



清华大学

综合论文训练

跨国公网链路传输优化方法研究

系 别： 电子工程系

专 业： 电子信息科学与技术

姓 名： 高 艺 轩

指导教师： 王 博 副教授

二〇二六年五月

关于论文使用授权的说明

本人完全了解清华大学有关保留、使用综合论文训练论文的规定，即：学校有权保留论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

作者签名：

导师签名：

日 期：

日 期：

摘 要

论文的摘要是对论文研究内容和成果的高度概括。摘要应对论文所研究的问题及其研究目的进行描述，对研究方法和过程进行简单介绍，对研究成果和所得结论进行概括。摘要应具有独立性和自明性，其内容应包含与论文全文同等量的主要信息。使读者即使不阅读全文，通过摘要就能了解论文的总体内容和主要成果。

论文摘要的书写应力求精确、简明。切忌写成对论文书写内容进行提要的形式，尤其要避免“第 1 章……；第 2 章……；……”这种或类似的陈述方式。

关键词是为了文献标引工作、用以表示全文主要内容信息的单词或术语。关键词不超过 5 个，每个关键词中间用分号分隔。

关键词：关键词 1；关键词 2；关键词 3；关键词 4；关键词 5

Abstract

An abstract of a dissertation is a summary and extraction of research work and contributions. Included in an abstract should be description of research topic and research objective, brief introduction to methodology and research process, and summary of conclusion and contributions of the research. An abstract should be characterized by independence and clarity and carry identical information with the dissertation. It should be such that the general idea and major contributions of the dissertation are conveyed without reading the dissertation.

An abstract should be concise and to the point. It is a misunderstanding to make an abstract an outline of the dissertation and words “the first chapter”, “the second chapter” and the like should be avoided in the abstract.

Keywords are terms used in a dissertation for indexing, reflecting core information of the dissertation. An abstract may contain a maximum of 5 keywords, with semi-colons used in between to separate one another.

Keywords: keyword 1; keyword 2; keyword 3; keyword 4; keyword 5

目 录

第 1 章 引 言.....	1
1.1 研究背景	1
1.2 研究现状	2
1.3 研究思路与贡献	2
1.4 论文内容	3
第 2 章 背景介绍与相关工作.....	4
2.1 云网络与覆盖网络	4
2.2 链路质量优化	6
2.2.1 简单复制冗余	7
2.2.2 分组冗余码	8
2.2.3 流式冗余码 (Streaming 码)	10
2.3 软件定义网络与网络调度	12
2.4 现有系统问题分析	14
2.4.1 网络路径的分段质量差异造成冗余编码效率低	14
2.4.2 现有链路调度方法忽略了低质量公网链路的可修复性	16
第 3 章 跨域云网络传输性能提升研究.....	18
3.1 设计目标与总体思路	18
3.2 系统总体架构	18
3.3 设计挑战	19
3.4 交织 XOR 前向纠错编码设计	20
3.5 基于丢包统计的自适应参数调整	21
3.5.1 丢包信道模型	21
3.5.2 参数估计与编码参数搜索	22
3.6 解码端输出速率控制设计	22
3.7 本章小结	23
第 4 章 实验验证与分析.....	24
参考文献.....	25
附录 A 补充内容	28
致 谢.....	29
声 明.....	30

插图清单

图 1.1 云网络为音视频通话提供服务.....	1
图 2.1 覆盖网络基于底层网络，将各类物理网络资源抽象为一个虚拟的覆盖网络	5
图 2.2 即使启用了快速重传机制，TCP 仍旧需要一个往返时延才能恢复丢包	7
图 2.3 早期 FEC 工作将冗余信息附加在后续发出的包中进行发送	8
图 2.4 分组码为 n 个数据包附加 k 个冗余包	8
图 2.5 交织编码示意.....	10
图 2.6 分组码需要暂停解码输出等待冗余包到来才能恢复丢包并继续解码过程..	11
图 2.7 混合 SDN 网络	12
图 2.8 跨域云网络路径中的链路分段质量差异.....	15
图 2.9 用户流量高峰与公网链路质量下降时段的重合.....	16
图 3.1 系统总体架构.....	18
图 3.2 交织编码矩阵结构示意 ($d = 4, k = 3$)	21
图 3.3 三状态丢包信道模型.....	22
图 3.4 Pacer 控制模型	22

附表清单

符号和缩略语说明

PI 聚酰亚胺

第 1 章 引 言

1.1 研究背景

云网络（Cloud Networking）是一种新型的网络部署与管理架构。云网络服务商通过预先在全球各地部署服务器与网络资源，其他软件服务的服务商可以通过租用这些计算和网络资源，并将他们进行互联，组建适用于自身业务需求的遍布全球的云网络。云网络是在物理网络及计算资源之上构建的逻辑网络，扩展或重新配置时只需要进行软件设置，而不需要对网络设备硬件进行更改。云网络易于配置、可靠性高、易于扩展等众多优点使得它被广泛应用于文件传输、实时音视频通话、企业资源管理等多种服务中。

对于跨区域的实时音视频通话业务，通话服务的服务商需要建立一条连接通话用户两端的双向连接。如图1.1，通常，通话服务商选择使用云网络为进行通话的用户建立连接。通话的用户各自选择距离自己最近云网络接入网关接入云网络，数据经由云网关进入云网络进行转发，再从接收端用户接入的云网关发至接收端用户。对于实时通信业务来说，传输的延迟以及传输视频的卡顿率极大地影响用户体验（Quality of Experience, QoE）^[1]，因此为了维持优秀的用户体验，云网络提供商必须确保云网络提供低延迟、低丢包率的转发链路。

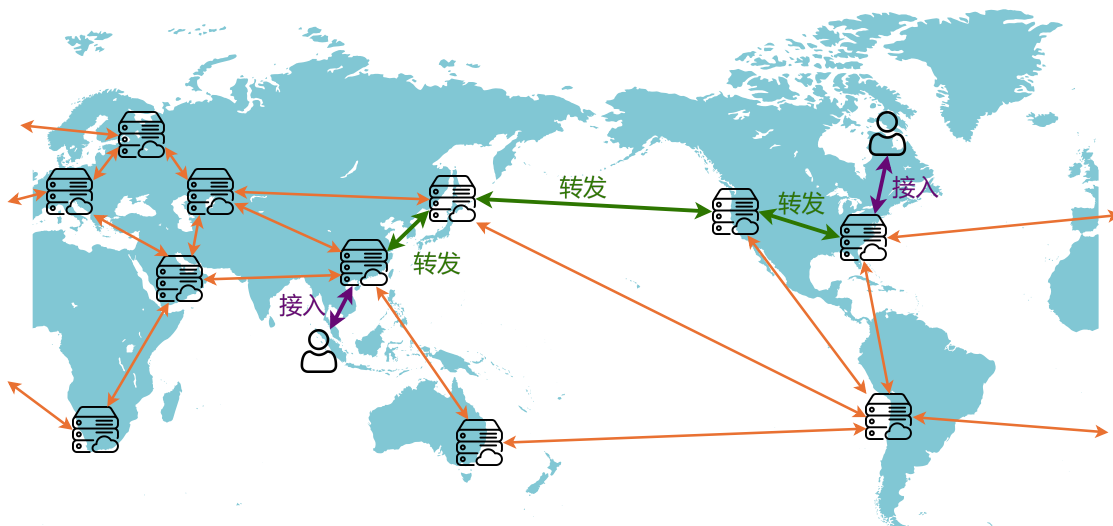


图 1.1 云网络为音视频通话提供服务

在云网络中，同一条逻辑链路的连接可以由多条物理链路抽象而成，云网络服务商通常同时提供专线与公网链路作为同一条物理链路的可选物理链路。通常来讲，专线的质量较高，延迟较低且稳定、丢包率低，能提供较好的传输质量和用

用户体验，但是高昂的价格对云服务商大规模使用带来挑战；与之相对地，公网链路的价格较低，但是容易受到网络中其它用户的影响，容易发生拥塞和竞争，传输质量容易发生波动，不能提供稳定优质的用户体验。因此，近期的研究主要聚焦在如何优化云网络的以在用户体验与运营成本之间取得平衡。

1.2 研究现状

物理链路调度类的工作^[1,2]通过不断监控同一逻辑链路下的公网链路和专线链路的质量，并在公网质量优秀可以为用户提供优质服务的时段将部分流量经由公网链路发送，从而希望能以此降低在专线上发送的数据流量，从而降低使用专线的成本。然而实际上，本研究的测量表明用户的高需求时段与公网链路质量下降时段基本重合，有大量流量需要提供服务时恰逢公网链路质量下降不能满足用户体验需求，公网链路的分流效果有限，大量流量仍旧通过专线转发，实际成本下降效果有限。

链路优化工作^[3-5]则放弃使用专线链路，通过在发送端加入冗余信息以对抗在公网链路上传输时，可能遭遇的丢包。然而，为了尽可能地恢复丢失的信息，算法必须悲观地估计在链路上传输时可能遭遇的最差情况，并据此估算需要加入的冗余信息量。这使得链路上为了恢复一小部分可能的丢包而额外传输了大量的冗余数据，造成了带宽的浪费，同时也提升了使用链路的流量成本。

已有的这些方法都没有很好地解决用户质量与运营成本的平衡问题，因为它们：

- 没有考虑用户需求升高与公网链路质量下降的时间相关性；
- 没有对公网链路丢包特性进行细致的建模。

这些工作都对真实公网的性质没有细致的研究，因而提出的方法不能很好地适应实际情况，造成运营成本居高不下。

1.3 研究思路与贡献

本文的作者通过在世界各地部署测试服务器，对真实的公网链路进行了持续的测量，并根据测量的结果，产生了两点观察：

1. 不同的公网链路的质量差距大。部分公网链路质量高，延迟稳定且丢包少，质量与专线几乎相同；还有一些公网链路的质量低，延迟不稳定，且易出现较为严重的丢包现象。这些低质量的公网链路通常是跨国链路。对于一条跨域的云网络连接，它通常在云网络内部也分为多段接力转发（见图1.1），而

