Quick-Start for TCP and IP

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

This draft outlines an optional Quick-Start mechanism for transport protocols to determine an optional allowed initial congestion window or initial sending rate at the start of a data transfer. This mechanism is designed to be used by a range of transport protocols; however, in this document we only describe its use with TCP and IPv4. By using Quick-Start, a TCP host, say, host A, would indicate its desired initial sending rate in packets per second in a Quick Start Request option in the IP header of the initial TCP SYN or SYN/ACK packet. Each router in turn could either approve the specified initial rate, reduce the specified initial rate, or indicate that nothing above the default initial rate for that protocol would be allowed. The Quick-Start mechanism also can determine if there are routers along the path that do not understand the Quick Start Request
option, or have not agreed to the initial rate described in the option. If all of the routers along the path have agreed to the initial rate in the Quick-Start Request, then TCP host B communicates this to TCP host A in a transport-level Quick-Start Response in the answering SYN/ACK or ACK packet. Quick-Start is designed to allow TCP connections to use high initial windows in circumstances when there is significant unused bandwidth along the path, and all of the routers along the path support the Quick-Start Request. We note that this is currently not a mature proposal, but an outline of one possible path of development in the IP protocol, and a request for feedback from the community.

1. Introduction

The life of a TCP connection begins with a question, that is, "With what rate I can transfer the data?" The question is not answered explicitly for TCP, but each TCP connection figures out the answer for itself. This is done by each connection starting with an initial congestion window (called cwnd) of from one to four MSS-sized segments, for an MSS the Maximum Segment Size in bytes. Currently, the TCP protocol allows an initial window of up to two segments [RFC2581], with an experimental proposal for an initial window of three or four segments [RFC2414]. The TCP connection then probes the network for available bandwidth using the slow-start procedure, essentially doubling its congestion window each round-trip time.

The probing mechanism of slow-start is time-consuming and causes an overhead in terms of queueing delay. It may take a number of round-trip times in slow-start before the TCP connection begins to fully use the available bandwidth of the network; it takes \log N \text{ round-trip times} to build up to a congestion window of \text{N segments}. This time in slow-start is not a problem for large file transfers, where the slow-start stage is only a fraction of the total transfer time. However, in the case of moderate-sized web transfers the connection might carry out its entire transfer in the slow-start phase. In an underutilized, high-bandwidth network, such a transfer can end up taking \log N \text{ round-trip times} to transfer the data, where one or two round-trip times might have sufficed.

A fair amount of work has already been done to address the issue of choosing the initial window for TCP, with RFC 2414 proposing an initial window of up to four segments [RFC2414]. Our underlying premise is that explicit feedback from all of the routers along the path would be required for best-effort connections to use initial windows higher than four segments.

The Congestion Manager proposes sharing congestion information among multiple TCP connections with the same endpoints [RFC2140]. With the
Congestion Manager, a new TCP connection could start with a high initial window if it was sharing the path with a pre-existing TCP connection, with a high congestion window, to the same destination. It is also possible that a newly-starting TCP connection could make use of congestion information from a recently-terminated TCP connection to the same destination. However, neither of these approaches are of any use for a connection to a new destination when there is no existing or recent connection to that destination.

Active Queue Management and Explicit Congestion Notification (ECN) are both based on the router detecting congestion before buffer overflow [RFC3168]. In a similar but somewhat simpler fashion, Quick-Start is based on the ability of the router to determine whether or not the output link is significantly underutilized.

2. Assumptions, General Principles, and Open Questions

This section describes the assumptions and general principles behind the design of the Quick-Start mechanism.

Assumptions:

* A router can determine reasonably easily if the output link has been significantly underutilized over a period of time.

* The data transfer in the two directions of a connection traverses different queues, and possibly even different routers. Thus, any mechanism for determining the allowed initial window or initial sending rate would have to be used independently for each direction.

* Any new mechanism must be incrementally deployed, and may not be supported by all of the routers and/or end-hosts. Thus, any new mechanism must be able to accommodate non-supporting routers or end-hosts without disturbing the current Internet semantics.

