在新兴的近地轨道卫星和云上合作构建成本效益高的内容分发网络

赖、李和武\*张、吴倩、吴建平

*清华大学网络科学与网络空间研究所，北京100084*

[zeqilai@tsinghua.edu.cn，](mailto:zeqilai@tsinghua.edu.cn)*{*lihewu，wuqian，jianping*}*@cernet.edu.cn，[qi-zhang19@mails.tsinghua.edu.cn](mailto:qi-zhang19@mails.tsinghua.edu.cn)

***摘要互联网内容提*供商通常利用基于云的内容交付/分发网络（CDN）来提供高可用性和低延迟的广域数据访问。然而，从全球角度来看，由于地面云基础设施的部署不足，很大一部分用户仍然面临高内容访问延迟的问题。**

2021年IEEE第29届国际网络协议会议（ICNP）| 978-1-6654-4131-5/21/31.00美元©2021 IEEE | DOI:10.1109/ICNP444.2021.9651950

**本文介绍了StarFront，这是一个经**济高效的内容分发框**架，用于优化全球CDN，并在任何地方实现低内容访问延迟。StarFront在新兴的近地轨道（LEO）星座和现有云平台上构建CDN，以满足低延迟要求，同时最大限度地降低运营成本。具体而言，StarFront利用了*一个关键见解，即*新兴**的巨型星座将由数千颗**配备高速数据链路和存储的低轨卫星组成，因此有可能作为“太空缓存”，实现普及和低延迟的数据访问。StarFront明智地将副本放置在LEO卫星或云上，并根据星座参数、云/用户分布和定价策略将用户请求动态分配给适当的缓存服务器。覆盖地理分布有利位置的广泛跟踪驱动评估表明：在具有代表性的CDN流量下，StarFront可以以可接受的运营成本有效降低全球内容访问延迟。**

***索引术语地理分布式内容*分发、低轨星座、综合卫星和地面网络。**

1. 介绍

内容分发网络（CDN）由数量可观的基于地理分布的云缓存服务器组成，其目标是在全球范围内提供高可用性和低内容访问延迟。CDN承载着大量的网络流量。预计到2022年，全球CDN将提供约72%的互联网总流量[5]。因此，优化CDN的网络性能可以显著提高基于其构建的各种应用程序（例如，Web服务和视频点播等）的体验质量。CDN的主要优势之一是低内容访问延迟，这通常通过向最终用户交付所需对象的时间消耗来量化。例如，内容访问延迟可以分别指Web或VoD应用程序的页面加载时间或视频初始化时间。低延迟的关键在于，用户可以访问缓存在靠近最终用户的地理分布缓存服务器上的内容副本。然而，我们对从183个国家的七家主要CDN运营商收集的大规模CDN跟踪进行的深入分析表明

*\**李是通讯作者。978-1-6654-4131-5/21/31.00美元©2021年IEEE

用户到最近的缓存服务器的往返时间（RTT）仍然很长，这可能会导致内容访问延迟很长。我们的进一步分析表明，这种高访问延迟在偏远或农村地区更为普遍，这是由于云基础设施部署不足，以及通往云服务器的地面路线曲折（例如，远程对等导致的互联网路径延长[28]）。

然而，上述根本原因在地面网络中都不容易解决。首先，当今互联网的部署本质上是一个不均衡的网络，网络资源聚集在许多发达地区（例如“热点地区”）。在偏远和/或其他欠发达地区，互联网接入有限，尽管这些地区的人口可能非常多，但提供和维护云服务器以提高网络性能的成本可能更高[19]。*其次，地面互联*网被划分为许多自治系统，由于不同自治系统的特定路由策略，AS之间的路由可能是曲折的，导致用户和指定的云服务器之间的路由曲折，这进一步增加了延迟。我们如*何应对上述挑*战？

拥有数千颗卫星在近地轨道（LEO）飞行的新兴巨型星座为优化全球CDN的网络性能提供了新的机会。现代卫星将配备“高通量通信组件”[34]、[56]、[60]和高容量存储器[40]、[42]，这些卫星有可能成为“太空缓存”。直观地说，在新兴星座上缓存内容副本是一种很有前途的方法，可以普遍实现低延迟[25]。然而，充分利用巨型星座的潜力需要解决几个基本挑战：（i）卫星从根本上是移动的，并且以高速移动。如何正确选择LEO卫星来缓存内容副本并避免间歇性连接的影响？（ii）通过卫星传输流量比通过地面网络传输流量更昂贵。如何以经济高效的方式明智地将用户请求分配给卫星或云缓存服务器？

我们提出了STARFRONT，这是一个内容分发框架，它协同利用LEO卫星和地面云中的缓存节点，以经济高效的方式优化内容访问延迟。特别是，STARFRONT采用了两项关键技术来实现具有成本效益的广域内容分发。首先，STARFRONT构建了一个动态的卫星云拓扑结构，它捕捉了时变的可达性

以及卫星云集成架构的性能。建造过程基于云分布信息、可预测的卫星轨迹及其估计的性能和定价模型。其次，STARFRONT为不同地区的用户提供了三种形式的分配。具体而言，用户请求可以是：（i）通过地面网络（如现有的基于云的CDN）直接分配给云服务器（即，地面缓存）；（ii）通过由一系列卫星构建的低延迟空间路线分配给云服务器；或者（iii）如果最近的云仍然太远，则分配给卫星高速缓存服务器。上述请求分配方法在实践中涉及不同的延迟性能以及相应的存储和流量成本。STARFRONT明智地推送内容并将副本放置在可用的云或卫星服务器上，并将用户请求分配到适当的缓存服务器，以满足不同应用程序的延迟要求，同时通过卫星云合作将内容分发的总成本降至最低。

我们已经实现了基于Apache Traffic Server（ATS）[3]的STARFRONT原型。我们还建立了一个试验台来模拟云站点和动态低轨卫星网络，以评估所提出的框架。基于覆盖地理分布有利位置集合的最先进星座（例如，星链和OneWeb）的跟踪驱动评估证明了STARFRONT的有效性。通过集成卫星和云，STARFRONT的性能优于现有的纯云方法，可以以可接受的运营成本满足应用程序的各种延迟要求。此外，我们发现星座的架构设计会显著影响STARFRONT的性能，因为轨道高度较低并配备星间链路（ISL）的星座可以进一步减少地面用户更多的*内容访问延迟*。

总之，本文做出了三个关键贡献：

* 我们通过对七个商业CDN运营商的测量研究（§ii）确定并分析了现有CDN中的高访问延迟问题，并设想了利用低地球轨道巨型星座协助普及、低延迟内容分发的可行性和挑战（§iii）。
* 我们设计了STARFRONT，并构建了它的原型。STAR-FRONT是一个内容分发框架，它协同利用云和低轨卫星中的存储和网络功能，以经济高效的方式明智地优化全球内容访问延迟（§IV）。
* 我们通过对STARFRONT原型的大量跟踪驱动模拟来评估STARFRONT在提高全球交付效率方面的有效性（§V）。

1. 背景和动机

**用于大面积内容分发的快速入门。内容交付/分发网络（CDN）是一个高度分布式的平台，具有全球分离的许**多缓存服务器。内容提供商通常利用公共云平台（例如，亚马逊AWS、微软Azure等）在终端用户附近部署缓存服务器。来自内容提供商拥有的源服务器的流行内容被分发到地理位置

云服务器，通常分发过程包括三个关键步骤：（i）将原始内容从源服务器推送到地理分布的缓存服务器集合（例如，通过构建分发树[29]）；以及（ii）配置所述区域↔服务器映射，其在运行时确定来自不同区域的用户请求应如何分配给特定缓存服务器；以及（iii）如果存在可从源服务器获得的新内容，则维护和更新每个高速缓存服务器中的数据。为内容提供商执行的上述操作主要根据其消耗的存储和带宽收*费，遵循不*同云提供商指定的具体定价策略（例如，[1]、[2]）。

理想情况下，通过将内容移动到用户附近，广域CDN有望在地理上实现低内容访问延迟，该延迟被定义为向最终用户交付一个或一批对象所需的时间。在实践中，内容评估延迟主要受用户和分配给他们的缓存服务器之间的往*返时间（RTT）以及可用带宽的影响。后者通常*取决于ISP或云提供商的定价，而前者受网络拓扑结构的限制。

