-Fi" scenario described in Section 2.4.1. The average jitter is set to 7ms, which implies multiple spikes of 100 to 200ms every second. The data rate is set to 20Mbps, and the base RTT before jitter is set to 2ms, i.e., simulating a local server. The test lasts for 5 video groups of frames, i.e. 5 seconds. The measurements start 200ms after the start of the connection. The expected average delay is set to 100ms, and the maximum delay is set to 680ms. The test is successful if 100 trials are all successful.

3. Tests

We need real life tests as well.

3.1. Loopback tests

Loopback tests were performed on Windows, downloading 10GB of data over a loopback connection. They showed picoquic using C4 achieving a data rate of 3Gbps, slightly more than the 2.9Gbps achieved when using Cubic or the 2.6 Gbps achieved when using BBR.

3.2. Webex prototype deployments

To do. Write down.

4. Security Considerations

This documentation of protocol testing does not have any particular security considerations.

We did not include specific security oriented tests in this document.

5. IANA Considerations

This document has no IANA actions.

6. Informative References

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[I-D.ietf-moq-transport] Nandakumar, S., Vasiliev, V., Swett, I., and A. Frindell, "Media over QUIC Transport", Work in Progress, Internet-Draft, draft-ietf-moq-transport-16, 13 January 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-moq-transport-16>.

[I-D.ietf-tsvwg-careful-resume] Kuhn, N., Stephan, E., Fairhurst, G., Secchi, R., and C. Huitema, "Convergence of Congestion Control from Retained State", Work in Progress, Internet-Draft, draft-ietf-tsvwg-careful-resume-24, 1 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-careful-resume-24>.

[RFC9743] Duke, M., Ed. and G. Fairhurst, Ed., "Specifying New Congestion Control Algorithms", BCP 133, RFC 9743, DOI 10.17487/RFC9743, March 2025, <https://www.rfc-editor.org/info/rfc9743>.

[Picoquic] Huitema, C., "Picoquic", GitHub Repository , 2025, <https://https://github.com/private-octopus/picoquic>.

[Picoquic_ns] Huitema, C., "Picoquic Network Simulator", GitHub Repository , 2025, <https://https://github.com/private-octopus/picoquic_ns>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Christian Huitema
Private Octopus Inc.
Email: huitema@huitema.net

Suhas Nandakumar
Cisco
Email: snandaku@cisco.com

Cullen Jennings
Cisco
Email: fluffy@iii.ca