* After the initial SYN exchange, a TCP data sender would be able to translate an initial sending rate in packets per second into an initial congestion window of MSS-sized segments.

General Principles:

* In order for best-effort connections to use initial windows higher than four segments, explicit feedback from all of the routers along the path would be required.

* A router should only allow an initial sending rate higher than the transport protocol’s default initial rate if the router is significantly underutilized. Any other approach will result in
either per-flow state at the router, or the possibility of either a (possibly transient) queue or of underutilization of the output link.

* No per-flow state is kept at the router for this mechanism. In Quick-Start, the only state kept at the router is the aggregate initial sending rate authorized over the most recent interval of time (e.g., quarter of a second).

There are also a number of open questions regarding the Quick-Start mechanism outlined in this draft.

Open Questions:

* Would be benefits of the Quick-Start mechanism be worth the added complexity?

One drawback of the Quick-Start mechanism is that the SYN and SYN/ACK packets containing the Quick-Start option would presumably be processed in the slow path in routers. This would reduce the capacity of routers to handle Quick-Start requests, and delay the initial SYN exchange for the connection.

Another drawback is that Quick-Start would require functionality in the router to estimate the current link utilization, and to keep track of the aggregate Quick-Start rate authorized over the last interval of time. Adding new functionality to routers should not be done lightly, and any mechanisms that would require new functionality in routers would have to be carefully considered.

* Apart from the merits and shortcomings of the Quick-Start mechanism, is there likely to be a compelling need to add explicit congestion-related feedback from routers over and above the one-bit feedback from ECN?

* If the answer to the question above is yes, should we be considering mechanisms that, while more complex, are also sufficiently more powerful than Quick-Start?

There are a number of such mechanisms that have been proposed in the literature for more fine-grained congestion-related feedback from routers. Quick-Start is extremely coarse-grained, in that in its current form it only applies to the initial window of the connection. Quick-Start also focuses in on the specific issue of allowing very high initial sending rates for connections over underutilized, high-bandwidth paths. Quick-Start might first be deployed, for example, in an overprovisioned high-bandwidth Intranet, to allow much quicker transfers of best-effort traffic.

Quick-Start would require end-points and routers to work together, with end-points requesting a higher initial sending rate in the Quick-Start Request (QSR) option in IP, and routers along the path approving, modifying, or discarding or ignoring (and therefore disallowing) the Quick-Start Request. The receiver would check the validity of the received Quick-Start Request, and if valid, would use transport-level mechanisms to inform the sender of the status of the Quick-Start Request. We note that the Quick-Start Request implicitly assumes a unicast transport protocol; we have not considered the use of the Quick-Start Request for multicast traffic.

3.1. The Quick-Start Request Option for IPv4

The Quick-Start Request for IPv4 would be defined as follows:

```
+----------+----------+----------+----------+----------+
| Option   | Length=5 |  QS TTL  | Initial  | QS Nonce |
|          |          |          | Rate     |          |
+----------+----------+----------+----------+----------+
```

Figure 1. The Quick-Start Request Option for IPv4.

The first byte contains the option field, which includes the one-bit copy flag, the 2-bit class field, and the 5-bit option number.

The second byte contains the length field, which indicates an option length of five bytes.

The third byte contains the Quick-Start TTL (QS TTL) field. If the sender decides to use Quick-Start, then the sender sets the QS TTL field to the same value as the TTL field in the IPv4 header. Routers that approve the Quick-Start Request decrement the QS TTL. The QS TTL is used by the receiver, and by downstream routers, to detect if there have been upstream routers that do not understand, or do not approve of, the Quick-Start option.

If a router receives a Quick-Start Request with a QS TTL field different from the TTL field in the IP header, then this Quick-Start Request is not valid, and should be ignored by the router. If it desires, the router could delete this option when it forwards the packet, thus saving unnecessary delay for the packet at downstream routers. If the Quick-Start Request is not approved, then the sender uses the default initial rate or initial window for that transport protocol.
The fourth byte is the requested Initial Rate (IR) field. The sender initializes the Initial Rate to the desired initial sending rate, formatted in packets per 0.1 sec. Routers can approve the Quick-Start Request for a lower Initial Rate by decreasing the Initial Rate in the Quick-Start Request.

