Computer Science and Application 计算机科学与应用, 2020, 10(1), 81-97
Published Online January 2020 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2020.101010

Hans汉斯

Towards Reliable Web Content Delivery against Network Failures: A Path-Aware Peer-Assisted Approach

Chaofeng Huang[1], Wen Hu[2], Lifeng Sun[1]

[1]Computer Science and Technology, Tsinghua University, Beijing
[2]Beijing IQIYI Science & Technology Co., Ltd., Beijing
Email: hcf14@mails.tsinghua.edu.cn

Received: Dec. 24th, 2019; accepted: Jan. 9th, 2020; published: Jan. 16th, 2020

Abstract

The characteristics including the fragmentation and non-cacheable dynamic content of network multimedia present a great challenge to traditional content distribution. Based user's web browsing patterns, in this paper, we propose a dynamic content distribution algorithm based on request forwarding, with WebRTC technology and browser-assist mechanism. We analyse user's browsing behaviors (e.g., including online duration and visiting frequency), application-layer web page request failure, network layer network condition change based on the real-world log system to design a condition-aware redirecting node selection algorithm. The algorithm utilizes the network-layer information, while considering the new characteristics of the Web, and proposes a request forwarding mechanism. By reconstructing the user-user-server distribution path and restoring the failure of web content distribution, "transparency" enhances the system reliability perceived by the client.

Keywords

Web Applications, Networks, WebRTC, Web Content Delivery, Peer-Assisted, Network Failures

针对网络故障的可靠Web分发：一种路径感知的节点协助方法

黄超峰[1]，胡　文[2]，孙立峰[1]

[1]清华大学计算机科学与技术系，北京
[2]爱奇艺科技有限公司，北京
Email: hcf14@mails.tsinghua.edu.cn

文章引用: 黄超峰, 胡文, 孙立峰. 针对网络故障的可靠 Web 分发: 一种路径感知的节点协助方法[J]. 计算机科学与应用, 2020, 10(1): 81-97. DOI: 10.12677/csa.2020.101010

收稿日期：2019年12月24日；录用日期：2020年1月9日；发布日期：2020年1月16日

摘　要

网络多媒体内容的碎片性、动态内容不可缓存性等特性对传统内容分发提出了极大挑战。基于用户网页内容访问模式，本文采用WebRTC技术，利用浏览器协助机制，提出基于请求转发的碎片化动态内容分发算法。本文基于实际系统日志，研究用户的网页浏览行为(包括在线时长分布，重复访问规律)、应用层网页访问故障分布以及网络层网络状态变化，设计节点状态感知的转发节点选择算法。该算法基于网络层信息，同时考虑网页分发的新特性，提出请求转发机制。通过重构用户－用户－服务器的分发路径，恢复网页内容分发故障，“透明”增强用户端感知的系统可靠性。

关键词

网络应用，网络，WebRTC，CDN，转发机制，网络故障

1. 引言

尽管基于 HTTP 的在线服务已经成为当前网络流量的主体部分，但仍然存在各种服务故障的情形(例如，服务器故障[1]，网络故障[2]，路由故障[3])。显著降低了提供给在线用户的服务质量(Quality of Service, QoS)。例如，2013 年谷歌服务器的中断导致 5 分钟内流量下降约 40%。当互联网提供的内容和信息由分布在不同地方的内容分发网络(Content Delivery Network, CDN)提供时，这个问题就更加严重。

针对这些故障，传统的解决方法是：1) 将内容项复制到不同位置的多个服务器，以避免服务器出现故障[4]。例如，当原始请求的服务器没有响应时，将指示用户从不同的服务器下载相同的内容。2) 部署边缘网络代理[5]，例如，用户可以从附近的网关代理下载内容项，该代理可以缓存所请求的内容项。然而，这些方法存在的问题包括：1) 需要额外复制内容，导致额外的部署和操作成本；2) 主要用于解决静态内容项分发的问题；3) 对于网站是不可行的，因为网站不仅包括诸如图像、视频和 SWF (Flash objects)等静态内容，也包含具有个性元素的动态内容项，而动态内容项所固有的不可缓存特性降低了这些方法的性能。

本文提出了一个基于 WebRTC (Web Real-Time Communication，网页即时通信)实现 CDN 和浏览器联合内容分发框架。WebRTC 由万维网联盟(W3C)和互联网工程工作组(IETF)定义，并得到了主要浏览器供应商，如 Google、Opera、Mozilla、Microsoft (正在开发中)的广泛支持[6]。如图 1 所示，通过战略性地选择适当的由 WebRTC 驱动的浏览器以形成“恢复”传递路径(表示为绿色虚线)，当用户未能从原始 CDN 服务器接收到请求的内容项时，这些广泛部署的浏览器可以协助内容分发。与传统的内容分发路径不同，基于浏览器的内容分发路径不仅是通过中枢网络从 CDN 服务器传递到最终用户，而且也包括从最终用户到最终用户的路径。这种转发机制，能够恢复网页访问故障，提升用户端感知的服务稳定性。

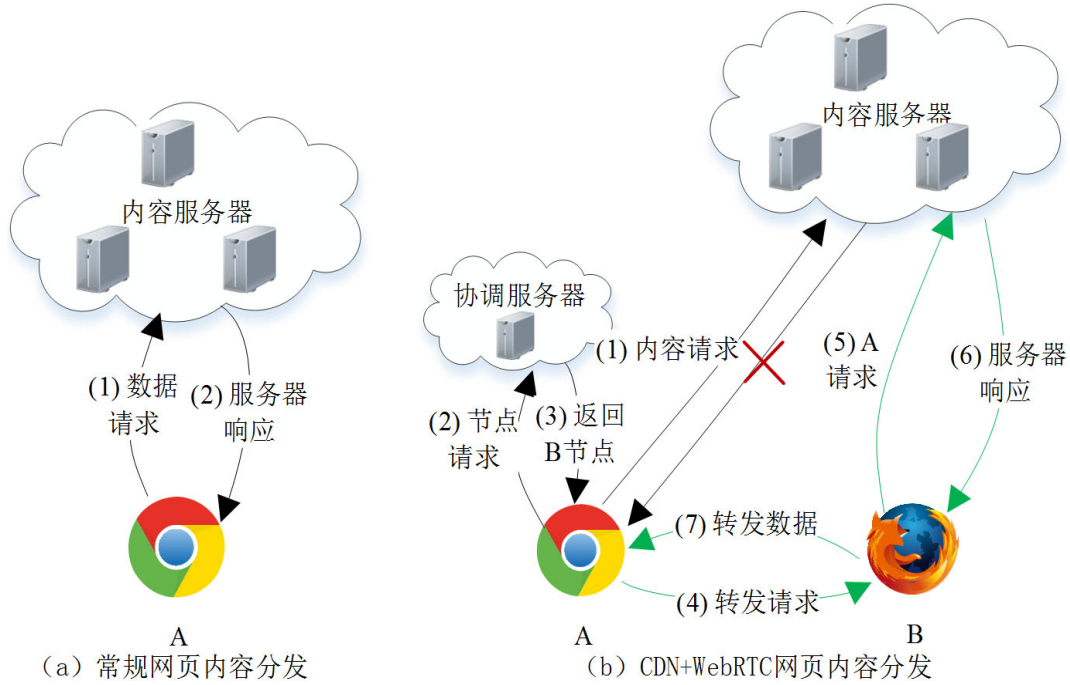


Figure 1. Comparison between conventional content delivery and CDN + WebRTC content delivery
图 1. 常规网页内容分发和 CDN + WebRTC 内容分发的比较

