doi:10.3969/j.issn.1000-436x.2013.07.003

数据中心间空闲带宽感知的内容分发算法

黄永锋[1,2]，董永强[1,2]，张三峰[1,2]，吴国新[1,2]
(1. 东南大学 计算机科学与工程学院，江苏 南京 210096；2. 东南大学 计算机网络和信息集成教育部重点实验室，江苏 南京 210096)

摘 要：针对数据中心链路上存在时间窗不重叠的空闲带宽的情况，提出了利用该带宽分发容迟数据的基本思路，进而设计了一种分布式可扩展的空闲带宽感知的节点选择算法 LBAPS，该算法避免了集中优化，适合目标节点较多的情况。为了匹配最优的带宽空闲节点，LBAPS 按综合度量进行节点选择；为了优先把文件块上传到空闲带宽大的节点以及尽早把不同的块分布到更多节点，LBAPS 按阈值预留资源以及按时间片退出上传。基于 LBAPS 实现了内容云原型系统 P2PStitcher。PlanetLab 上的实验表明，LBAPS 算法所提出的策略可以有效地减少平均分发时间。
关键词：内容云；P2P；CDN；平均分发时间；PlanetLab
中图分类号：TP393 **文献标识码：**A **文章编号：**1000-436X(2013)07-0024-10

Leftover bandwidth-aware peer selection algorithm for inter-datacenter content distribution

HUANG Yong-feng[1,2], DONG Yong-qiang[1,2], ZHANG Shan-feng[1,2], WU Guo-xin[1,2]
(1. School of Computer Science and Engineering, Southeast iversity, Nanjing 210096, China; 2. Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing 210096, China)

Abstract: Due to the fact that leftover bandwidth appears during non-overlapping time intervals, an approach of using such bandwidth to distribute delay tolerant data was proposed, and then a distributs and scalable leftover bandwidth-aware peer selection algorithm named LBAPS was designed. LBAPS avoids centralized optimization method that fails to effectively utilize leftover bandwidth when multiple destinations occur. In LBAPS, a node selection strategy based on synthetical evaluation was presented in order to find appropriate nodes h leftover bandwidth currently. In addition, two other strategies, i.e., resource reservation based on threshold and exiting upload upon the length of time slice, were put forward. With these two strategies, nodes with more leftover bandwidth get higher priority to obtain file blocks; besides, different file blocks can be delivered to different nodes as soon as possible. On the basis of LBAPS, a content cloud prototype, P2PStitcher was implemented. Experimental results on PlanetlLab show that the strategies proposed in LBAPS are effective to decrease the average delivery time.
Key words: content cloud; P2P; CDN; average delivery time; PlanetLab

1 引言

由于云计算以较低的费用提供了弹性的资源供给，一些内容提供商开始借助云提供商的存储云（如亚马逊的 S3[1]）发布其内容，并通过与云无缝集成的内容分发网络（如 CloudFront[2]）把内容分发至各边缘节点以减少终端用户的访问延迟。内容分发网络（CDN）的这些边缘节点通常是分布在全球不同地方的数据中心，云提供商一般通过租用 ISP 链路实现这些节点的互联网接入，节点之间的数据传输需要大量的带宽费用，研究表明，网络传输支出已占云提供商整个运营支出的近

收稿日期：2012-09-02；**修回日期：**2013-01-15
基金项目：国家高技术研究发展计划(“863”计划)基金资助项目(2013AA013503)；国家自然科学基金资助项目(61272532)；江苏省自然科学基金资助项目(BK2011335)
Foundation Items: The National High Technology Research and Development of China (863 Program) (2013AA013503); The National Natural Science Foundation of China(61272532); The Natural Science Foundation of Jiangsu Province(BK2011335)

15%[3]，降低互联网传输费用将给云提供商节省大量运营支出。

由于节点所服务区域的流量需求每日遵循一定的模式[4]，假定 ISP 链路采用普遍使用的、基于峰值的定价策略，如“95th Percentile[5,6]”，则 ISP 链路上存在随时间变化的已付费未使用的带宽[4]（下文简称空闲带宽）。内容分发中通常有部分数据，它们对分发完成时间的要求较为宽松，文献[7]中称之为容迟数据。如果能够感知上述空闲带宽并利用其分发容迟数据，则该部分数据分发不会抬高计费峰值，可节省传输费用。然而，分布于全球范围的数据中心节点由于跨越了多个不同时区，空闲带宽资源出现的时间窗口呈现大量不重叠的特征，即当源节点带宽出现空闲的时候，目标节点未必空闲，如何有效利用这些空闲带宽来完成到多个边缘节点的容迟数据分发就是本文要解决的问题。

本文主要贡献如下：首先，在单源节点到多目标节点内容分发中提出了如何分布式利用空闲带宽分发容迟数据的基本思路；其次，分析了减少平均分发时间的关键因素，认为把文件优先传输到上行空闲带宽大的节点以及尽早把不同的块分布到更多节点可减少系统中节点后续的传输时间，并在此基础上设计了空闲带宽感知的节点选择(LBAPS, leftover band width-aware peer selection)算法，提出了 3 种具体减少平均分发时间的策略；最后，在 Slurpie[8]基础上实现了内容云原型系统 P2PStitche[9]，PlanetLab[10]上的实验表明，LBAPS 算法提出的策略有效减少了系统的平均分发时间。

2　相关工作

随着服务提供商对用户体验的重视，20 世纪 90 年代末，MIT 研究者引入了 CDN 概念，如今，CDN 已从单纯的 C/S 结构，如早期的 Akamai[11]网络，P2P 结构如 BitTorrent[12]、Slurpie[8]，发展到混合分发结构[13]及内容云[2, 14]，CDN 分发需要消耗大量的带宽资源，不过，很少有关于面向计费策略优化的 CDN 方面的研究。

近年来，随着基于峰值使用的百分位计费在 ISP 中的普遍使用，Goldenberg 等人[5]和 Wang 等人[15]分别提出了针对该种计费策略优化 Internet 资费的一些办法。文献[5]针对与多个 ISP 连接的节点,通过智慧路由的方式把流量优化分布到多条 ISP 链路上以降低单条链路上的峰值，从而降低链路资费；文献[15]对部分流量进行延迟传输，使用优化流量规划解决了峰值下降带来的资费收益和不同延迟招致的惩罚之间的权衡问题。不过，这些优化资费的策略均是针对单独的传输节点提出的，而内容分发理想的情况下，要同时能够降低源和目标节点的峰值，即节点之间需要进行上下行空闲带宽互相感知的联合优化调度。