**从全球角度观察到的高内容访问延迟。为了定量了解最先进的广域商业CDN的可实现性能，我们收集了来自183个国家的17402个探针的**RTT测量结果，这些探针距离七个最受欢迎的CDN平台（即Akamai、Azure、Cachefly、Cloudflare、Cloudfront、Fastly和GoogleCloud）提供的最近的云站点。测量是在2019年3月7日至4月1日期间使用RIPE Atlas测量平台进行的[10]。对于每个探测，我们对所考虑的七个CDN中的每一个执行20多次ping测试，并收集最佳探测结果（访问延迟），以避免网络拥塞或临时链路故障对测量结果的影响。我们还使用traceroutes工具来跟踪数据包在用户和*云服务器之间*的路由。图1绘制了不同地理分布用户到某个CDN运营商最近的云服务器的RTT结果的CDF。如图1所示，尽管CDN具有优势（例如，40%的Akamai测量值的RTT小于23ms），但仍有很大一部分CDN用户的RTT高于50ms，长尾时间高达约300ms，即使选择了最近的云服务器。

高网络RTT会严重损害用户感知体验，尤其是对于具有一批顺序请求的时间敏感应用程序（例如，网络浏览、视频点播等）。例如，如[27]所示，即使是几十毫秒的额外RTT也可能大大缩短网络浏览页面加载时间并降低用户体验。

为了进一步了解高RTT观测结果，我们将每个用户的测量结果按其原始大陆分组，如图2所示。我们发现，尽管许多人口稠密和发达地区的延迟很低，但即使选择了最近的云服务器，仍有大量用户遭受高访问RTT的困扰。对于那些偏远或欠发达地区的用户来说尤其如此。具体来说

1

Akamai Azure缓存

Cloudflare Cloudfront Fastly公司

谷歌云

0.75

0.5

**CDF公司**

0.25

0

0 50 100 150 200 250 300

**RTT（毫秒）**

图1：现有CDN运营商的延迟分布。

1

欧盟为纳

AF自组织

所有

0.75

0.5

**CDF公司**

0.25

0

0 50 100 150 200 250 300

**RTT（毫秒）**

图2：不同大陆的潜伏期分布。

总的来说，约有53.58%/23.93%的用户患有50ms/100ms以上的RTT。即使在欧盟和北美等发达地区，我们仍然观察到3.4%和4.5%的用户与高于50ms的RTT相关。AF中的延迟问题更为严重，超过86.6%/49.8%的用户患有高于50ms/100ms的RTT。

**在偏远/农村地区大规模部署缓存服务器是困难的。优化全球不均衡CDN部署并降低访问延迟的一个简单方法是，通过大规模增加偏远和农村地**区的云/边缘部署，扩大地面云站点的覆盖范围，为这些地区的用户提供服务。然而，地面部署在很大程度上受到地理因素（如地形）或大规模运营和维护成本的限制，尤其是在发展不足的地区。例如，在一些地区，云/边缘站点可能没有足够的电力基础设施支持，或者由于政治或管辖原因，这些地区的部署变得困难。当今基于地面云的CDN的现状促使我们探索一个具有挑战性但重要的问题：我们如何应对上述挑战，并在全球范围内实现普及和低延迟的CDN访问？

1. 未来LEO中的低延迟机遇

兆合金钢

近年来，近地轨道（LEO）卫星的研究和部署再次受到欢迎[25]，[26]，[32]，[35]-[38]，[40]，[44]，[48]，[54]。星载技术的快速发展提高了最先进的低轨卫星的硬件能力。与使用地球静止卫星进行通信的第一代卫星网络相比，计划由数千颗大规模生产的低空飞行卫星组成的新兴巨型星座（例如，星链[15]、OneWeb[8]）将拥有进化的网络和存储能力，从而（可能）为在全球构建低延迟CDN提供新的机会。

* 1. **不断发展的车载通信能力。现代卫星配备了“高通量通信组件”[34]、[56]、[**60]，能够提供数十甚至数百Gbps的数据速率[50]。许多计划

星座建议使用射频或激光卫星间链路，低轨卫星可以通过这些链路连接到可见的相邻卫星，并在太空中构建网络。此外，由于其低飞行特性（即500-1200公里的高度），与在约36000公里的高度运行的传统地球静止卫星相比，低地球轨道星座也有望实现低延迟互联网连接。实现降低延迟的另一个关键因素是，地面光纤中的光速比空气或真空中的光速慢约33%[35]。最近的研究概述了太空中低延迟路由的愿景[36]-[38]，[*48]，揭示*了通过低地球轨道航天器上的太空路由减少端到端延迟的潜力，特别是对于远程通信。

* 1. **大数据存储在太空中。机载容量的另一个演变是太空存储。**最近的工作[40]，[42]已经设想了基于卫星的大数据存储。太空带[13]建造的云星座公司（CCC）是一种使用低轨卫星的数据存储服务。CCC系统包含一个由10颗低轨卫星组成的环，位于650公里的赤道轨道上，其中三颗是数据存储，自2018年12月以来提供了约5PB的存储容量[12]。

因此，航天工业的上述发展描绘了一幅卫星云合作内容分发架构的前景图，该架构可以（潜在地）提高现有CDN的可访问性和网络性能。直观地说，低轨卫星可以通过以下方式帮助当前基于云的CDN：（i）构建连接地面云和最终用户的低延迟、接近最佳的空间路径，以避免可能延长接入延迟的曲折光纤/电缆路线；和

（ii）启用一种新的模式“低轨卫星上的缓存服务器”*，为即使是最近的云站点也离得太远的地区的用户提供更低的*访问延迟。然而，尽管前景看好，但我们认为，由于低轨卫星网络的两个特定特征，在云数据中心和巨型星座上构建这样一个合作CDN仍然面临着几个尚未解决的挑战。

1. **稀缺且昂贵的空间资源。尽管已经进化，但与优化良好的地面云平台相比，网络**和存储资源在空间上仍然相对有限且成本高昂。例如，对于现有CDN运营商（例如Amazon CloudFront）的用户来说，传输1GB的数据流量大约需要0.1美元[1]。由于大多数新兴的低轨星座仍在大量部署中，其定价政策尚不可用，因此我们根据当前运行的宽带卫星系统估计了卫星数据传输的成本。例如，根据提供高速卫星宽带服务的卫星提供商ViaSAT[17]的定价政策，通过卫星网络传输1GB的数据量大约需要1美元。此外，对于卫星网络的用户来说，用户终端（例如，小型碟形卫星天线）需要额外的前期成本[14]。由于内容提供商通常*有通过CD*N分发其数据的成本预算，因此应以具有成本效益的方式明智地将内容分发到卫星和云，即，构建的合作CDN有望满足应用程序的延迟要求，同时涉及可接受的运营成本。



**（1） StarFront[控制器]**

在线请求分配

**源云段（2）StarFront[缓存]**

**卫星段**

**ś**

放置和分配决策

战略决策

**地理的**

**Ś**

**③**

历史性能信息

**最终用户**

将内容从源推送到缓存服务器

**从**云服务器请求内容

**ś**通过空间路径向云服务器请求内容③向卫星服务器请求内容

图3：STARFRONT框架概述。

1. **低轨卫星的高机动性。低轨卫星相对于地球高速移动，导致地面通信不稳定**。只有低轨卫星进入视线（LoS），地面上的终端主机或云站点才能与低轨卫星通信。在一个轨道周期内（例如，星链一期低轨卫星约为90分钟），一颗低轨卫星仅在几分钟内的特定地面有利位置可见。如果请求被分配给离开LoS的卫星，则应用级会话可能被中断。因此，应正确分配用户请求，以避免间歇性连接导致的性能下降。
2. StarFront框架

为了应对上述挑战，我们提出了STARFRONT，这是一个合作的内容分发框架，它利用LEO卫星或全球分布的云数据中心上托管的许多缓存服务器，以经济高效的方式构建普遍的低延迟CDN。在高层，STARFRONT利用了以下关键思想：（i）利用云分布和可预测卫星轨迹的信息，以及它们的估计性能和定价模型，构建一个动*态卫星云拓扑结构，该拓扑结构捕*捉了卫星云集成架构的时变可访问性和性能；因此（i*i）明智地将内容放置在动态卫星云拓扑上*，并将用户请求分配给适当的缓存服务器，以满足各种应用程序的延迟要求，同时最大限度地降低内容分发的总成本。