基于实验室之前的研究[7] [8]，本文对用户的网页浏览行为和网页会话特征进行了详尽的测量研究，并详细阐述了原型实现。本文的主要贡献总结如下：

►本文进行大规模的测量研究，以证明在当下的 web 内容分发中网页访问故障现象，从而确定本文设计的必要性。还对用户的网页浏览行为和网页会话特征进行了测量，包括：1) 在线节点的高转换率；2) “重新加入，即稍后再次访问同一网页”的概率很低；3) 网页内容的碎片化，即各种内容类型(如 JavaScript、HTML、CSS、image、Flash 等)放置在不同的服务器上。这些都促使本文提出一种针对 CDN 和节点协助 web 内容分发设计的新的节点选择策略，而不是直接采用传统的 P2P 策略。

►基于测量研究，本文提出了一个联合 CDN 和节点协助的 web 内容分发框架，并将节点选择问题建模为一个优化问题。针对一组有 WebRTC 节点的用户，他们可以在网络故障时形成传递路径。设计了两种启发式算法来解决这个问题。

►通过 WebRTC 实现设计，允许本文策略在流行的浏览器上运行，而不需要用户安装任何插件。此外，在腾讯 QZone(QQ 空间)的两个测试网页中部署了本文的设计，并对其在不同类型网页下的性能进行了评估。本文的设计可以显著地恢复失败的内容分发，例如，在测量中失败的会话比率高达 2%，通过部署本文的设计可以检测和“恢复”失败的事件。此外，还进行了基于仿真的实验，以测试本文的设计在动态和极端运行场景下的有效性。该设计将内容下载率提高到 60%，即便用户位于无法连接到区域 CDN 服务器的区域(例如一个城市)。

2. 测量驱动动机和设计原则

2.1. 测量方法

首先介绍如何进行测量研究。基于腾讯 QZone 在全国范围内部署的 CDN 服务器采集的真实历史数据，研究了用户的网页访问模式。这些记录包括腾讯 QZone 中两个测试网页的用户访问信息，包括 118，

707 个网页访问，其中 2300 个会话遇到网络故障。表 1 列出了每个数据项的信息：1) 用户标识符；2) 用户开始请求网页时的时间节点；3) 用户离开当前网页时的时间节点；4) 内容获取失败的指示器(即网页中包含的内容项是否无法从原始服务器下载)。基于这些线索，本文能够研究节点网络内容分发的挑战和设计原则。

Table 1. Trace items collected from Tencent QZone
表 1. 从腾讯 QZone 收集到的数据项

项目	定义
ID_u	用户的唯一标识符
$C_{request}$	网页 URL
T_b	用户开始请求网页时的时间点
T_e	用户离开当前网页的时间点
Fail	指示用户是否未能下载当前网页中的内容的标识符

2.2. Web 内容分发失败的强随机性

在收集的数据中，本文从时间、位置和互联网服务提供商(ISP)的角度分析了故障分布。在图 2(a)中，绘制每个时间段(6 小时)记录网络故障内容的数量。从图中可以看出，在一天中的不同时间都可能发生。接下来，在图 2(b)中绘制用户在一段时间内遭受下载失败的区域(城市级别)的累积数量。用户经历过网络故障的累积位置不断增加，表明故障位置是地理分布的，即位于不同位置的用户可能遇到下载失败问题。最后，在图 2(c)中，绘制了在一天内(24 小时，4 个时间段)每个时段中经历故障的用户的 ISP (Internet Service Provider，互联网服务提供商)分布。图中看到分布是动态的，这表明故障不受一个 ISP 完全中断的影响。复制/缓存策略的内容传递方案不再适用。

接下来，通过测量用户的网页访问模式，研究了 CDN 和 WebRTC 联合辅助网页分发框架的设计原则。

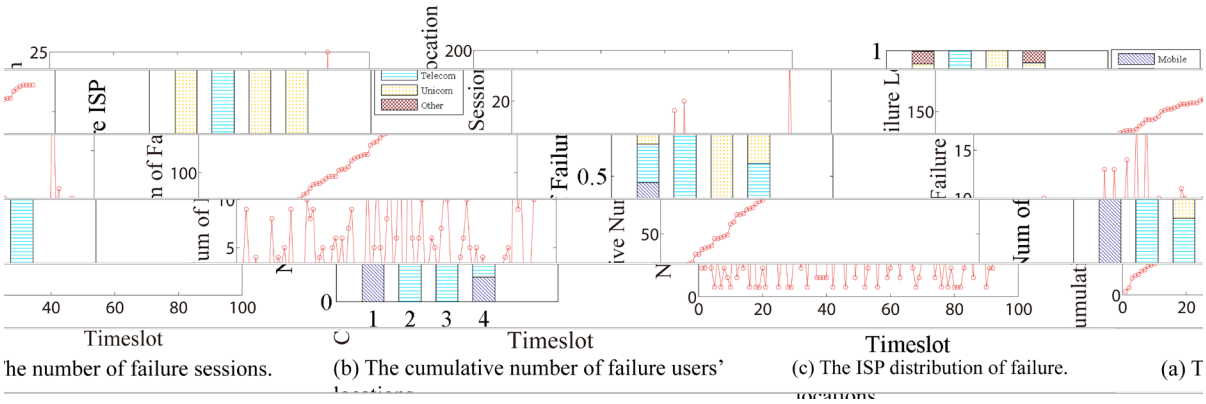


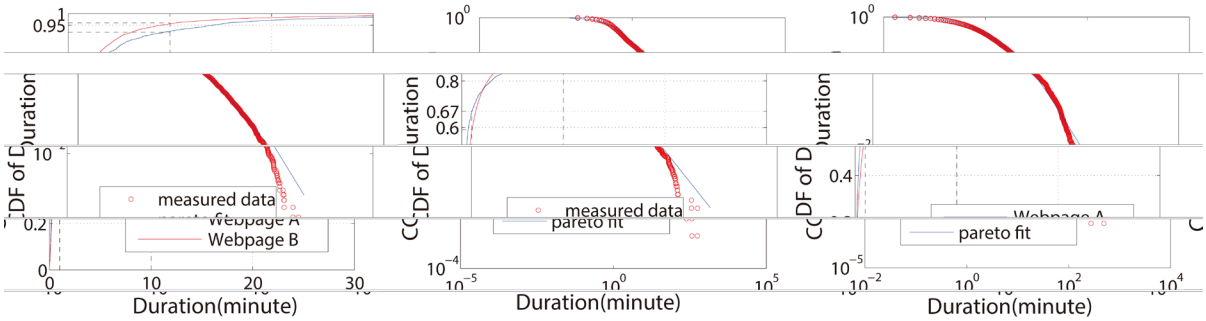
Figure 2. Failure distribution over time. Each timeslot is six hours
图 2. 按时间间隔分发失败的分布。每个时间间隔是 6 小时

2.3. 网页浏览模式

在本文的设计中，节点辅助内容分发是基于 WebRTC 实现的，它要求作为内容分发节点的用户在网页上停留。因此，用户的网页浏览模式对节点的资源可用性有着重要的影响。

2.3.1. 在线时长分布

根据收集的数据记录，每个网页查看事件都被定义为“会话”。首先研究网页浏览会话的持续时间，即用户在网页上停留的时间。本文抽样调查了 97,905 名浏览两种不同类型网页(称为网页 A 和网页 B)的用户。在图 3(a)中，绘制了网页会话持续时间的 CDF (Cumulative Distribution Function，累积分布函数)。



(c) Webpage B. Pareto with α = 1.43 and b = 1.21. (a) CDF of time users spend on viewing Webpage A (b) Webpage A. Pareto with α = 0.88 and b = 0.45.

