QUIC Q. An
Internet-Draft Alibaba Inc.
Intended status: Standards Track Z. Li
Expires: January 7, 2021 ICT-CAS
Y. Liu
Alibaba Inc.
July 6, 2020

Enabling application policy-awareness in Multipath QUIC
draft-an-multipath-quic-application-policy-00

Abstract

This document describes an application policy-awareness method for Multipath QUIC, to enable taking applications' demands into account when managing paths or scheduling packets in Multipath QUIC.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
2. Notational Conventions 3
3. What is Application Policy-awareness in Multipath QUIC . . . 3
4. Per-connection Policy 4
5. Per-stream Intent 6
6. IANA Considerations 7
7. Informative References 8
Authors' Addresses 9

1. Introduction

The availability of multiple network interfaces in devices enables the use of multiple paths between two end points for robust and (possibly) fast data delivery. Multi-path TCP (MPTCP) was designed to leverage off such an availability. It has been adopted by operating systems and ISPs. A major design principle of MPTCP is that it must be usable by applications through existing SOCKET APIs. That said, it shields the underlying multiple paths for the applications, which may be not even aware the use of multiple path transmission. While this implements the backward compatibility with TCP, it also leaves the applications out of the control loop of the transport. Nevertheless, applications may have completely different QoE requirements---the interactive applications are delay sensitive, while the video streaming are more throughput sensitive. There is thus a trend of cross-layer design that tries to take applications' demands into account when managing paths or scheduling packets in MPTCP [MP-DASH][WN][AD-MPTCP][SMAPP].

We thus advocate the idea of application policy awareness in multi-path transport, which is to separate the 'control plane' from the `data plane' like in software defined networking. The 'control plane' takes applications' high-level demands (a.k.a intent) as input to generate the corresponding policies, which later are deployed on the 'data plane'. The 'data plane' maps users policies to the 'actions', which control the path management, packet scheduling and other functionalities that the transport implements. For instance, the 'actions' for path management may include "backup", "full mesh" and "ndiffports", while the 'actions' for packet scheduling may designate whether ECF [ECF] or DEMS [BSC] is used for path selection, and may also impose a traffic split ratio among paths. Each connection then can have a customized multipath transport.

Since MPTCP is implemented in OS kernel, it is not easy for application providers to incorporate the above idea into MPTCP to improve the applications' performance. QUIC, on the other hand, implements a UDP-based transport, which enables the transport to be bundled into applications and quickly deployed in end users' devices. MPQUIC extends QUIC to multipath scenarios. Nevertheless, the basic MPQUIC [MPQUIC-Design] acts much like MPTCP, except it uses QUIC (as opposed to TCP). That said, the current MPQUIC implementation does not support application policy-aware transport.

A straight solution for MPQUIC to enable the interaction between applications and transport is to let applications to access/change every single logic of the packet scheduling and path management, or export callback functions as in [SMAPP]. This however will result in a complex mix between applications and transport because of limited abstraction. And perhaps more importantly, it requires application developers to understand the details of the underlying transport.

This document describes an application policy-awareness method for Multipath QUIC, to enable taking applications' demands into account when managing paths or scheduling packets in Multipath QUIC. The method remains compliant with the current Multipath Extensions for QUIC [I-D.deconinck-quic-multipath] design.

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. What is Application Policy-awareness in Multipath QUIC

We advocate a more practical alternative (See Figure 1), where 1). the `control plane' provides primitives to applications to express their policies (intent); 2). the `data plane' abstracts execution models for application policies and generates the corresponding composed `actions'.


```
               per-stream intent                   per-conn policy
    Control e.g. stream prioirity      e.g. path preference, path mode,
    Plane                                        deadline-aware etc
               |                                        |
    -----------|----------------------------------------|-------------
               V                                        V
    Data  +--------+ -                      +------------------------+
    Plane |        |  |                     |                        |
          +--------+  |                     |                +-----+ |
                      |                     |             +->| ... | |
          +--------+  |                     |             |  +-----+ |
          |        |  |                     |             |  +-----+ |
          +--------+  |                     |             +->| ... | |
                      |                     |             |  +-----+ |
          +--------+  | +----------+ chunks |+----------+ |          |
          |        |  ->|  stream  |------->||  packet  |-+    ...   |
          +--------+  | |scheduling|        ||scheduling| |          |
             ...      | +----------+        |+----------+ |  +-----+ |
          +--------+  |                     |             +->| ... | |
          |        |  |                     |             |  +-----+ |
          +--------+  |                     |             |  +-----+ |
                      |                     |             +->| ... | |
          +--------+  |                     |                +-----+ |
          |        |  |                     |           path manager |
          +--------+ -                      +------------------------+
            streams
```