Note that the one-byte Initial Rate field limits the Initial Rate to at most 2560 packets/sec. For 1500-byte packets, this corresponds to an initial rate of 30 Mbps. A larger option field, or a different encoding for the one-byte requested Initial Rate option, would be needed to allow a higher range for the requested initial rate.

The fifth byte contains the QS Nonce. The sender initializes this value to a non-zero, ‘random’ eight-bit number. A router can deny the Quick-Start request by failing to decrement the QS TTL field, by zeroing the QS Nonce field, or by deleting the Quick-Start Request from the packet header. The QS Nonce is included to provide some protection against broken downstream routers, or against misbehaving TCP receivers who might be inclined to lie about the Initial Rate.

A router that modifies or deletes the Quick-Start Request in the IPv4 header also has to update the IPv4 Header checksum.

3.2. Processing the Quick-Start Request at a Participating Router

Each participating router can either terminate or forward the Quick-Start Request. The router terminates the Quick-Start Request if:

* The QS TTL doesn’t match the TTL field in the IP header.
* The router is not underutilized, and therefore has decided not to grant the Quick-Start Request.

The preferable method for a router to terminate the Quick-Start Request is to delete the Quick-Start Request from the IP header. A less preferable but possibly more efficient method is to simply forward the packet with the Quick-Start Request unchanged, or with the QS Nonce zeroed.

If the participating router has decided to approve the Quick-Start Request, it does the following:

* It decrements the QS TTL by one.
* If the router is only willing to approve an Initial Rate less than that in the Quick-Start Request, then the router puts the new Initial Rate in that field of the Quick-Start Request.
A non-participating router forwards the Quick-Start Request unchanged, without decrementing the QS TTL. Of course, the non-participating router still decrements the TTL field in the IP header, as is required for all routers. As a result, any subsequent routers and the receiving end-host can detect that the Quick-Start Request is not valid.

3.3. Deciding the Permitted Initial Rate at a Router

In this section we briefly outline how a router might decide whether or not to approve a Quick-Start Request. As an example, the router could ask the following questions:

* Has the router’s output link been underutilized for some time (e.g., several seconds).

* Would the output link remain underutilized if the arrival rate was to increase by the aggregate initial rate that the router has approved over the last fraction of a second?

Answering this question requires that the router have some knowledge of the available bandwidth on the output link for that output queue. It also requires that the router keep a counter indicating the total aggregate Initial Rates that have been approved over the recent interval of time. Thus, if an underutilized router experiences a SYN flood, then the router would begin to deny Initial Rate requests, even if the router remains underutilized.

* If the answer to both of the previous questions is Yes, then the router could approve the Quick-Start Request. The router could allow an Initial Rate that was a small fraction of the available unused bandwidth of the output link.

4. The Quick-Start Mechanisms in TCP

This section describes how the Quick-Start mechanism would be used in TCP. If a TCP sender, say host A, would like to request Quick-Start, the TCP sender puts the requested initial sending rate in packets per second in the Quick-Start Request option in the IP header of the SYN or SYN/ACK packet, as described above. The TCP host B returns the Quick-Start Response option in the TCP header in the responding SYN/ACK packet or ACK packet, respectively, informing host A of the results of their request.

If the returning packet does not contain a Quick-Start Response, then host A has to assume that its Quick-Start request failed. In this case, host A uses TCP’s default initial window.
If the returning packet does contain a Quick-Start Response, then host A uses the information in the response, along with its measurement of the round-trip time, to determine the initial congestion window. In order to use Quick-Start, the TCP host would also be required to use rate-based pacing to pace out the packets in the initial window at the rate indicated in the Quick-Start Response.