Figure 3. Distribution of webpage sessions' duration
图 3. 网页会话持续时间的分布

从图 3(a)中观察到：与传统的 P2P 服务和应用相比，网页会话的在线时长要短得多，超过 60%的会话时间小于 1 分钟，超过 90%的会话时间小于 10 分钟。特别是，在表 2 中比较了不同服务和应用程序中节点方的会话持续时间。观察到 P2P 共享文件平均有一个小时的在线时长[9]，P2P 流媒体的平均会话在线时长约为 26 分钟[10]，而网页浏览会话的平均在线时长仅为数十秒。

Table 2. Comparison of session durations in different applications
表 2. 比较不同应用程序中的会话持续时间。

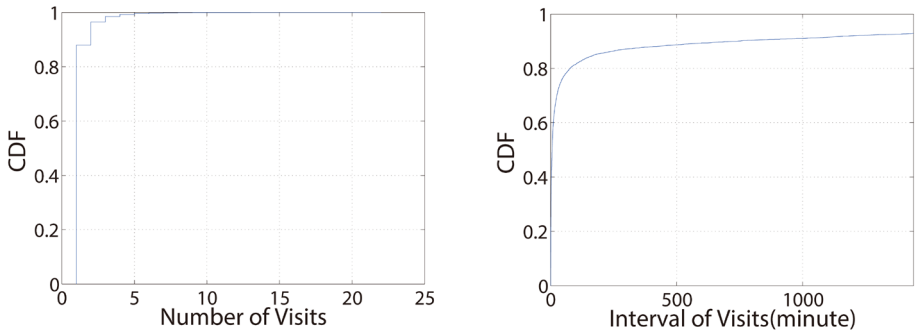
	Applications	Mean Lifetime(s)	Median Lifetime(s)
File Sharing	Napster		3600
	Gnutella		3600
Stream Media	PPLive	393	
	PPStream	1222	
	SOPCast	1861	
	TVAnts	2778	
Webpage Viewing	QZone	158	41

内容对会话持续时间的影响。在测量中，网页 A 和网页 B 包含不同类型的多媒体内容项：网页 A 包含图像和 HTTP 嵌入视频；网页 B 包含用户上传的照片。测量中发现会话持续时间会受内容类型的不同影响，例如，访问网页 B 的用户停留的时间比访问网页 A 的用户稍长。

会话持续时间的帕累托分布(Pareto Distribution)。基于之前对会话持续时间的研究[11]，本文使用移位帕累托分布来拟合网页会话持续时间的分布。数学表达式如下：

$$F(x)=1-\left(1+\frac{x}{b}\right)^{-a},0<x, \tag{1}$$

其中 α 是形状参数，b 是尺度参数。在图 3(b)~(c)中，两个网页在线时长的拟合结果分别是：对网页 A 而言，$\alpha=0.88$，$b=0.45$；对网页 b 而言，$\alpha=1.43$，$b=1.21$。进一步证实了用户停留时间与网页内容有关。



(a) CDF of the number of visits to the same webpage.

(b) CDF of the interval of visits to the same webpage.

Figure 4. Users' re-join pattern in terms of rejoin frequency and rejoin interval
图 4. 用户重复网页访问分布

2.3.2. 重复访问规律

用户重复访问相同网页意味着用户对该网页一些缓存资源可以被重复利用。图 4(a)显示同一用户 14 天内对同一网页的访问次数的 CDF。用户几乎不会重新访问已经浏览过的网页，例如，不到 12% 的用户会在 14 天内返回相同的网页。原因是实验中的两个网页包含“静态”内容项，在测量期间不会发生变化，用户也不会重新访问这些网页来查看更新。对于更新更加频繁的网页，重新加入的可能性会增加。

此外，本文还研究这些重新加入的用户重新访问网页的时间。在图 4(b)中，曲线表示用户对同一网页的两次连续访问之间的间隔的 CDF。超过 60%的重新加入用户在 10 分钟内重复访问同一网页，并且大多数重新加入用户在 1 天内重新访问同一网页。

这些观察结果表明，与传统的 P2P 共享系统(其客户端可能每天返回系统多达数十次(例如，60 次/天[12])相比，网页重新访问的可能性要小得多。这也对本文的 CDN 和节点辅助的 web 内容分发设计提出了挑战，特别是对于更新频率较低的网页。

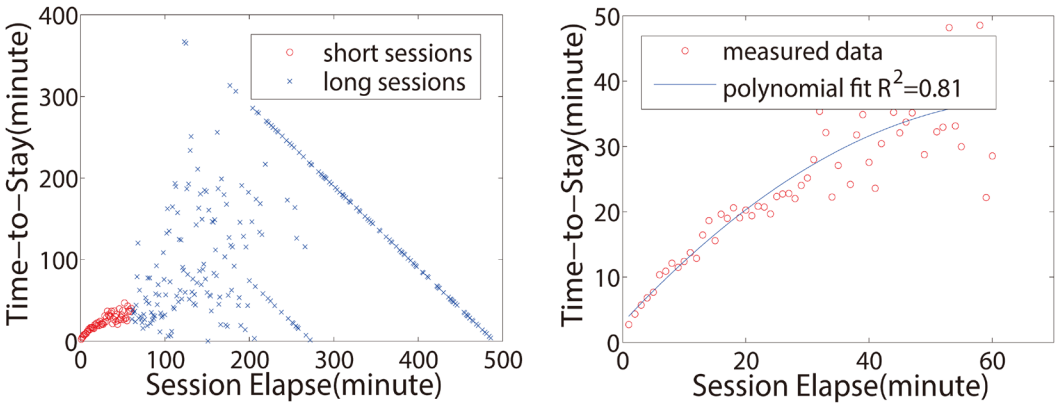
2.3.3. 停留时间与时间相关性较高

由于较短的会话持续时间可能会对本文所提出的 CDN 和节点协助网页内容分发框架的性能产生负面影响，接下来研究如何判断哪些节点可能在系统中停留很长时间，哪些节点可能很快离开网页。深入查看了数据集中的网页会话在线时长，通过将浏览网页的用户的会话在线时长随机分为两部分：已在线时长(定义为用户已在网页上停留的持续时间)和剩余在线时长(定义为用户将在网页上停留的持续时间)，实验生成一组(已在线时长，剩余在线时长)的会话样本。如图 5 所示，S_a、S_b、S_c 和 S_d 是由随机选择的时间点 t_s 分割的会话，并且($t_s - b_i$、$e_i - t_s$)是生成的样本对。

b_i: 会话 i开始时间节点
t_s: 随机分割点
e_i: 会话 i 结束时间节点
Li: 会话 i持续时间

Figure 5. Illustration of session split. elapse = $t_s - b_i$ and time-to-stay = $e_i - t_s$
图 5. 会话切割方法，已在线时长 $= t_s - b_i$，剩余在线时长 $= e_i - t_s$

在图 6(a)中，绘制了 30,000 对样本随机落入 330 个 1 分钟时长的时间段(即 30,000 对的会话经过时间随机落入 330 个一分钟长度的经过段中)。将短于 60 分钟的会话标记为红色，长于 60 分钟的会话标记为蓝色。可以观察到，对于短会话，停留时间与经过时间高度相关，这表明如果用户在网页上已经花费了很长的时间，他/她倾向于继续停留在网页上。



(a) All sessions. (b) Sessions elapse in [0, 60].