丢包通常只集中发生在跨域的一段链路中，使得端到端测量得到的丢包率较高；

2. 在丢包率较高的链路中，连续丢包的情况较为常见。一般的冗余算法直接在一数据包发送之后立刻附上与之相匹配的冗余包，在公网链路上因连续丢包而导致同时丢失了数据包和冗余包而导致最终无法恢复丢失的数据。

基于这两点观察，本文设计了一套新的基于交织前向纠错编码（Interleaved Forward Error Correction, Interleaved FEC）的跨国公网链路优化方法。本文提出的方法使用公网实现云网络中所有节点的互联，但只在低质量的链路片段上应用 FEC 冗余编码进行链路质量修复。本方法不需要使用专线连接，极大地降低了链路的使用成本，同时又有选择性地地在低质量链路上使用冗余编码，避免了在高质量链路上添加额外带宽。另外，应用交织编码技术，将冗余包与数据包间隔其它数据包发送，极大地降低了链路连续丢包对丢包恢复的影响。

本文作者使用 Rust 语言实现了基于本方法的分布式云网络转发与针对低质量链路的冗余包计算及丢包恢复算法。经过对真实网络的模拟实验，本文提出的方法将端到端带宽提升了 xxx，流完成时间减少了 xxx。

总结而言，本文主要的贡献是：

- 通过对公网链路的真实测量，指出了长距离跨域公网链路质量差的核心在于单段跨域链路片段存在链路质量差、连续丢包多的特性；
- 提出了通过针对性地对低质量链路片段加入冗余，以最低的额外带宽开销实现对链路整体质量的提升；
- 实现并测量了本文提出的链路优化方法在跨国公网链路场景下对端到端性能的提升。

1.4 论文内容

(To be filled)

第 2 章 背景介绍与相关工作

2.1 云网络与覆盖网络

云网络的核心思想是服务商将计算和网络基础设施作为一种服务进行售卖（Infrastructure as a Service, IaaS）的新型计算范式^[6]。它的核心思想是云网络的服务商出资搭建数据中心并购买网络资源将数据中心内的计算、存储等单元连接互联网，其他服务提供商或者个人用户可按需要购买云服务商中提供的资源，并通过互联网访问。与传统的网络依赖与本地硬件进行部署不同，云网络通过虚拟机、虚拟路由器、虚拟交换机、负载均衡、虚拟防火墙等多种技术将已有的物理网络和计算资源抽象为虚拟化的计算资源，提供给不同的用户进行访问。通过网络虚拟化技术，云网络同时减少了计算资源的提供商与用户的成本，因为云网络的虚拟化特性使得资源可以按需用户需求动态分配与计费，用户只会为自己真正使用的资源付费，而云服务商可以通过对虚拟资源在硬件上的整合避免资源分配后的浪费，高效地满足所有用户的资源需求，降低运营成本^[7]。

在云网络模型下，用户所能访问的几乎所有资源如计算资源、存储资源以及网络互联资源都是虚拟资源而非物理资源。在云网络中，云服务器、云存储等实例可能分布在不同的物理节点上，所有的资源都需要通过高质量的网络进行互联，因此高性能的虚拟网络是云网络的基础^[8]。

覆盖网络（Overlay Network）是一种广泛应用于云网络结构的网络虚拟化设计，它基于物理的底层网络（Underlay Network）上通过对资源的逻辑整合而形成的逻辑网络。如图2.1，覆盖网络在已有的硬件网络上构建一个虚拟的网络层，使得使用云网络服务的企业和用户可以获得更灵活与稳定的虚拟网络连接。近年来，企业对虚拟化和云网络的需求不断增长，因而将分布在全球各地的云资源进行互联的需求也不断提升。不同的云资源所处的基础设施可能出现异构的情况，使用覆盖网络可以有效地将这些区别隐藏在相同的虚拟网络层抽象之后，极大地简化了部署和配置网络的成本。

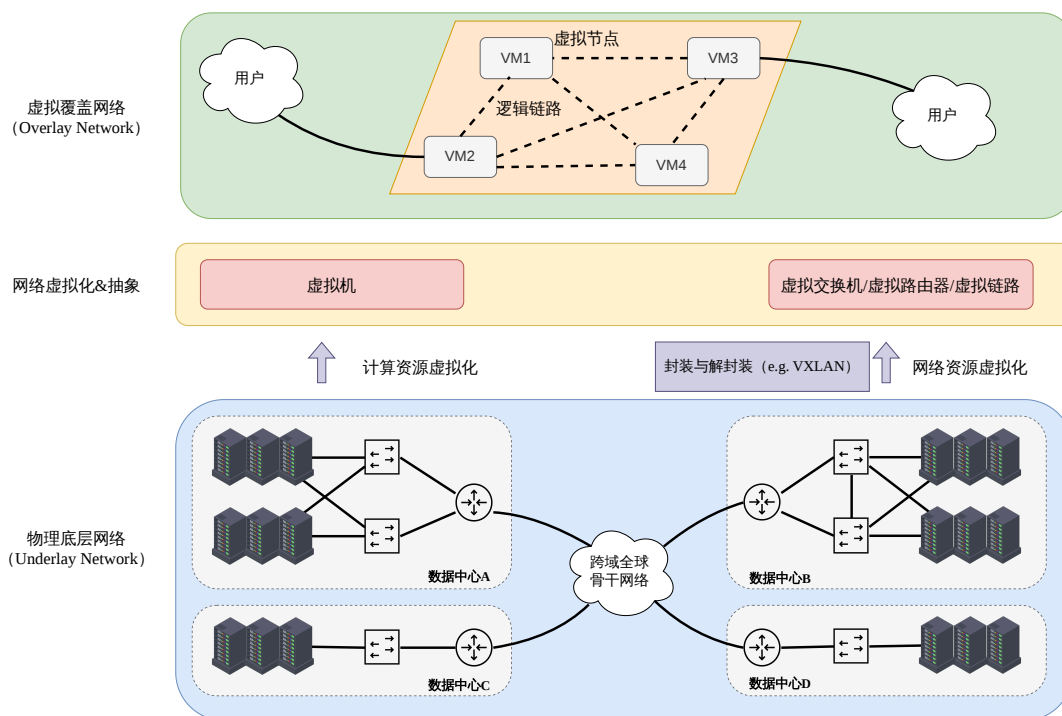


图 2.1 覆盖网络基于底层网络，将各类物理网络资源抽象为一个虚拟的覆盖网络

覆盖网络的实现依赖于隧道封装技术，其基本原理是将原始的二层或三层报文封装在另一种网络协议中进行传输，从而在底层的 IP 网络上构建虚拟的二层网络。当前主流的 Overlay 隧道技术主要包括 VXLAN^[9]、NVGRE^[10] 和 Geneve^[11] 等，它们在封装格式、协议机制和适用场景上各有特点。

VXLAN (Virtual eXtensible Local Area Network, 虚拟可扩展局域网)^[9] 是由 IETF 制定的虚拟网络技术之一，广泛应用于在数据中心和云网络中。VXLAN 通过 MAC over UDP 的方式，将二层的以太网帧封装在 UDP 报文中通过公网传递，对虚拟的二层网络在三层网络的基础上进行扩展。VXLAN 使用 24 比特的虚拟网络标识 (VXLAN Network ID, VNI) 来区分不同的虚拟以太网，可以突破传统 VLAN 的 4096 个虚拟网络数量限制，提供约 1600 万个各自独立的虚拟局域网。VXLAN 协议将普通的二层网络数据帧添加上 VXLAN 的包头，之后再将数据包装上外层的以太网、IP 和 UDP 报文头后发送至公网。VXLAN 包的封装和解封装由 VXLAN 隧道端点 (VXLAN Tunnel End Point) 进行，VTEP 负责将从虚拟机进入隧道的包进行封装，也负责将从隧道接收到的包进行解封装后交付给虚拟机。这使得 VXLAN 隧道对虚拟机透明，便于与其他网络系统集成。VXLAN 利用已有的 UDP 传输机制在网络中建立隧道，成熟度高，当前已广泛应用于数据中心。

NVGRE (Network Virtualization using Generic Routing Encapsulation, 基于路由封装的网络虚拟化)^[10] 是另一种主要的虚拟隧道协议。该协议主要应用于微软的

Hyper-V 虚拟环境中^[12]。NVGRE 将二层的 MAC 包封装在 GRE 隧道包内通过公网传递，利用 GRE 协议中的 Key 字段传递包所属的虚拟子网标识（Virtual Subnet ID, VSID）以及流标识（FlowID）。NVGRE 同样以 24 比特标识虚拟网络的名称，因此也可以支持最多约 1600 万个虚拟子网。同时，NVGRE 支持在同一子网内进一步通过流标识来区分不同的数据流，为更精细地管理流量和流量均衡提供了支撑。然而，这要求物理网络设备具备识别和处理这些字段的能力，对在公网部署带来了一定的挑战。

Geneve（Generic Network Virtualization Encapsulation，通用虚拟化网络封装技术）^[11]是 IETF 新提出的通用网络虚拟化封装协议，旨在以单一、可扩展的封装格式取代碎片化的 VXLAN、NVGRE 等多种隧道协议，以维持生态统一。Geneve 也采用 MAC over UDP 的封装，通过灵活配置的元数据传递机制满足多种网络虚拟化需求。Geneve 也使用 24 比特的虚拟网络标识（Virtual Network Identifier, VNI）来区分不同的虚拟网络，支持的网络数量与 VXLAN、NVGRE 等协议相当。与 VXLAN 等协议不同，Geneve 允许在头部后添加可变长度和数量的控制位和控制信息，可以有效满足不同虚拟网络的需求，增强了可扩展性。Geneve 协议通过设计可选的元数据空间，允许在不修改协议的前提下引入新功能，自推出以来已经逐步得到各类虚拟网络平台的支持^[13,14]，但是协议较为复杂，适配难度较大。

