TSVWG
Internet-Draft
Intended status: Informational
Expires: 3 September 2026

Y. Song
Pengcheng Laboratory
C. Li
East China Normal University
Q. Li
Pengcheng Laboratory

2 March 2026

Consistency-Aware Multipath Transport (CAMP) toward Interactive Multimodal LLM-Based Systems
draft-song-tsvwg-camp-00

Abstract

With the prosperity of generative large language models (LLMs), interactive LLM-based services, such as digital humans, have imposed new stringent requirements on low latency and high multimodal consistency. Traditional interactive LLM-based systems typically transmit multimodal content over a single network path, thereby failing to exploit the advantages offered by multipath networks. Even when multipath transport mechanisms are adopted, single-stream encapsulation does not enable differentiated management of heterogeneous modalities. However, naively separating modalities into multiple streams further introduces inter-modal arrival inconsistency. To address these challenges, this document specifies CAMP, a consistency-aware multipath transport design over the Multipath QUIC (MPQUIC) protocol. First, CAMP defines a three-stream separation encapsulation format to support modality-differentiated transmission. Second, it incorporates a transport-layer consistency-aware multipath scheduler to reduce inter-modal arrival time deviation across network paths. Third, it specifies a client-side application-layer alignment mechanism that operates in coordination with the transport scheduler. To the best of our knowledge, this is the first specification to address multipath-enabled multimodal consistency guarantees for interactive LLM-based systems.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction 3
2. Background and Challenges 4
  2.1. Characteristics of LLM-generated Multimodal Data 4
  2.2. Traditional Streaming Protocol 4
  2.3. Multipath Transport Basics 5
  2.4. Key Challenges 5
3. Problem Statement and Design Principles 6
  3.1. Problem Statement 6
    3.1.1. Multimodal Data Model 6
    3.1.2. Multipath Network Model 7
    3.1.3. Scheduling Function 7
    3.1.4. Performance Metrics 7
    3.1.5. Overall Problem Definition 7
  3.2. Design Principles 8
4. Consistency-Aware Multipath Transport (CAMP) Design 8
  4.1. Overall Architecture 8
    4.1.1. Server Side 10
    4.1.2. Multipath Transport Network 10
    4.1.3. Client Side 11
  4.2. Three-Stream Separation Format 11
  4.3. Consistency-Aware Scheduler 12
  4.4. Client-Side Alignment Mechanism 12
5. Security Considerations 13
6. IANA Considerations 14
7. References 14

7.1. Normative References 14
7.2. Informative References 14
Authors' Addresses 15

1. Introduction

Generative large language models (LLMs) have enabled a new generation of interactive systems, including digital humans, conversational agents, and multimodal assistants. These systems continuously generate and transmit multimodal (i.e., text, audio, and video) content in response to user inputs. Unlike traditional media streaming or static content delivery, interactive LLM-based systems operate under strict real-time constraints and require strong semantic and temporal coupling across modalities.

Existing transport approaches for interactive LLM-based systems predominantly rely on single-path transmission, which fails to leverage the bandwidth aggregation, redundancy, and latency diversity benefits offered by multipath networks. In addition, current implementations typically encapsulate multiple modalities within a single transport stream, which preventing fine-grained multimodal data management. For example, text and audio generally require reliable and in-order delivery, whereas video streams may tolerate controlled frame loss to reduce latency.

A straightforward improvement is to separate text, audio, and video into independent streams to enable differentiated transmission. However, this may occur significant inter-modal arrival time, introducing data incosistency probelm. Existing consistency mitigation mechanisms are primarily implemented at the client-side application layer through post-arrival timestamp alignment. However, they operate reactively and lack coordination with transport-layer multipath scheduler, leading sub-optimal performance. Furtheremore, existing multipath transport mechanisms primarily optimize for throughput or per-stream latency; they do not consider multimodal consistency requirements intrinsic to interactive LLM-based systems.

Based on the above challenges, this document specifies Consistency-Aware Multipath Transport (CAMP), a transport-layer design built on Multipath QUIC (MPQUIC). CAMP introduces structured modality separation, a consistency-aware scheduler, and a coordinated client-side alignment mechanism. The design does not modify the underlying MPQUIC protocol and is deployable over existing MPQUIC implementations.