1. StarFront*系统概述*

图3描绘了STARFRONT的概述，它由两个核心组件组成：STARFRONT地理分布式缓存和STARFRONT控制器。具体而言，STARFRONT缓存包含一个基于公共云提供商（如亚马逊AWS）提供的地理分布式云数据中心（即地面缓存服务器）的云段，以及一个基于SpaceX或OneWeb等卫星运营商管理的新兴巨型星座的卫星段。云数据中心通过地面网络相互通信，而配备卫星间链路（IS*L）的卫星可以通过空间路由算*法构建的卫星路径将数据传输到其他卫星（例如，[36]）。此外，领先的云提供商也在积极部署地面即服务（GSaaS）[*4]，[21*]，这可以实现现收现付的地面通信，以互连云数据中心和卫星。利用GSaaS基础设施，

如果卫星进入LoS，地面云能够按需与卫星通信。

内容提供商可以在STARFRONT框架中选择一组缓存服务器，并使用它们将内容分发给最终用户，因为可以选择云数据中心和带存储的卫星来缓存内容。在运行时，STARFRONT框架允许内容提供商执行两个基本操作：（i）内容推送和放置操作，即通过云间/卫星路径将内容从源服务器推送到选定的缓存服务器集；和

（ii）*请求分配操作，其将用户请求从地理分*布区域重定向到适当的缓存服务器以获得低访问延迟。根据云和卫星运营商的专用定价政策，这两项操作都涉及费用，因为分发过程可能会消耗相应云或卫星平台中的存储和带宽资源。

总之，STARFRONT支持三种形式的请求分配，具有不同的网络性能和相应的成本。首先，与传统的云CDN一样，如果云性能能够满足应用程序要求，STARFRONT允许用户直接从附近的云缓存服务器（例如，图3中的\*1）请求内容。其次，如果用户到最近的云缓存服务器的访问延迟很高（例如，由于云站点部署不足、通信距离很长或地面光纤路由曲折），STARFRONT允许用户通过自由空间卫星路由加速内容访问（例如，图3中的\*2）。在这种形式下，相关卫星只是在用户和相应的云缓存服务器之间转发流量，而不需*要在卫星上缓存*副本。第三，当最近的可用云仍然太远时，请求也可以直接分配给卫星缓存（例如，图3中的\*3）。请注意，在实践中，上述三种形式的请求分配会导致不同的存储和带宽成本，空间中的资源可能更宝贵。特别是，第一种形式只消耗地面存储和带宽，第二种形式使用地面存储和卫星带宽，而最后一种形式在太空中消耗更昂贵的存储和带宽。

1. StarFront*工作流*

为了利用新兴巨型星座的低延迟潜力，同时不给内容提供商带来高运营开销，STARFRONT控制器以成本效益高的方式明智地计算布局和分配决策，即决定如何从所有可用的云和卫星中选择一组缓存服务器以形成内容分发网络？以及如何将来自不同地理分布区域的请求分配到适当的缓存服务器，同时不超过内容提供商的成本预算？为了应对星座拓扑的高动态性，STARFRONT控制器利用卫星运动的周期性和可预测性，定期执行以下操作，以实现具有成本效益的分布目标。每个操作将在以下小节中详细描述。

* （i） 在每个周期开始时构建动态卫星云拓扑来对时变可用性进行建模，

网络性能以及可用云和卫星的成本，基于云和用户分布、星座模式以及云和卫星提供商的定价政策。

* （ii）一旦获得卫星云拓扑，执行明智的放置和分配算法，并根据计算出的决定将内容推送并放置在所有选定的缓存服务器上，然后配*置区域↔ 服务器分配。在实践中，请求分配通常可以通过本地DNS重定向应*用[59]。
* （iii）在此期间，根据地区将用户请求重定向到适当的缓存服务器↔ 服务器分配。直观地，STARFRONT可以通过地面网络为低延迟云平台供应良好的发达地区（例如大都市）的用户分配请求*到附近的云缓存服务器，并通过卫星*路径或直接向可用的卫星缓存服务器向云服务器发出请求，-对于那些偏远或农村地区的用户来说，那里的地面网络和云非常有限。

1. *时变卫星云拓扑建模*
   1. *制定动态卫星云拓扑：尽管之前的许多工作已经研究了云平台中具有成本效益的内容分发问题（即，如*何在静态部署的云存储服务器上部署副本[29]，[47]，[58]），低轨卫星的高动态性使得内容放置和请求分配问题与以往的工作有根本不同。因此，我们构建了一个图来对时变动态卫星云拓扑进行建模。

**顶点集。在地面云CD**N中，内容提供商将内容的副本放置在云运营商拥有的地理分布式云数据中心上。让Src表示内容提供商的源/原始服务器。我们假设有M个由地面云运营商提供的用于缓存内容的可用云数据中心，即C={C1，C2，…，CM}。

此外，内容也可以由低轨卫星缓存，我们假设该星座由N颗在轨道高度H运行的低轨卫星组成。卫星集表示为S={S1，S2，…，SN}。此外，为了描述低轨卫星的高动态性，我们假设T是给定星座的会合周期，它指的是从地球表*面观*察到的卫星的重复周期。假设一个周期T被分成多个时隙。我们将卫星集S推广到S={S1？8746；S*2*，…，？8746ST}，ST={S1t，S2t，…，SNt}，

其中，Sit*表示时*隙t中第i颗卫星的快照。正如我们之前所介绍的，新*兴的云*提供商可以在其数据中心附近部署地面站（例如，亚*马*逊地面站服务[21]和Azure轨道[4]），以提供按需地面通信，将云数据中心连接到巨型星座。因此，每个云也可以通过低轨卫星构建的时变空间路线访问。因此，我们对不同插槽中的云数据中心集C进行了逻辑快照，并设置了CSt=

*{CS*1*t，CS*2*t，…，CSMt}，C*S*={*C*S1Ş*C*S2，*…*，ŞCST}为*

通过卫星可到达的一组云层。CSt是指通过时隙t中的卫星路径可访问的M个云站点*的所有*快照的集合，更具体地说，CSit表示

可通过时隙t中的卫星路径访问的云数据*中*心i。物理上，Ci和CSit指的是同一个云数据中心，而逻辑上，总云集为C={CŞC*S*}。我们将槽t中的云的快照表*示为*：Ct=*{CŞC*St}。CDN接收和发送地理用户请求，这些请求通常由HTTPGET/POST请求封装。在实践中，请求调度通常通过基于DNS或基于任意播的*客户*端-*服务器映射来*实现[16]，[59]。为了对用户分布进行建模，我们假设地球表面被离散为不重叠的网格*状区*域，总共*有J*个*区域，并使用R=*{R1，R2，…，RJ}来描述所有服务区域的集合。用户请求将根据其所在的地区进行调度。总之，假设Src是源服务器，则槽t中卫星云拓扑图的顶点集可以表示为：Vt={SrcŞStŞCtŞR}。此外，在周期T中设置的顶点相应地表示为：

*VT*=*{V*1*ŞV*2*Ş…ŞVT}。*

**边缘设置。对于顶点**集VT中的每两个不同节点i和j，它们之间可能存在无向边（即，*（*i，j）*）*。我们将周期T中的边缘集表示为ET。*具体*地说，连接两个云/卫星节点的边缘表示它们之间存在通过陆地或太空网络的数据传输路径。云/卫星节点和*区域之间的*边*缘指示*来自该区域的用户请求可以被分配给云/卫星结点。两个不同区域之*间*不存在*边，*因为它们代表最终用户。源服务器类似于云节点。

每个边缘都与costraffic（i，j）属性相关联，该属*性指示在i和j之间传输*一*个数据*单元的成本。costraffic（i，j）和latency（i，j）的具体值取决于相关的顶点类型，如下所述。