Figure 6. Relationship between time-to-stay and elapse
图 6. 用户已在线时长和剩余在线时长的关系

此外，还深入研究了短时间样本(即在线时长 < 60 分钟)的更多细节，这些样本占所有数据的 97%。如图 6(b)所示，多项式模型 $y=-0.0076x^2+0.97x+3.5$ 与这些样本非常吻合，进一步验证了停留时间与流逝时间之间的相关性。

这些观察和分析表明，可以准确预测网页会话的停留时间，并在联合 CDN 和节点协助分发设计中选择“稳定”的同伴。

3. 测量驱动动机和设计原则

在这一部分中，将分析 CDN 和节点辅助框架中的节点选择问题。本文将节点选择问题描述为一个整数规划问题，并设计了两个启发式算法来求解。

3.1. 框架

本文提出了路径感知的节点辅助 Web 内容分发的框架，WebRTC 节点能够以最大化成功的传递率和系统吞吐量帮助其他用户从 Web 服务器获取 Web 内容项。本文将内容分发框架与图 1 中的传统 web 内容分发范式进行了比较。图 1(a)示出了传统的内容交付模式，其中用户直接从相应的 CDN 服务器下载包含在网页中的内容项。图 1(b)展示了我们的对等辅助交付模式：在分发失败(表示为红十字)的情况下，用户(例如，Alex)将尝试通过恢复的路径(表示为绿色虚线)从其他 WebRTC 对等方(例如，Bob)接收 web 内容项。

3.2. 问题表述

本文设计的主要思想是战略性地将具有可用节点资源(如上行带宽容量)的 WebRTC 对等节点分配给遭遇传输失败的用户。因此，为每个失败的用户通过仔细的节点选择生成候选节点列表在设计中至关重要。在深入了解更多细节之前，先介绍了公式中使用的几个定义，并总结了表 3 中的重要符号。

Table 3. Trace items collected from Tencent QZone
表 3. 从腾讯 QZone 收集到的数据项

符号	定义
B	浏览器间带宽矩阵
$b_{ij}(t)$	在时刻 t，浏览器 j 到浏览器 i 的上传带宽
P	转发节点选择矩阵
$p_{ij}(t)$	t 时刻节点 j 是否选择节点 i 为其转发
$b_{wi}(t)$	t 时刻节点 i 的总上传带宽
p_s	请求节点
p_r	转发节点
τ_i	节点 i 的已在线时长
R	候选节点转发列表
R_I	算法选择转发列表的第一部分
R_II	算法选择转发列表的第二部分
α	R_I 与 R_II 的比例
γ	转发节点的负载阈值
ζ	转发节点列表的长度

3.2.1. 符号定义

首先，定义了一个所有节点间的带宽矩阵 $B^{M\times N}$，每个元素 $b_{ij}(t)$表示在 t 时刻，从 WebRTC 节点 j 到用户节点 i 的上传带宽。将节点选择策略定义为节点选择策略矩阵 $P\in\{0,1\}^{N\times M}$，P 的每个元素都是一个二值变量，即 $p_{ij}(t)=1$，表示在 t 时刻节点 j 选择节点 i 为其转发；否则，$p_{ij}(t)=0$。M 表示发生故障的节点个数，N 表示提供转发服务的节点个数。在本文的设计中，节点选择是随着时间推移产生的，即节点选择矩阵 P 根据预计的 P2P 带宽而改变。

3.2.2. 优化方案

本文将节点选择问题建模成一个条件约束优化问题，如下所示

$$\max_{P_{ij}(t)}\sum_{i=1}^{n}\sum_{j=1}^{m}p_{ij}(t)b_{ji}(t) \tag{2}$$

服从

$$\sum_{i=1}^{n}P_{ij}(t)\geq 1 \quad \forall j\in\{1,\cdots,m\} \tag{3}$$

$$\sum_{j=1}^{m}P_{ij}(t)b_{ji}(t)\leq b_{wi} \quad \forall i\in\{1,\cdots,n\} \tag{4}$$

优化的基本原理是合理选择最优的可转发节点，为每个获取内容失败的节点快速恢复网络故障，提升用户服务体验。条件(3)保证所有发生故障的节点都能找到一个转发节点。条件(4)保证中间转发节点所提供的上传带宽不能超过自身的上传容量限制。

为了解决这个整数规划问题，本文设计了如下启发式算法：请求节点 p_s 从协调服务器获取转发节点列表 R，该列表根据转发节点 p_r 和 p_s 之间的容量、当前工作负载和 p_r 的停留时间对列表进行排序。然后

p_s 按照排序顺序请求列表中的每个 p_r。通过这种方式，p_r 将获得更高的优先级，以确保更高的成功率和带宽来完成分发失败的内容项，p_r 具有更大的备用容量，并有望保持更长的时间。

3.3. 路径感知节点选择策略

启发式和分布式算法工作如下：1) 基于各种节点选择因素，为请求用户生成请求节点列表；2) 然后，请求用户主动尝试这些候选节点下载失败的 web 内容。接下来分别阐述这两个步骤。

3.3.1. 节点选择因素

由于网络状态的实时测量开销过大，一些基本信息(地理位置和互联网服务提供商(ISP)等)可为网络状态提供了有价值的依据[13]。本文的设计中，选择与请求节点处于同一位置的转发节点和 ISP，以获得更好的网络性能。特别是，同一 ISP 还降低了跨 ISP 成本[14]。

另外，只有确保转发节点在整个服务的过程中持续在线，才能利用其资源，帮助恢复网络故障。因此，必须估计一个节点在线的时间。在第 2.3.3 节的测量研究中，发现会话的剩余在线时长与会话的已在线时长高度相关。基于这一观察，根据候选转发节点已经浏览网页的持续时间来对其进行优先级排序。因此，所选择的转发节点更可能在内容分发阶段处于联机状态。

根据测量研究，内容分发失败在时间、地点和 ISP 方面表现出随机性。为了增加潜在内容分发路径的多样性，本文 1) 选择位于不同位置的转发节点来克服区域网络故障；2) 选择具有不同 ISP 的转发节点，以便转发节点可以改进与原始内容服务器的连接；3) 选择不同负载的节点，实现系统中所有转发节点的负载均衡。

3.3.2. 转发节点列表的生成

基于上述因素，当存在来自节点的转发请求时，涉及两个步骤。

- 请求节点 p_s 在候选集合 R 中选择转发节点 p_r，候选集从候选服务器中获取；
- 根据候选转发节点的工作量和停留时间排列列表。

算法 1 总结了由调度服务器执行的节点选择算法。给定输入参数 ζ，即转发节点列表的长度，生成包含两个节点集的列表：仔细选择的节点(R_{I})和随机选择的节点(R_{II})。α 是每组的比率，可根据特定的网络环境进行调整。请注意，R_{I} 由位于同一城市的节点组成，由与请求节点相同的 ISP 提供服务(第 3 行)，以实现更好的网络性能；R_{I} 由随机选择的节点组成(第 7 行)，以增加潜在内容传递路径的多样性。在两个集合(R_{I}, R_{II}) (第 10 行，第 11 行)过滤那些经历了一些内容获取失败或更高工作负载超过阈值 γ (即它已经转发了数个节点)的转发节点。此外，为了选择具有更多空闲带宽且在线时间更长的节点，本文将这两部分按停留时间 τ_i (第 12 行)的降序排序。