The remainder of this document is organized as follows. Section 2 provides background and summarizes the key challenges. Section 3 defines the problem and presents the design principles. Section 4 specifies the CAMP architecture and mechanisms. Section 5 discusses security considerations, and Section 6 covers IANA considerations.

2. Background and Challenges

2.1. Characteristics of LLM-generated Multimodal Data

Interactive LLM-based systems generate heterogeneous yet semantically coupled data streams, typically including text, audio, and video. These modalities differ significantly in their traffic patterns, reliability requirements, delay sensitivity, and tolerance to loss.

Text data is discrete, low-bandwidth, and generally requires reliable and in-order delivery to preserve semantic correctness. Audio data is continuous and delay-sensitive, requiring bounded jitter and reliable delivery within strict playout deadlines. Video data is bandwidth-intensive and latency-sensitive but can tolerate controlled frame loss or quality adaptation in exchange for reduced delay.

Although these modalities exhibit distinct transport requirements, they are derived from a unified generative process and therefore require tight temporal alignment. For example, synthesized speech must correspond to generated text, and video frames representing facial animation must remain synchronized with spoken content. Consequently, multimodal transmission for interactive LLM systems requires both differentiated per-modality management and bounded inter-modal arrival skew.

2.2. Traditional Streaming Protocol

Traditional real-time streaming protocols, such as the Real-Time Messaging Protocol (RTMP), encapsulate audio and video data within logical channels over a single transport connection, typically based on TCP. These protocols were primarily designed for media broadcasting and content distribution rather than interactive, generative workloads.

Such designs exhibit two fundamental limitations in the context of interactive LLM-based systems. First, single-path transport cannot leverage multipath bandwidth aggregation, redundancy, or latency diversity. Second, multiple modalities are often multiplexed within a single transport stream, preventing differentiated congestion control, retransmission strategies, and scheduling policies. As a result, heterogeneous modality requirements cannot be independently satisfied.

These limitations make traditional streaming approaches unsuitable for interactive LLM-based systems that demand both real-time responsiveness and modality-aware management.

2.3. Multipath Transport Basics

Multipath transport protocols enable a single logical connection to utilize multiple network paths concurrently. Multipath QUIC (MPQUIC, as defined in [RFC9221]) extends QUIC (as defined in [RFC9000]) by allowing multiple paths under a single connection context, each with independent path characteristics. Multipath transport can provide bandwidth aggregation, path redundancy, and potentially reduced latency through dynamic path selection.

However, existing multipath scheduling algorithms primarily optimize metrics such as throughput maximization, congestion window utilization, or per-stream delivery latency. They do not explicitly account for cross-stream temporal relationships. When independent streams are distributed across heterogeneous paths with different delay and jitter characteristics, arrival-time divergence across streams may increase.

Therefore, while multipath transport improves network utilization and resilience, it does not inherently guarantee multimodal consistency. Moreover, existing multipath schedulers do not consider multimodal inconsistency problem.

2.4. Key Challenges

Based on the above background, four key challenges are concluded in transporting LLM-generated multimodal data:

1. *Inability of Single-Path Transmission to Exploit Multipath Advantages:* Traditional single-path transport cannot leverage multipath bandwidth aggregation, redundancy, or latency diversity. This limitation constrains achievable performance in real-time interactive environments.

2. *Lack of Differentiated Multimodal Management under Single-Stream Encapsulation:* Encapsulating text, audio, and video within a single transport stream prevents modality-specific congestion control, reliability policies, and scheduling strategies. As interactive LLM systems impose heterogeneous requirements across modalities, single-stream designs cannot satisfy differentiated quality-of-service objectives.

3. *Inter-Modal Arrival Inconsistency under Naive Stream Separation with Multipath:* Separating modalities into independent streams enables differentiated management. However, when such streams are scheduled independently across heterogeneous paths, significant inter-modal arrival skew may occur. This timing divergence degrades perceptual synchronization and interactive quality.

