Performance Evaluation of QUIC with BBR in Satellite Internet

Yue Wang*, Kanglian Zhao*, Wenfeng Li*, Juan Fraire†, Zhili Sun‡ and Yuan Fang*
*School of Electronic Science and Engineering, Nanjing University, Nanjing, Jiangsu, 210023, P.R. China
†Laboratorio de Comunicaciones Digitales, Universidad Nacional de Córdoba. Códorba, Argentina
‡Institute for Communication Systems, University of Surrey, Guildford GU27XH, UK

***Abstract*—Quick UDP Internet connection(QUIC) protocol and bottleneck bandwidth and round-trip propagation time (BBR) based congestion control algorithm are two major contributions out of Google's persistent attempts to make the Internet faster. QUIC is an user level reliable protocol running on top of UDP to enable evolution of transport mechanisms, while BBR is designed for better TCP congestion control. Although in two different directions, they also converge with BBR support in QUIC. Many researches have proved the performance of QUIC and BBR in different network scenarios, but the evaluation in satellite Internet with long propagation delays and relatively high error rates is still missing.In this paper, we have a preliminary evaluation of the performance of QUIC with BBR via GEO satellite Internet access on dedicated network emulation testbeds. The results and our analysis confirms that for satellite Internet QUIC with the new BBR congestion control has improvements compared with the classic CUBIC.**

***Index Terms*—Satellite Internet, QUIC, BBR, HTTP, transmission protocol, congestion control**

I. Introduction

Quick UDP Internet connection(QUIC) protocol [1] and bottleneck bandwidth and round-trip propagation time (BBR) [2] based congestion control algorithm are two major contributions out of Google's persistent attempts to make the Internet faster. Different from TCP, QUIC provides encrypted, multiplexed, and low-latency reliable transport which is originally dedicated to improve performance for HTTPS traffic. As an user-space transport protocol running on top of UDP, QUIC is designed from the ground up to enable rapid deployment and continued evolution of transport mechanisms. QUIC has been globally deployed at Google on thousands of front-end servers which handle billions of requests from web browsers and mobile apps across a wide range of services. On the client side, QUIC is deployed in Chrome browser and YouTube mobile video streaming app, and in the Google Search app on Android. It is reported QUIC accounts for over 30% of Google's total egress traffic in bytes and consequently an estimated 7% of global Internet traffic [1].

For clients relying on TCP, BBR is proposed to improve the performance of TCP at data sender with advanced congestion control. Different from other loss- or delay-based congestion control algorithm, with estimation of available bottleneck link bandwidth and minimum round-trip time (RTT), BBR tries to achieve high link utilization while avoiding to create queues in bottleneck buffers. BBR has also been widely deployed in Google's own production platforms like the B4 wide-area network and YouTube [2]. BBR is also opensourced and has been integrated into the Linux kernel (available since version 4.9). Currently TCP CUBIC is still the default congestion control in QUIC implementation, but BBR is also optional congestion control in QUIC.

QUIC and BBR are both reported by Google for the impressive improvement in performance and have drawn broad interest. Many independent researches have evaluated the performance of QUIC and BBR in different network scenarios, such as mobile Internet or terrestrial backbone networks [3]–[7]. The benefits and flaws in QUIC and BBR are better revealed and understood through these works, but the evaluation in satellite Internet is still missing in the literature.

Satellite Internet or Internet access via satellite is one of the most important components in current Internet for its wide coverage, high bandwidth and broadcasting capability. It is particularly suitable for sky, marine and undeveloped areas without communications infrastructures [8]. Satellite Internet also helps emergency response when communication systems and terrestrial infrastructures are damaged during earthquake, flood, typhoon or other disasters. It can be envisioned that more and more traffic will be delivered by satellite Internet in the near future.

The inherent long propagation delays and the relatively high bit error rates [9] of the satellite links are challenging for the networking protocols, especially TCP, which usually deteriorates the quality of experience in web access. HTTP protocol over TCP is particularly impaired over satellite links. For web browsing, the quality of experience is not just decided by the transport layer protocols, another solution is to change the HTTP proxy, but few work has been done yet to evaluate the performance of higher layer protocols over different variants of transport layer protocols [5].