与本文工作最相近的是文献[4,7]，它们系统地提出了在 2 个数据中心节点间利用空闲带宽传输容迟数据的办法。文献[7]提出了在时差较小的节点间通过双方时间上重叠的空闲带宽传输“大容量容迟数据”，在时差较大的节点间借助部署额外的中间节点、通过存储转发解决源和目标节点因时间窗不重叠导致的空闲带宽无法利用的问题，然而，它只给出了单个额外中间节点的简单情况。文献[4]的工作更进一步，当在 2 个时差较大的节点间传输数据时，首先收集所有数据中心节点（包括多个中间节点）的空闲带宽、空闲存储等相关信息，在此基础上对这些信息的未来值进行预测，通过集中优化的方式得到最优调度，然后依此进行存储转发传输，它解决了单个目标节点下的分发问题，然而，在内容分发中往往有多个目标节点，虽然可通过与文献[4]类似的建模得到全局最优化调度，但这种调度需要进行大量集中运算，当预测值与实际值出现偏差时还要不断重算，特别是随着目标节点数量的增长（Akamai 目前在全球 1 000 多个网络中拥有节点），运算量迅速膨胀，集中优化在实践中并不可行，需要寻找一种可扩展的分布式优化调度方法。

3　内容分发的基本思路

为了简化问题，下面对本文问题中的相关条件做进一步假设：1) 各节点均只有 1 条互联网接入链路，链路上行和下行均采用“95th Percentile”的计费策略；2) 链路上流量需求每日遵循一定的模式，更确切地说，链路上当前由于用户访问节点产生的流量与节点所在的地理位置密切相关，且随时间而变化，因此，每个节点均会有与节点位置相关的时变空闲带宽。

为了充分利用各节点的空闲带宽资源，减少分发时间，初步的想法是在分发中利用 P2P 技术，基本思路如下：在内容源节点完成文件分块和发布后，便广播通知各边缘节点择机进行下载，各边缘节点开始周期性地监控 ISP 链路，一旦预测到下行

空闲带宽，便寻找若干具有上行空闲带宽的节点下载本节点没有的文件块，下载的同时继续监控本节点空闲带宽的变化情况，通过增加或删除链接的方式保证既不浪费空闲带宽，也不抬高计费峰值。

由于任何一个已完成分发的节点不需要等到其他节点分发完成便可向其覆盖区域提供访问，因此，P2PStitcher 的设计目标是追求节点平均分发时间的最小化，这样，节点间可采用互相合作的方式，在所有节点完成分发之前，始终提供上传服务。如何设计分布式的、代价较小的分发算法，在不抬高计费峰值的情况下减少容迟数据的平均分发时间是本文研究的重点。

4 最小化平均分发时间

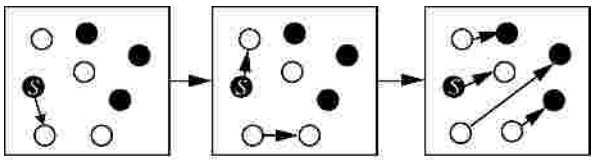
4.1 平均分发时间实例分析

为便于讨论，引入一个简化的分发模型，模型各节点上行和下行带宽相等，且不随时间而变化，模型中内容源节点 S 的带宽容量为 c_s，需要分发的文件大小为 f，分成 k 块，其他 n 个边缘节点的带宽分别为 $c_1, c_2, \cdots, c_n$，所有节点间均可直接传输数据。设节点 j 的分发时间为 dt_j，下面看 2 种特殊情况下不同调度的平均分发时间 $m(dt)$。

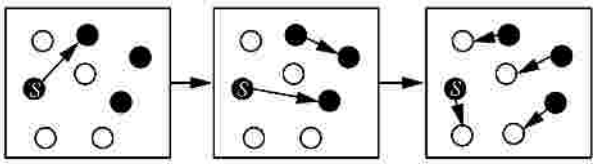
情况 1 节点间的传输速率仅受节点上下行带宽限制，相关参数值具体为 f=1，k=1，n=7。各节点带宽为 $c_1=c_2=c_3=1$, $c_s=c_4=c_5=c_6=c_7=2$。

最小带宽优先调度 调度如图 1(a)所示，空心节点表示带宽为 1 的节点，实心节点表示带宽为 2 的节点，调度优先完成带宽容量小的节点的分发，该调度下系统的平均分发时间为

$$m(dt)=\frac{1}{7}\sum_{j=1}^{7}dt_j=\frac{(1+2\times2+3\times4)}{7}=\frac{17}{7} \quad (1)$$



(a) 最小带宽优先调度



(b) 最大带宽优先调度

图 1 情况 1 下的 2 种调度

最大带宽优先调度 调度如图 1(b)所示，空心节点与实心节点意义与最小带宽优先调度相同，调度优先完成带宽容量大的节点的分发，该调度下系统的平均分发时间为

$$m(dt)=\frac{1}{7}\sum_{j=1}^{7}dt_j=\frac{0.5+1\times2+2\times4}{7}=\frac{10.5}{7} \quad (2)$$

显然，最大带宽优先调度的平均分发时间小于最小带宽优先调度，因为该调度先满足了带宽大的节点的分发，减少了后续节点的分发时间。

情况 2 节点间传输速率不仅受节点上下行带宽限制，还受节点间网络传输链路带宽的限制，相关参数值具体为 f=4，k=4，n=4，$c_s=c_1=c_2=c_3=c_4=2$，节点间网络传输链路带宽均为 1。

从条件看，情况 2 意味着文件被分成 4 块，每块大小为 1，节点无须等待整个文件下载完成便可开始上传。受网络传输链路带宽限制，每文件块传输需要 1 个单位时间，由于节点带宽为链路容量的 2 倍，节点可同时从 2 个节点下载不同的文件块。

顺序传输调度 调度如图 2(a)所示，边缘节点内数字表示该节点目前已有的文件块号，传输边上的数字表示当前时间正在传输的文件块号，该调度下，源节点给目标节点顺序传输完所有的文件块后才终止链接，且给各目标节点并发传输时选择的是相同的文件块，该调度下系统的平均分发时间为

$$m(dt)=\frac{1}{4}\sum_{j=1}^{4}dt_j=\frac{(4\times2+5\times2)}{4}=\frac{18}{4} \quad (3)$$