2.2 链路质量优化

低质量的互联网链路由于负载较大出现拥塞或部分设备运行故障，容易出现丢包或者延迟波动。在这些低质量的链路上进行传输时，即使链路还有可用的传输带宽，也会出现丢包或是延迟波动。即使 TCP^[15]等可靠传输协议通过重传确保了所有数据都能可到达，但性能较差。这是因为 TCP 协议依靠超时重传来在确保所有数据都最终送达至接收端，即使使用了基于重复 ACK 的快速重传机制，如图2.2，恢复单个丢失的包也至少要经历接收端检测丢包——请求发送端重传——发送端重传包送达恢复的过程，至少需要一个往返时延（Round Trip Time, RTT）才能恢复。对于一条在云网络中的跨域链路，往返时延可能达到300 ms 或更长，如此缓慢的丢包恢复不仅会阻塞后续数据包的发送，也会极大地影响实时媒体服务如影视直播、视频通话等应用的用户体验。

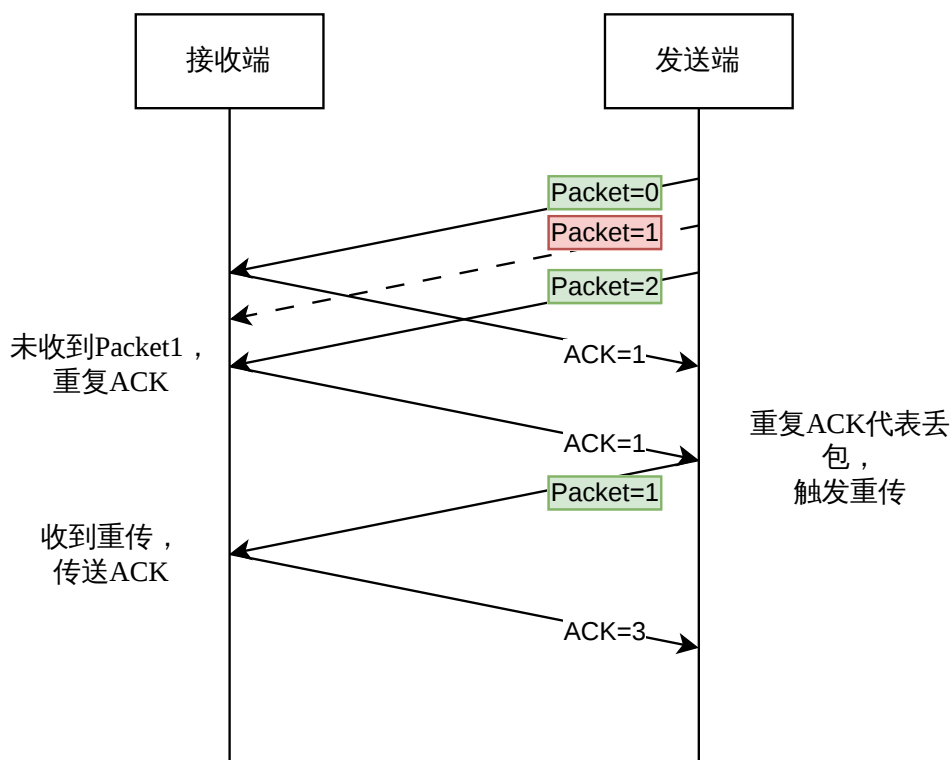


图 2.2 即使启用了快速重传机制，TCP 仍旧需要一个往返时延才能恢复丢包

针对此问题，研究者们提出了多种解决方案，其中前向纠错编码（Forward Error Correction, FEC）被广泛地用于应对链路传输中的丢包。其基本思想是，在发送数据时直接加入一部分冗余信息，以确保在部分信息丢失时，接收端无需请求发送端重新传送任何信息，而可以利用已经接收到的信息配合冗余信息推算出丢失的信息。使用前向纠错编码进行丢包恢复时，与重传机制需要经历一整个往返时延的长时间反馈路径不同，利用前向纠错编码的冗余包进行恢复只需要等待后续冗余包送达后即可进行，错误恢复时间短，能更好地适应延迟敏感型应用如视频通话等应用的需求。

为了实现高效地前向纠错编码，研究者们提出了多种编码方式，它们针对不同的目标进行了设计和优化。

2.2.1 简单复制冗余

早期的 FEC 工作主要通过对时间敏感的数据包进行简单地复制和多次传输进行错误恢复。如图2.3所示，通过将每个数据包复制多份，每次发送新的数据的同时，在同一个数据包中同时捎带发送之前已经发送过的一些数据包，这样可以在

一部分数据包丢失的同时仍旧确保接收端收到了所有数据。Bolot 等人^[3]基于此思路提出可以利用实时语音通话应用中已经存在的平均丢包率监控字段对传输链路的丢包模式和丢包律进行估计，从而动态地选重复发送包的发送间隔和次数，优化通话用户的用户体验。之后 Gandikota 等人^[16]在此基础上提出可以通过多路径传输，进一步提升冗余包和原始数据包中至少有一个送达的概率。Gandikota 等人工作中提出通过估算网络中的丢包率，动态地调整冗余参数以实现对语音流中的重要子流进行保护，同时再将编码后的数据包以及其他次要子流经过两个最大程度节点不相交路径在网络上与重要数据流分开传输，以降低数据传输丢失概率、提升用户体验。Huang 等人^[4]通过测量 Skype 应用在不同丢包网络条件下的行为，印证了相关冗余编码在提升实时语音通话用户体验方面的积极作用。

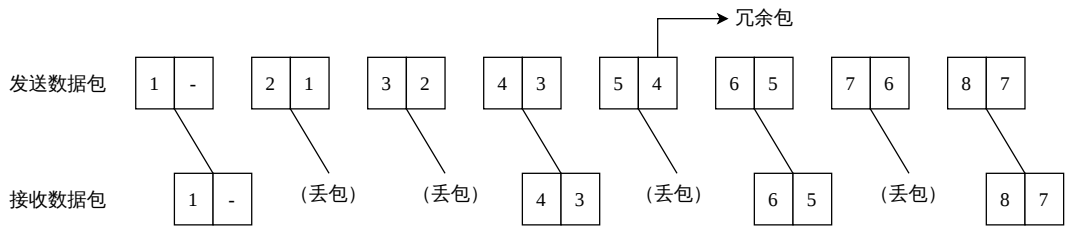


图 2.3 早期 FEC 工作将冗余信息附加在后续发出的包中进行发送

2.2.2 分组冗余码

通过重复发送数据包的方式添加冗余虽然简单，但是会带来较高的冗余开销，为了提高冗余信息的恢复效率，研究者们进一步提出了基于分组冗余码的前项纠错机制。XOR 码和 R-S 码是较为主要的冗余纠错恢复机制。如图2.4，这两种编码都是线性分组码，将原始数据分为 n 个数据包一组，对于每一组数据再加入 k 个冗余数据包并将 $n + k$ 个数据一并发送，接收端同样以组为单位进行丢包的恢复。

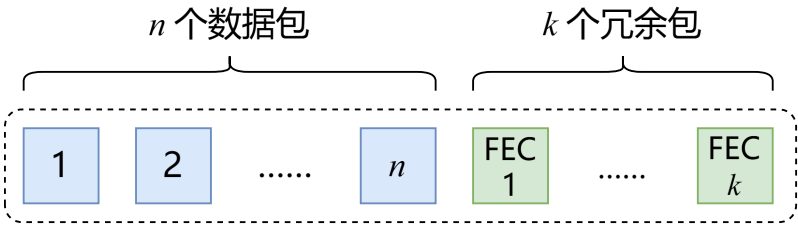


图 2.4 分组码为 n 个数据包附加 k 个冗余包

在 XOR 编码中， n 可以为任意值，而固定 $k = 1$ ，冗余包通过将所有组内的数据包按位进行异或运算得到。如果接收端只接收到了一组数据包共 $n + 1$ 个包中的 n 个，则丢失的包可以通过对已经接收到的包按位进行异或运算恢复得到。XOR 编码可以在一组数据共 $n + 1$ 个包丢失任意一个时通过剩余的 n 个包将丢失的包恢

复，但是如果丢失了两个或更多包，则完全不能恢复丢失的数据。XOR 码的计算简单，冗余包生成和丢失数据包恢复都只需要使用异或运算即可完成，运算开销小，但是只能恢复固定模式的少量丢包，面对组内多个丢包的情况效果有限。

为应对 XOR 编码的缺点，在 1960 年，Reed 与 Solomon 提出了 R-S 编码^[17]。R-S 编码保证，对于 n 个数据包和 k 个在有限域上计算出的冗余包共 $n + k$ 个数据包，接收端只要接收到了其中的任意 n 个，就能完整地恢复出所有的原始数据包。相较于 XOR 编码，R-S 编码的恢复能力有较大的提升，能够在同一个编码组里出现较多的丢包的恶劣情况下进行恢复，从能承受最多 1 个丢包增加至能承受最多 k 个丢包。RS 编码被广泛用于传输音视频流媒体，Lin 等人^[18]通过在无线局域网链路上对数据包进行 FEC 编码，提升了视频传输的效果。更多的其他研究者选择结合视频编码自身以帧和画面组（Group of Pictures, GOP）进行编码的特性，进行 FEC 编码以提升视频传输质量。Shih 等人^[19]通过对视频中的关键帧进行 FEC 保护，提升了关键帧以及后续多个依赖关键帧的画面质量，有效提升了视频传输质量。Xiao 等人^[20]使用贪心算法动态决定每个冗余组需要包含的帧数量及冗余度，在不牺牲延迟的情况下提升了视频质量。Yang 等人^[21]通过估算不同的数据包丢包后对解码视频的影响时间，动态选择 FEC 参数以提升用户的视频观看体验。Kurdoglu 等人^[22]则将 FEC 冗余率与编码帧率、编码量化参数及编码方式等联合优化，以最佳化用户观看体验而非追求更高的单一量化指标。总体而言，XOR 码和 R-S 编码相比简单复制冗余具有更高的冗余恢复效率，因此被广泛应用于实时音视频传输等场景。