In a similar manner, if TCP host B requests Quick-Start in the IP header of the TCP SYN/ACK packet, then TCP host A returns the Quick-Start Response in the TCP header of the answering ACK packet. The two TCP end-hosts can independently decide whether to request Quick-Start.

4.1. The Quick-Start Response Option in the TCP header

The Quick-Start Response option is as follows:

```
+----------+----------+----------+----------+
| Kind     | Length=4  | Initial  | QS Nonce |
|          |           | Rate     |          |
+----------+----------+----------+----------+
```

Figure 2. The Quick-Start Response option in the TCP header.

The first byte of the Quick-Start Response option contains the option kind, identifying the TCP option.

The second byte of the Quick-Start Response option contains the length field, specifying the option length in bytes. The length field is set to four bytes.

The third byte of the Quick-Start Response option contains the allowed Initial Rate, formatted as in the Quick-Start Request option.

The fourth byte of the TCP option contains the QS Nonce. The TCP end-node returns the QS Nonce exactly as it was received in the Quick-Start Request option.

4.2. Sending the Quick-Start Response

An end host, say host B, that receives a TCP SYN or SYN/ACK packet containing a Quick-Start Request first has to determine if the Quick-Start Request is valid. The request is valid if the QS TTL field in the Quick-Start Request matches the TTL field in the IP header. A valid Quick-Start Request is passed to the receiving TCP layer.

If the TCP host is willing to permit the Quick-Start request, then a
Quick-Start Response option is included in the TCP header of the answering acknowledgement packet. The Initial Rate in the Quick-Start Response option is set to the received value of the Initial Rate in the Quick-Start Request option, or to a lower value if the TCP receiver is only willing to allow a lower Initial Rate. Similarly, the QS Nonce value in the Quick-Start Response is set to the value of the QS Nonce received in the Quick-Start Request.

If the Quick-Start Response is being sent on the SYN/ACK, in response to a Quick-Start Request on the SYN, then the Quick-Start Response will be resent if the SYN/ACK has to be retransmitted. If the Quick-Start Response is being sent on the ACK, in response to the Quick-Start Request on the SYN/ACK, then the Quick-Start Response has to be resent on data packets until that TCP host receives an acknowledgement for a data packet from the other end.

4.3. Receiving and Using the Quick-Start Response

A TCP host, say TCP host A, that sent a Quick-Start Request in a SYN or SYN/ACK, and receives an answering Quick-Start Response in the acknowledgement, first checks that the Quick-Start Response is valid. The Quick-Start Response is valid if it contains the same value for the QS Nonce as was sent in the Request, and an equal or lesser value for the Initial Rate. If this check is not successful, then the Quick-Start request failed, and the TCP host uses the default TCP initial window.

If the checks of the QS Nonce and the Initial Rate are successful, then the TCP host sets its initial congestion window to R*T*MSS bytes, for R the Initial Rate in packets per second and T the measured round-trip time in seconds. The TCP host sets a flag that it is in Quick-Start mode, and while in Quick-Start mode the TCP sender uses rate-based pacing, pacing out packets at the specified Initial Rate.

Because the initial SYN packet with the Quick-Start Request was not able to use the fast path in routers, the initial round-trip time measurement might be unnecessarily large. The Quick-Start mode ends when the TCP host first receives an ACK for one of the data packets. If the initial congestion window has not been fully used, then the initial congestion window is decreased to the amount that has actually been used so far. This should address the problem of an overly-large congestion window from an overly-large initial measurement of the round-trip time.
4.4. An Example Quick-Start Scenario with TCP

The following is an example scenario in the case when both hosts request Quick-Start:

* The TCP SYN packet from Host A contains a Quick-Start Request in the IP header.

* Routers along the forward path modify the Quick-Start Request as appropriate.

* Host B receives the Quick-Start Request in the SYN packet, and verifies that the QS TTL matches the TTL in the IP header. If it does, and Host B approves the Quick-Start Request, then Host B sends a Quick-Start Response in the TCP header of the SYN/ACK packet. Host B also sends a Quick-Start Request in the IP header of the SYN/ACK packet.