* （i） 对于任意时隙中的云节点i*，j∈C，*它们之间的数据流量通过地面互联网传输。如果i和j在物理上相等，例如i＝Ck和j＝CSkt，则我们具有costraffic（i，j）＝0。否则，costraffic（i，j）＝c，即，在i和j*之间传输一个单元数据的*成*本为$*c。
* （ii）对于任意时隙中的云i*∈C和卫*星j∈S，在实践中，通过L*EO卫星*序列和地面站，可以通过最短路径源路由方法[36]建立连接它们的边缘。考虑到空间路径上的数据传输可能会消耗更多宝贵的星地通信资源，我们表示costraffic（i，j）=α·c，i∈c和j∈S，其中α是放大因子，α>1。
* （iii）对于任意时隙中的卫星节点i，j∈S，*（i/=j）*，

我们设置co*straffic（i，*j*）=β*·c。*与α类似*，β是一个放大因子，因为*卫*星*之*间的数据传输可能比地面互联网中的数据传输成本更高。（β>1）

* （iv）对于区域i∈R和云*j∈C*，costraf*fic*（*i，j）=*

*c*、 如果*j∈c*，并且co*straffic（i，*j*）=α*·c*如果j≠*CS*，因为利*用空间路由来提供对云站点的内容访问的成本更高。此外，假设用户

区域感知到与同一高速缓存服务器相似的延迟，并且延迟（i，j）是区域i和服务器j之间的区域到服务器的访问延迟。该*值指示高速缓存*服*务器j*对来自区域r的请求的响应速度。

* （v） 对于区域i∈R和*卫星j*∈S，（i，j）表示

来自r的请求可以分配给卫星缓存服务器j*。我*们设置costraffic（i，j）=α·c。具体而言，如果j=Skt，则latency（*i*，j）是指时*隙t中区域i和卫星k之*间*的访问*延迟*。注意*，随着LEO卫星在轨道上移动，区域*和卫*星缓*存之间*的延*迟可能会随着时*间*的推移*而相应变化。

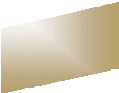
通过与顶点和边相关的定义，我们将周期T中的连通图定义为GT=<VT，ET>。

* 1. *在卫星云拓扑上制定具有成本效益的内容分发问题：在给定上述时变拓扑的情况下，接下来我们基于卫星云模型制定具有成本效率的内容分配问题。对于mally，问题可以转化为形成分布*

卫星云拓扑G*T上*的图（DG）。假定

**延迟要求：**

* + - **≤50ms**



**洛杉矶**

S-2型

狮子座

S-1

**RTT=40毫秒**

R-1型

来源

**时隙1**

**RTT=100毫秒**

R-2型

**RTT=20ms**

C1

S-2型

R-3型

S-1

**RTT=20ms**

**洛杉矶**

**RTT=80毫秒**

狮子座

**时间**

R-2型

**RTT=20ms**

R-1型

**RTT=100毫秒**

**时隙2**

C1

R-3型

**复制副本放置：**

* + - **C1，S1**

**插槽1中的分配：**

* + - **R1****C1，通过卫星**
    - **R2****C1，R3**C**1**

**插槽2中的分配：**

* + - **R1****S1**
    - **R2****C1，R3**C**1**

图4：从源服务器分发内容和相应分配的示例。

用户体验，在每个周期T中，每个区域与其分配的缓存服务器之间的平*均*用户感知延迟预计低于阈值Thdlatency。具体而言，假设时隙t中区域r的用户感知延迟为：延迟（r，a（r，t）），因此延迟约束可以是

*T*

*t*

*λ*（*i，j*）是一个二进制值，λ（i、j）=1表*示*边

公式为：∑

*潜伏期（r，a*（*r，t*）*）≤T*hd*latency，∀t∈t。*

（*i。假*设x（i）是服务器选择

如果在DG中选择了顶点i*，*则*x*（i）＝1。则DG可以*表*示为DG=<VDG，ED*G>*，其中x（i）*=x*（j）=1，λ（i，j）=1、∀*i，*j*∈VDG，（i，j*）∈EDG，以*及*V*D*G⊂VT*，*E*D*G⊆ET。

*DG揭*示了如何在选定的缓存服务器集合上推送和放置内容，以及如何将来自某个区域的请求分配给缓存服务器。首先，VDG中的所有节点都被选为缓存服务器，并且必须存储内容副本。其次，假设r是一个区域节点，则连接一个区域和一个服务器的边（r，j）∈EDG，j∈{CŞCStŞSt}表示T期间的时变请求分配。具体地，设a（r，t）表示为时隙t中的区域r分配的高*速缓存*服务器。a（r、t）计算如下：（i）如果在时隙t中EDG中存在连接r和CSt或St中的服务器的边缘，则在时隙t中将通过空间路由将来自r的请求分配给高速缓*存*服务器；（ii）否则，如果在时隙t中EDG中只有一条边缘将r连接*到C中*的*地面云，则来自r的请求将通过地面路由分配给连接*的云；（iii）否则，不存在可连接节点

图4绘制了一个显示内容过程的示例

放置和请求分配。假设在两个时隙中有三个区域（R1-3）、一个云站点（C1）和两个LEO卫星（S1-2）。在每个时隙中，从R2和R3到云C1的RTT约为20ms，低于50ms的延迟要求。在时隙1中，R1和C1之间的RTT通过地面网络为100ms，通过卫星为40ms。在时隙2中，随着S2移出R1的传输范围，从R1到C1的空间路径更新，R1和C1之间的*R*TT通过卫*星*从40ms增加到约8*0*ms。基于可预测的条件，保证延迟要求的可行解决方案是在S1和C1上推送并放置副本，并且：（i）在每个时隙中将R2和R3的请求分配给C1；（ii）在时隙1中经由卫星将请求从R1分*配*给C1，以*及*（iii）在时隙2中将请求从R2分配给S1。

**成本分析。注意，内容放置和请求**分配操作都可能涉及内容提供商的成本。在放置过程中，将内容从源服务器分发到所有缓存服务器会消耗地面和太空网络中的带宽，成本为CT1=

Σ

对于槽*t*中的r，a（*r*，t）=*∅*。*因此，*问题在于

*i、 j*

*成本流量（i，j）=8*7*27；*0*99*5*5；i*，*j；=87277；W，©8704i*；i，j=8712；{Src=8746；C=8746；S}，

转换为在满足如下所述的几个约束的同时在周期T中从GT找到子图DG。

**C1：每个区域必须在每个时隙中分配给一个缓存服务器。分布图DG应保证每个区域中的用户在时间段T期间被分配到缓存服务器，即a（r，T）/=∅，∀r**∈r，⇼T∈T。

# C2：对于每个选定的缓存服务器，应该有

其中W是内容*大*小。在云或卫星上缓存内容也涉及额外的存储成本，我们将coststorage（i）表示为在缓存服务器i上存储一个数据单元的成本。如果i是云服务器，我们将设置coststoragei=d，如果i是卫星，则设置coststore（i）=*γ\*d，其中γ≥1表示*在*广*泛的资源上相比更昂贵

地面云层。因此计算总存储成本

*k*

# 从所述源服务器到所述缓存服务器的至少一个路径。

式中：*CT*2=∑*coststorage*（*k*）*\*x*（*k*）*\*W，k∈{CŞS}*。

最初，内容是在源服务器上生成的，并且必须推送到每个选定的缓存服务器（即，其中x（i）=1）。因此，在DG中，应该有一个从源服务器到每个缓存节点的分发路径。

设req*（r，*t*）表示*时隙t中来自区域r的用户请求的总数。然后，分配过程的成本通过消耗用户和分配的缓存服务器之间的带宽而涉及。粗略地*说，*可以计算分配成本