然而，R-S 编码通常以数据组为单位进行编码与恢复，其恢复能力依赖于单个编码组中的丢包数量不超过冗余包数量 k 。实际网络上的丢包并不是独立的，在部分链路上可能由于链路拥塞、无线信号衰减等原因出现连续的突发丢包。在这些场景下如果使用 R-S 编码进行丢包恢复，为了能成功恢复数据，必须按照最差的可能情况决定 n 与 k 的相对取值，而这通常使得算法对网络的状态产生过于悲观的估计，为了应对短暂出现的连续丢包而将 k 的值始终维持在较高水平。这导致在其他未遭遇连续丢包的数据组中，大量的冗余包被浪费，占用了传输带宽而未能有效地提升传输质量。为解决此问题，研究者们提出了交织（Interleave）技术。如图2.5所示，交织技术将多个编码组交替地在网络上发出，使得当传输过程中出现了连续丢包时，丢包被分散在多个不同的编码组中分别应对，使得单个编码组需要应对的丢包比例大大下降，从而降低了整体需要的冗余率。

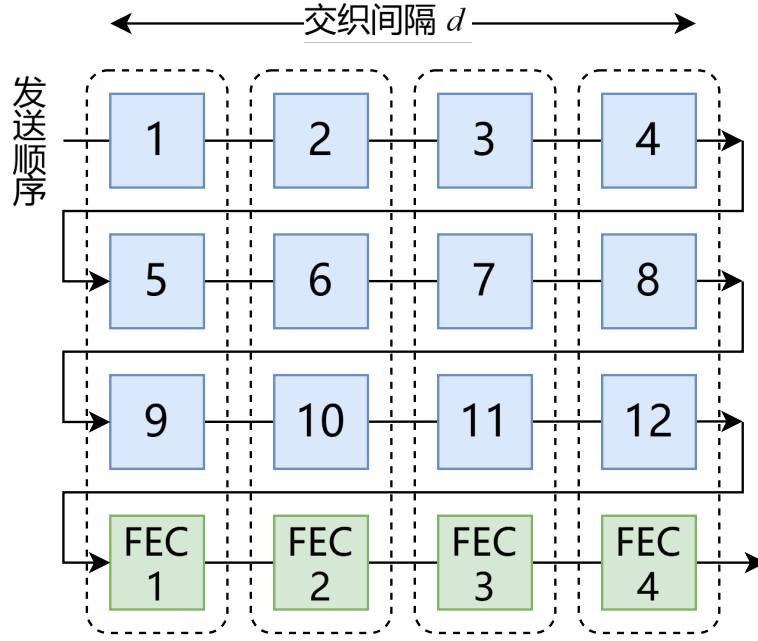


图 2.5 交织编码示意

Liu 等人^[23]提出了一种在开放光通信场景下利用交织应对连续丢包的方法。该方法通过马尔科夫链对网络的状态进行建模，通过测量信道的“开启时间”和“中断时间”，估计信道的连续丢包长度和数据包接收时间特性，同时综合考虑缓冲区分大小、FEC 恢复丢包数量上限等因素，联合优化交织参数和 FEC 参数。Yin 等人^[24]将 FEC 交织编码应用于多跳无线网络的物联网场景中，在每一跳的转发设备上利用上游设备的 FEC 编码对发送内容进行恢复后再重新编码发送至下游。作者提出了一种利用力学中势能概念衡量交织性能的方法，同时提出了一种基于此指标对交织参数进行优化的算法。

分组冗余码通过设计比简单复制更复杂的冗余信息计算和丢失包解算机制，允许通过调整参数动态变化冗余率以适应不同丢包率的网络环境。结合交织技术，可以有效地应对真实网络中存在的连续丢包等特性，得到了广泛的应用。

2.2.3 流式冗余码（Streaming 码）

XOR、R-S 等分组码结合交织已经能较好地应对网络中的丢包问题，但是这些编码仍旧不能满足一些实时性需求高的应用。如图2.6，由于分组码的冗余包通常是通过对所有组内的数据包进行计算得到，因此冗余信息必须在所有数据包已经发出后才能够计算并在网络中发出，这导致如果接收端在接收数据包时如果检测到了丢包且需要利用冗余信息进行恢复，为了保证数据包按发送顺序连续交付至上层应用，接收端通常需要暂停后续数据的解码输出，直至对应的冗余包到达并完成恢复。由此产生的恢复等待时间会显著增加端到端时延。对于实时

性要求较高的应用，即使最终能够恢复出丢失数据，其对应的视频帧或音频数据也可能已经错过播放时限，从而无法有效改善用户体验。

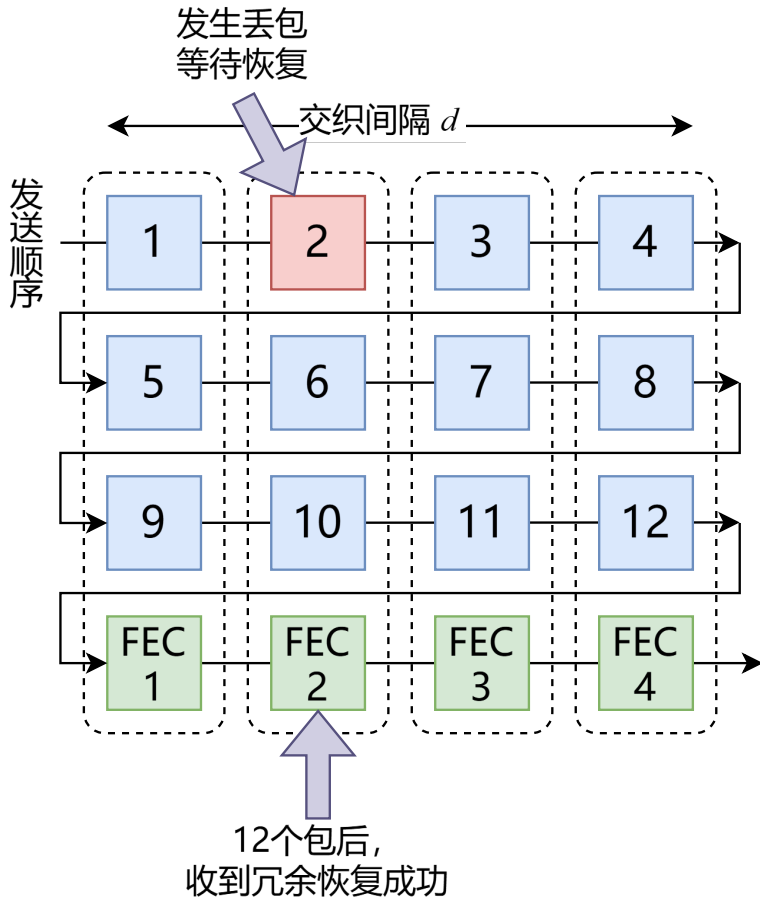


图 2.6 分组码需要暂停解码输出等待冗余包到来才能恢复丢包并继续解码过程

基于此，Martinian 等人提出了流式编码（Streaming Code）^[25]。与传统基于固定编码块的 FEC 不同，流式编码采用跨时间窗口的卷积式编码结构，将同一时刻数据包的冗余信息分散嵌入到后续多个时刻发送的数据包中，从而在时间维度上持续提供保护。例如，某一时刻 $t = 0$ 发送的数据，其相关冗余不仅存在于当前数据包中，还会被逐步附加到 $t = 1, 2, 3$ 等后续时刻发送的数据包内。当 $t = 0$ 时刻的数据包发生丢失时，接收端可以利用后续若干时刻收到的数据包逐步恢复其内容，并在预设的有限解码时延内完成恢复，而无需等待整个编码块全部发送完成。该机制能够在保证连续突发丢包恢复能力的同时，显著降低恢复延迟，更适用于实时流媒体等低时延传输场景。Martinian 等人进一步证明了，在给定码率与突发丢包长度条件下，流式编码能够达到理论上的最小恢复时延下界。已经有一些研究工作^[26,27]尝试将流式编码应用于实时音视频通信领域，获得了一定的效果提升。

前向纠错技术逐渐从早期基于简单重复发送的冗余机制，发展到结合有限域运算的分组纠删码，并进一步演化出结合交织技术与时间维度编码的低时延流式

编码结构。不同类型的 FEC 机制在冗余开销、连续丢包恢复能力以及恢复时延等方面各有侧重：简单复制具有实现简单、恢复迅速的特点，但冗余效率较低；XOR 码与 R-S 码等分组码能够显著提高冗余恢复效率，但通常需要等待整个编码组完成后才能进行恢复；而流式编码则通过跨时间窗口的连续冗余保护，在保证突发丢包恢复能力的同时进一步降低了解码等待时延，更适用于实时音视频通信等低时延场景。因此，如何在冗余率、恢复能力与恢复时延之间取得平衡，已经成为当前链路质量优化与实时媒体传输中的重要研究方向。

尽管现有 FEC 技术已经能够有效提升低质量网络环境中的数据恢复能力，但大多数研究主要关注编码结构本身的恢复性能、冗余效率以及恢复时延等问题，通常默认数据传输路径已经固定，而较少进一步考虑不同网络链路之间的质量差异与成本差异。在跨域云网络场景下，不同链路可能同时具有显著不同的传输性能与租赁成本，如何结合链路状态动态选择冗余保护策略，并进一步联合流量调度共同优化整体传输性能与网络成本，仍然是值得进一步研究的问题。

2.3 软件定义网络与网络调度

软件定义网络（Software defined networking, SDN）指的是将网络中各个转发设备的数据平面与控制平面解耦，集中进行控制的网络。SDN 网络大大简化了网络的管理和控制流程。对于跨域云网络及其中部署的虚拟网络，尽管有部分的网络设备由 SDN 统一控制，但是各个设备间的跨域互联通常仍仍由传统的网络设备提供连接，形成了混合形软件定义网络（hybrid SDN network），如图2.7^[28]。