随机传输调度 调度如图 2(b)所示，与顺序传输相同的是，源节点给目标节点传输完所有的文件块后才终止链接，与顺序传输不同的是，源节点并发传输时是随机选择文件块，该调度下系统的平均分发时间为

$$m(dt)=\frac{1}{4}\sum_{j=1}^{4}dt_j=\frac{(3\times2+4\times2)}{4}=\frac{14}{4} \quad (4)$$

非持续传输调度 调度如图 2(c)所示，与前 2 种调度不同的是，源节点并没有在完成到其他节点的所有块的传输后才断开链接，而是采取了一种尽快把不同的块分布到不同节点的调度策略，该调度下系统的平均分发时间为

$$m(dt)=\frac{1}{4}\sum_{j=1}^{4}dt_j=\frac{(3\times4)}{4}=\frac{12}{4} \quad (5)$$

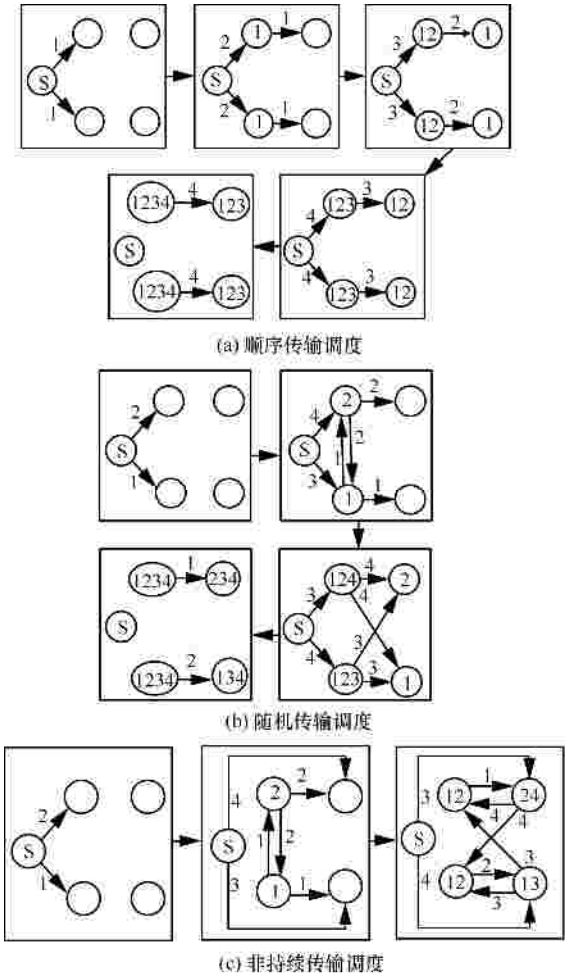


图 2　情况 2 下的 3 种调度

由调度过程分析，随机传输调度的平均分发时间较顺序传输调度更优，根本原因在于该调度下随机文件块的选择减少了系统后续的传输时间；另外，非持续传输调度优于顺序传输调度和随机传输调度，根本原因在于通过把块尽早分布到不同的节点，减小了后续分发对上传带宽的竞争。

4.2　内容云平均分发时间建模

4.1 节给出了以下启示：把文件优先传输到上行空闲带宽大的节点以及尽早把不同的块分布到不同的节点，可以改善系统后续的传输，下面从更一般的情况来对内容云平均分发时间进行建模分析。

根据文献[16]，可把本文内容云分发分成 2 个阶段讨论。1)初始时，内容云中可用的文件资源远远无法满足下载需求，减少平均分发时间的关键是让文件资源快速增长。2)随着分发的进行，可用文件资源已足够充分，此时，由于节点缺乏全局信息，找到更好的资源来满足节点需求就成为减少分发时间的关键。为讨论方便，本文分别称内容分发的这 2 个阶段为瞬态阶段和持久阶段。

先分析瞬态阶段影响可用文件资源增长的关键因素。假设文件被分成 k 块，为尽可能分布不同的块，节点采取随机选择块下载策略，即块被下载的概率相等，则任一时刻各文件块在系统中数量的期望应相等，因此，t 时刻系统中某一文件块数量可看作系统可用文件资源的标志，用 N_t 来表示。设该文件块的某一拷贝在经过时间 T_i 后变成 V 个拷贝（V 是一与 T_i 无关的随机变量且满足 $\mathrm{E}(V)=v$，文件分发中显然有 $v\geq 2$），在瞬态阶段，可认为 T_i 独立同分布[16]，$T_i \sim T$，$F_{\mathrm{T}}(t)$ 为 T 的概率分布函数，$F_{\mathrm{T}}(t)$ 反映了节点端到端链路带宽以及节点空闲带宽的情况。图 3 表示了 t 时刻的 N_t，N_t 为一个标准的年龄相依的分支过程[17]，根据文献[16]可得引理 1。

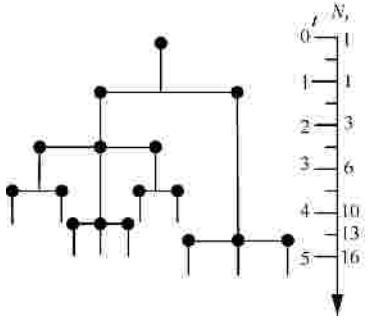


图 3　t 时刻系统中某文件块的数量

引理 1　在分发的瞬态阶段，$v>1$ 时，N_t 的期望满足式(6)。

$$\mathrm{E}(N_t)=d\,\mathrm{e}^{l t} \tag{6}$$

其中，$?>0$ 且满足

$$F(x)=\int_0^x v\mathrm{e}^{-l t}\mathrm{d}F_{\mathrm{T}}(t) \tag{7}$$

为某一随机变量的分布函数，即 $\int_0^{\infty} v\mathrm{e}^{-l t}\mathrm{d}F_{\mathrm{T}}(t)=1$，$d=\dfrac{v-1}{ml\ v}$，其中，$\mu$ 为该随机变量的数学期望。

由引理 1 可知，在瞬态阶段，$?$ 是决定系统可用文件资源增长率的关键，也是减少平均分发时间的关键因素。

为叙述方便，根据节点当前的空闲带宽，把节点状态分为上行空闲带宽饱和状态及上行空闲带宽非饱和状态，节点处于上行空闲带宽饱和状态，即节点当前上行空闲带宽已用完，增加新的上传链接将占用非空闲带宽或改变原有其他传输终端到端链路带宽的大小；节点处于上行空闲带宽非饱和