Figure 1: Application Policy Awareness in Multi-path QUIC framework

4. Per-connection Policy

An application imposes per-connection policy through the primitives provided by the control plane. Some examples of policies are path preference (e.g. a Wi-Fi path is preferred twice than a cellular path), path model (e.g. full mesh, ndiffports, backup). The packet scheduling component and the path manager at the data plane will act based on these indications for path selection and data transmission. We assume the policies are 'soft'---the policies are not a must. Instead, the data plane will follow the policies as much as possible.

Let us take real-time interaction applications as an example to illustrate the basic idea. The applications are indeed delay sensitive but data volume is often low. 3 types of policies may be used by different applications, as shown in Table 1 where we assume only two paths are available (Wi-Fi and Cellular)

No.	Application defined policy: Path mode	Application defined policy: Path Preference	Underlying action: Packet Scheduling	Underlying action: Path mngm.
1	Wi-Fi=full, Cellular=full	Wi-Fi=1, Cellular=1	full redundant	/
2	Wi-Fi=full, Cellular=backup	Wi-Fi=1, Cellular=1	full redundant	activate backup interfaces when the active one's performance is lower than X for 5s
3	Wi-Fi=full, Cellular=full	Wi-Fi=2, Cellular=1	partially redundant	/

Table 1: Example policies of a real-time interaction application

The first type of policies would like to use two paths equally, and because the applications are delay sensitive, the actions will be 'full redundant' for the packet scheduling---two paths send the same data. The second type of policies, on the other hand, would like to use the Wi-Fi interface (possibly because of data charge) if possible. But if two paths have to be activated at the same time due to the lower performance of Wi-Fi, then the two paths are with same the preference. So, the corresponding actions will be will be 'full redundant' for the packet scheduling, and the path management component will monitor the performance of the active interface and activate the backup one if the performance is lower than a threshold for a short period of time (e.g. 5 seconds). The third type of policies would like to use the two interfaces at the same time, but Wi-Fi is preferred twice as the cellular one. The actions will take this into consideration, and implement partial redundant transmission over the cellular Interface.

Likewise, we can define a mapping between the policies of different types of applications and the actions in the data plane. We leave the design of such a mapping to the designers.

TODO: Add design of such a mapping.

5. Per-stream Intent

Per-stream intent is a unique feature provided by (MP)QUIC---it is implemented through the multiple streams in QUIC. Streams can be associated with priorities to implement applications intent. For instance, objects in a web page may be dependent on others and thus have different priorities [MPQUIC-Scheduler]. A priority-aware packet scheduling algorithm will improve the performance notably.

```
 High priority  /\  +----------+
                ||  |          |
                ||  +----------+
                ||  +----------+
```

```
               ||   |          |
               ||   +----------+
               ||      ...           User-defined priority
               ||   +----------+
 Low priority  ||   |          |
               ||   +----------+
 -----------------------------------------------------------
 High priority /\   +----------+
               ||   |          |
               ||   +----------+
               ||   +----------+
               ||   |          |
               ||   +----------+
               ||      ...           Default priority
               ||   +----------+
 Low priority  ||   |          |
               ||   +----------+
```

Figure 2: Stream priority

We envision a priority management scheme of two separated priority ranges (see Figure 2). The user-defined priority ranges are those streams that the applications explicitly designate the priorities, where the default priority ranges include the streams with no priority values set by the applications. Only when the streams in the user-defined ranges have no data sent, the data in the streams in the default priority ranges can be sent. In the same range, one can use the weighted round robin for scheduling---the higher-priority streams get more quantum for data sending in each round. One can also dynamically set/change the priorities of the streams in the default priority ranges to enable short stream first if needed.

We list in Table 2 some applications that can be benefited from the above priority-aware stream management scheme.