4. *Insufficiency of Pure Application-Layer Alignment without Transport-Layer Coordination:* Existing multimodal consistency mechanisms are primarily implemented at the client-side application layer through post-arrival timestamp alignment. While such approaches can partially compensate for skew, they operate reactively and lack coordination with transport-layer multipath scheduling decisions. Consequently, latency and consistency cannot be jointly optimized. Moreover, existing transport-layer schedulers are not designed to be consistency-aware and do not consider cross-modal timing constraints.

These challenges motivate the need for a consistency-aware multipath transport design tailored to interactive multimodal LLM-based systems.

3. Problem Statement and Design Principles

3.1. Problem Statement

In interactive LLM-based systems, multimodal outputs (i.e., text, audio, video) are generated continuously and transmitted over heterogeneous multipath networks. We aim to achieve low latency and cross-modal consistency under path heterogeneity.

3.1.1. Multimodal Data Model

Let $M = \{T, A, V\}$ denote the set of modalities.

For each modality m in M, let $S_m = \{s_{m,1}, s_{m,2}, ...\}$ denote the ordered sequence of generated data units.

Each data unit $s_{m,k}$ is associated with:

* generation time $g_{m,k}$

* size $\sigma_{m,k}$

* deadline δ_m

* reliability indicator r_m, where r_m is either 0 or 1

For a given semantic synchronization index k, the set {s_T,k, s_A,k, s_V,k} represents cross-modal correlated data units.

3.1.2. Multipath Network Model

Let P = {p_1, p_2, ..., p_N} denote the set of available paths.

Each path p_i is characterized by:

* delay function D_i(t)

* available bandwidth b_i(t)

* loss probability l_i(t)

We assume heterogeneous paths, i.e., there exist i != j such that D_i(t) is not equal to D_j(t).

3.1.3. Scheduling Function

Define a scheduling function pi(.) such that: pi(s_m,k) = p_i indicates that data unit s_m,k is transmitted over path p_i.

The arrival time of s_m,k is defined as: a_m,k = g_m,k + D_i(g_m,k).

3.1.4. Performance Metrics

End-to-end latency L is defined as: L = average(a_m,k - g_m,k).

Inter-modal arrival skew for synchronization index k is defined as: Delta_k = max_m(a_m,k) - min_m(a_m,k).

Expected cross-modal skew C is defined as: C = average(Delta_k). A smaller C indicates better cross-modal consistency.

3.1.5. Overall Problem Definition

Given:

* modality set M

* multimodal data streams {S_m}

* heterogeneous path set P

* path dynamics {D_i(t), b_i(t), l_i(t)}

Find a scheduling function pi(.) such that:

* L is minimized

* C is minimized

* Reliability constraint holds: If r_m = 1, the loss probability of s_m,k MUST be zero

* Deadline constraint holds: a_m,k - g_m,k <= delta_m

* Bandwidth constraint holds for each path p_i: The total transmitted size assigned to p_i at time t MUST NOT exceed b_i(t)

3.2. Design Principles

Based on the above problem statement, the consistency-aware multipath transport (CAMP) design for LLM-generated multimodal data is guided by the following principles:

1. *Modality Differentiation:* Different modalities (text, audio, and video) exhibit heterogeneous requirements in reliability, latency sensitivity, and rate characteristics. The transport layer MUST support differentiated handling across modalities rather than enforcing a single homogeneous policy.