状态，即节点仍有上行空闲带宽，增加新的上传链接并不会改变原有其他传输中的端到端链路带宽的大小。

同理，根据内容云系统中所有节点的状态，把系统在瞬态阶段下的状态分为空闲带宽全局饱和状态和空闲带宽非全局饱和状态，系统的状态称为空闲带宽全局饱和状态当且仅当系统中所有节点均处于上行空闲带宽饱和状态；系统的状态称为空闲带宽非全局饱和状态即系统中至少存在一个节点处于上行空闲带宽非饱和状态。

定理 1 对任何含有空闲带宽非全局饱和状态的分发过程 DS_1，可通过重新调度，生成一个不含空闲带宽非全局饱和状态的分发过程 DS_2，且 DS_2 在瞬态阶段下可用文件资源的增长率大于 DS_1。

证明 对任何含有空闲带宽非全局饱和状态的分发过程，设为 DS_1，在瞬态阶段，存在某个时刻 t，至少存在一个节点处于上行空闲带宽非饱和状态，通过重新调度，逐步增加链路并行上传数量直至其饱和，重复以上过程，至分发过程中不含任何空闲带宽非全局饱和状态，设此时对应的新分发过程为 DS_2，增加的并行上传并未改变单条链路上的 $F_T(t)$，由于多个节点并行上传数量增加，显然有 $v_2>v_1$，v_1、v_2 分别为 DS_1 和 DS_2 的 V 值的期望，由引理 1，得到

$$\int_0^{\infty} v_1 e^{-l_1 t} dF_T(t) = 1 ,\ \int_0^{\infty} v_2 e^{-l_2 t} dF_T(t) = 1 \qquad (8)$$

$?_1$ 和 $?_2$ 分别为引理 1 式(6)中对应 DS_1、DS_2 分发过程瞬态阶段的资源增长率指数，根据式(8)，得到

$$\int_0^{\infty} v_1 e^{-l_1 t} dF_T(t) - \int_0^{\infty} v_2 e^{-l_2 t} dF_T(t) = \int_0^{\infty} (v_1 e^{-l_1 t} - v_2 e^{-l_2 t}) dF_T(t) = 0 \qquad (9)$$

设 $?_1 \geq ?_2$，由于 $t>0$，$v_1<v_2$，则 $v_1 e^{-l_1 t} - v_2 e^{-l_2 t} < 0$，由于 $dF_T(t)$ 非负，根据定积分性质，有 $\int_0^{\infty} (v_1 e^{-l_1 t} - v_2 e^{-l_2 t}) dF_T(t) < 0$，矛盾，因此 $?_1 < ?_2$，DS_2 的资源增长率大于 DS_1，证毕。

根据定理 1 证明过程，易得到以下推论。

推论 1 任何分发过程 DS_1，如果在不改变 $F_T(t)$ 的情况下能够重新调度得到新的分发过程 DS_2，如果 $v_2>v_1$，v_1、v_2 分别为 DS_1 和 DS_2 的 V 值的期望，则 DS_2 在瞬态阶段的文件资源增长率大于 DS_1。

推论 2 在瞬态阶段，把文件优先传输到上行空闲带宽大的节点将产生更大的文件资源增长率。

证明 由于把文件优先传输到上行空闲带宽大的节点并不改变 $F_T(t)$ 的分布，并能够带来更大的 v，根据推论 1，将产生更大的文件资源增长率，证毕。

推论 3 如果所有节点的上行空闲带宽相等且所有节点间端到端链路的带宽也相等，在瞬态阶段，更早把更多不同的块分布到更多不同节点的分发过程将有更大的文件资源增长率。

证明 由于上述策略并不改变 $F_T(t)$ 的分布，在所有节点上行空闲带宽相等且所有节点间端到端链路带宽相等的情况下，把更多不同的块分布到不同的节点，提高了系统整体的实际可用空闲带宽，增加了并行块传输的机会，这样的分发过程能获得更大的 v 值，从而获得更快的资源增长率，证毕。

因此，通过一般情况下的分析，在瞬态阶段，得到了与简化分发模型实例分析中相同的结论。

当系统中可用文件资源足够充分时，引理 1 成立的条件不再满足，从实际经验来看，在大部分节点已完成分发的情况下，系统中的可用文件资源也不再呈指数增长。由于完成分发的节点并不退出系统，上行空闲带宽已不在多个节点间竞争，此时减少平均分发时间的关键就转变为让节点找到最快的资源尽快完成本节点的分发。

4.3 内容分发算法面临的挑战

本文的目标是实现代价较小的分布式算法，其选择策略需要在缺乏全局信息的情况下作出，鉴于理论分析多基于理想条件，实际算法设计还须面对下列挑战。

1) 分布式算法如何在瞬态阶段下让内容云中具有空闲带宽的节点尽快进入上行空闲带宽饱和状态，同时又保证节点尽快完成自己的分发。

2) 节点在缺乏全局信息的情况下获取哪些文件块是节点根据本身已有的块信息决定的，因此，数据传输采用的是拉的方式，即节点先进行块请求，待请求被接受后才进行块传输，由于拥有文件块的节点无法预知后面的块请求，算法如何保证让上行空闲带宽大的节点能够优先获得块。

3) 瞬态阶段下，把不同的块尽早分布到不同的节点可带来更大的文件资源增长率，这基于所有节点上行空闲带宽均相等的情况，实际中不同节点的上行空闲带宽存在较大差别，且传输链接的建立需要开销，把不同的块尽早分布到不同的节点有时不如把所有的块都优先传输到上行空闲带宽大的节点，算法如何在这两者之间权衡。