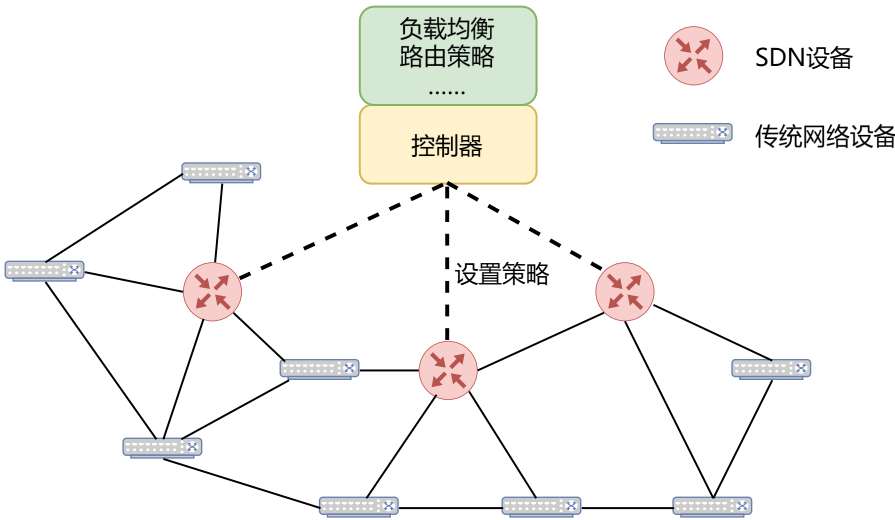


图 2.7 混合 SDN 网络

随着云计算与实时互联网应用的发展，现代云网络中的跨域流量规模持续增

长，用户对于传输质量与服务稳定性的要求也不断提高。在跨地域云网络场景中，不同节点之间通常并非只存在单一的物理连接路径，而是可能存在多种不同质量、不同价格的传输链路。例如，许多云服务商同时提供公网链路互联与专线链路互联^[29,30]。其中，专线链路通常具有更稳定的传输性能、更低的丢包率与时延，但部署成本与使用成本较高；而公网链路虽然成本较低，却容易受到网络拥塞、跨域路由波动等因素影响，出现高丢包、时延抖动等问题^[1]。与此同时，链路质量与网络负载往往还会随着时间动态变化，使得不同链路在不同时间段内呈现出不同的性能特征。

在这种场景下，仅依赖传统网络静态地选择固定传输路径，难以同时满足不同业务对吞吐、时延、可靠性以及成本控制等方面的需求。相比之下，基于 SDN 的覆盖网络能够通过集中控制的方式，对网络中的链路状态、节点负载以及业务需求进行统一管理，并动态地对流量进行调度与路径选择，从而更灵活地利用不同网络资源，在传输性能、可靠性与成本之间取得平衡。因此，如何基于覆盖网络与 SDN 架构实现高效的网络调度，逐渐成为跨域云网络与实时媒体传输领域的重要研究方向。

最初的一些工作主要集中在覆盖网络的建立与路由绕行方面。覆盖网络的概念最初由 Anderson 等人提出^[31]，该工作中介绍了 RON 这一实验性覆盖网络。该工作提出了将公网中并不直接相连的一些节点重新抽象为一个覆盖网络中的相邻节点，称为 RON 节点。各个 RON 节点之间通过公网建立连接，形成 Overlay 网络中的虚拟链路。除了转发功能，RON 节点间还可以通过主动探测的方式，对建立虚拟链路所依靠的物理链路质量进行实时测量，并将测量结果汇总至控制器。当客户端希望通过 RON 网络进行连接时，控制器将综合考虑覆盖网络中所有可用的连接的质量，选择最符合客户端的传输需求的链路对流量进行调度。Roy 等人^[32]根据路由可靠性和 TCP 性能建立指标，并以此为标准优化转发节点的选择。作者提供了多种不同的算法，包括贪心算法、随机算法及两者的混合算法，分别对两种指标存在不同的侧重，供用户灵活根据需要选择。

之后的一些研究进一步研究了通过流量调度实现对资源利用的优化。CRONets^[33]提出了利用云网络服务商提供的虚拟机网络网络链路建立覆盖网络的方案，并利用多路径 TCP 在覆盖网络节点间提升性能。在覆盖网络资源规模进一步扩大的背景下，研究者开始关注如何通过集中式调度提升资源利用效率。B4^[34]则提出了通过流量调度和流量工程，有效分配不同链路的负载以最大化链路使用率的方法。BDS^[35]使用统一的中央控制器持续监控不同覆盖网络间节点的可用资源，动态调度传输路径以实现链路的带宽的充分利用。除了单纯追求更高链路利用

率，一些研究开始进一步联合考虑性能与部署成本之间的平衡。Skyplane^[36]则观察到云网络提供商不同地域资源的定价差异，将追求文件传输最大吞吐量与追求更低租赁成本建模为一个线性优化问题，给定其中一个指标的限制，利用算法最优化另一个指标。Titan^[1]则针对持续运行的流媒体服务将租用云网络互联资源成本纳入考量，在维持用户体验在一定水平之上的前提下，动态调度流量与计算资源，降低整体的网络部署成本。与此同时，随着实时媒体等对服务连续性要求更高的应用出现，部分工作开始关注链路状态变化时的快速恢复能力。Troia 等人^[37]利用 eBPF 技术实时检测各个覆盖网络节点的传输状态，并在检测到链路拥塞或其他链路质量变化事件时，快速重新触发流量调度算法以维持高质量连接。XRON^[2]则同时结合链路成本优化、资源利用与快速恢复，主动探测可用的公网链路和专线链路质量，结合未来用户流量需求预测，持续计算和更新成本最佳的流量调度策略。为保持所承载音视频通话的服务质量，计算多个备用调度方案以确保故障条件的快速恢复。

总体而言，现有覆盖网络与 SDN 调度相关研究主要通过动态路径选择、流量工程以及资源调度等方式，尽可能规避低质量链路对服务质量的影响。然而，在跨域云网络场景下，公网链路虽然存在较大的性能波动，但同时具有覆盖范围广、部署灵活以及成本较低等优势。现有工作通常倾向于在网络质量下降时将流量迁移至专线或其他高质量链路，而较少进一步考虑结合冗余编码等机制，提升对低质量公网的利用率，在保证服务质量的同时进一步降低整体网络部署成本。因此，如何联合考虑链路调度与冗余编码策略，充分利用低成本公网资源，成为当前覆盖网络优化中的一个值得关注的问题。

2.4 现有系统问题分析

前文分别介绍了链路质量优化和覆盖网络调度两类相关工作。前者通常利用 FEC 等机制直接提升低质量网络中的丢包恢复能力，后者则利用 SDN 和覆盖网络能力在多条候选链路之间进行路径选择和流量调度。两类方法分别从“修复链路”和“选择链路”的角度改善传输性能，但在跨域云网络场景下，仍然存在与真实公网结构不匹配的问题。本节结合本文的测量观察，对现有方法的局限性进行进一步分析，并引出第三章的系统设计目标。

2.4.1 网络路径的分段质量差异造成冗余编码效率低

现有 FEC 相关工作大多将发送端到接收端之间的网络路径视为一条整体链路，并基于端到端测得的丢包率、突发丢包长度或应用层质量指标选择冗余策略。

这种建模方式适用于端到端路径不可控的普通互联网应用，但并不完全适用于跨域云网络。云网络中的一条端到端连接通常由多个云节点接力转发形成，逻辑上的端到端路径可以拆分为若干相邻云节点之间的链路片段。不同片段的物理距离、跨运营商情况和跨境路由情况不同，其丢包率和时延抖动可能存在显著差异。

本文的测量显示，端到端质量下降并不意味着路径上的每一段公网链路都同样低质。相反，许多域内或近距离公网链路的质量较高，丢包率低且时延稳定；端到端丢包往往集中出现在少数跨域或跨境链路片段上。图2.8展示了这一现象：在节点分布于全球多地的真实云网络中，不同链路片段的质量差异明显，低质量片段通常只占有互联路径的一部分。

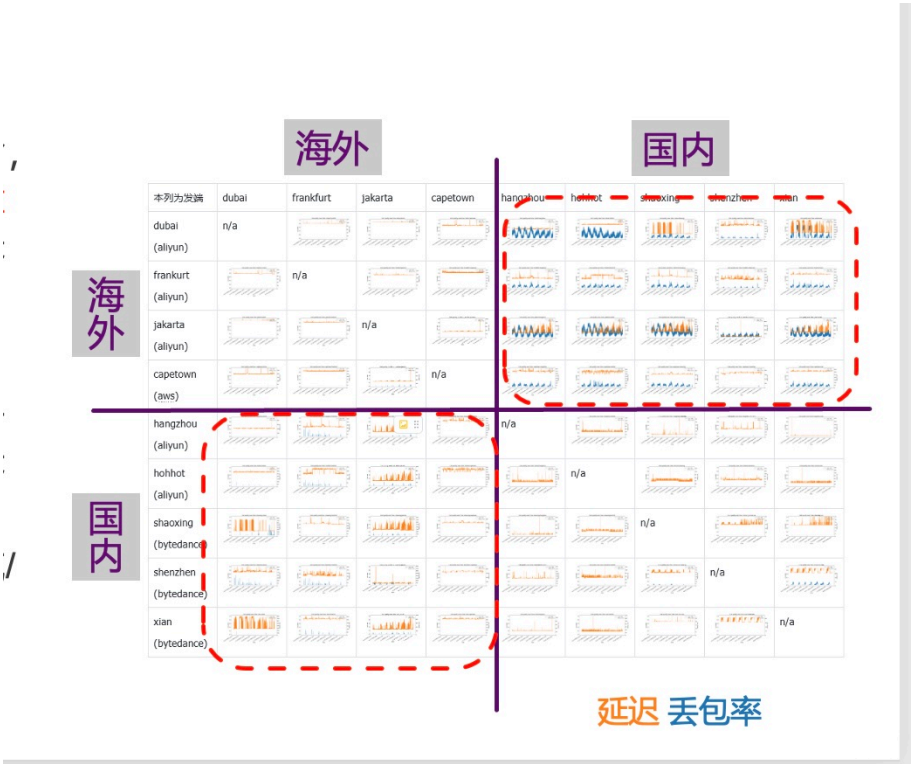


图 2.8 跨域云网络路径中的链路分段质量差异

如果仍然按照端到端链路整体设计 FEC 策略，系统只能根据整条路径的丢包情况加入冗余。一方面，为了恢复由少数差链路片段造成的丢包，系统会在整条端到端路径上承担冗余开销，使得质量良好的链路片段也传输额外冗余包；另一方面，端到端恢复需要等待冗余信息跨越整条路径到达接收端，恢复延迟由完整端到端时延决定，而不是由实际发生丢包的短链路片段决定。因此，现有端到端 FEC 方法没有充分利用云网络路径可分段、节点可控的特点，难以在恢复能力、额外带宽和恢复延迟之间取得更细粒度的平衡。