In this paper, the performance of QUIC with BBR, via GEO satellite Internet access are evaluated on dedicated network emulation testbeds. The results of emulation and our analysis confirms that QUIC with BBR outperforms QUIC with BBR on satellite links.

II. Brief introduction for QUIC and BBR

In this section, a brief introduction of the mechanism of QUIC and BBR is given.

A. Quick UDP Internet connection (QUIC) protocol

Currently, QUIC is designed as a modern general-purpose transport optimized for HTTP/2 application, which achieves the ability similar to TCP+TLS+HTTP/2 over UDP while maintaining the feasibility of evolution. Efforts are taken to reduce latency in the underlying TLS/TCP mechanisms, which commonly run into limitations, such as handshake delay and head-of-line blocking delay. Many researches have proved that QUIC has advantages over HTTP/2 and HTTP/1.1 in terrestrial Internet. Our earlier work [5] also proves that when applying QUIC in satellite network, the page load time in web browsing is also reduced. The main features of QUIC are 0-RTT connection establishment and multiplexing, which also benefits satellite Internet.

(1) 0-RTT Connection

TCP establish a connection with a three-way handshake incur at least one RTT delay. If an encrypted connection is required, TLS adds another two RTT and it costs three RTT totally. To reduce the handshake delay, QUIC has its own encryption system and combines connection establishment and secure key agreement into only 1 RTT handshake. After a successful handshake, the two end nodes caches the client and the server information with a global connection ID (64bits). Afterwards, one node can send encrypted application data to the other node without extra initialization, which means 0-RTT. 0-RTT connection is beneficial for satellite links with long propagation delays.

(2) Multiplexing

Similar to the multiplexing in HTTP/2 [10], QUIC multiples several QUIC streams over a single UDP connection which can reduce redundant connections effectively, while in HTTP/2, multiplexing is conducted over a single TCP connection. With multiplexing, HTTP/2 and QUIC are able to overcome the problem of head-of-line blocking in TCP. Considering the long delay of satellite network, reducing connections can help reduce the response time.

B. Bottleneck bandwidth and round-trip propagation time (BBR) based congestion control

The classical loss-based congestion control algorithms, such as TCP Reno and TCP CUBIC, take packet loss as an indication of network congestion. With higher data rates and larger memory capacity, the relationship between packet loss and congestion is more tenuous [2]. When the bottleneck buffers are large, congestion control will keep buffers full and bring bufferbloat. When the buffers are small, packet loss in buffers will be misinterpreted as a signal of congestion, leading to low throughput.

The new congestion based BBR algorithm reacts to actual congestion, not packet loss or transient queue delay. It only cares about two factors, the bandwidth of the bottleneck and the minimum RTT of the connection. When it comes to bandwidth of the bottleneck (BtlBw), BBR probes the largest delivery rate during a time window. Delivery rate (deliveryRate) is calculated by the packets which are acknowledged to the time elapsed. RTT is usually estimated by the interval from sending a data packet until it is acknowledged. BBR also chooses the minimum round trip time it probes as an estimate of round trip propagation time (RTprop) during a time window. The delay caused by the network noise is represented by the parameter η. Elements with subscript t indicates that they are measured at time t.

$$deliveryRate = \Delta Packets_{Acked}/\Delta t \tag{1}$$

$$BtlBw = max(deliveryRate_t),\ \forall t \epsilon [T - W_B, T] \tag{2}$$

$$\begin{aligned} RTprop &= RTprop_t + min(\eta_t) \\ &= min(RTT_t), \forall t \epsilon [T - W_B, T] \end{aligned} \tag{3}$$

By probing the maximum bottleneck bandwidth and minimum RTT of a link, BBR tries to adapt its sending rate to achieve the best link utilization without creating queues at bottleneck, which should be the optimal operating point.