* Routers along the reverse path modify the Quick-Start Request as appropriate.

* Host A receives the Quick-Start Response in the SYN/ACK packet, and checks the QS Nonce and Initial Rate for validity. If they are valid, then Host A sets its initial congestion window appropriately, and sets up rate-based pacing to be used with the initial window. If the Quick-Start Response is not valid, then Host A uses TCP’s default initial window.

Host A also checks the QS TTL in the Quick-Start Request in the incoming SYN/ACK packet, and if it is valid, sends a Quick-Start Response in the TCP header of the ACK packet.

* Host A repeats sending the Quick-Start Response in data packets at least once per round-trip time until it receives an acknowledgement from Host B for one of those data packets.

* Host B receives the Quick-Start Response in an ACK packet, and checks the QS Nonce and Initial Rate for validity. If the Quick-Start Response is valid, then Host B sets its initial congestion window appropriately, and sets up rate-based pacing to be used with the initial window. If the Quick-Start Response is not valid, then Host B uses TCP’s default initial window.

Note that the Quick-Start Request mechanism is self-terminating, and it terminates at the first participating router after a non-participating router has been encountered on the path. This minimizes unnecessary overhead incurred by routers because of option processing for the Quick-Start Request.
5. Evaluation of Quick-Start

The main benefit of Quick-Start is to the transport connection itself. For a small TCP transfer of one to five packets, Quick-Start is probably of very little benefit; at best, it might shorten the connection lifetime from three to two round-trip times (including the round-trip time for connection establishment). Similarly, for a very large transfer, where the slow-start phase would have been only a small fraction of the connection lifetime, Quick-Start would be of limited benefit. Quick-Start would not significantly shorten the connection lifetime, but it might eliminate or at least shorten the start-up phase. However, for moderate-sized connections of N packets in well-provisioned environments that allow Quick-Start requests of M packets per second, the use of Quick-Start could shorten the connection lifetime from log N round-trip times to N/M + 1 round-trip times. For large values of N and M, this would be a dramatic shortening of the connection lifetime.

The main cost of Quick-Start concerns the costs of added complexity at the routers. The added complexity at the end-points is moderate, and might easily be outweighed by the benefit of Quick-Start to the end hosts. The added complexity at the routers is also somewhat moderate, in that it involves estimating the unused bandwidth on the output link over the last several seconds, and keeping a counter of the aggregate Quick-Start rate approved over the last fraction of a second. However, this added complexity at the routers adds to the development cycle, and could prevent the addition of other competing functionality to routers. Thus, careful thought would have to be given to the addition of Quick-Start to IP.

Another drawback of Quick-Start is that the Quick-Start Request message presumably would not take the fast path in routers. This would mean extra delay for the end hosts, and extra processing burden for the routers. This extra burden is mitigated somewhat by the following factors: only SYN and SYN-ACK packets would carry the Quick-Start Request option; very small flows of, say, one to five packets would receive little benefit from Quick-Start, and presumably would not use the Quick-Start Request; flows from end hosts with low-bandwidth access links would receive little benefit from Quick-Start, and hopefully could be configured not to use the Quick-Start Request. Nevertheless, it is still conceivable, in the worst case, that up to 10% of the packets were SYN or SYN/ACK packets using a Quick-Start Request, and this could slow down the processing of SYN or SYN/ACK packets in routers considerably.

One limitation of Quick-Start is that it presumes that the data packets of a connection will follow the same path as the SYN or SYN/ACK packet. If this is not the case, then the connection could
be sending the initial window, at the permitted initial rate, along a path that was already congested, or that became congested as a result of this connection. This is, however, similar to what would happen if the connection’s path was changed in the middle of the connection, when the connection had already established the allowed initial rate.

A problem of any mechanism for feedback from routers at the IP level is that there can be queues and bottlenecks in the end-to-end path that are not in IP-level routers. As an example, these include queues in layer-two Ethernet or ATM networks. The hope would be that an IP-level router adjacent to such a non-IP queue or bottleneck would be configured to reject Quick-Start requests if that was appropriate.