由此可见，跨域云网络中的链路质量优化不应只关注端到端路径的整体丢包

率，而应识别丢包集中发生的低质量链路片段，并仅在这些片段上加入冗余编码与丢包恢复。这样既能够避免在高质量片段上浪费冗余带宽，也能够将丢包恢复限制在更短的链路范围内。

2.4.2 现有链路调度方法忽略了低质量公网链路的可修复性

覆盖网络与 SDN 调度类工作通常将不同链路视为质量和成本不同的候选资源，并根据实时链路状态、业务需求和成本约束选择转发路径。在跨域云网络中，这类方法常见的策略是：当公网链路质量较好时，将部分流量调度到公网以降低成本；当公网链路质量下降时，则将流量迁移到专线或其他高质量链路，以维持用户体验。这种方法能够避免直接在差链路上传输，但也默认低质量公网链路只能被规避，而不能被修复。

这一假设在成本优化场景下会带来明显限制。实时音视频等业务的用户访问量具有明显的时间规律，用户高峰时段通常也是区域网络负载较高、跨域公网质量更容易下降的时段。即系统最希望利用低成本公网分担流量的时间，往往正是公网链路质量最不稳定的时间。图2.9展示了用户使用量与公网链路质量变化之间的关系：公网链路丢包或时延抖动升高的时段，与用户流量高峰存在较强重合。

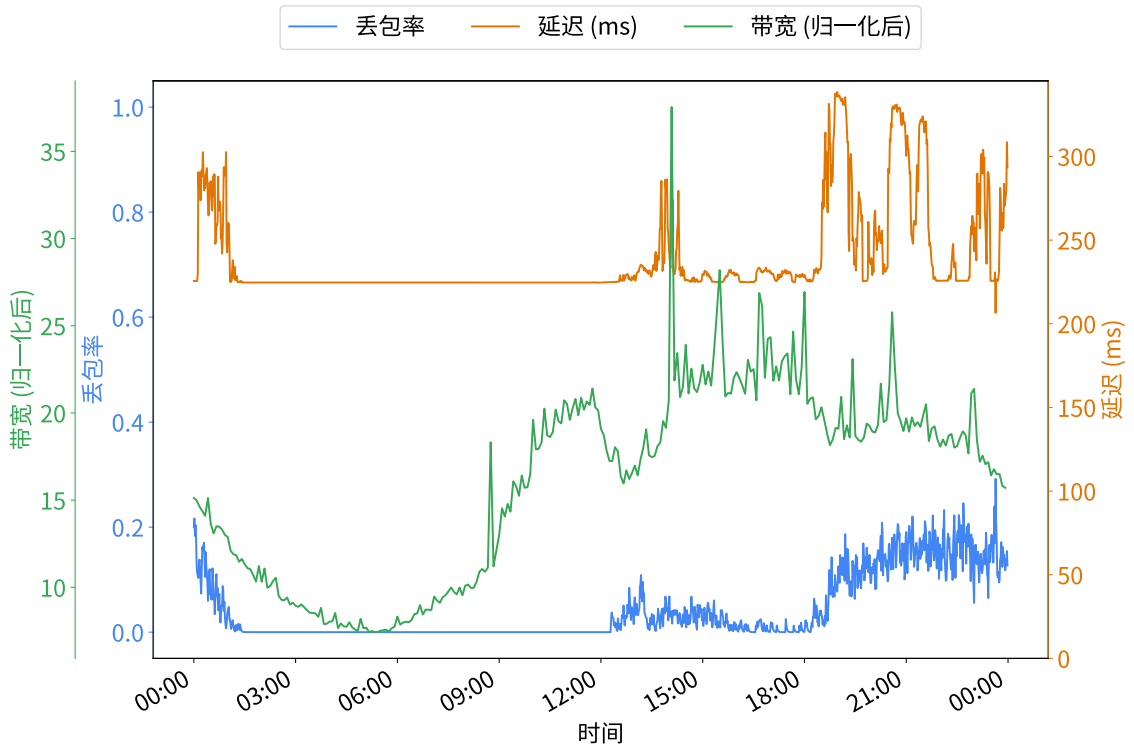


图 2.9 用户流量高峰与公网链路质量下降时段的重合

在这种情况下，单纯依赖调度规避差链路会削弱公网的成本优势。当用户需

求较低、公网质量较好时，即使将流量调度到公网，能够节省的专线流量也有限；而当用户需求较高、需要承载大量流量时，公网质量下降又迫使系统回退到专线，导致整体成本仍然由高价链路主导。因此，现有调度方法虽然能够在给定链路质量条件下做出较优路径选择，但没有进一步考虑通过 FEC 等机制修复低质量公网链路，使其在高需求时段也能承担更多流量。

综上所述，现有 FEC 方法和现有调度方法分别存在互补但尚未结合的局限：FEC 方法具备修复丢包的能力，却通常忽略跨域云网络路径的分段质量差异；调度方法具备覆盖网络视角和集中控制能力，却通常将低质量公网链路视为需要规避的资源，而没有考虑其可修复性。本文第三章基于这一观察，提出一种面向跨域公网链路的传输优化方法：在全公网互联的前提下，以链路片段为粒度识别和修复低质量公网链路，通过交织 FEC 和自适应参数调整提升差链路质量，从而在保证实时业务体验的同时降低对专线资源的依赖。

第 3 章 跨域云网络传输性能提升研究

3.1 设计目标与总体思路

前两章的分析表明，跨域云网络中的核心矛盾在于：专线链路能够提供稳定的传输质量，但使用成本较高；公网链路成本低廉、覆盖灵活，却在部分跨域片段上存在明显的丢包和抖动。若继续沿用“公网质量良好时走公网、质量恶化时切回专线”的以资源调度为中心的策略，系统在用户需求高峰期仍将大量依赖专线，成本优化空间十分有限，而已有的前向纠错编码等链路质量优化类工作则又将传输链路当作不可改变的低质量链路，只进行端到端的冗余添加与丢包恢复，没有考虑网络传输过程中不同分段的网络质量不同，需要不同程度的丢包恢复。因此，本文提出，依靠云网络配置灵活、路由选择多样的特点，通过冗余编码对云网络中的部分低质量链路进行针对性的质量修复，以达到传输性能与运营成本的平衡。

在可变的用户包大小下设计 FEC 编码方案-> 使用 block code

在网络质量动态变化的情况下，自适应地选择 FEC 参数-> 使用 markov 链估计网络参数

FEC 解码时出现 burst 造成速率测量不准，干扰 CCA 决策-> 使用 PI 控制器稳定发送速率

3.2 系统总体架构

系统由一个中心控制器（Coordinator）和多个部署在不同地域的转发节点（Node）组成，如图3.1所示。中心控制器维护节点和连接状态，为端到端连接分配流标识，向相关节点下发转发表项，并根据解码端上报的丢包统计调整低质量链路片段上的 FEC 编码参数。转发节点负责数据面的实际转发，在本地根据控制器下发的配置，对不同流分别执行普通转发、FEC 编码或 FEC 解码。

图片内容：系统总体架构。上方 Coordinator 通过 TCP 连接（控制面，虚线）连接多个 Node。Node 之间通过 UDP 隧道互联（数据面，实线）。标注一条端到端路径：用户 A → TUN → Node1（FEC 编码器）→ UDP 隧道 → Node2（FEC 解码器 + Pacer）→ TUN → 用户 B。Coordinator 标注“FEC 参数计算”，数据面上标注编码后数据包传输，控制面上标注丢包统计上报与参数下发。

图 3.1 系统总体架构

每个转发节点上为每条端到端连接创建独立的 TUN 虚拟网络设备，用户应用

只感知到一条普通 IP 链路，而不需要感知底层覆盖网络和 FEC 机制。节点之间通过 UDP 隧道传输数据包，每个数据包的头部携带 `flow_id` 以标识所属的数据流。系统为每个 `flow_id` 创建独立的处理线程，线程中包含入方向解码器和出方向编码器：当流量进入一个需要修复的低质量链路片段时，出方向编码器添加 FEC 冗余；当流量离开该链路片段时，入方向解码器根据收到的数据包和冗余包恢复丢包。对于不需要修复的链路片段，编码器和解码器退化为普通转发逻辑。

3.3 设计挑战

在“全公网承载、差链路修复”的总体目标下，系统设计需要进一步回答三个具体问题。首先，FEC 机制必须能够插入通用覆盖网络转发路径，而不能改变用户报文本身的大小假设；其次，低质量公网链路的丢包程度会随时间变化，系统需要决定何时启用冗余以及使用多少冗余；最后，FEC 解码按组恢复数据会改变报文交付节奏，如果不加处理，可能反过来干扰端系统的拥塞控制。由此，本文需要解决以下三个核心设计挑战。

挑战一：如何在用户包大小可变的场景下设计链路片段级 FEC 编码方案。系统作为通用转发平台，承载的上层应用可能产生任意大小的数据包。当用户数据包接近 MTU 时，若 FEC 编码产生的冗余信息追加在用户数据包内一同发送，则封装后的数据包可能超出 MTU，导致 IP 层分片或传输失败。因此，FEC 编码方案必须能够在不影响用户数据包大小的情况下独立添加冗余信息。同时，由于本文只在低质量公网片段上进行修复，编码方案还需要在单个链路片段上有效应对连续突发丢包，而不是依赖端到端重传。本文采用交织 XOR 分组编码方案应对此挑战（详见第3.4节）。

