# 特征相关的结构化 P2P 结点编址和数据副本分发

兰明敬
(信息工程大学 郑州 450002)

**摘 要** 传统结构化 P2P 网络中,随机或顺序产生结点标识,结点分布与结点位置、安全性等特征间缺乏相关性,无法有效应对"错误相关"现象,存在数据丢失的风险。提出一种新的编址方法和相关联的副本分发算法,该算法将结点的位置、安全性等特征信息融入到结点标识中,使结点按特征分布,在副本分发过程中依据标识来识别结点特征,避开或靠近具有特定特征的结点,解决错误等问题,提高分发效率。仿真实验给出了改进后的结点分布和备份点选择结果,表明了方法的有效性。

**关键词** 结构化对等网,结点特征,结点标识,编址方法,副本分发

**中图法分类号** TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.10.037

## Feature-related Node Address and Replica Distribution over Structured P2P Networks

LAN Ming-Jing
(College of Information Engineering, Zhengzhou 450002, China)

**Abstract** In the traditional structured P2P network, node identifications are generated randomly or based on order. There is a lack of correlation between the features such as node distribution, node location, and safety. The phenomenon of "wrong correlation" cannot be dealt with effectively, and there is a risk of data corruption. A new node addressing method was proposed in this paper, together with its correlated replica distribution algorithm. By combining characteristic information such as location and security of nodes into node identification, the nodes can be distributed according to their features. According to these features, nodes can be avoided or approached during the replica distribution procedure. Based on this, problems such as "wrong correlation" are solved, and the distribution efficiency is improved. The node distribution and backup nodes selection result after improvement were given out in the emulation experiment, which shows the effectiveness of the method.

**Keywords** Structured peer-to-peer, Node feature, Node identification, Address method, Replica distribution

## 1 背景

结构化 P2P 技术已经被广泛应用于音乐、视频分享中。由于结点自由加入和退出的特性,P2P 系统必须采用多点冗余备份的策略保证数据的持久存储。随之而来的问题是,如何选取一个最合适的结点集合来存放冗余数据以达到最好的数据持久性。不适当的结点组合将可能极大地消耗系统带宽,甚至威胁系统中数据的持久性[1]。例如,将数据的多个副本存放在一个错误相关的结点集合上,即集合中的结点可能由于区域断网或断电而同时离线,这样即便存在多个数据副本,也容易出现数据不可用的情况。

在 P2P 网络中,主要存在 DHT(Distributed Hash Table)直接分发和基于目录的间接分发两种备份点选择方法。DHT 直接分发指将数据及其副本直接存放在负责其名字的结点和其后续结点集上,如 MIT 的 CFS。这种方法实现简单,但无法确保后续结点间不存在错误相关性,在有些应用场景下,标识距离相近的结点往往物理位置也相近,错误相关的可能性更大。

基于目录的分发方式则将数据随意分发至网络中的任意结点集上,最后将这个结点集合的位置信息作为目录信息存放在负责数据名字的结点和后继结点集上。当需要读取数据时,先读取目录信息,然后再定位到实际的数据存放结点,如 Berkeley 的 OceanStore[2,3],UCSD 的 Total Recall[5]。与前者相比,此方法改善了结点变动时数据移动的效率,但在数据可靠性方面并无提升,反而可能有所降低。首先,数据的目录信息同样采用 DHT 直接分发的方式存储和备份,一旦目录信息丢失则数据也将随之无法访问。其次,采用随机的方法分发数据也无法确保数据存储点不存在错误相关。因为考虑到较小的副本数目(一般为 3),这种随机性是很有限的。即使如 amazon S3 这样全球性的存储网络,其区域数目也仅仅是 9 个[6],按上述方法随机选择 3 个结点,则 3 个结点在同一区域的概率为 1/729,考虑到 S3 庞大的访问量,发生位置相关的次数将是非常庞大的。如果进一步考虑 S3 主机数目分布的严重不均衡性(超过 70%的主机位于弗吉尼亚州北部[7]),位置相关概率将大幅增加至 30%以上。

除上述避开错误相关结点等"避开"需求外,在有些应用

到稿日期:2013-12-12 返修日期:2014-03-13 本文受国家 863 计划项目(2009AA012417)资助。
兰明敬(1982—),男,博士生,讲师,主要研究方向为 P2P 系统、服务计算,E-mail:lanmingjing@126.com。

中，我们还存在“靠近”需求，即希望将数据及其副本存储在具有共同特征的结点上。比如为加快传输速度，将数据及其副本存放在同一区域中；或者为保证安全性，将数据及其副本存放在相同的安全域中。上述两种数据分发策略显然无法满足这种“靠近”需求，除非建立某种机制，存储、管理结点特征，在分发前查询这些特征，依据特征按“避开”或靠近“策略”分发数据。然而结点特征的存储和管理又会带来新的问题，集中式存储存在单点失效和瓶颈问题，分布式存储则使得问题更为复杂，需要大量的维护操作和带宽消耗。