III. EXPERIMENT DESIGN AND PERFORMANCE EVALUATION

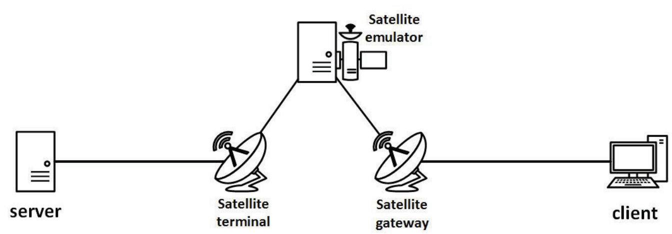
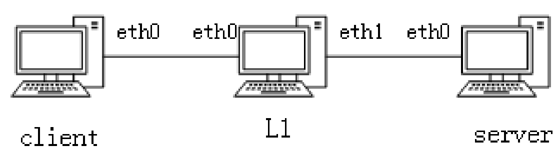
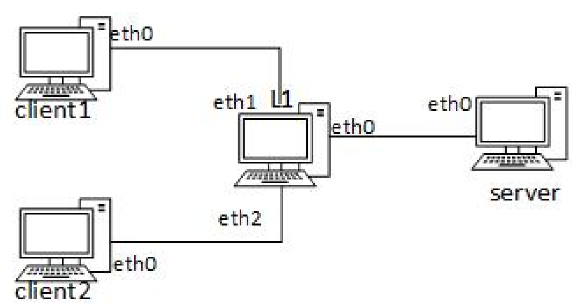


Fig. 1: The scenario of satellite Internet [5]



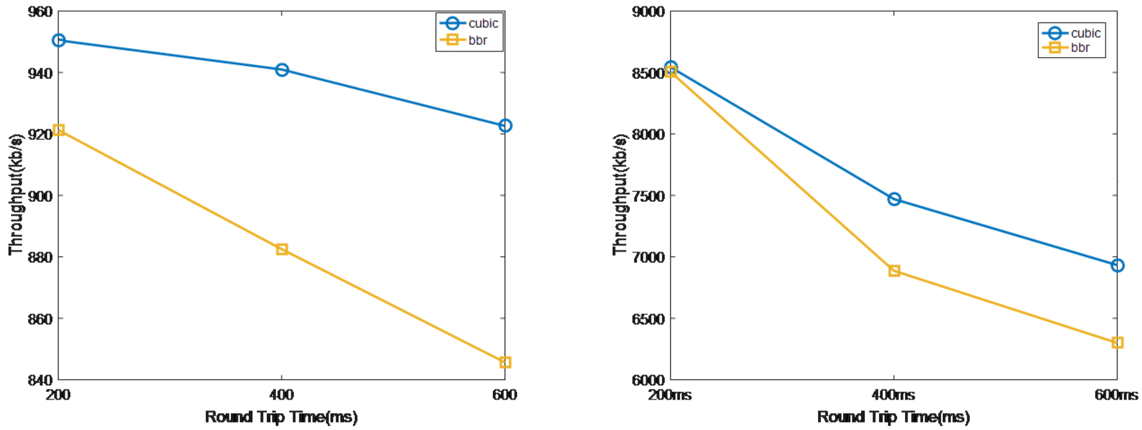
(a) First mode for emulation of single connection



(b) Second mode for emulation of bandwidth competition

Fig. 2: The laboratory testbed for emulation

The satellite Internet scenario studied in this paper is shown in Fig. 1. To evaluate the performance of QUIC with BBR, the



(a) 1M bandwidth with various RTTs

(b) 10M bandwidth with various RTTs

Fig. 3: Throughput performance of QUIC+BBR with different RTTs

laboratory testbed are in two modes, one for emulation of a single connection as shown in Fig. 2(a) and one for emulation of bandwidth competition with two clients sharing one satellite link which causes congestion at the bottleneck as shown in Fig. 2(b). Based on this testbed, we evaluate the performance and the fairness of QUIC with BBR for different delays and packet loss rates in satellite Internet.

A. *Test environment*

(1) The satellite links:

As shown in Fig. 2, a computer running Tc Netem was inserted between the server and the client nodes as a satellite link emulator. As shown in Table I, the range of RTT is from 200ms to 1000ms, as common parameters of different kinds of satellite links and random packet loss is also introduced to evaluate the impact of channel error rate.