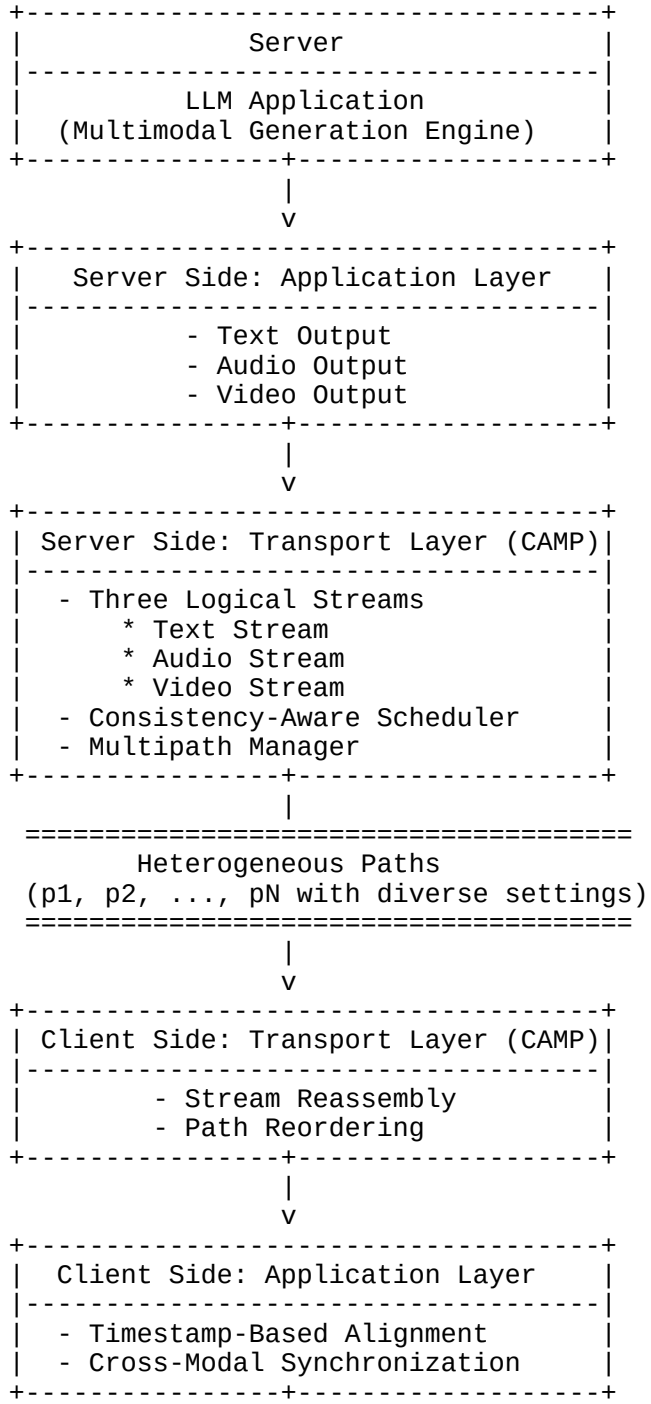
2. *Consistency Awareness*: In interactive multimodal systems, cross-modal arrival consistency is a first-order requirement. Scheduling decisions MUST consider inter-modal arrival skew in addition to per-flow latency. Minimizing latency alone is insufficient.

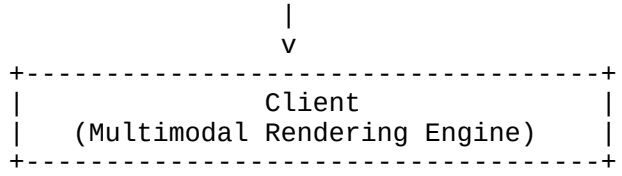
3. *Coordination Between Transport and Application Layers*: Cross-modal consistency is an end-to-end property that depends on both transport-layer scheduling and receiver-side application-level alignment mechanisms. Transport-layer multipath scheduling decisions influence arrival ordering and skew, while the receiver performs timestamp-based alignment and buffering. These mechanisms SHOULD be designed to cooperate, rather than operate independently.

4. Consistency-Aware Multipath Transport (CAMP) Design

4.1. Overall Architecture

CAMP adopts an end-to-end architecture that integrates multimodal generation, transport-layer consistency-aware scheduling, and receiver-side alignment. The high-level architecture is illustrated below.


Server
LLM Application
(Multimodal Generation Engine)
Server Side: Application Layer
- Text Output
- Audio Output
- Video Output
Server Side: Transport Layer (CAMP)
- Three Logical Streams
* Text Stream
* Audio Stream
* Video Stream
- Consistency-Aware Scheduler
- Multipath Manager
Heterogeneous Paths
(p1, p2, ..., pN with diverse settings)
Client Side: Transport Layer (CAMP)
- Stream Reassembly
- Path Reordering
Client Side: Application Layer
- Timestamp-Based Alignment
- Cross-Modal Synchronization



The above architecture consists of two primary components: Server Side and Client Side, with clear separation of responsibilities between the transport layer and the application layer at both ends. The end-to-end design ensures that the transport layer and application layer work in coordination to meet the low-latency and consistency requirements of multimodal interactive systems.