3.3.3. 与转发节点进行内容转发

算法 2 总结了请求节点所携带的算法。候选转发列表中的第一个节点充当主节点，其余节点充当后备。请求节点 p_s 首先请求主节点获取失败的内容(第 3 行)，然后逐一尝试候选列表中的其他候选对象，直到最终获取内容(第 7 行)。

Algorithm 1. Relay peer list generation
算法 1. 候选节点列表生成算法

1：	procedure relay-peer-selection(α, γ , ζ)
2：	for i = 0 to $\zeta * \alpha$ do
3：	select p_r which has the same location and ISP with p_s, from online peer set

Continued

4：	$R_{\mathrm{I}} \leftarrow R_{\mathrm{I}} \cup p_r$
5：	end for
6：	for $i = (\zeta * \alpha + 1)$ to ζ do
7：	select p_r from online peer set randomly
8：	$R_{\mathrm{II}} \leftarrow R_{\mathrm{II}} \cup p_r$
9：	end for
10：	filter peers which experienced some content fetching failure in R_{I}, R_{II}, respectively
11：	filter peers whose workload exceed the threshold γ in R_{I}, R_{II}, respectively
12：	sort R_{I}, R_{II} in descending order of τ_i
13：	$R \leftarrow R_{\mathrm{I}} \cup R_{\mathrm{II}}$
14：	return R
15：	end procedure

Algorithm 2. Content fetching with relay candidates
算法 2. 基于请求转发的内容获取算法

1：	procedure fetch-content (R)
2：	for each relay peer $p_r \in R$ do
3：	send the content url to the first relay peer in R
4：	if p_r can get the content from server and relay to requesting peer p_s then
5：	break
6：	else
7：	try the next relay peer in R
8：	end if
9：	end for
10：	end procedure

3.4. 实现与讨论

在 WebRTC 浏览器和一个专用的调度服务器上实现本文的设计。

3.4.1. 系统组件

系统的三个主要组件的功能如下所示。

- 内容服务器：为用户提供内容的服务器。注意，为了支持节点协助的内容转发，需要在正常的 web 页面源代码中插入一些 javascript 代码。
- 调度服务器：本系统最重要的模块是：1) 当用户访问特定网站时分配用户标识符 ID_u，并记录用户行为，如表 1 和表 4 所示；2) 当一些失败用户请求转发时，根据节点选择算法生成候选转发节点列表 R；3) 负责在请求用户(p_s)和转发节点方(p_r)之间建立直接连接。
- 客户：由 WebRTC 支持的 web 浏览器，例如 Chrome、Firefox。不需要安装任何插件或其他第三方软件，以便 web 浏览器相互通信。

3.4.2. 系统实现方式

本文利用 peerjs [15]库，将许多本地 WebRTC 函数封装到自定义 Javascript 中，实现浏览器通信。为了整合节点选择策略，网页需要包含进一步定制 Javascript 库的参考，并在网页源代码中做一些修正。例如，传统用于加载图片的 HTML 代码如下。

```
<img id="-id-" src="-src-">
```

结合本文的设计，即内容传递失败检测和候选节点选择策略，修改后的 HTML 代码如下。

```
<img id="-id-">
<script type="text/javascript">
if (!$("#id").load("-normal-load-function-")
$("#id").load("-peer-assisted-load-function-")
</script>
```

一旦内容下载失败，浏览器将运行本文的策略并从协调服务器请求转发节点。请注意，网络地址转换(Network Address Translation, NAT)问题是传统 P2P 系统面临的主要挑战，大约 24%的节点[16]是在对称 NAT 环境下发现的。该问题超出了本文讨论范围，留待以后探讨。本文简单地通过在候选转发节点列表中的节点之间建立多个并行连接来减轻问题，以提高成功转发率。

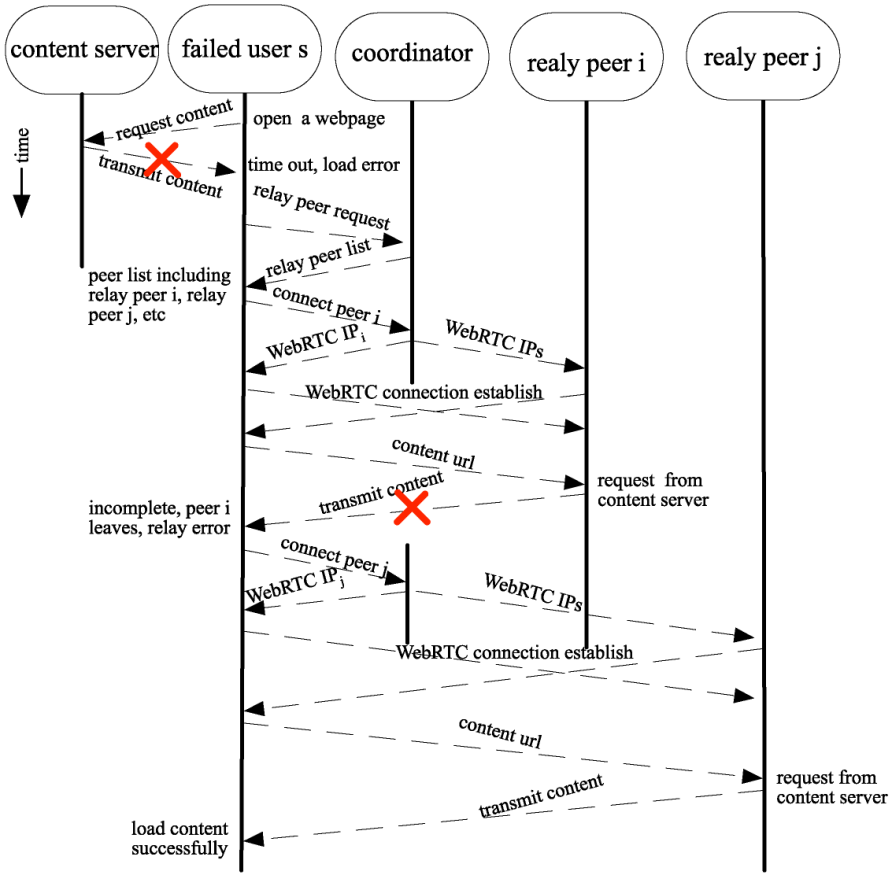


Figure 7. The workflow of our prototype implementation
图 7. 原型系统工作流程图

4. 性能评估

通过在腾讯 QZone 上运行的原型实现和仿真实验，验证了设计的有效性和性能。

4.1. 原型试验

根据第三节提出的算法，在腾讯 QZone 上实现了本文的设计，并收集了算法工作的轨迹。

如图 7 所示，在从原始服务器(在图 7 中表示为第一个红十字)检测到内容下载失败之后，用户联系协调服务器以获得候选中继节点列表；然后，用户尝试向列表中的这些候选节点发出请求以获取中继失败的内容(注意，候选节点也可能遭受失败，在图 7 中表示为第二个红十字)，直到最终下载 web 内容为止。基于这些记录，能够绘制出内容下载的失败轨迹。如图 8 所示，每个标记表示提交给 ACM 的样本的其中一个失败案例。在本文的记录中，大约 2%的内容下载遇到分发失败，本文实现能够检测并“恢复”这些故障。

Figure 8. The locations where content download failures occur
图 8. 系统探测到并恢复的网页访问故障地理分布