4.4　空闲带宽感知的节点选择算法

本文提出了一种分布式 LBAPS 算法以应对上述挑战。针对挑战 1)，LBAPS 算法提出了一种综合度量的方式进行节点选择，节点周期性地预测本节点下行空闲带宽，一旦感知到下行空闲带宽，依据下列 *Rank* 值选择节点，增加下载链接。

$$Rank = a\,Lb + b\,Nb + g\,Bd \tag{10}$$

其中，*Lb*、*Nb*、*Bd* 分别为被选节点当前上行空闲带宽指数、与本节点端到端链路带宽指数以及与本节点块差异指数，三者均已做了归一化，范围在[0,1]之间，*a*、ß、*?* 为三者权重。该度量综合考虑了影响分发速度的各种因素，能够在整个分发阶段尽快完成本节点分发。通过设置较大的 *g* 权重，可让瞬态阶段下的节点尽快进入上行空闲带宽饱和状态，同时，随着分发的进行，由于节点间块差异越来越小，链路端到端带宽指数在度量中将发挥更重要的作用，减少了节点持久阶段分发的时间。为了有效避免部分节点请求过载，而部分节点空闲带宽闲置的情形，实现中根据 *Rank* 排名挑选多个候选节点，并从这些候选节点中随机选择进行下载。

分发起始时，链路端到端带宽指数可通过定期简单的测量 RTT 获得[18]，随着分发的进行，可使用历史下载中实际产生的速率进行替代。在 PlanetLab 环境下，由于链路端到端带宽与节点地理位置距离存在一定的相关性，通过节点反时区差指数来简单替代链路端到端带宽指数，即节点所在地理位置相距越近，时区差越小，反时区差指数越大，链路端到端带宽越大。

针对挑战 2)和挑战 3)，LBAPS 提出了一种按阈值预留资源和按时间片退出上传的策略。按阈值预留资源策略如下：当上传者收到下载链接请求时，根据节点当前上行空闲带宽值决定是否接受请求，在已使用的上行空闲带宽到达一定阈值时，只接受上行空闲带宽满足条件的节点，让上行空闲带宽大的节点有更多被接受的机会，该部分算法具体描述如下。

算法 1　LBAPS 算法处理初始下载请求

输入：当前节点上行空闲带宽 *lbdup*，控制参数 C_T_1,C_T_2

约束：$0 \le C\_T_1 \le 1 \le C\_T_2 \le 2$

1) 计算已实际使用上行空闲带宽值 *usedbd*；
2) //计算阈值 t_1,t_2
3) $t_1 = C\_T_1 \times lbdup$；
4) $t_2 = C\_T_2 \times lbdup$；
5) if (*usedbd* > t_2){
6) 发送上行空闲带宽不可用消息；
7) } else if (*usedbd* 位于[t_1，t_2]之间){
8) if (请求节点上行空闲带宽值>活动链
9) 接上行空闲带宽平均值)
10) 接受下载请求，链接建立成功；
11) else
12) 发送上行空闲带宽不可用消息；
13) } else{
14) 接受下载请求，链接建立成功；
15) }

按时间片退出上传策略如下：为了在瞬态阶段把更多不同的块分布到不同的节点，上传节点在感知到本节点空闲带宽已进入上行空闲带宽饱和状态时按时间片选择节点退出上传，以给其他节点创造更多的下载机会。为了保证块传输的完整性，上传节点不采用强制删除链接的办法，而是在链接的当前数据块传输完毕后，在原有链接上传新的数据块时根据时间片大小判断是否继续当前上传。时间片的大小通过时间片常数和时间片系数决定，时间片常数一般根据内容云具体的环境和链接开销进行估计，时间片系数与节点的上行空闲带宽成正比，在节点尽早把不同的块分布到更多节点的同时，让上行空闲带宽大的节点获得更多传输机会，该部分算法描述如下。

算法 2　LBAPS 算法处理新的数据块请求

输入：当前节点上行空闲带宽 *lbdup*、已使用上行空闲带宽 *usedbd*、控制参数 C_T_2，归一化的对端上行空闲带宽值 *normalclbdup*、时间片常数 TD、链接持续的最小时间片 *LST*。

1) //上传节点处于上行空闲带宽饱和状态
2) if(*usedbd* > l*bdup* and *usedbd* < $C\_T_2 \times lbdup$){
3) 计算该链接上传已持续时间 *timeelapse*；
4) //根据对端上行空闲带宽值计算时间片大小
5) *timethreshold*=TD×*normalclbdup*；
6) if (*timethreshold*<*LST*)
7) *timethreshold=LST*；
8) if(*timeelapse*>*timethreshold*){ //时间片到
9) 发送时间片用完消息，退出上传；
10) }else{
11) 接受新块请求；

12) }
13) //上传节点处于上行空闲带宽非饱和状态
14) }else if(*usedbd*<*lbdup*){
15) 接受新块请求；
16) }else{
17) 发送上行空闲带宽不可用消息，退出上传；
18) }

明确区分系统处于瞬态阶段还是持久阶段是非平凡的，LBAPS 算法中没有显性判断系统阶段，对于分发的上传节点来说，只在瞬态阶段才频繁进入上行空闲带宽饱和状态，时间片退出才发生作用，否则，上传节点会一直等下载节点完成所有块传输后才断开链接，确保了节点处于上行空闲带宽非饱和状态时，不浪费空闲带宽。

为了评估按阈值预留资源和按时间片退出上传的策略对全局块分布的影响，需要一个度量系统当前状态块分布的办法，因此，引入块分布计数 *Bdv*，该值是系统中所有块的单块块分布计数(*Sbdv*)的和。设文件分成 k 块，包含第 i 块的节点的数量为 n，节点 j 上行空闲带宽为 *lbdupj*，处于上行空闲带宽饱和状态时可并行上传的链接数为 *plinkj*，节点 j 当前拥有的块数为 *bcj*，首先计算第 i 块在节点 j 的块上行空闲带宽值 *blbdupj*，如式(11)所示。

$$blbdup_j=\begin{cases} lbdup_j / plink_j, & bc_j \le plink_j \\ (lbdup_j / plink_j)\cdot(plink_j / bc_j), & bc_j > plink_j \end{cases} \tag{11}$$

然后，按照 $blbdup_j$ 的大小进行排序，重新命名排序后第 s 个大小值为 $blbdup_s$，则 i 块的 $Sbdv_i$ 为

$$Sbdv_i=\sum_{s=1}^{s=n}(2^{-s}\times blbdup_s) \tag{12}$$

最后系统的块分布计数计算为

$$Bdv=\sum_{i=1}^{i=k}Sbdv_i \tag{13}$$

从 *Bdv* 计算过程可以看出，该度量充分体现了系统当前状态下可真正使用的上行空闲带宽。

5 空闲带宽预测

实际中，节点需要预测 ISP 链路的空闲带宽，为评估 P2PStitcher 的分发性能，下面给出空闲带宽预测的一个仿真模型。根据文献[4]，分布于全球的数据中心的流量每日均遵循一定的模式，模型必须反映这个特征，本文采用时间序列仿真以符合该特征。在“95th Percentile”的计费策略下，以月为计费周期，5 min 采样一次流量，定义随机变量时间序列 Q_t 作为数据中心仿真采样流量值，表示第 t 个 5 min 时生成的采样流量，Q_t 定义为

$$Q_t=m_t+X_t \tag{14}$$

模型中假设 Q_t、μ_t、X_t 均是周期为 1 天的函数，其中，μ_t 是时变均值函数，反映了一天中 t 时刻流量的期望，X_t 表示高斯噪声，反映了 Q_t 在 μ_t 附近的随机波动。