挑战二：如何判断差链路所需的冗余强度并自适应地选择编码参数。全公网方案不能简单地对所有链路长期使用高冗余率，否则低成本公网节省下来的费用会被额外流量开销抵消。系统承载的应用中又包含实时音视频流媒体等对延迟敏感的业务，FEC 编码引入的冗余包不仅占用额外带宽，也会引入解码等待延迟。因此，系统需要根据链路片段上的实时丢包统计，在丢包恢复能力、额外带宽开销和恢复延迟之间取得平衡。由于公网链路的丢包率与丢包模式随时间动态变化，固定的编码参数无法同时适应不同质量状态的链路。本文通过建立丢包信道模型并据此进行约束搜索来解决此挑战（详见第3.5节）。

挑战三：如何消除 FEC 解码按组突发输出对拥塞控制算法的干扰。FEC 解码器按编码组为单位批量恢复和交付数据包。当一个编码组恢复完成后，组内的所有数据包被一次性连续交付给上层应用。这种突发式的输出模式会使得上游的拥

塞控制算法收到密集的 ACK 确认包，从而错误地估计链路可用带宽，引发发送速率的震荡。速率震荡不仅影响应用的传输性能，还会使 FEC 编码器的输入节奏不稳定，进一步干扰吞吐量统计和参数估计的准确性。本文在解码端设计了基于 PI 控制器的输出速率控制器来消除此问题（详见第3.6节）。

3.4 交织 XOR 前向纠错编码设计

本节针对挑战一，介绍链路片段级交织 FEC 编码方案的设计。核心需求是：FEC 冗余信息必须以独立包的形式发送，不嵌入用户数据包内部，从而避免影响用户数据包的大小；同时编码方案需能在低质量公网片段上有效应对连续突发丢包。

如第二章所述，现有的前向纠错编码方案主要包括简单复制冗余、XOR 码、R-S 码以及流式编码等。本文选择基于 XOR 运算的分组编码结合交织技术作为 FEC 编码方案，其核心考量如下。

分组码天然适应可变包大小的场景。在分组码中，冗余包是独立于数据包的单独包：编码器先将用户数据包逐个作为数据包发出，在编码组填满或超时后再生成独立的冗余包并追加发送。由于冗余信息不嵌入在用户数据包内部，用户数据包的大小不受 FEC 编码的影响，因此即使数据包本身已接近 MTU，也不会因为 FEC 而产生封装溢出的问题。相比之下，流式编码将同一时刻的冗余信息分散嵌入后续多个数据包中，要求在数据包内部预留冗余空间，在用户包大小不可控的通用转发场景下难以适用。分组码的另一个优势是边界清晰：编码器和解码器可以部署在某一段公网链路的两端，只修复该链路片段，而不会要求整条端到端路径都采用同一种传输机制。

在多种分组码中，本文选择 XOR 编码而非 R-S 码，理由是结合交织技术的 XOR 编码已足以应对公网链路上观察到的丢包模式。根据第一章的分析，公网链路的主要丢包特征是偶发的孤立丢包和有限长度的连续突发丢包。交织技术将连续的突发丢包分散到不同的恢复列中，使得每列至多丢失一个数据包，恰好匹配 XOR 编码“每列可恢复一个丢包”的能力。同时，XOR 编码的编码和解码均只需要按位异或运算，无需有限域上的矩阵运算，计算开销极低，适合高吞吐量的转发场景。

具体地，本文提出的交织 FEC 编码将数据包组织为一个二维矩阵结构，如图3.2所示。设交织深度为 d ，保护包数为 k ，则每个编码组包含 $d \times k$ 个数据包和 d 个冗余包。矩阵共有 d 列、 $k+1$ 行，其中前 k 行为数据包，第 $k+1$ 行为冗余包。每个数据包在矩阵中的位置由其组内序列号唯一确定：对于序列号为 s 的数据包，

其所在列号为 $s \bmod d$ ，行号为 $\lfloor s/d \rfloor$ 。每个冗余包通过对同一列中所有数据包进行按位异或运算得到。交织技术的关键优势在于：当网络上发生长度不超过 d 的连续丢包时，由于相邻数据包被分配到不同的列中，这些丢失的数据包被分散到最多 d 个不同的列中，每个列至多丢失一个数据包，因此每个列都可以独立恢复。

图片内容：交织编码矩阵示意。以 $d = 4, k = 3$ 为例，矩阵为 4 列、4 行（3 行数据 + 1 行冗余）。数据包标注为 $D_{0,0}, D_{1,0}, D_{2,0}, D_{3,0}, D_{0,1}, D_{1,1}, \dots, D_{3,2}$ ，冗余包标注为 F_0, F_1, F_2, F_3 。其中 $F_j = D_{j,0} \oplus D_{j,1} \oplus D_{j,2}$ 。下方用箭头展示实际发送顺序： $D_{0,0}, D_{1,0}, D_{2,0}, D_{3,0}, D_{0,1}, D_{1,1}, \dots, F_0, F_1, F_2, F_3$ 。右侧用虚线框标注一个连续丢包的例子：假设连续丢失 $D_{1,0}$ 和 $D_{2,0}$ ，由于交织深度 $d = 4$ ，两个丢包分别落在第 1 列和第 2 列，每列恰好只丢失一个包，各自可用 F_1 和 F_2 恢复。

图 3.2 交织编码矩阵结构示意图（ $d = 4, k = 3$ ）

编码参数（ d 和 k ）可以在编码组之间动态切换，编码器在结束当前组时加载最新收到的参数配置，下一编码组立即使用新参数。

3.5 基于丢包统计的自适应参数调整

本节针对挑战二，介绍如何根据实时丢包统计自适应地选择编码参数。如前所述，本文并不希望在所有公网链路上持续加入固定冗余，而是希望仅在差链路上使用足够但不过量的冗余。因此，FEC 编码参数需要同时满足流媒体应用的延迟需求、控制额外带宽开销，并适应链路质量的动态变化。本文的解决思路是：首先为公网链路的丢包行为建立一个数学模型，然后从接收端的丢包观测量中估计模型参数，最后在延迟与残余丢包率的约束下搜索最优编码参数。

3.5.1 丢包信道模型

根据第一章中对公网链路丢包特性的分析，公网链路上的丢包行为可以大致分为两类：一类是偶发的孤立丢包，丢失一个包后链路随即恢复正常；另一类是连续的突发丢包，由于链路拥塞等原因连续丢失多个数据包。基于这一观察，本文提出一个简化的三状态丢包信道模型，如图3.3所示。模型定义三个状态： S_0 （正常状态，当前数据包正常接收）、 S_1 （孤立丢包状态，丢失一个包后立即恢复）、 S_2 （突发丢包状态，连续丢失多个包）。模型通过三个参数 p_{21} （孤立丢包触发概率）、 p_{23} （突发丢包触发概率）和 p_{33} （突发延续概率）完整描述链路的丢包行为。

图片内容：三状态马尔科夫丢包模型状态转移图。三个状态： S_0 （正常接收）， S_1 （孤立丢包）， S_2 （突发丢包）。转移概率标注。

图 3.3 三状态丢包信道模型

3.5.2 参数估计与编码参数搜索

解码端通过全局序列号间隔检测丢包，维护一个滑动窗口统计近期的丢包事件，并将统计量定期上报至中心控制器。中心控制器收到统计量后，首先根据丢包事件的突发长度分布估计三状态模型的参数 p_{21} 、 p_{23} 和 p_{33} ，进而得到丢包事件的总发生率 $\lambda = p_{21} + p_{23}$ 。

在估计出模型参数后，中心控制器在给定的性能约束下搜索最优的编码参数 (d, k) 。算法考虑交织深度 $d \in \{1, 2, 3, 4\}$ 的候选值，对于每个 d ，确定满足以下两个约束的最大保护包数 k ：

1. **延迟约束：**编码组引入的额外等待延迟不应超过阈值，以满足流媒体应用的实时性需求。
2. **残余丢包率约束：**编码后未被恢复的随机丢包率应低于阈值，确保应用层的丢包率在可接受范围内。

最终在所有可行的 (d, k) 组合中，选择 k 值最大的组合作为最优编码参数，以提供最强的冗余保护。上述参数调整过程构成了一个完整的反馈闭环：解码端持续检测丢包并上报统计信息，中心控制器计算最优参数并下发至编码端，编码端在下一编码组生效新参数。

3.6 解码端输出速率控制设计

本节针对挑战三，介绍解码端的输出速率控制器（Pacer）设计。如前所述，FEC 解码器按编码组为单位批量交付数据包，这种突发输出会使上游 CCA 收到密集的 ACK，导致错误的带宽估计和速率震荡。本文通过在解码端引入 PI 速率控制器，将突发输出平滑为匀速流来解决此问题，其控制模型如图3.4所示。

图片内容：Pacer 控制模型框图。左侧“数据包输入”（来自 FEC 解码器，标注“突发式到达”）→ 进入“输出缓冲区”（buffer）→ 受控输出 paced 数据包（标注“匀速输出”）。缓冲区下方有一个“PI 控制器”模块，输入为 $\text{error} = \text{depth} - \text{target}$ ，输出为 pacing rate 。

图 3.4 Pacer 控制模型

Pacer 的核心是一个基于缓冲区深度的 PI（比例-积分）控制器。FEC 解码器恢复后的数据包首先进入一个输出缓冲区，Pacer 根据缓冲区的当前深度与目标深度的偏差计算出发包速率，并按照该速率匀速地从缓冲区中取出数据包交付给上层应用。比例项提供快速的瞬态响应：缓冲区超过目标时加快发包消耗积压，低于目标时减慢发包等待新数据。积分项消除稳态误差：即使数据包到达速率发生变化，积分项也能逐步学习新的稳态速率，确保缓冲区深度收敛到目标值。启动时，Pacer 先以直通模式运行并测量实际的到达速率，待积累足够的观测后切换到 PI 控制模式，切换时将测量速率编码为积分项的初始值以实现平滑过渡。