4.2. 模拟实验

4.2.1. 实验建立

为了研究其在极端场景下的性能细节，本文进行了仿真实验。实验模拟了分布在中国五个城市的 5000 个节点，根据城市的地理位置，即经纬度信息，计算出每个节点对之间的物理距离。使用距离和延迟之间的相关性估计节点对之间的延迟[17]。为了模拟真实的互联网环境，根据[18]统计数据，在上行链路和下行链路容量方面对节点进行了异构配置。如表 4 所示。每个用户的 ISP 被随机分配给 3 个 ISP。根据[19]中的调查，平均网页为 1600 KB，由 112 个对象组成。这里分析了由于网页大小的快速增长，内容大小从 500 KB 到 16,000 KB 不等。

Table 4. Bandwidth capacity and distribution of users
表 4. 带宽容量和用户分布

Category I	Category I	Category II	Category III
Downlink(kbps)	784	1500	3000
Uplink(kbps)	128	384	1000
Fraction of Users	20%	50%	30%

用户行为：在实验中，每个节点访问网页模式由 Poisson 过程(参数 = 30)生成；在线时长由拟合的 Pareto 分布产生。如无其他规定，$\alpha = 0.2$，$\gamma = 0.8$，$\zeta = 10$ [20]。

网页访问故障：假设用户从区域 CDN 服务器下载 web 内容项，其中用户根据其位置和 ISP 下载内容项被重定向到区域节点服务器。模拟网络故障如下。网络故障[21]是指用户所在区域的网络遇到一些问题，并且该区域的一小部分用户无法连接到服务器。图 9 示出了这样的故障的示例：用户 u1 和用户 u2 不能从服务器下载内容并相互连接；但是，其他用户/节点(例如 u3、u4)能够连接到 u1、u2 和服务器。本文的实验中，如果没有特别说明，网络内的故障率是 60%。

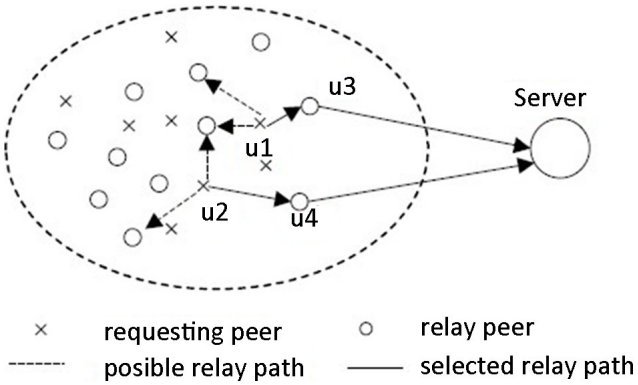


Figure 9. Illustration for in-network failures
图 9. 网络中间故障示例

对比算法：将设计与两个方案进行比较，即：1) 无转发机制的内容分发算法：用户直接从 CDN 服务器下载内容项；2) 转发节点随机选择策略：即发生故障的用户随机从在线的节点中选择节点发送转发请求；3) 本文的设计：发生故障的用户根据策略向节点发送转发请求。接下来，介绍实验结果。

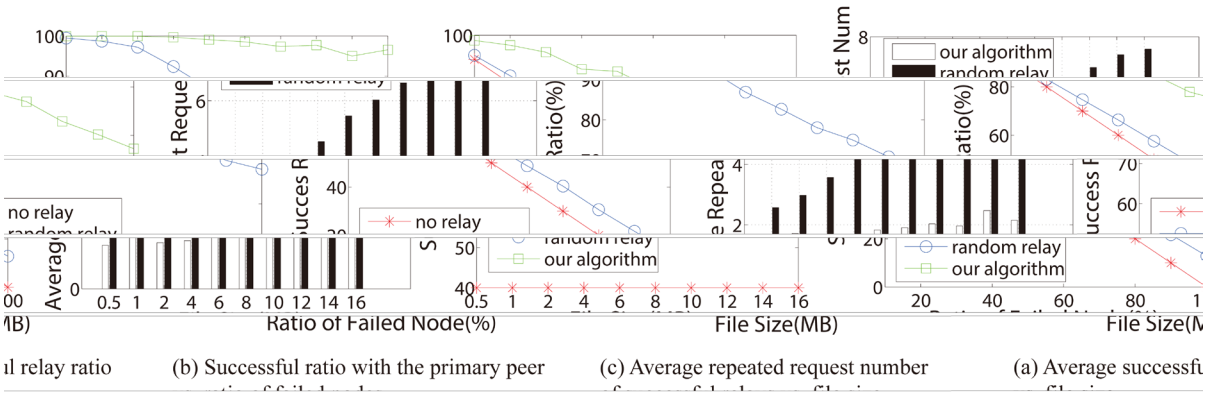


Figure 10. The performance comparison among different algorithms when in-network failures occur
图 10. 在发生网络故障时不同算法性能对

4.2.2. 实验结果

内容获取成功率定义为CDN服务器或节点端最终可以服务的web内容请求的比率。如图10(a)所示，曲线表示不同策略的内容获取成功率与 web 内容大小的关系。本文的设计优于转发节点随机选择策略和无转发机制的内容分发算法。特别地，当 web 内容大小约为 16 MB 时，该设计相对于转发节点随机选择策略将成功的内容下载率提高了 25%以上。

研究网络内故障率对主节点(即候选转发节点列表中的第一个转发节点)的成功转发率的影响。如图10(b)所示，曲线表示成功转发率与网内故障率。随着网络故障率的增加，成功转发率降低。特别是，与本文策略相比，转发节点随机选择策略和无转发机制的内容分发算法的成功转发率下降得更快。当网内故障率达到100%时，即某一特定区域内的所有用户都无法连接到CDN服务器，本文的设计最终分别比两种参照算法好50%和60%。

进一步研究故障用户成功获取内容所尝试发送转发请求的次数，它定义为请求用户向候选转发节点发送的重复请求数，直到成功获取内容为止。在图10(c)中，条形图表示请求用户尝试的请求数与web内容项的文件大小。本文提出的算法最好的情况下，尝试次数是随机算法的1/3。原因是，在设计本文算法时，尽可能将“优质”节点放置在候选转发节点列表的开始位置，从而减少用户从上到下尝试的次数，增强了网页访问故障恢复的实时性，提升用户体验。

5. 相关工作

在这一部分中，将对对等网络内容分发的相关工作进行调查。

5.1. WebRTC支持的内容分发

WebRTC允许直接的浏览器到浏览器的通信，从而导致产生越来越多的节点协助web应用程序。2017有超过10亿个端点在使用中支持WebRTC [3]；2018年有约47亿个设备将支持WebRTC [22]。学者一直在研究使用WebRTC浏览器来协助内容分发，如下所示。

Zhou等人[23]开发了WebCu云，它通过利用用户浏览器来分发内容来分散Web内容交换；然而，它需要部署在每个ISP区域内的重定向代理。Zhang等人[24]提议，该系统可以自动招募网络访客来协助服务器，以降低网站运营成本。费尔南德斯等人[25]旨在在不同平台(如桌面WWW平台和智能手机)上实现当前碎片化顶级(OTT)实时通信服务(如Skype、谷歌Hangouts、苹果FaceTime)之间的有效融合。他们基于GStreamer多媒体堆栈[26]构建了一个支持WebRTC的高级媒体服务器，并演示了WebRTC能够以无缝和简单的方式与其他移动和桌面实时通信服务进行互操作。Vogt等人[27]引入了一种分散的、基于名称的内容共享架构，即基于浏览器的开放发布(BOPlish)，以利用WebRTC API的web浏览器内置功能。Wichtlhuber等人[28]采用群体工作的方法研究了基于WebRTC的分布式流媒体系统中的激励问题。