(2) The server and the client:

The server and the client are two 64-bit machines running Ubuntu 14.04 O.S with Linux kernel 3.19.0. We use a TCP initial window size of ten packets as the default setting, which is suggested by HTTP/2's best practice. QUIC-server and Chromium browser are run separately in the server and the client. The QUIC version is 39, which is the latest version available at the time of writing. The congestion control algorithm in QUIC is configured and recompiled in the chrome browser. Apache server2.4.18 with TLS/1.2 are installed in the server. Module for HTTP/2 is employed to support HTTP/2 over SSL/TLS, while QUIC utilizes its embedded encryption. In the client, Chromium browser is set in three different configurations to fetch web pages from server.

(3) The web pages:

To simplify the experiments, a simple web page index.html is linked with jpeg images used as an example website. Chromium browser is employed to fetch three kinds of web pages respectively with QUIC with default CUBIC or QUIC with BBR. Web pages only contain tow kinds of images, the small one is 3.7KB and the big one is 86.1KB. As shown in Table I, the small page contains 18 small images and 3 big images, the medium page has 40 small ones and 7 big ones, while the big page has 200 small pages and 17 big pages.

TABLE I: Test environment parameters

Parameters	Values
RTT	200ms, 400ms, 1000ms
packet loss rate	10^{-5},10^{-4},10^{-3}, 10^{-2}, 2×10^{-2}, 5×10^{-2}, 10^{-1},2×10^{-1}
page size	344KB, 784KB, 2.3MB
Num of obj	21, 47, 217

(4)The tools for performance evaluation:

Wireshark is used to collect all the traffics on both the server and the client.

B. *Throughput performance of QUIC with BBR*

The throughput performance of QUIC with BBR is evaluated on the testbed in mode one. The impact of long delay on the throughput performance is first tested. The RTT is set to $200ms, 400ms$ and $600ms$ for evalutaion and no packet loss is set during the test. To minimize the impact of noise in network, the results are averaged over five experiments. The results of the emulation is shown in Fig. 3. It is obvious that when RTT increases, the performance of CUBIC and BBR both deteriorates. QUIC with CUBIC achieves better throughput than QUIC with BBR in all the test. The reason is when BBR probes the RTT every 10 seconds there will be only 4 max segment size packets on-the-fly, which means the longer the propagation delay is, the lower the transmission rate will be, so as the throughput. The data rate of QUIC with BBR is also shown in Fig. 4, in which we can observe that the data rate decreases during the probing stage.

We also evaluate the impact of packet loss introduced by channel error. The results are shown in Fig. 5. The RTT is fixed to $600ms$ for emulation of GEO Internet access. With the effect of packet loss, the performance of the loss based CUBIC congestion control deteriorates very seriously. As BBR

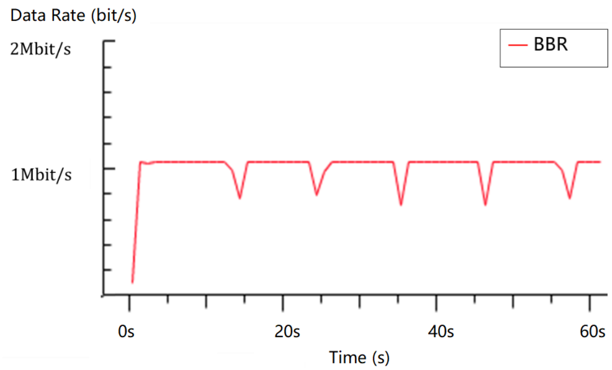


Fig. 4: Data rate performance of QUIC+BBR

is congestion based, the non-congestion packet loss has very limited effect on its congestion window. When the packet loss rate is larger than 1%, the throughput performance of QUIC with BBR is much better than QUIC with CUBIC, which is beneficial for GEO satellite Internet.