Pacer 与 FEC 编码的自适应参数调整形成了协同关系。Pacer 将突发输出平滑为匀速流，使得 CCA 能够基于稳定的 ACK 反馈正确估计带宽，维持稳定的发送速率。稳定的发送速率使得 FEC 编码器以稳定的节奏填满编码组，进而使吞吐量统计值更加准确，提升了参数调整的准确性。

3.7 本章小结

本章围绕“全公网承载、差链路修复”的总体思路，介绍了本文提出的跨域公网链路传输优化方法。首先明确了本文不依赖专线兜底，而是在公网覆盖网络中针对低质量链路片段进行 FEC 修复；随后介绍了集中控制、分布转发的系统总体架构，并将总体目标细化为三个设计挑战。针对通用转发场景下可变包大小与 MTU 约束的挑战，本文选择了交织 XOR 分组编码方案，冗余包作为独立包发送不影响用户数据包大小，交织技术则将连续丢包分散到不同的恢复列中。针对满足实时应用延迟需求的同时自适应选择编码参数的挑战，本文提出了三状态丢包信道模型，介绍了从丢包观测量估计模型参数以及在延迟与残余丢包率约束下搜索最优编码参数的方法。最后，针对 FEC 解码突发输出干扰拥塞控制算法的挑战，本文设计了解码端 PI 速率控制器，将突发输出平滑为匀速流，并与 FEC 参数调整机制形成协同。

第 4 章 实验验证与分析

参考文献

- [1] Kataria B, Lnu P, Bothra R, et al. Saving private wan: Using internet paths to offload wan traffic in conferencing services[J]. Proceedings of the ACM on Networking, 2024, 2(CoNEXT4): 1-22.
- [2] Wu B, Qian K, Li B, et al. Xron: A hybrid elastic cloud overlay network for video conferencing at planetary scale[C]//Proceedings of the ACM SIGCOMM 2023 Conference. 2023: 696-709.
- [3] Bolot J C, Fosse-Parisis S, Towsley D. Adaptive fec-based error control for internet telephony[C/OL]//IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320): Vol. 3. 1999: 1453-1460 vol.3. DOI: 10.1109/INFCOM.1999.752166.
- [4] Huang T Y, Huang P, Chen K T, et al. Could skype be more satisfying? a qoe-centric study of the fec mechanism in an internet-scale voip system[J]. IEEE Network, 2010, 24(2): 42-48.
- [5] Holmer S, Shemer M, Paniconi M. Handling packet loss in webrtc[C]//2013 IEEE international conference on image processing. IEEE, 2013: 1860-1864.
- [6] Azodolmolky S, Wieder P, Yahyapour R. Cloud computing networking: Challenges and opportunities for innovations[J]. IEEE Communications Magazine, 2013, 51(7): 54-62.
- [7] Luong N C, Wang P, Niyato D, et al. Resource management in cloud networking using economic analysis and pricing models: A survey[J]. IEEE Communications Surveys & Tutorials, 2017, 19(2): 954-1001.
- [8] Mogul J C, Popa L. What we talk about when we talk about cloud network performance[J]. ACM SIGCOMM Computer Communication Review, 2012, 42(5): 44-48.
- [9] Mahalingam M, Dutt D, Duda K, et al. Request for comments: No. 7348 Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks[M/OL]. RFC Editor, 2014. <https://www.rfc-editor.org/info/rfc7348>. DOI: 10.17487/RFC7348.
- [10] Garg P, Wang Y S. Request for comments: No. 7637 NVGRE: Network Virtualization Using Generic Routing Encapsulation[M/OL]. RFC Editor, 2015. <https://www.rfc-editor.org/info/rfc7637>. DOI: 10.17487/RFC7637.
- [11] Gross J, Ganga I, Sridhar T. Request for comments: No. 8926 Geneve: Generic Network Virtualization Encapsulation[M/OL]. RFC Editor, 2020. <https://www.rfc-editor.org/info/rfc8926>. DOI: 10.17487/RFC8926.
- [12] Microsoft. Network virtualization using generic routing encapsulation (nvgre) task offload [EB/OL]. Microsoft Learn, 2023[2026-05-13]. <https://learn.microsoft.com/en-us/windows-hardware/drivers/network/network-virtualization-using-generic-routing-encapsulation--nvgre--task-offload>.
- [13] OVN Project. General — ovn documentation[EB/OL]. OVN Project, 2026[2026-05-13]. <https://docs.ovn.org/en/latest/faq/general.html>.

- [14] VMware. Nsx-t: Routing where you need it (multi-hypervisor & multi-cloud)[EB/OL]. VMware, 2017[2026-05-13]. <https://blogs.vmware.com/networkvirtualization/2017/09/nsx-t-routing-where-you-need-it.html/>.
- [15] Eddy W. Request for comments: No. 9293 Transmission Control Protocol (TCP)[M/OL]. RFC Editor, 2022. <https://www.rfc-editor.org/info/rfc9293>. DOI: 10.17487/RFC9293.
- [16] Gandikota V R, Tamma B R, Murthy C S R. Adaptive fec-based packet loss resilience scheme for supporting voice communication over ad hoc wireless networks[J/OL]. IEEE Transactions on Mobile Computing, 2008, 7(10): 1184-1199. DOI: 10.1109/TMC.2008.42.
- [17] Reed I S, Solomon G. Polynomial codes over certain finite fields[J]. Journal of the society for industrial and applied mathematics, 1960, 8(2): 300-304.
- [18] Lin C H, Shieh C K, Hwang W S. An access point-based fec mechanism for video transmission over wireless lans[J]. IEEE Transactions on Multimedia, 2012, 15(1): 195-206.
- [19] Shih C H, Kuo C I, Chou Y K. Frame-based forward error correction using content-dependent coding for video streaming applications[J]. Computer Networks, 2016, 105: 89-98.
- [20] Xiao J, Tillo T, Lin C, et al. Dynamic sub-gop forward error correction code for real-time video applications[J]. IEEE Transactions on Multimedia, 2012, 14(4): 1298-1308.
- [21] Yang X, Zhu C, Li Z, et al. Unequal loss protection for robust transmission of motion compensated video over the internet[J]. Signal Processing: Image Communication, 2003, 18(3): 157-167.
- [22] Kurdoglu E, Liu Y, Wang Y. Perceptual quality maximization for video calls with packet losses by optimizing fec, frame rate, and quantization[J]. IEEE Transactions on Multimedia, 2017, 20(7): 1876-1887.
- [23] Liu J, Zhang X, Blow K, et al. Performance analysis of packet layer fec codes and interleaving in fso channels[J]. Iet Communications, 2017, 11(13): 2042-2048.
- [24] Yin H H, Ng K H, Zhong A Z, et al. Intrablock interleaving for batched network coding with blockwise adaptive recoding[J]. IEEE Journal on Selected Areas in Information Theory, 2021, 2(4): 1135-1149.
- [25] Martinian E, Sundberg C E. Burst erasure correction codes with low decoding delay[J]. IEEE Transactions on Information theory, 2004, 50(10): 2494-2502.
- [26] Emara S, Fong S L, Li B, et al. Low-latency network-adaptive error control for interactive streaming[J]. IEEE Transactions on Multimedia, 2021, 24: 1691-1706.
- [27] Rudow M, Yan F Y, Kumar A, et al. Tambur: Efficient loss recovery for videoconferencing via streaming codes[C]//20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23). 2023: 953-971.
- [28] Amin R, Reisslein M, Shah N. Hybrid sdn networks: A survey of existing approaches[J]. IEEE Communications Surveys & Tutorials, 2018, 20(4): 3259-3306.
- [29] Microsoft. Azure bandwidth pricing[EB/OL]. Microsoft, 2026[2026-05-15]. [https://https://azure.microsoft.com/en-us/pricing/details/bandwidth/](https://azure.microsoft.com/en-us/pricing/details/bandwidth/).
- [30] Google. Google cloud bandwidth pricing[EB/OL]. Google, 2026[2026-5-15]. <https://cloud.google.com/vpc/network-pricing>.

- [31] Andersen D, Balakrishnan H, Kaashoek F, et al. Resilient overlay networks[C]//Proceedings of the eighteenth ACM symposium on Operating systems principles. 2001: 131-145.
- [32] Roy S, Pucha H, Zhang Z, et al. On the placement of infrastructure overlay nodes[J]. IEEE/ACM Transactions on networking, 2009, 17(4): 1298-1311.
- [33] Cai C X, Le F, Sun X, et al. Cronets: Cloud-routed overlay networks[C]//2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2016: 67-77.
- [34] Jain S, Kumar A, Mandal S, et al. B4: Experience with a globally-deployed software defined wan[J]. ACM SIGCOMM Computer Communication Review, 2013, 43(4): 3-14.
- [35] Zhang Y, Jiang J, Xu K, et al. Bds: A centralized near-optimal overlay network for inter-datacenter data replication[C]//Proceedings of the Thirteenth EuroSys Conference. 2018: 1-14.
- [36] Jain P, Kumar S, Wooders S, et al. Skyplane: Optimizing transfer cost and throughput using {Cloud-Aware} overlays[C]//20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23). 2023: 1375-1389.
- [37] Troia S, Mazzara M, Savi M, et al. Resilience of delay-sensitive services with transport-layer monitoring in sd-wan[J]. IEEE Transactions on Network and Service Management, 2022, 19 (3): 2652-2663.

附录 A 补充内容

A.1 插图

A.2 表格

A.3 数学表达式

A.4 文献引用

参考文献

致 谢

衷心感谢导师 ××× 教授和物理系 ×× 副教授对本人的精心指导。他们的言传身教将使我终生受益。

在美国麻省理工学院化学系进行九个月的合作研究期间，承蒙 Robert Field 教授热心指导与帮助，不胜感激。

感谢 ××××× 实验室主任 ××× 教授，以及实验室全体老师和同窗们学的热情帮助和支持！

本课题承蒙国家自然科学基金资助，特此致谢。

声 明

本人郑重声明：所呈交的综合论文训练论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：_____ 日 期：_____