根据以上数据特征的要求，在一个周期内，即 $0\le t<24?$ 时，μ_t 定义如式(15)所示。

$$m_t=\begin{cases} q_1, & t\in[0,8q) \\ q_1+(q_h-q_1)\times(t-8q)/q, & t\in[8q,9q) \\ q_h, & t\in[9q,12q) \\ q_h-(q_h-q_m)\times(t-12q)/q, & t\in[12q,13q) \\ q_m+(q_h-q_m)\times(t-13q)/q, & t\in[13q,14q) \\ q_h, & t\in[14q,17q) \\ q_h-(q_h-q_n)\times(t-17q)/q, & t\in[17q,18q) \\ q_n, & t\in[18q,22q) \\ q_n-(q_n-q_1)\times(t-22q)/q, & t\in[22q,23q) \\ q_1, & t\in[23q,24q) \end{cases} \tag{15}$$

其中，?=12，代表每小时采样次数。式(15)反映了下列趋势：流量在当地上下午工作时间段内达到峰值，在深夜和凌晨处于谷值并且在午间和下班后也处于较低值的一种趋势。式(15)中 q_l 表示流量谷值时刻的 μ_t，q_h 表示流量峰值时刻的 μ_t，q_m、q_n 分别表示午休和大部分人下班后的 μ_t。

实际中，X_t 的方差与 t 时刻流量均值呈现出一定的相关性，因此，模型中假设 X_t 服从如下正态分布

$$X_t\sim N(0,(m_t/w)^2) \tag{16}$$

其中，? 为一常数。

以上模型在下文中称为简易流量模型（STM），P2PStitcher 原型中，节点的上下行均采用 STM 模拟空闲带宽。图 4 为 q_h=1 000，q_l=100，q_m=600，q_n=300，?=8 时 MATLAB 仿真的 2 天的 Q_t 值。

定理 2 简易流量模型 STM 生成的流量数据的付费带宽在“95th Percentile”计费策略下的预测值 Q_p 为 $q_h+(q_h\times F^{-1}(0.8))/w$，其中，$F$ 为标准正态分布的分布函数。

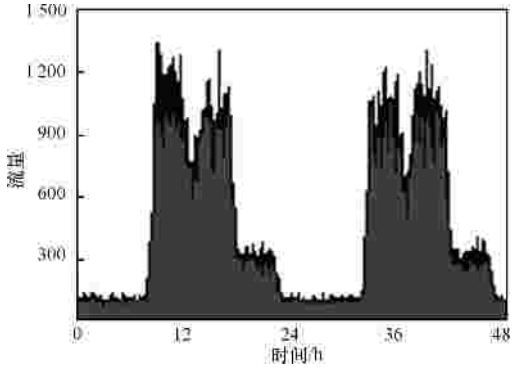


图 4　仿真生成的 2 天的 Q_t

证明　由于 STM 模型大于 Q_p 的流量在每天是均匀分布的，5%的峰值流量主要集中在[9?, 12?)和[14,17?)这 6 个小时之间，则一天内 5%的最高流量占了约一天内这 6 个小时的 20%的流量，得到

$$P(Q_t > Q_p) \approx 0.2 \quad t \in [9q, 12q) \text{或} t \in [14q, 17q) \tag{17}$$

根据式(14)、式(15)、式(16)，有

$Q_t \sim N(m_t, (m_t/w)^2)$，令 $Z_t = \dfrac{Q_t - m_t}{(m_t/w)}$，有

$Z_t \sim N(0,1)$，根据式(17)可得

$P(Z_t > \dfrac{Q_p - m_t}{(m_t/w)}) \approx 0.2$，设 $s = \dfrac{Q_p - m_t}{(m_t/w)}$，有

$P(Z_t \leq s) \approx 0.8$，即 $s = F^{-1}(0.8)$，代入 s,有

$$Q_p = m_t + (F^{-1}(0.8) \times m_t)/w \tag{18}$$

代入 t 在[9?，12?)和[14,17?) 时 u_t 的值 q_h，有

$$Q_p = q_h + (q_h \times F^{-1}(0.8))/w \tag{19}$$

证毕。

这样，t 时刻的空闲带宽 Bf_t 即为

$$Bf_t = Q_p - Q_t \tag{20}$$

6　实验及结果分析

本文在 Slurpie[8]的基础上实现了内容云的原型 P2PStitcher，源代码已发布在 SourceForge[9]上。

实验在 PlanetLab 环境下进行，内容源节点(planetlab-1.pdl.nudt.edu.cn)从事先配置好的 WEB 服务器(Apache 2.0.52 运行于 RedHat AS 4.7 上，位于大学校园网内)下载需要分发的内容，在源节点完成部署后，启动其他节点并发下载，实验中链路端到端带宽使用反时区差指数(4.4 节)进行替代，节点的时区通过 plc_api 接口获取的节点经度信息计算得到，系统的主要参数及默认值设置如表 1 所示。

表 1　P2PStitcher 主要参数说明及默认值

参数	说明	默认值
(q_h q_l q_m q_n ?)	空闲带宽参数	(2 0.2 1.2 0.6 8)/Mbit/s
(a ß ?)	式(15)	(1 2 2)
C_T1	算法 1 参数	1
C_T2	算法 1 参数	1.2
TD	算法 2 参数	50/s
LST	算法 2 参数	3/s
FSIZE	分发文件大小	50 MB

6.1　基本性能分析

为了对 P2PStitcher 的性能有一个认识，首先对空闲带宽感知分发的 P2PStitcher 和普通 P2P 分发 Slurpie 在文件分发的平均时间上作了一个比较，为了消除不同实验时间对于结果的影响，实验固定在北京时间 8：30 左右进行，一天在每种节点规模下对 P2PStitcher 每种设置和 Slurpie 实验各进行 1 次，连续实验 5 天，最后结果为 5 次实验的平均值。