C. Goodput Performance of QUIC with BBR

Compared with CUBIC, BBR doesn't grow congestion window until it gets a signal of packet loss. BBR detects the largest bandwidth and smallest round trip time and decides the packet pacing rate. In the environment of satellite networks, non-congestion loss rate is high, BBR can keep pace in packet sending without paying attention to these interference signals. As we can see from Fig.6, when the loss rate is 0%, the goodput rates(the total size of web page/page load time) of both algorithm are almost the same. When the loss rate comes to 1%, the rate of CUBIC will decay rapidly, while BBR can still keep a considerable goodput. BBR outperforms CUBIC when the loss rate is between 1% to 10% which can be normal in satellite networks. The results on goodput conforms with our observation in evaluation of throughput.

D. Fairness of QUIC with BBR

After evaluating the performance in transmission, we also evaluate the bandwidth fairness of QUIC with BBR in satellite Internet. When several users compete with each other on the bottle neck, the fairness of the congestion control algorithm is important. With the second mode of the testbed, we emulate two clients compete for the 1Mbps satellite link for Internet acess. The two clients are equipped with different congestion control algorithms, one with CUBIC and one with BBR. The results with $100ms$ and $300ms$ RTTs are shown in Fig. 7. Basically, BBR is CUBIC friendly in transmission stage, but every 10 seconds when BBR transits into the probing stage, it gives out bandwidth and CUBIC will take over. As BBR has better goodput on average, the unfairness in the probing stage seems kind of "fair" for CUBIC.

IV. Conclusion

In this paper, the performance of QUIC with BBR, via satellite Internet access are preliminarily evaluated on dedicated network emulation testbeds. Our results and analysis demonstrate the effectiveness of QUIC with new BBR congestion control algorithm over satellite links with large RTTs and high packet loss rate. QUIC with BBR does help to shorten the transmission delay and keep transmission speed when the packet loss rate grows. In the future, further researches would be taken to improve the algorithm to make BBR more effective in satellite networks.

References

[1] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017, pp. 183–196.

[2] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: Congestion-based congestion control," *Queue*, vol. 14, no. 5, p. 50, 2016.

[3] A. M. Kakhki, S. Jero, D. Choffnes, C. Nita-Rotaru, and A. Mislove, "Taking a long look at quic: an approach for rigorous evaluation of rapidly evolving transport protocols," in *Proceedings of the 2017 Internet Measurement Conference*. ACM, 2017, pp. 290–303.

[4] S. Cook, B. Mathieu, P. Truong, and I. Hamchaoui, "Quic: Better for what and for whom?" in *IEEE International Conference on Communications (ICC2017)*, 2017.

[5] H. Zhang, T. Wang, Y. Tu, K. Zhao, and W. Li, "How quick is quic in satellite networks," in *International Conference in Communications, Signal Processing, and Systems*. Springer, 2017, pp. 387–394.

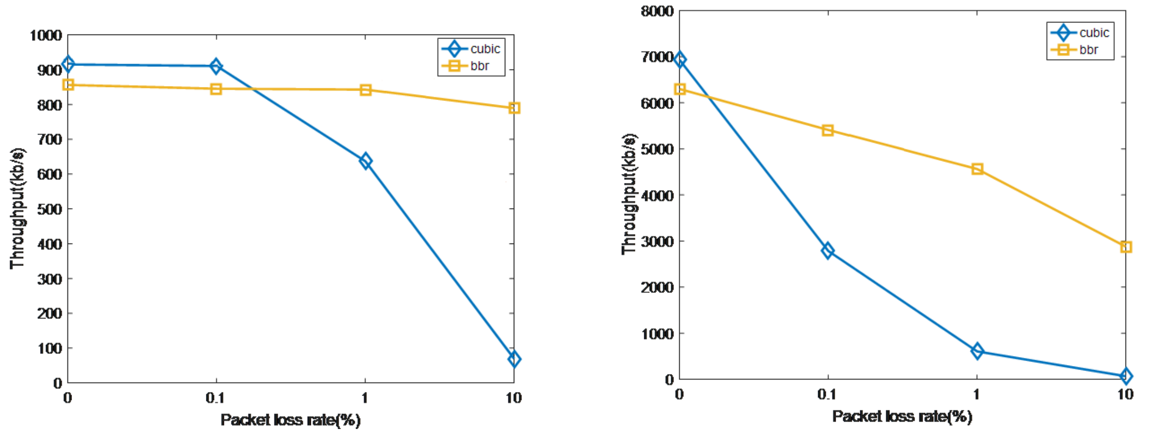
[6] M. Hock, R. Bless, and M. Zitterbart, "Experimental evaluation of bbr congestion control," in *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. IEEE, 2017, pp. 1–10.