以往基于WebRTC的研究工作的局限性在于，主要集中在缓解基于P2P理念的原始内容服务器，即节点之间共享带宽，未能找到解决网络故障问题的方法，这是本文研究的重点。

5.2. 弹性内容分发

针对一些突发性的网络故障，人们已经开展了一些工作来寻求弹性内容分发网络环境。

为了获得更高的可靠性和更好的性能，许多内容发布商采用了内容多宿主和ISP多宿主技术。Adhikari等人[29]对使用内容多宿主的Netflix进行了测量，得出结论认为，使用内容多宿主确实有潜在的性能优势。Akella等人[30]提出了双ISP多宿主网络可以将非多宿主网络的可用性提高9%左右。然而，多宿主也需要增加更多的基础设施投资，从而带来更多的经济成本，并且不适用于动态内容分发。而ISP多址技术也提出了inter-AS路由中的非聚集性问题，这是导致路由表爆炸的主要原因，需要新的协议[31]来实现实际部署。

谢等人[32]利用软件定义网络(SDN)技术构建抗灾网络。然而，这些系统需要用支持SDN的路由器替换网络中的普通路由器，即这些系统不能在任意IP网络上使用。

5.3. 节点协助的内容分发

从节点协助的内容分发方面来看，Ly 等人[33]开发了一个间接转发节点系统(IRS)，通过迂回路径转发游戏状态更新，减少玩家之间的往返时间。Wang 等人[34]提出了有效的点对点机制，同时考虑到了玩家之间的社会关系和虚拟游戏的历史贡献水平。Xu 等人[35]研究了无线电蜂窝网络中的转发协助下行链路多用户视频流，其中将基站和多个转发节点协同用于流式视频。在之前的研究中，他们都假设节点在网上停留足够长的时间来转发内容项，这在 web 内容分发中是不正确的。因为节点的在线时间是内容分发的关键因素[36]。本文通过主动预测节点在系统中的停留时间来选择节点协助内容分发。

Cheng 等人[37]表明，短视频剪辑表现出极大的差异性，这使得现有传统的视频项目内容用点对点流媒体的解决方案次优，并提出了 NETBube，一个基于新的节点协助分发框架利用社会网络的短视频共享。Xu 等人[38]开发了一个工作平台 PPVA，它可以全面探索聚合视频和客户端资源站点，以实现普遍和透明的点对点加速。邱等人[39]针对大规模点对点视频点播(P2P-VoD)流媒体应用，提出了 InstantLeap，它根据节点的回放点将节点划分为多个组，每个节点保持与不同组节点的连接。他们声称 InstantLeap 可以同时实现低维护成本和快速的同行搜索。Moraes 等人[40]提出了一种 P2P 视频点播系统的特定节点选择机制，称为 LIPS，它根据候选节点的到达时间来选择节点，以增加寻找到感兴趣的节点的概率。他们还展示了 LIPS 相对于随机选择机制在块可用性和存储成本方面的优势。Cui 等人[41]同时考虑了节点的带宽和选择节点的延迟。然而，这些工作主要集中在视频内容的传递上，不适用于 web 内容的传递，特别是网站中的动态内容传递。

6. 结论

在前网络上基于 HTTP 的 web 分发通常被用来分发各种多媒体内容项，但它却受到网络和服务器上发生故障的困扰。本文提出了一个联合 CDN 和节点协助的 web 内容分发框架，实现了基于请求转发的动态内容算法。与传统的以减轻 CDN 服务器带宽负载为主的节点协助方法不同，本文的贡献在于首次研究了一种基于浏览器的节点协助方案来解决内容发布故障问题。在对用户访问和浏览网页的大规模测量研究的基础上，本文不仅展示了传统 P2P 策略无法直接解决的设计挑战(例如，节点停留在网页上的时间极短)，还学习了网页浏览模式和设计原则。另外，本文将节点选择问题描述为一个优化问题，并设计基于测量洞察力的启发式解决算法。本文在腾讯 QZone 的原型实现和仿真实验证明了该设计的有效性，与转发节点随机选择策略和无转发机制的内容分发算法相比，本文的设计显著提高了网络故障下的成功交付率。

参考文献

[1] Facebook Outage (2010) https://www.facebook.com/notes/facebook-engineering/more-details-on-todays-outage/431441338919

[2] http://www.w3.org/2011/04/webrtc-charter.html

[3] WebRTC Plugin-Free Realtime Communication (2013). http://gotocon.com/dl/goto-aar-2013/slides/SamDutton_RealtimeCommunicationWithWebRTC.pdf

[4] Sivasubramanian, S., Szymaniak, M., Pierre, G. and van Steen, M. (2004) Replication for Web Hosting Systems. ACM Computing Surveys (CSUR). https://doi.org/10.1145/1035570.1035573

[5] Ihm, S. (2011) Understanding and Improving Modern Web Traffic Caching. Ph.D. Thesis, Princeton University, Princeton, NJ.

[6] Alexandru, C. (2014) Impact of WebRTC (P2P in the Browser). Internet Economics VIII (2014), 39.

[7] Hu, W., Wang, Z. and Sun, L.F. (2015) Path-Aware Peer-Assisted Web Content Delivery against Network Failures. 2015 *IEEE* 23*rd International Symposium on Quality of Service*, Portland, OR, 15-16 June 2015, 79-80. https://doi.org/10.1109/IWQoS.2015.7404695

[8] Hu, W., Wang, Z. and Sun, L.F. (2016) Towards Network-Failure-Tolerant Web Content Delivery: A Path-Aware Peer-Assisted Approach. 2016 *IEEE Global Communications Conference*, Washington DC, 4-8 December 2016, 1-6. https://doi.org/10.1109/GLOCOM.2016.7842347

[9] Saroiu, S., Gummadi, K.P. and Gribble, S.D. (2003) Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts. *Multimedia Systems*, **9**, 170-184. https://doi.org/10.1007/s00530-003-0088-1

[10] Silverston, T. and Fourmaux, O. (2007) Measuring p2p IPTV Systems. 2017 *ACM NOSSDAV*, 1-6.

[11] Wang, F., Liu, J.C. and Xiong, Y.Q. (2008) Stable Peers: Existence, Importance, and Application in Peer-to-Peer Live Video Streaming. *IEEE INFOCOM* 2008-*The* 27*th Conference on Computer Communications*, Phoenix, AZ, 13-18 April 2008, 1364-1372. https://doi.org/10.1109/INFOCOM.2008.194

[12] Stutzbach, D. and Rejaie, R. (2006) Understanding Churn in Peer-to-Peer Networks. *Proceedings of the* 6*th ACM SIGCOMM Conference on Internet Measurement*, October 2006, 189-202. https://doi.org/10.1145/1177080.1177105

[13] Hu, W., Wang, Z. and Sun, L.F. (2015) Guyot: A Hybrid Learning- and Model-Based RTT Predictive Approach. 2015 *IEEE International Conference on Communications*, London, 8-12 June 2015, 5884-5889. https://doi.org/10.1109/ICC.2015.7249260

[14] Bindal, R., Cao, P., Chan, W., Medved, J., Suwala, G., Bates, T. and Zhang, A. (2006) Improving Traffic Locality in BitTorrent via Biased Neighbor Selection. 26*th IEEE International Conference on Distributed Computing Systems*, Lisboa, Portugal, 4-7 July 2006, 66.