P2PStitcher-small 和 P2PStitcher-large 是 P2PStitcher 2 种实验设置，其(q_h q_l q_m q_n ?)分别为(2 0.2 1.2 0.6 8)和(10 1 6 3 8)，单位均为 Mbit/s，分别表示 2 种不同的空闲带宽水平，其余参数均采用默认设置。如此设置的原因在于 P2PStitcher-small 设置在 PlanetLab 下足以形成对空闲带宽的竞争。不同节点规模下 P2PStitcher 的平均分发完成时间和 Slurpie 的对比情况如图 5 所示。

从图 5 可以看出，在空闲带宽水平较高时，即采用 P2PStitcher-large 设置的情况下，不同节点规模下的平均分发时间均低于 Slurpie,特别是在节点数量较多的情况下，两者之间的差距更明显，这从一定程度上说明 LBAPS 算法是有效的，即使各节点的空闲带宽水平较低，系统实际可用带宽远远小于 Slurpie，随着节点数量的增长，平均分发时间也能逼近 Slurpie。

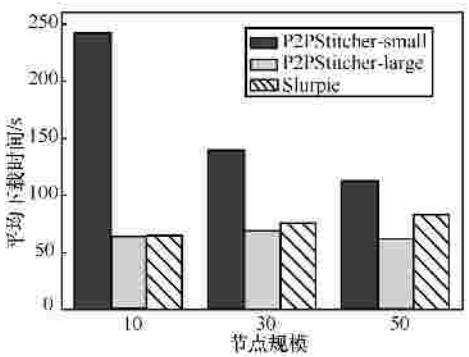


图 5　P2PStitcher 与 Slurpie 平均分发时间对比

50 个节点并发下载完成时间的累积分布函数如图 6 所示。在 P2PStitcher-small 的设置情况下，80%的节点均能在 150s 内完成下载，可见，虽然空闲带宽较少，但在节点数量较多的情况下，也能够利用空闲带宽较快地完成内容分发。

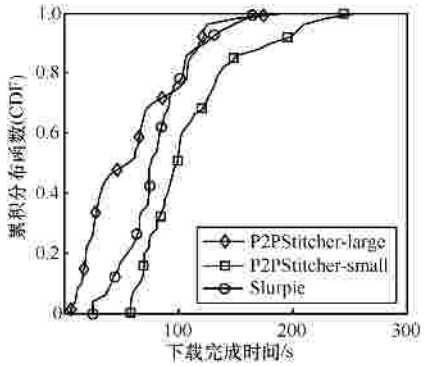


图 6 50 个节点并发下载完成时间的 CDF

6.2 LBAPS 算法评估

为了有效评估 LBAPS 算法，下面观察不同参数设置对平均分发时间的影响。实验分为 4 组，每组同样固定在北京时间 8：30 左右开始，在每种节点规模下各进行一次，连续实验 5 天，最后结果为 5 次实验的平均值。为了让算法效果更明显，各组实验(q_h q_l q_m q_n ?)的取值与 P2PStitcher-small 设置相同，针对特定的实验目的，每组实验改动了 LBAPS 算法部分默认参数值，如表 2 所示，表中未列项仍采用 P2PStitcher 的默认设置。

PSNF 实验主要验证节点选择时不考虑节点上行空闲带宽指数的影响；PSNNB 实验主要验证节点选择时不考虑链路端到端带宽指数的影响；BCNTS 实验中，节点不再设置时间片(通过设置 TD 和 LST 为大常数)，主要验证按时间片退出策略的影响；BCSTS 实验让时间片常数比默认值更小，主要验证频繁退出上传对平均分发时间的影响。

表 2 LBAPS 算法验证实验参数设置

实验名	参数	值
PSNF	(a、ß、?)	(0 2 2)
PSNNB	(a、ß、?)	(1 0 2)
BCNTS	TD、LST	10 000、10 000
BCSTS	TD、LST	5、2

从图 7(a)可以看出，PSNF 下节点由于空闲带宽不可用被拒绝请求的平均次数(包括下载链接请求被拒绝和新数据块请求被拒绝)高于 P2PStitcher-small，节点数量较多时更显著，表明综合度量考虑上行空闲带宽指数对于合理引导节点选择是有效的。

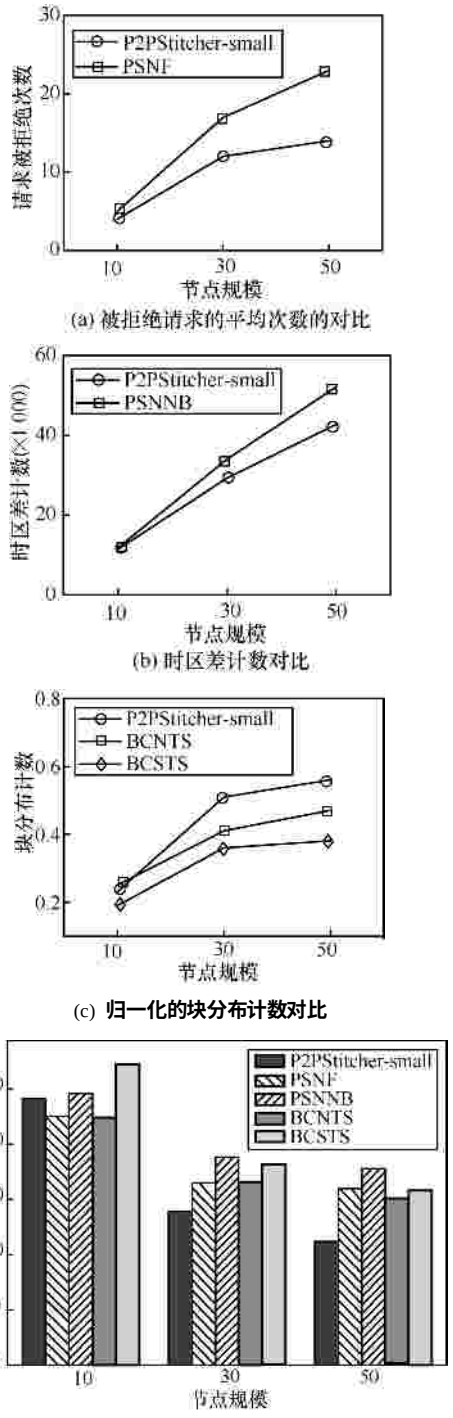


图 7 P2PStitcher 不同参数设置下的分发实验对比

为了考查综合度量中链路端到端带宽因素的作用，实验中引入了时区差计数，各个节点计数初始为 0，当完成某一块下载后，计数就累加本节点与对端节点的时区差值，图 7(b)是 P2PStitcher-small