# C3：每个区域的请求分配应满足

式为：*CT*3=∑

*r、 吨*

*成本流量（r，a（r，*t*）87*2*7英镑*；r*eq（r*，*t）等*于*87277；W，↓8704；r=8712；*

# 特定于各种应用程序的延迟要求。像

许多应用程序都有保持良好性能的延迟要求

*R、 t∈t*。计划周期T内的总成本

可以公式化为CT=CT1+CT2+*CT*3*，其*中C*T1*

是将源内容分发到每个选定的缓存服务器的带宽成本，CT2是指所有缓存服务器的存储成本，CT3表示在运行时为用户请求提供服务的带宽成本。

**具有成本效益的内容分发问题。通常，内容提供商可能有自己的预算来利用云/卫星基础设施来分发内**容。在我们的STARFRONT框架中，具有成本效益的内容分发问题是从GT中找到一个分发图DG，该分发图最小化总成本CT，同时满足所有上述约束C1、C2和C3。成本效益问题本质上是整数线性规划问题的一个例子。我们实现了ILP公式，并用Python MIP[9]对其进行了求解。随着卫星数量和用户请求的增加，实现的典型处理时间需要几分钟到几小时，这太长了，不实用。这促使我们设计和实施更高效、更适用的解决方案。

1. *明智的副本放置和请求分配*

我们提出了一种启发式方法来有效地解决内容分发问题，其关键思想是明智地探索适当的缓存服务器分配，以满足应用程序级别的延迟要求，同时将带宽和存储成本降至最低。特别是，利用缓存服务器为最终用户提供服务通常涉及三部分成本。假设s是为区域r分配的缓存服务器，并回忆内容分发过程中每个步骤的成本。首先，必须将副本从存储副本的另一台服务器推送到s。此步骤消耗云间/卫星带宽成本。其次，在云或卫星上配置复制副本也会产生存储成本。对于每个选定的缓存服务器，上述两个步骤只需要在周期T开始时执行一次。最后，在周期T期间和每个时隙中，来自区域r的服务请求也涉及s和r之间的带宽成本。形式上，设Costassign（r，s）表示将请求从r分配给s的成本估计。

算法1显示了STARFRONT的副本放置和请求分配算法的细节。在每个周期T中，STARFRONT的控制器（图3）首先根据当前周期内卫星的云分布和可预测轨迹生成连接图GT。然后，控制器运行算法来决定如何将内容推送*到*每个选定的缓存服务器并正确分配用户请求。对于每个区域r，STARFRONT首先搜索能够满足访问延迟要求的所有可用云或卫星服务器（即候选服务器）的集合（*第7*-12行）。在Candidate中的所有条目中，STARFRONT贪婪地选择涉及最小成本的服务器（第13-14行）。在算法运行期间，如果x（s）=1，则已经在s上提供了副本，并且该成本仅包括使用s来服务来自r的用户请求所产生的带宽成本。

**算法1明智的放置和请**求分配。

1：**输入：连接图**GT=<VT，ET>，延迟要求Th*d。*

2：**输出：分布图D**G，用x（k）表示，k∈

*{CŞS}*和λ（*i*，*j），*i*，j∈VT。*

3:/\*贪*婪地分配满足Thd的请求，同时将总成本降至最低\*/*

4:*x*（*Src*）=1/\*源服务器上的原始内容\*/

5：**对于所有区域**r∈Rdo

6: /\*查找所有满足Thd要求的可用服务器\*/

7： *候选人=[UNK]*

8： **对于所有i∈***VT-d***o**

9： **如果***潜伏期（r，i*）*≤Th*d

10： *候选人。添加（i）*

11： **如果结束**

12: **结束**

13： *选择→argminj∈VT Costassign（r，j）*

14: *x*（*选定）=1，λ（*r，选*定）*=*1*

15：**结束**

16：**返回所有x（**k）和*λ*（*i*，j）。

1. 绩效评估

由于大多数新兴的巨型星座仍处于早期阶段，很难在实时卫星网络上进行实验。虽然之前有许多网络模拟或仿真的工作，但现有的工作要么无法模拟/仿真LEO卫星的高动态性（例如，NS3、Mininet），要么无法支持对真实内容分布的评估（例如，[44]、[48]）。为了解决先前评估方法的这一局限性，我们建立了一个基于公共轨道数据模拟地理分布式云数据中心和星座的试验台，并实现了STARFRONT的原型。我们进一步进行了跟踪驱动的模拟，以验证STARFRONT在降低成本效益延迟方面的有效性，并将对实际部署的系统级影响（如延迟、损失、系统参数等）的系统评估作为我们未来的工作。

1. *模拟卫星云综合网络*

在高层，我们的试验台包含了一个拓扑生成器，该生成器加载云分布和时变卫星轨迹的信息，*以生成卫星云网络拓扑。此外，测试台利*用在物理机器上运行的多个容器来模拟内容分发的软件行为（例如，接收用户请求、查询缓存中所需的数据并将响应发送回用户）。

**拓扑生成器。拓扑生成器如下模拟卫星云**集成架构。首先，它使用亚马逊AWS云站点的分布作为可用的云数据中心[22]。我们基于亚马逊配置云分发，因为亚马逊已经部署了大量全球云站点，最近还部署了地面站服务来互连云和卫星。其次，拓扑生成器计算时变卫星轨迹，其包括LLA位置（即，

|  |  |
| --- | --- |
| **参数描述** | **价值** |
| 期间 | 7天 |
| #共个请求 | 390万 |
| #请求的总字节数 | 4137结核 |
| 第50/90个对象大小 | 256KB/1.3 GB |

表一：评估中使用的CDN跟踪摘要。

经度和高度）。具体而言，轨道信息由第三方轨道计算工具根据[11]生成的双线元素（TLE）数据计算，用于从其他节点（例如，邻居卫星、地面站或地面用户）的角度估计每个卫星的能见度和距离。在我们的实验中，我们在SpaceX的星链一期[15]和OneWeb[8]星座的第一个外壳下评估了STARFRONT，因为这两个星座都计划部署数百或数千颗低轨卫星，以提供广域覆盖和互联网服务。截至2021年8月，前一个星座由72个轨道平面上的1584颗卫星组成，高度约550公里，而后一个星座是计划在约1200公里高度的648颗卫星星座。星链和OneWeb的会合周期根据其公共星座信息分别配置为5731s和6557s。最后，我们根据其相关可见性设置拓扑*中每个节点的*连接性，即，如果卫星进入传输范围，则卫星可连接到地面站。

**云和卫星模拟。我们使用在物理机器上运行的Docker contain**-ers[6]来支持基于云/卫星的缓存服务器的模拟。具体来说，我们在每台物理机器上运行多个Docker容器，并使用每个容器和网络软件堆栈来模拟可用的云/卫星服务器。容器使用macvlan[7]连接到物理NIC，该方法将物理NIC虚拟化为多个虚拟NIC。我们使用tc来控制时变RTT、卫星间/卫星地面连接和每个链路的带宽。云间网络条件是基于真实AWS云站点的测量值进行配置的。卫星的连接性和性能是基于[48]所述的结果进行配置的。

**数据集和请求生成器。我们的评估利用了2015年2月24**日从一家商业云CDN运营商收集的真实世界CDN跟踪，共包含55.2万条流量记录。表I描述了所选跟踪的详细信息。此外，我们编写了一个请求生成器来模拟用户客户端。它从跟踪中提取信息，并生成获取对象数据的HTTP请求。最终用户发出的每个HTTP请求都按如下方式处理。首先，用户向位置DNS服务器发出DNS查询。其次，DNS服务器将分配的缓存服务器（Docker容器之一）的IP地址返回给用户。最后，客户端向缓存服务器发送获取内容数据的请求。

1. StarFront*原型*

STARFRONT控**制器。STARFRONT**的控制器是用大约1100行Python代码实现的。控制器定期读取卫星位置信息和历史网络性能信息，并计算

内容放置和请求分配的决策。然后，根据计算出的决定，通过HTTP连接将内容复制推送到缓存服务器。我们遵循现有云和卫星运营商（例如，CloudFront[1]和ViaSAT[18]）的定价政策来估计通过云或卫星交付内容的成本函数。

STARFRONT**缓存服务器。我们已经实现了**基于Apache Traffic Server（ATS）[3]的STAR-FRONT缓存。ATS是一个多线程、基于事件、模块化、高性能缓存和前向代理服务器，用C++编写。ATS作为一种商业产品分发，并已在许多生产级系统中使用。我们修改了ATS以连接到STARFRONT控制器并执行布局决策。接下来，本节中的评估旨在回答以下两个问题：（i）与其他最先进的内容分发方法相比，在具有代表性的CDN跟踪和星座模式下，STARFRONT能否满足地理分布用户的各种延迟要求？和（ii）