[15] (2017) http://peerjs.com

[16] Huang, Y., Fu, T.Z.J., Chiu, D.-M., Lui, J.C.S. and Huang, C. (2008) Challenges, Design and Analysis of a Large-Scale P2P-VOD System. *ACM SIGCOMM Computer Communication Review*, **38**, 375-388. https://doi.org/10.1145/1402958.1403001

[17] Katz-Bassett, E., John, J.P., Krishnamurthy, A., Wetherall, D., Anderson, T. and Chawathe, Y. (2006) Towards IP Geolocation Using Delay and Topology Measurements. *Proceedings of the* 6*th ACM SIGCOMM Conference on Internet Measurement*, October 2006, 71-84. https://doi.org/10.1145/1177080.1177090

[18] Zhang, M., Xiong, Y.Q., Zhang, Q., Sun, L.F. and Yang, S.Q. (2009) Optimizing the Throughput of Data-Driven Peer-to-Peer Streaming. *IEEE Transactions on Parallel and Distributed Systems*, **20**, 97-110.

[19] (2014) http://www.websiteoptimization.com/speed/tweak/average-web-page

[20] Magharei, N. and Rejaie, R. (2006) Understanding Mesh-Based Peer-to-Peer Streaming. *Proceedings of the* 2006 *International Workshop on Network and Operating Systems Support for Digital Audio and Video*, May 2006. https://doi.org/10.1145/1378191.1378204

[21] Dahlin, M., Chandra, B.B.V., Gao, L. and Nayate, A. (2003) End-to-End WAN Service Availability. *IEEE/ACM Transactions on Networking*, **11**, 300-313. https://doi.org/10.1109/TNET.2003.810312

[22] (2014) http://www.websiteoptimization.com/speed/tweak/average-web-page

[23] Zhou, F.F., Zhang, L., Franco, E., Mislove, A., Revis, R. and Sundaram, R. (2012) WebCloud: Recruiting Social Network Users to Assist in Content Distribution. 2012 *IEEE* 11*th International Symposium on Network Computing and Applications*, Cambridge, MA, 23-25 August 2012, 10-19. https://doi.org/10.1109/NCA.2012.41

[24] Zhang, L., Zhou, F.F., Mislove, A. and Sundaram, R. (2013) Maygh: Building a CDN from Client Web Browsers. *Proceedings of the* 8*th ACM European Conference on Computer Systems*, April 2013, 281-294. https://doi.org/10.1145/2465351.2465379

[25] Lopez Fernandez, L., Paris Diaz, M., Benitez Mejias, R., Lopez, F.J. and Santos, J.A. (2013) Kurento: A Media Server Technology for Convergent WWW/Mobile Real-Time Multimedia Communications Supporting WebRTC. 2013 *IEEE* 14*th International Symposium on* "*A World of Wireless*, *Mobile and Multimedia Networks*", Madrid, Spain, 4-7 June 2013, 1-6. https://doi.org/10.1109/WoWMoM.2013.6583507

[26] GStreamer (2017) http://www.gstreamer.net/

[27] Vogt, C., Werner, M.J. and Schmidt, T.C. (2013) Content-Centric User Networks: WebRTC as a Path to Name-Based Publishing. 2013 21*st IEEE International Conference on Network Protocols*, Goettingen, Germany, 7-10 October 2013, 1-3. https://doi.org/10.1109/ICNP.2013.6733652

[28] Wichtlhuber, M., Aleksandrov, N., Franz, M., Hinz, O. and Hausheer, D. (2016) Are Incentive Schemes Needed for WebRTC Based Distributed Streaming?: A Crowdsourced Study on the Relation of User Motivation and Quality of Experience. *Proceedings of the* 7*th International Conference on Multimedia Systems*, May 2016, 1-12. https://doi.org/10.1145/2910017.2910598

[29] Adhikari, V.K., Guo, Y., Hao, F., Varvello, M., Hilt, V., Steiner, M. and Zhang, Z.-L. (2012) Unreeling Netflix: Under-

standing and Improving Multi-CDN Movie Delivery. 2012 *Proceedings IEEE INFOCOM*, Orlando, FL, 25-30 March 2012, 1620-1628. https://doi.org/10.1109/INFCOM.2012.6195531

[30] Akella, A., Pang, J., Maggs, B., Seshan, S. and Shaikh, A. (2004) A Comparison of Overlay Routing and Multihoming Route Control. ACM SIGCOMM, 1-14.

[31] Gummadi, R. and Govindan, R. (2005) Practical Routing-Layer Support for Scalable Multihoming. *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, FL, 13-17 March 2005, 248-259.

[32] Xie, A., Wang, X.L., Wang, W. and Lu, S.L. (2014) Designing a Disaster-Resilient Network with Software Defined Networking. 2014 *IEEE 22nd International Symposium of Quality of Service*, Hong Kong, 26-27 May 2014, 135-140. https://doi.org/10.1109/IWQoS.2014.6914312

[33] Ly, C., Hsu, C.-H. and Hefeeda, M. (2011) IRS: A Detour Routing System to Improve Quality of Online Games. *IEEE Transactions on Multimedia*, **13**, 733-747. https://doi.org/10.1109/TMM.2011.2114645

[34] Wang, Z., Wu, C., Sun, L.F. and Yang, S.Q. (2011) Peer-Assisted Online Games with Social Reciprocity. 2011 *IEEE Nineteenth IEEE International Workshop on Quality of Service*, San Jose, CA, 6-7 June 2011, 1-9. https://doi.org/10.1109/IWQOS.2011.5931316

[35] Xu, Y., Hu, D. and Mao, S. (2014) Relay-Assisted Multiuser Video Streaming in Cognitive Radio Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, **24**, 1758-1770. https://doi.org/10.1109/TCSVT.2014.2313898

[36] Bishop, M.A., Rao, S.G. and Sripanidkulchai, K. (2006) Considering Priority in Overlay Multicast Protocols under Heterogeneous Environments. *Proceedings IEEE INFOCOM* 2006. 25*TH IEEE International Conference on Computer Communications*, Barcelona, 23-29 April 2006, 1-13. https://doi.org/10.1109/INFOCOM.2006.140

[37] Cheng, X. and Liu, J.C. (2009) NetTube: Exploring Social Networks for Peer-to-Peer Short Video Sharing. *IEEE INFOCOM* 2009, Rio de Janeiro, 19-25 April 2009, 1152-1160. https://doi.org/10.1109/INFCOM.2009.5062028

[38] Xu, K., Li, H.T., Liu, J.C., Zhu, W. and Wang, W.Y. (2010) PPVA: A Universal and Transparent Peer-to-Peer Accelerator for Interactive Online Video Sharing. 2010 *IEEE 18th International Workshop on Quality of Service*, Beijing, 16-18 June 2010, 1-9. https://doi.org/10.1109/IWQoS.2010.5542762

[39] Qiu, X.J., Wu, C., Lin, X.L. and Lau, F. (2009) Instant Leap: Fast Neighbor Discovery in P2P VoD Streaming. *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, June 2009, 19-24. https://doi.org/10.1145/1542245.1542251

[40] Moraes, I.M. and Duarte, O.C.M.B. (2010) A Lifetime-Based Peer Selection Mechanism for Peer-to-Peer Video-on-Demand Systems. 2010 *IEEE International Conference on Communications*, Cape Town, 23-27 May 2010, 1-5. https://doi.org/10.1109/ICC.2010.5501745

[41] Cui, L.Z., Jiang, Y. and Wu, J.P. (2011) Employing QoS Driven Neighbor Selection for Heterogeneous Peer-to-Peer Streaming. 2011 *IEEE International Conference on Communications*, Kyoto, Japan, 5-9 June 2011, 1-6. https://doi.org/10.1109/icc.2011.5962867