与 PSNNB 时区差计数对比，由此看出，P2PStitcher-small 较 PSNNB 更能够让节点趋向于选择地理位置临近的节点，即 PlanetLab 环境下那些通常端到端带宽更大的节点。

为了验证按阈值预留资源策略和按时间片退出策略对全局块分布情况的影响，图 7(c)为 P2PStitcher-small、BCNTS、BCSTS 在下载开始后的第 10 s 时归一化的块分布计数 *Bdv*(见 4.4 节)的对比，可以看出，节点数量较多时，P2PStitcher-small 的块分布明显优于 BCNTS，这也证明了瞬态阶段按阈值预留资源策略和按时间片退出策略发挥了作用。另外，从实验看出，BCSTS 的归一化块分布计数最小，这也说明过短的时间片在实际分发情况下反而不利于文件块更好地分布。

图 7(d)展示了各组实验在不同节点规模并发下载时平均分发时间与采用 P2PStitcher-small 设置的比较。在空闲带宽没有形成竞争时，LBAPS 算法效果不明显，随着节点数量的增长，空闲带宽在节点间形成足够竞争，算法体现出充分的优势。

7　结束语

数据中心是近年来的研究热点之一，本文提出了一种在不抬高数据中心节点 ISP 链路计费峰值的情况下分发容迟数据到多个边缘节点的思想，通过理论分析影响平均分发时间的关键因素，设计了一种旨在改善平均分发时间的算法 LBAPS，并在此基础上实现了原型系统 P2PSticher。PlanetLab 上的实验结果表明，如果各节点空闲带宽水平较高，在仅利用空闲带宽的情况下，P2PStitcher 平均分发完成时间甚至优于普通 P2P 分发，即便节点空闲带宽水平较低，在节点数量较多时，P2PStitcher 的平均分发完成时间也可逼近普通分发。大量的评估实验表明，空闲带宽感知的节点选择算法 LBAPS 采取的按综合度量进行下载节点选择、按阈值预留资源和按时间片退出的策略可让上行空闲带宽大的节点优先获得文件块，在分发的初始阶段能够尽快把不同的文件块分布到更多不同的节点，这些策略有效地减少了系统的平均分发时间。

参考文献：

[1] Amazon simple storage service(S3) [EB/OL]. http://aws.amazon.com/s3/, 2012.

[2] CloudFront [EB/OL]. http://aws.amazon.com/cloudfront/, 2012.

[3] GREENBERG A, HAMILTON J, MALTZ D A, *et al*. The cost of a cloud: research problems in data center networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 39(1):68-73.

[4] LAOUTARIS N, SIRIVIANOS M, YANG X, *et al*. Inter- datacenter bulk transfers with netstitcher[J]. ACM SIGCOMM Computer Communication Review, 2011, 41(4):74-85.

[5] GOLDENBERG D K, QIUY L, XIE H, *et al*. Optimizing cost and performance for multihoming[J]. ACM SIGCOMM Computer C ation Review, 2004, 34(4):79-92.

[6] ISP bandwidth billing-how to make more or pay less[EB/OL]. http://servicelevel.net/rating_matters/newsletters/ issue13.htm, 2012.

[7] LAOUTARIS N, SMARAGDAKIS G, RODRIGUEZ P, *et al*. Delay tolerant bulk data transfers on the internet[A]. Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems[C]. Seattle, WA, USA, 2009.229-238.

[8] SHERWOOD R, BRAUD R, BHATTACHARJEE B. Slurpie: a cooperative bulk data transfer protocol[A]. IEEE INFOC HongKong, China, 2004.941-951.

[9] P2PStitcher [EB/OL]. http://sourceforge.net/p/p2pstitcher/, 2012.

[10] PlanetLab [EB/OL]. https://www.planet-lab.org/, 2012.

[11] NYGREN E, SITARAMAN R K, SUN J. The akamai network: a platform for high-performance internet applications[J]. SIGOPS Operation Systems Review, 2010, 44(3):2-19.

[12] COHEN B. Incentives build robustness in BitTorrent[A]. Proceedings of the Workshop on Economics of Peer-to-Peer Systems[C]. Berkeley, CA, USA, 2003.68-72.

[13] XU D, KULKARNI S, ROSENBERG C, *et al*. Analysis of a CDN-P2P hybrid architecture for cost-effective streaming media distribution[J]. Multimedia Systems, 2006, 11(4):383-399.

[14] BJORKQVIST M, CHEN L Y, VUKOLIC M, *et al*. Minimizing retrieval latency for content cloud[A]. IEEE INFOCOM[C]. Shanghai, China, 2011. 1080-1088.

[15] WANG J, CHEN J, YANG M, *et al*. Traffic regulation with single-and dual-homed ISPs under a percentile-based pricing policy[J]. Journal of Combinatorial Optimization, 2009, 17(3):247-273.

[16] YANG X Y, VECIANAG D. Service capacity of peer to peer networks[A]. IEEE INFOCOM[C]. HongKong, China, 2004. 2242-2252.

[17] GRIMMETT G R, STIRZAKER D R. Probability and Random Processes[M]. Oxford,2001.

[18] CARTER R L, CROVELLA M E. Server selection using dynam path characterization in wide-area networks[A]. IEEE INFOCOM[C]. Washington, DC, USA, 1997.1014-1021.

作者简介:

黄永锋（1978-），男，江苏溧阳人，东南大学博士生，主要研究方向为云计算环境下的内容分发、时间演化的动态连通、网络的传输机制等。

董永强（1973-），男，河南渑池人，博士，东南大学副研究员，主要研究方向为网络体系结构、移动网络计算、网络融合环境、高效内容分发、动态连通网络等。

张三峰（1979-），男，山东金乡人，博士，东南大学讲师，主要研究方向为 P2P 网络、信任管理、移动自组网等。

吴国新[通信作者]（1956-），男，安徽芜湖人，硕士，东南大学教授、博士生导师，主要研究方向为网络协议、网络安全、自组网等。E-mail: gwu@seu.edu.cn。