使用STARFRONT的相应成本是多少？

1. *验证满足延迟要求的能力*

我们比较了四种不同策略在全局内容分发方面的延迟减少：（i）现有云CDN中最先进的低延迟内容放置和分配方案（表示为基于云的SoA）（例如，TailCutter[47]、GRP[29]、CosTLO[58]等）；（ii）星光大道，

我们提出的框架合法利用云和卫星服务器放置内容，并将用户请求分配给适当的缓存服务器，以满足各种应用程序的延迟要求，同时将运营成本降至最低。此外，为了全面了解将LEO卫星集成到基于云的CDN的增量效率，我们在两种特定配置下评估了Starfront：

1. 副本缓存在云服务器上，用户只能通过卫星路径访问云（即IV-C中的C=∅，S=∅但CS/=∅）；以及（iv）副本缓存在云服务器上，用户可以通过地面或卫星路径获取副本（即，S=*∅，但第iv*-C*节中*的C/=∅和*CS/=*∅）。战略（三）和（四）指的是方法

只利用卫星网络来扩展陆地云的连接，而不使用卫星的存储能力来缓存空间中的内容。我们将（iii）表示为卫星路径（CCS）访问的云缓存，并将（iv）表示为地面和卫星路径（CCTS）访问的云和缓存。图5绘制了不同延迟需求下不同内容分发策略的用户感知RTT的CDF。STARFRONT的结果是在星链星座的配置下得到的。OneWeb的结果相似，但由于页面限制而被省略。由于STARFRONT集成云和卫星在全球范围内存储和分发内容，因此在RTT要求10ms/30ms/50ms/70ms/100ms的情况下，其平均性能优于最先进的基于云的策略90.51%/66.63%/52.82%/35.62%/15.45%。更具体地说，我们提出了一些意见。首先，对于严格的RTT要求（例如≤10ms），利用卫星网络加速云访问，甚至直接

云系统星光（CC S）星光（CCTS）星光

1

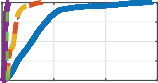
0.75

**CDF公司**

0.5

0.25

0



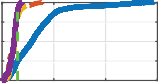
厚度=10ms

0 100 200 300

**RTT（毫秒）**

1

0.75



厚度=30ms

**CDF公司**

0.5

0.25

0

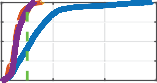
0

100 200 300

**RTT（毫秒）**

1

0.75



Thd=50ms

**CDF公司**

0.5

0.25

0

0

100 200 300

**RTT（毫秒）**

1

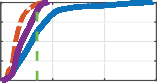
0.75

**CDF公司**

0.5

0.25

0



厚度=70ms

0 100 200 300

**RTT（毫秒）**

1

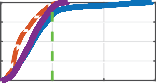
0.75

**CDF公司**

0.5

0.25

0



0毫秒

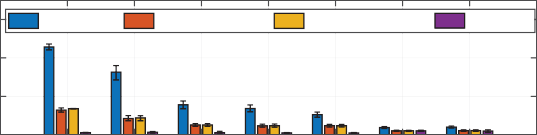
厚度=10

0 100 200 300

**RTT（毫秒）**

图5:RTT在不同延迟要求（10/30/50/70/100ms）下通过不同策略实现的CDF。

300



云系统星光（CC S）星光（CCTS）星光

**RTT（毫秒）**

200

100

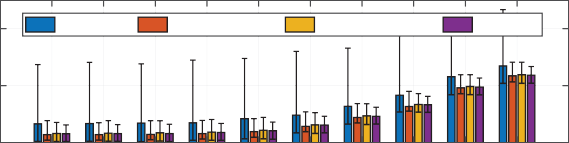
0

400

200

**延迟（毫秒）**

0



云系统星光（CC S）星光（CCTS）星光

20千克50千克80千克250千克500千克1.5千克3千克5千克8千克10千克

**平均复制副本大小**

图6：七个地理分布有利位置的RTT统计数据。

在空间中提供缓存可以显著提高满足广域用户请求的延迟要求的能力。这是因为纳入LEO卫星补充了陆地CDN，并从全球角度实现了对云和卫星服务器的低延迟访问。其次，随着所需RTT的增加（例如10ms→ 100ms），STARFRONT的延迟性能越来越接近于仅基于云的方法。这一结果表明，在宽松的延迟约束下，STARFRONT优选使用更多基于云的资源来节省卫星所涉及的运营成本。第三，卫星上的缓存可以进一步帮助减少延迟，但不可避免地会涉及更多的运营成本。在我们的进一步分析中，我们发现低轨卫星更适合缓存国际用户要求的内容。这是因为低轨卫星本身具有高动态性，而具有区域内容的卫星缓存在高速绕地球运行时可能会受到低缓存利用*率的影响。*

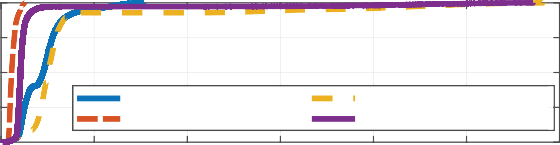
此外，我们将重点放在一组地理分布区域的RTT上。图6显示了世界各地一组有利位置在不同内容分发策略下的延迟比较。我们观察到，STARFRONT实现的延迟增益在不同地区有所不同。对于偏远或欠发达地区的用户，STARFRONT可以实现更多的延迟改进，因为这些地区的云部署和地面网络基础设施可能服务不足，而低轨卫星扩展了地面云平台的可用性和性能。具体而言，与纯云策略相比，STARFRONT为帕皮提和马朱罗等偏远地区的用户减少了90%以上的RTT。对于堪萨斯城和成都等人口稠密地区的用户来说，由于附近云基础设施的充分部署，所有策略都能获得类似的延迟结果。

1. *不同复制副本大小下的延迟减少*

根据具体的应用程序类型（例如，静态文本、文件或视频剪辑等），来自内容提供商的复制品在实践中可能具有不同的对象大小。我们评估内容访问延迟，以指示在不同的复制品大小下，请求的对象可以以多快的速度传递给用户。如图所示

图7：不同副本大小下的延迟结果。

1



星光大道（CCTS）-星光大道（CCTV）-On eWeb星光大道星光大道OneWeb

0.75

**CDF公司**

0.5

0.25

0

0 50 100 150 200 250 300

**RTT（毫秒）**

图8：不同星座下的延迟结果。

在图7中，我们观察到，与其他较大的请求相比，具有较小对象大小的较小请求更容易实现延迟减少。STARFRONT的主要好处是利用新兴的低轨卫星将内容推送到更接近用户的地方，并实现更低的客户端到内容RTT。在更深入的分析中，我们发现，对于小请求，总的内容访问延迟由RTT决定，内容访问延迟受可实现吞吐量和RTT的共同影响，并且在低RTT情况下，小请求的响应更快。

1. *不同星座模式下的时延降低*

不同的卫星运营商有其特定的星座设计，其轨道参数（如轨道高度、倾角等）不同，完整的星座需要很长时间才能完全部署。接下来，我们研究不同星座模式下的延迟减少。具体来说，在我们的实验中，我们比较了两个最先进的星座星*链和One*Web下的延迟。如图8所示，我们发现与OneWeb相比，与S*tarl*ink相关的STARFRONT可以实现更低的延迟。原因有三方面。首先，与星链相比，OneWeb卫星在更高的高度上工作，因此，当它作为缓存服务器或提供与地面云的网络连接时，会遭受更高的传播延迟。其次，OneWeb卫星没有卫星间数据链路，因此它限制了使用卫星构建太空路线以扩展云服务器可访问性时的延迟改进。第三，由于星链星座由比OneWeb更多的LEO卫星组成，并且如果这些卫星具有足够的存储能力来缓存内容，那么与星链相关的STARFRONT可以获得更高的延迟增益，因为密度更大的星座可以实现更多的缓存服务器和多样化的低延迟空间路线来从云中获取内容。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **成本（$/GB）方法**  （*a、b、T h*d） | 云SoA | 财政司司长 | 财政司司长 | 科幻小说 |
| *<*  2*,*  2*,*  1*0ms >* | 0.108 | 0.208 | 0.203 | 0.250 |
| *<*  2*,*  2*,*  5*0ms >* | 0.105 | 0.202 | 0.162 | 0.163 |
| *<*  2*,*  5*,*  1*0ms >* | 0.107 | 0.211 | 0.183 | 0.240 |
| *<*  5*,*  5*,*  1*0ms >* | 0.108 | 0.507 | 0.485 | 0.611 |
| *<*  5*,*  5*,*  5*0ms >* | 0.104 | 0.503 | 0.299 | 0.398 |