The discussion in this paper has largely been of the Quick-Start mechanism with default, best-effort traffic. However, Quick-Start could also be used by traffic using some form of differentiated services, and routers could take the traffic class into account when deciding whether or not to grant the Quick-Start request. However, we would note that routers should be discouraged from granting Quick-Start requests for higher-priority traffic when this is likely to result in significant packet loss for lower-priority traffic.

The Quick-Start proposal, taken together with the recent proposal for HighSpeed TCP [F02], would go a significant way towards extending the range of performance for best-effort traffic in the Internet. However, there are many things that the Quick-Start proposal would not accomplish. For example, Quick-Start would not allow flows to ramp-up quickly in the middle of the connection. Quick-Start would not help in making more precise use of the available bandwidth, that is, of achieving the goal of very high throughput with very low packet loss rates. Quick-Start would not give routers any additional power in allocating bandwidth in the interests of greater fairness, or in having more control over slow decrease rates of active connections. One of the open questions is whether the limited capabilities of Quick-Start are sufficient to warrant standardization and deployment, or whether more work is needed to explore the space of potential mechanisms.

6. Conclusions

We are presenting the Quick-Start mechanism not as a mechanism that we believe is urgently needed in the current Internet, but rather as a straw proposal for a simple, understandable, and incrementally-deployable mechanism that would be sufficient to allow connections to start up with large initial rates, or large initial congestion windows, in overprovisioned, high-bandwidth environments. We are not making any predictions about what we think typical environments are
likely to be like in the future Internet. However, we expect there will be an increasing number of overprovisioned, high-bandwidth environments where the Quick-Start mechanism, or another mechanism of similar power, could be of significant benefit to a wide range of traffic. We are presenting the Quick-Start mechanism as a request for feedback from the Internet community in considering these issues.

7. Acknowledgements

The authors wish to thank Mark Handley for discussions of these issues. The authors also thank the End-to-End Research Group and a long list of other people for feedback, both positive and negative; we will add this list later. This draft builds upon the concepts described in [RFC2414], [AHO98], [RFC2415], and [RFC3168].

This is a modification of a draft originally by Amit Jain for Initial Window Discovery.

8. Normative References


9. Informative References


10. Security Considerations

The only security consideration would be if the use of Quick-Start resulted in the sender using an Initial Rate that was inappropriately large, resulting in congestion along the path. Such congestion would result in an unacceptable level of packet drops along the path. Such congestion could also be part of a Denial of Service attack.

A misbehaving TCP sender could use a non-conformant initial congestion window even without the use of Quick-Start, so we restrict our attention to problems with Quick-Start with conformant TCP senders. (We also note that if the TCP sender is a busy web server, then the TCP sender has some incentive to be conformant in this regard.)

If a router that understands the Quick-Start Request deletes the Request, or zeroes the QS Nonce in the request, then the chances of a downstream router or misbehaving receiver guessing the value of the QS Nonce is at most 1/256. Thus, deleting the Quick-Start Request makes it unlikely that the receiver would be able to send a valid Quick-Start Response back to the sender.

If there are routers along the path that do not understand or approve of the Quick-Start Request, and that forward the Quick-Start Request unchanged, it would be easy for a downstream router or the receiver to cheat and modify the QS TTL field so that the request was considered valid.

It would also be easy for routers or for the receiver to increase the Initial Rate, making the Initial Rate be higher than that approved by an upstream router. However, this higher Initial Rate will only be considered valid by the TCP sender if it is no higher than the Initial Rate originally requested by the sender.
11. IANA Considerations

The only IANA Considerations would be the addition of an IP option to the list of IP options, and the addition of a TCP option to the list of TCP options.

AUTHORS’ ADDRESSES

Amit Jain
Netscaler
Email: ajain@netscaler.com

Sally Floyd
Phone: +1 (510) 666-2989
ICIR (ICSI Center for Internet Research)
Email: floyd@icir.org
URL: http://www.icir.org/floyd/

This draft was created in June 2002.