4.1.1. Server Side

At the server side, the system is responsible for generating multimodal data, including text, audio, and video, which are produced by the LLM-based application. These modalities are generated by the LLM Application (Multimodal Generation Engine), which processes user inputs and produces multimodal outputs in real-time.

Once the data is generated, the Application Layer handles the preparation of each modality for transmission. This layer separates the modalities (text, audio, video) into different logical streams, enabling modality-specific handling. For example, while text and audio may require reliable transmission, video might be able to tolerate some packet loss, and thus can be transmitted with a different reliability strategy.

The Transport Layer (CAMP) then encapsulates each modality into its own logical stream. The transport layer is responsible for consistency-aware scheduling, managing how data is sent over multiple paths. This layer uses a Consistency-Aware Scheduler to make scheduling decisions that minimize inter-modal arrival skew and optimize the use of available bandwidth. It takes into account the heterogeneity of the transport paths and the varying needs of different data types (modalities).

4.1.2. Multipath Transport Network

The network is composed of multiple heterogeneous paths (denoted as p1, p2, ..., pN) that offer diverse characteristics such as delay, bandwidth, and packet loss. These paths are utilized by the CAMP Transport Layer to maximize the available bandwidth and optimize latency, while ensuring cross-modal consistency.

CAMP uses multipath transport to enable the aggregation of bandwidth and path redundancy, but the scheduling of data across these paths is done in a way that minimizes the inter-modal arrival skew. The transport layer takes the different path characteristics into account, ensuring that data from different modalities arrives at the receiver in a synchronized manner.

4.1.3. Client Side

On the client side, the data streams are received and reassembled by the Transport Layer (CAMP). This layer is responsible for reordering data packets that may have arrived out of order due to the diverse paths used for transmission. The Multipath Manager ensures that the transport layer maintains synchronization across all streams and guarantees consistency in the arrival times of the different modalities.

Once the data is reassembled, the Application Layer at the client side performs timestamp-based alignment and cross-modal synchronization. The timestamp-based alignment mechanism ensures that the data from all modalities (text, audio, video) is correctly synchronized before being presented to the user. This synchronization is essential for interactive multimodal applications like digital humans or virtual assistants, where inconsistencies in the timing of different modalities would disrupt the user experience.

The Client then renders the synchronized multimodal data, ensuring a seamless experience for the user. The interaction between the transport layer and application layer ensures that latency is minimized, and consistency between modalities is maintained throughout the system.

4.2. Three-Stream Separation Format

The Three-Stream Separation Format is designed to handle multimodal data (text, audio, and video) in a manner that ensures independent management and optimized transmission for each modality. In this format, each modality is encapsulated in its own logical stream within the CAMP protocol. By separating these modalities, we can apply different transmission policies tailored to the unique needs of each type of data, such as reliability, latency, and throughput.

Each stream is identified by a unique stream identifier (e.g., T for text, A for audio, V for video), allowing for differentiation and independent handling. The encapsulation of each modality also includes important metadata that is used to ensure the correct delivery and synchronization of the streams. This metadata can include generation timestamp, modality-specific parameters, synchronization index, etc.

By using separate streams and attaching the appropriate metadata, we enable more efficient and precise transmission of each modality. The CAMP protocol leverages multipath transport to distribute these streams across different network paths, optimizing the delivery of each modality based on its specific transmission needs.

4.3. Consistency-Aware Scheduler

The Consistency-Aware Scheduler is responsible for optimizing the transmission of multimodal data streams while ensuring that the inter-modal timing remains consistent. By leveraging multipath transport, the scheduler assigns each modality to the most appropriate network path based on its specific needs. It considers factors such as bandwidth, latency, and reliability, ensuring that each stream (text, audio, or video) is transmitted under the conditions that best suit its characteristics.