表二：不同定价策略和延迟要求下的内容分发成本。

1. *运营成本*

最后，我们评估了运营成本，包括内容分发过程中涉及的存储和带宽。表II总结了不同策略下内容分发的成本（存储和带宽的总和）（即向用户分发一GB数据的平均成本），以及系数α、β和Thd的各种配置。如表II所示，成本随着成本系数α和β的增大而增加。与纯云策略相比，STARFRONT以更高的内容分发费用为代价，实现了更低的延迟。虽然成本更高，但成本可能不是不可接受的，对于具有非常严格延迟要求的高优先级区域或任务来说应该是值得的。

1. 相关工作

我们将在本节中简要讨论相关工作。

**CDN中的副本放置和请求分配。大量先前的研究已经研究了基于云的CDN中的副本放置或请求**分配问题[20]，[29]，[41]，[43]，[47]，[49]，[53]，[58]，

[59]。CostTLO[58]结合了发出冗余请求的使用，以从不同的云存储节点获取内容。NetSession[61]是一个对等辅助CDN系统，它利用专用的、集中管理的基础设施和客户端来分发内容。NetSession可以提供基于基础设施和对等CDN的几个关键好处。STARFRONT与现有基于云的工作之间的根本区别在于：STARFRONT是一个协同利用云（静态）和低轨卫星（动态）上的网络和存储资源来优化全球内容交付效率的框架。与之前的工作（例如，NetSession）不同，STARFRONT表征了低轨卫星的高移动性，并利用其预测轨迹来帮助计算内容放置和请求分配决策。如评估所示，通过利用配备存储和高数据率ISL的LEO卫星，STARFRONT可以以具有成本效益的方式提高CDN的交付效率，特别是对于偏远地区的大量人口。

**探索建立在“新空间”基础上的网络。由于STARFRONT的设计部分基于新兴的低轨星座，我们的工作受到了最近关于表征和理解“新空间”**星座网络性能的一些研究的启发[24]，[25]，[30]，[31]，[35]-[38]，[44]，[45]，[48]，[55]。几个

研究了卫星网络中的路由算法。例如，为了了解LEO的延迟特性

卫星网络，作者在[36]中构建了一个模拟器来评估如何使用激光链路来实现卫星上的低延迟通信。类似地，[37]中提出了一种基于地面中继的路由策略。作者在[26]中提出了一种路径感知网络架构，以优化综合卫星和地面网络上的互联网路由。此外，Motifs[26]被提出在太空中动态建立网络连接，解决低轨卫星的高动态性问题。所有这些工作都是对我们工作的补充。

**在轨计算。最近的工作还探索了利用新兴纳米**卫星星座中改进的计算能力来构建一类新型计算机系统的可能性[25]，[32]，[46]。作者在[25]中定性地研究了在轨计算的机遇和挑战。提出了一种轨道边缘计算（OEC）[32]架构，以解决现有“弯管”架构的局限性，并优化边缘处理延迟。我们的工作不是只关注纳米卫星，而是提出了一种合作架构，联合利用陆地云数据中心和低轨卫星上的资源，以明智地提高全球范围内的内容分发性能。

**缓存算法。确定如何更新或替换缓存在每**个边缘节点中的对象的算法也是CDN的一个重要问题。最先进的算法通常是基于启发式的（例如，最近最少使用及其变体[23]、[33]、[39]、[51]）或基于学习的[52]、[57]。Learning Relaxed Belady（LRB）是最近提出的一种CDN缓存设计，它利用一种新的度量良好决策率来优化类缓存系统中的字节未命中率。这些专注于专用边缘节点中的缓存策略的*研究补充了我*们的工作。

1. 结论

本文研究了在新兴的低轨卫星和云上构建低延迟、低成本的CDN的问题。我们分析并量化了卫星云集成CDN改善内容访问延迟的潜在好处、可行性和挑战。我们介绍了STARFRONT，这是一个具有成本效益的框架，它以卫星/云运营商的动态网络拓扑、工作负载分布和定价策略为输入，并优化了内容分布的访问延迟。跟踪驱动的评估表明，与现有的基于云的方法相比，通过合作和明智地在卫星和云上放置副本，STARFRONT可以以可接受的成本满足应用程序的各种延迟要求。

1. 鸣谢

我们感谢我们的牧羊人Ignacio Castro和匿名的ICNP评审员的评论和建议。本研究部分得到了国家重点研发计划（No.2020YFB1806001）、国家自然科学基金（No.61832013.）、中国博士后科学基金（No.2021M691786）和清华大学-中国移动通信集团有限公司有限公司联合研究所的支持。

说到的事

1. 亚马逊云计算定价。https://aws.amazon.com/cloudfront/pricing/.
2. 亚马逊s3定价。https://aws.amazon.com/s3/pricing/.
3. Apache流量服务器。https://trafficserver.apache.org/.
4. Azure轨道：连接到Azure的卫星地面站和调度服务，用于快速下载数据。https://azure.microsoft.com/en-美国/服务/轨道公司/。
5. 思科视觉网络指数（内部），完整预测更新（2017-2022）。https://www.cisco.com/c/dam/m/en美国/ [网络-情报/服务提供商/数字化转型/知识网络-网络研讨会/pdf](http://www.cisco.com/c/dam/m/en)s/1213-business-services-ckn.pdf。
6. Docker容器。https:/[/www.docker](http://www.docker.com/).com/.
7. 从宏引 擎开发人员开始 。htt ps://do cs.dock er.com/network/macvlan/.
8. Oneweb.  [https://ww](http://www.oneweb.world/)w.onewe[b.world/.](http://www.oneweb.world/)
9. Python mip （混合整数线性编程）工具。h ttps:/ /pypi.org/pr oject/mip/.
10. 成熟的图集测量平台。https://atlas.ripe.net/.
11. 太空追踪网站。https://w[ww.space-track.org/.](http://www.space-track.org/)
12. 太空带旨在将数据存储在卫星中。https://blocksandfiles.com/2020/04/21/spacebelt-store-data-in-satellites-分析/。
13. 太空带云星座公司。http://spacebelt.com/.
14. 太空探索技术公司的星链卫星互联网服务价格为每月99美元。https://www.cnbc.com/2020/10/27/spacex-starlink-se[rvice-priced-每月99次-公开-测试-博客.html。](http://www.cnbc.com/2020/10/27/spacex-starlink-service-priced-)
15. 星链。https:[//www.starlink.com/.](http://www.starlink.com/)
16. 使用dns策略对主服务器进行基于地理位置的流量管理。https://docs.microsoft.com/en-us/windows-server/networking/dns/deploy/primary地理位置。
17. Viasat：为您的家庭或企业提供高速互联网。https://www.viasat.com/internet[.](http://www.viasat.com/internet)
18. 通过互联网计 划和定价。htt ps:// www .satelli[teinternet.com/providers/viasat/.](http://www.satelliteinternet.com/providers/viasat/)
19. 2020年世界互联网使用和人口统计。https://www.internetworldstats.com[/stats.html.](http://www.internetworldstats.com/stats.html)
20. S.Agarwal、J.Dunagan、N.Jain、S.Saroiu、A.Wolman和H.Bhogan。Volley：用于地理分布式云服务的自动化数据放置。在第7届USENIX网络系统设计与实现会议论文集，NSDI’10，第2页，美国，2010年。USENIX协*会。*
21. 亚马逊。AWS地面站。https://aws.amazon.com/ground-station/，2021年。[在线；2021年8月30日访问]。
22. 亚马逊。全球基础设施。https://aws.amazon.com/about-aws/全球基础设施/，2021年。[在线；2021年8月30日访问]。
23. H.Bahn、K.Koh、S.H.Noh和S.Lyul。高效替换web缓存中的非均匀对象。计算机，35（6）：65-732002。
24. D.Bhattacherjee、W.Aqiel、I.N.Bozkurt、A.Aguirre、B.Chandrasekaran，