[7] Z. Zhong, I. Hamchaoui, R. Khatoun, and A. Serhrouchni, "Performance evaluation of cqic and tcp bbr in mobile network," in *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE, 2018.

[8] J. Farserotu and R. Prasad, "A survey of future broadband multimedia satellite systems, issues and trends," *IEEE Communications Magazine*, vol. 38, no. 6, pp. 128–133, 2000.

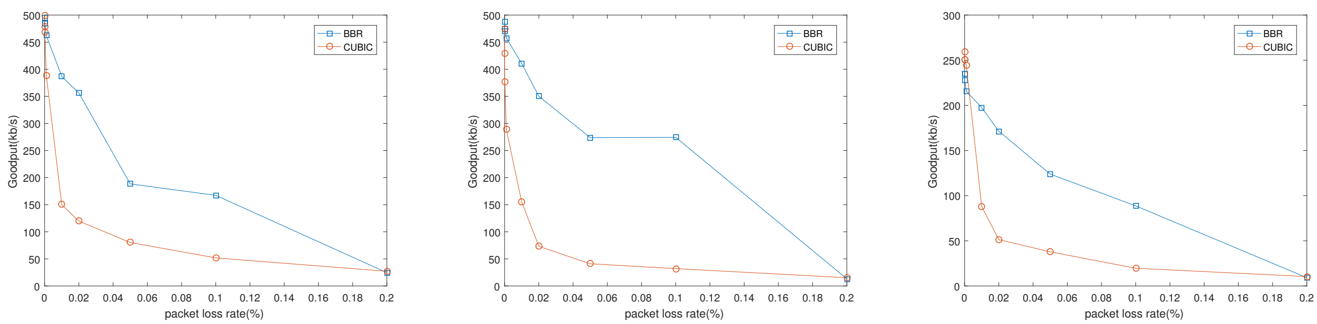
[9] C. Roseti, M. Luglio, and F. Zampognaro, "Analysis and performance evaluation of a burst-based tcp for satellite dvb rcs links," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 3, pp. 911–921, 2010.

[10] X. S. Wang, A. Balasubramanian, A. Krishnamurthy, and D. Wetherall, "How speedy is spdy?" in *Usenix Conference on Networked Systems Design and Implementation*, 2014.



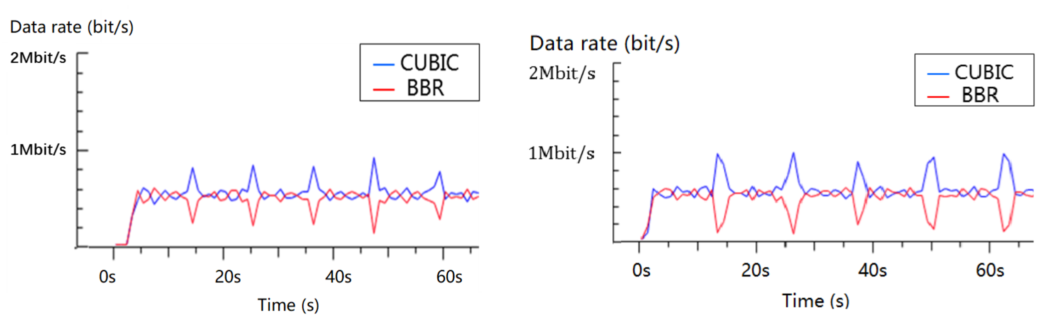
(a) 1M bandwidth 600ms RTT and various packet loss rates (b) 10M bandwidth 600ms RTT and various packet loss rates

Fig. 5: Throughput performance of QUIC+BBR with different packet loss rates



(a) RTT 200ms (b) RTT 400ms (c) RTT 1000ms

Fig. 6: Goodput performance of QUIC+BBR with different loss rate



(a) RTT 100ms (b) RTT 300ms

Fig. 7: Fairness performance of QUIC+BBR