The scheduler dynamically adjusts to changing network conditions, such as congestion or fluctuating bandwidth, while also ensuring that the timing between different streams remains consistent. It operates by coordinating the scheduling of streams in a way that minimizes timing skew between modalities, thus improving the overall synchronization. This coordination with the transport layer enables a consistent delivery of multimodal data, ensuring that all streams are delivered in sync for interactive LLM-based systems.

4.4. Client-Side Alignment Mechanism

The Client-Side Alignment Mechanism is responsible for ensuring that multimodal data streams received by the client are synchronized correctly before being rendered. Upon receiving the streams, the client uses timestamps and synchronization indices, which were embedded during the transmission process, to align the streams (text, audio, and video) based on their respective generation times. This ensures that the modalities are presented in a synchronized manner, with minimal temporal divergence.

The alignment process involves using the application layer to adjust the playback of each modality based on the synchronization index and timestamps. In the event of minor discrepancies caused by network

conditions or transmission delays, the client employs techniques such as buffering to mitigate these differences and ensure smooth playback. This mechanism relies on continuous feedback from the transport layer to adjust the synchronization strategy dynamically, ensuring that the multimodal data remains consistent across all modalities throughout the interaction.

5. Security Considerations

The CAMP protocol design includes several elements to ensure secure transmission of multimodal data across different network paths. However, as with any transport protocol, security considerations must be addressed to prevent potential security threats.

1. *Data Integrity and Confidentiality*
   Given that CAMP uses multipath transport to transmit multimodal data across multiple paths, ensuring data integrity and confidentiality is critical. We recommend leveraging the existing encryption mechanisms of MPQUIC, which supports encryption of data in transit using standard algorithms like advanced encryption standard (AES). This ensures that data is not compromised during transmission, preventing eavesdropping and unauthorized access.

2. *Authentication and Authorization*
   As with any network protocol, ensuring that both sender and receiver are authenticated is vital. CAMP relies on MPQUIC’s built-in authentication mechanisms, which leverage public key infrastructure (PKI) or certificate-based authentication to prevent unauthorized parties from participating in the communication session.

3. *Replay and Man-in-the-Middle Attacks*
   CAMP utilizes time-stamped data and synchronization indices to coordinate multimodal streams. These timestamps should be protected from tampering to prevent replay attacks or man-in-the-middle (MITM) attacks. To mitigate this risk, CAMP ensures that all communication channels are protected by end-to-end encryption, and utilizes integrity protection schemes (e.g., HMAC) to verify the authenticity of the timestamps and synchronization indices.

4. *DoS and DDoS Attacks*
   The CAMP protocol should also consider protection against denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks, where malicious users could flood the system with excessive data, causing network congestion and delaying or disrupting data transmission. To mitigate such attacks, CAMP should implement rate limiting and congestion control mechanisms as defined in the MPQUIC specification.

5. *Trust and Privacy Considerations*
   Since CAMP is designed to handle multimodal data streams, it is important to consider the privacy of the data being transmitted. Any sensitive data, such as personally identifiable information (PII), should be encrypted and anonymized when possible. Additionally, careful access controls should be implemented to ensure that only authorized entities can access the data.

These are just some of the key security considerations related to the CAMP protocol. Future versions of the protocol may introduce additional security measures as new threats emerge.

6. IANA Considerations

This document does not define any new protocol numbers, port numbers, or other IANA-managed resources. However, future revisions of the CAMP protocol may define new types, identifiers, or registry entries, in which case the relevant considerations will be added.

If any new IANA registries or assignments are required in future versions of this document, the authors will request the necessary changes from IANA.

7. References

7.1. Normative References

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

7.2. 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>.

[RFC9221] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", RFC 9221, DOI 10.17487/RFC9221, March 2022, <https://www.rfc-editor.org/info/rfc9221>.

Authors' Addresses

Yuhong Song
Pengcheng Laboratory
Email: songyh@pcl.ac.cn

Changlong Li
East China Normal University
Email: clli@cs.ecnu.edu.cn

Qing Li
Pengcheng Laboratory
Email: liq@pcl.ac.cn