P.B.戈弗雷、G.劳克林、B.马格斯和A.辛格拉。为21世纪的太空竞赛做好准备。第17届ACM网络热点研讨会论文集，HotNets'18，第113–119页，美国纽约州纽约市，2018年。计算机*协会。*

1. D.Bhattacherjee、S.Kassing、M.Licciardello和A.Singla。在轨计算：一个古怪的思想实验？第19届ACM网络热点研讨会论文集，HotNets’20，第197–204页，2020。
2. D.巴塔彻吉和A.辛格拉。27000公里/小时的网络拓扑设计。《第15届新兴网络实验与技术国际会议论文集》，第341–354页，2019年。
3. I.N.Bozkurt、A.Aguirre、B.Chandrasekaran、P.B.Godfrey、G.Laughlin，

B.马格斯和A.辛格拉。为什么互联网这么慢？！无源和有源网络测量国际会议，第173-187页。斯普林斯，2017年。

1. I.卡斯特罗、J.C.卡多纳、S.戈林斯基和P.弗朗索瓦。远程对等：在没有互联网扁平化的情况下进行更多对等。2014年，第10届ACM国际新兴网络实验与技术会议论文集，第185–198页。
2. 陈，郭，林，拉波塔。云内闪电：在云中构建cdn。在INFOCOM中。IEEE，2012年。
3. 一、 德尔波蒂略，B.卡梅隆和E.克劳利。EHF频段带有馈线链路的大型LEO星座的地面段结构。2018年IEEE航空航天会议，第1-14页，2018年。
4. 一、 德尔波蒂略，B.G.卡梅隆和E.F.克劳利。提供全球宽带的三个低地球轨道卫星星座系统的技术比较。《宇航员法》，159:123-1352019年。
5. B.丹比和B.露西亚。轨道边缘计算：纳米卫星星座作为一类新的计算机系统。《第二十五届程序设计语言和操作系统体系结构支持国际会议论文集》，ASPLOS’20，第939-9542020页。
6. B.范、D.G.安徒生和M.卡明斯基。Memc3：紧凑并发的memcache，具有更愚蠢的缓存和更智能的哈希。作为第十届USENIX网络系统设计与实现研讨会（NSDI 13）的一部分，2013年第371–384页。
7. H.Fenech、S.Amos、A.Tomatis和V.Soumpholphakdy。高通量卫星系统：一种分析方法。IEEE航空航天与电子系统汇刊，51（1）：192-2022015。
8. G.Giuliari、T.Klenze、M.Legner、D.Basin、A.Perrig和A.Singla。太空中的互联网骨干。SIGCOMM计算机常见的修订版，50（1）：2020年3月25日*至37日。*
9. 汉德利先生。延迟不是一种选择：空间中的低延迟路由。第17届ACM网络热点研讨会论文集，HotNets’18，第85-91页，美国纽约州纽*约市，2018年。计算机协会。*
10. 汉德利先生。在巨型星座中使用地面中继进行低延迟广域路由。第18届ACM网络热点研讨会论文集，HotNets’19，第125–132页，美国纽约州纽约市，2019年。计算机协会。
11. Y.Hauri、D.Bhattacherjee、M.Grossmann和A.Singla。'；从太空上网”，没有卫星间链接。第19届ACM网络热点研讨会论文集，HotNets’20，第205–211页，2*020。*
12. 胡，王，李，周，罗，丁，姜，王。Lama：优化了键值缓存的位置感知内存分配。2015年USENIX年度技术会议（USENIX ATC 15），第57–69页，2015年。
13. 黄，郭，王。设想用于近地轨道卫星云的无线大数据存储。IEEE无线通信，25（1）：26-312018。
14. 黄，李，刘，戴。云下载：使用云工具为不受欢迎的视频实现高质量的内容分发。在MM.ACM，2011年。
15. 贾，吕，何，黄。利用低轨卫星网络中的卫星间链路进行协作数据下载。IEEE无线通信汇刊，16（3）：1523-1532017。
16. Joshi先生、Y.Liu先生和E.Soljanin先生。关于从编码分布式存储系统下载内容的延迟存储权衡。JSAC，2014年。
17. S.Kassing、D.Bhattacherjee、A.B.A´guas、J.E.Saethre和A.Singla。与hypatia一起探索“太空互联网”。在ACM互联网测量会议论文集中，IMC&20，第214-2292020页*。*
18. T.Klenze、G.Giuliari、C.Pappas、A.Perrig和D.A.Basin。人间天堂，人间天堂。第17届ACM网络热点研讨会论文集，HotNets 20*18，美国华盛顿州雷德蒙德，2018年11月15-16日，第22-28页。ACM，2018年。*
19. V.Kothari、E.Liberis和N.D.Lane。最后一个前沿：太空深度学习。在HotMobile’20：第21届移动计算系统与应用国际研讨会上，美国德克萨斯州奥斯*汀，2020年3月3-4日，第45-49页。ACM，2020年。*
20. 赖，崔，李，李，戴，陈。Tailcutter：在成本限制下，明智地减少云cdn中的尾部延迟。在INFOCOM中。IEEE，2016年。
21. 赖，李，李。星性能：新兴巨型星座的网络性能表征。2020年IEEE第28届国际网络协议会议（ICNP），2020年。
22. 李，黄，刘，王，张，戴。云转码器：弥合互联网视频和移动设备之间的格式和分辨率差距。在NOSSDAV中。ACM，2012年。
23. N、Pachler、J.J.G.Luis、M.Guerster、E.Crawley和B.Cameron。使用粒子群优化分配多波束卫星系统中的功率和带宽。
24. 朴、荣、康、金和李。Cflru：闪存的替换算法。《2006年嵌入式系统编译器、体系结构和综合国际会议论文集》，第234–241页，2006年。
25. 宋，伯格，李，劳埃德。学习内容分发网络缓存的轻松迟来。在第17届USENIX研讨会上

*关于网络化系统设计与实现（NSDI 20），第529–544页，2020年。*

1. A.Sundarajan、M.Kasbekar、R.K.Sitaraman和S.Shukla。支持Midgress的流量供应，用于内容交付。2020年USENIX年度技术会议，USENIX ATC 2020，2020年7月15*-17日，第页*

543–557。USENIX协会，2020。

1. D.瓦西许和R.钱德拉。用于近地轨道卫星的分布式混合地面站网络。HotNets’20，第190–196页，2020年。
2. F.Vidal、H.Legay、G.Gousseis、M.Garcia Vigueras、S.Tubau和J.D.Gayrard。一种在巨型星座用例中对灵活的有效载荷架构进行基准测试的方法。《国际卫星通信与网络杂志》，2020年。
3. O.Vidal、G.Verelst、J.Lacan、E.Alberty、J.Radzik和M.Bousquet。下一代高通量卫星系统。2012年，IEEE首届AESS欧洲卫星通信会议（ESTEL），第1–7页，2012年。
4. 威尔克斯先生。从存储器和动态存储分配。IEEE电子计算机汇刊，（2）：270–2711965。
5. 吴、俞、中央邦H.V.、河滨U。Costlo：具有成本效益的冗余，可降低云存储服务的延迟差异。在NSDI中。USENIX，2015年。
6. 薛、乔夫内斯和王。Cdns meet cn中国cdn部署的实证研究。IEEE接入，5:5292–53052017。
7. H.Zech、F.Heine、D.Trondle、S.Seel、M.Motziemba、R.Meyer和

S.菲利普·梅。EDRS的LCT：Alphasat和Sentinel 1a之间1.8 Gbps的LEO到GEO光通信。在E.M.卡拉佩扎，

P.G.Datskos、C.Tsamis、L.Laycock和H.J.White，《无人/无人值守传感器和传感器网络XI》编辑*；和《高级自由空间光通信技术与应用》，第9647卷，第85-92页。国际光学与光子学会，SPIE，2015。*

1. M.Zhao、P.Aditya、A.Chen、Y.Lin、A.Haeberlen、P.Druschel、，

B.马格斯、B.维申和M.波内克。akamai网络会话中的同行辅助内容分发。2013年互联网测量会议论文集，第31-42页，2013年。