Application	Priority setting	Benefits
Web	Setting stream priority based on the objects'	Reduced page load time

	priorities [MPQUIC-Scheduler]	
Photo album	Default setting; dynamical changing priority to mimic shorted stream first like in flow scheduling in data centers [CDC]	Reduced load time of the first image; images of small size will be displayed first
Live streaming (viewing) / VoD	The chunks or GoPs with the closest deadline are put into the stream with higher priorities. This essentially implements deadline-aware data transmission	Low startup delay, low buffering ratio
Live streaming: broadcaster->server	The original segments are put into the streams in the user-defined priority range, while the enhanced segments are put into the streams in the default priority range	Enhanced shift-viewing quality of services [Vantage]

Table 2: Example applications benefited from priority-aware stream management scheme

TODO: Add priority management solution.

6. IANA Considerations

This document makes no request of IANA.

7. Informative References

[AD-MPTCP]
Froemmgen, A., Rizk, A., Erbshaeusser, T., Weller, M., Koldehofe, B., Buchmann, A., and R. Steinmetz, "A programming model for application-defined multipath TCP scheduling", Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference (Middleware '17). Association for Computing Machinery, New York, NY, USA, 134-146., 2017, <https://doi.org/10.1145/3135974.3135979>.

[BSC] Guo, Y., Nikravesh, A., Mao, Z., Qian, F., and S. Sen, "Accelerating Multipath Transport Through Balanced Subflow Completion", Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom '17). Association for Computing Machinery, New York, NY, USA, 141-153., 2017, <https://doi.org/10.1145/3117811.3117829>.

[CDC] Bai, W., Chen, L., Chen, K., Han, D., Tian, C., and H. Wang, "Information-agnostic flow scheduling for commodity data centers", Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation (NSDI'15). USENIX Association, USA, 455-468., 2015.

[ECF] Lim, Y., Nahum, E., Towsley, D., and R. Gibbens, "ECF: An MPTCP Path Scheduler to Manage Heterogeneous Paths", Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '17). Association for Computing Machinery, New York, NY, USA, 147-159., 2017, <https://doi.org/10.1145/3143361.3143376>.

[I-D.deconinck-quic-multipath]
De Coninck, Q. and O. Bonaventure, "Multipath Extensions for QUIC (MP-QUIC)", draft-deconinck-quic-multipath-03 (work in progress), August 2019.

[MP-DASH] Han, B., Qian, F., Ji, L., and V. Gopalakrishnan, "MP-DASH: Adaptive Video Streaming Over Preference-Aware Multipath", Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies (CoNEXT '16). Association for Computing Machinery, New York, NY, USA, 129-143., 2016, <https://doi.org/10.1145/2999572.2999606>.

[MPQUIC-Design]
De Coninck, Q. and O. Bonaventure, "Multipath QUIC: Design and Evaluation", Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '17). Association for Computing Machinery, New York, NY, USA, 160-166., 2017, <https://doi.org/10.1145/3143361.3143370>.

[MPQUIC-Scheduler]
Wang, J., Gao, Y., and C. Xu, "A Multipath QUIC Scheduler for Mobile HTTP/2", Proceedings of the 3rd Asia-Pacific Workshop on Networking 2019 (APNet '19). Association for Computing Machinery, New York, NY, USA, 43-49., 2019, <https://doi.org/10.1145/3343180.3343185>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[SMAPP] Hesmans, B., Detal, G., Barre, S., Bauduin, R., and O. Bonaventure, "SMAPP: towards smart multipath TCP-enabled applications", Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT '15). Association for Computing Machinery, New York, NY, USA, Article 28, 1-7., 2015, <https://doi.org/10.1145/2716281.2836113>.

[Vantage] Ray, D., Kosaian, J., Rashmi, K., and S. Seshan, "Vantage: optimizing video upload for time-shifted viewing of social live streams", Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM '19). Association for Computing Machinery, New York, NY, USA, 380-393., 2019, <https://doi.org/10.1145/3341302.3342064>.

[WN] Cui, Y., "Wireless Network Instabilities in the Wild: Measurement, Applications (Non)Resilience, and OS Remedy", 2019 IEEE/ACM Transactions on Networking, vol. 27, no. 1, pp. 214-230., 2019, <10.1109/TNET.2018.2885872>.

Authors' Addresses

Qing An
Alibaba Inc.

Email: anqing.aq@alibaba-inc.com

Zhenyu Li
ICT-CAS

Email: zyli@ict.ac.cn

Yanmei Liu
Alibaba Inc.

Email: miaoji.lym@alibaba-inc.com