本文在结点编址层面解决按需分发问题，无需附加的结点特征存储和管理机制。本文的贡献概括如下：1）提出包含特征码的标识结构和编址方法，据此，结点按特征分布；2）提出支持特征避开和接近的副本分发算法；3）讨论了强制和非强制特征，提出了按特征满意度分发数据的方法；4）讨论了方法对 P2P 系统地址空间和系统性能的影响，以及对方法的扩展、改进；5）通过实验，验证了方法的有效性。

本文第 2 节从单“避开”特征开始，逐步深入，分别探讨多“避开”特征、“靠近”特征、特征优先级问题，给出相应的编址方法和备份点选择算法；第 3 节描述测试环境和测试数据，反映出方法是有效的；最后总结全文并讨论了方法对地址空间、性能的影响，讨论了对方法的扩展。

## 2 方法

“避开”和“靠近”都是针对结点的某种特征而言的，如地理位置、安全域、结点类型等。所谓特征相关的结点编址方法，简言之，就是将结点特征信息融入到结点标识中。特征相关的副本分发算法则读取这些特征信息，判断结点具有哪些特征，按“避开”或“靠近”需求为数据副本选择合适的存储点。

为表述方便，做如下定义：称负责数据名称的结点为数据的存储点；称负责存储数据副本的结点为副本存储点或备份点；称副本分发过程中需要“避开”的特征为“避开特征”，需要“靠近”的特征为“靠近特征”；称共有某特征的结点具有“某特征相关关系”；使用某特征相关关系可将结点空间划分为若干结点集合，称之为“特征域”，本节不考虑特征域交叉的情况，第 3 节会扩展讨论此问题；称结点标识中表示每个特征的字段为该特征的“特征码”；对于避开特征，若两个特征码相同，则称这两个特征码“不相容”或“互斥”，否则称之为“相容”；对于靠近特征，若两个特征码相同，则这两个特征码相容，否则互斥；多个特征码按顺序组合在一起形成“特征序列”，若两个特征序列中的所有特征码均对应相容，则这两个特征序列相容，否则互斥。

设 P2P 系统标识空间大小为 $S$，每一数据的副本数目（包含数据及其备份）为 $K$。

先从“地理位置”这种特征入手，讨论具有单避开特征的编址方法和相应的副本分发算法。

### 2.1 单避开特征

依据地理位置相关关系，将 P2P 系统中的结点集合划分为 $M(M<<S)$ 个特征域，编号为 0 到 $M-1$。先假设 $K$ 小于 $M$，后文会讨论 $K$ 大于等于 $M$ 的情况。

采用二进制的形式表示结点标识，由高位到低位表示如下：

$$f_m, f_{m-1}, \cdots, f_1, a_n, a_{n-1}, \cdots, a_1 \quad (1)$$

称 $f_m, f_{m-1}, \cdots, f_1$ 为特征码，$m=\lceil \log_2^M \rceil$；$a_n, a_{n-1}, \cdots, a_1$ 为名称码，$n=|\log_2^S|-m$。

某结点加入 P2P 网络时，采用如下方法为其产生结点标识。

方法 1：单避开特征编址方法

1）使用 P2P 系统原有的标识符分配方法（如随机分配）为结点分配长度为 $n$ 的名称码；

2）将结点所属特征域的编号值转换为二进制形式的特征码；

3）按式（1）的形式顺序连接特征码和名称码，得到完整的二进制标识序列；

4）将二进制的标识序列转换为 P2P 算法所需的形式，得到结点标识符。

按照上述编址方法，具有相同特征的结点聚集在一起。通过比较标识符的前 $m$ 位特征码，即可判断任意两个标识符对应的结点是否位置相关。

基于此，很容易在已有 DHT 直接分发方法的基础上进行改进，得到避免位置相关的副本分发方法。设数据存储点的结点标识为 $nd$，选取备份点的算法描述如下。

**算法 1** 单避开特征备份点选择算法

```
k←0, nlist←NULL, flist←NULL;
ac←nd 名称码值加 1;
将 nd 特征码放入 flist;
循环 A: k<K
  fc←产生一个与 flist 中所有相容特征码均相容的特征码;
  若 fc 为空，则选择失败，返回;
  将 fc 放入 flist;
  nc=使用 fc 和 ac 生成结点标识;
  ns←使用 nc 查找结点;
  若 ns 不为空，则将 ns 放入 nlist，k 值加 1;
循环 A 结束;
返回 nlist;
```

传统 DHT 直接分发方法将数据副本存储在数据存储点的后继结点上，与之相比，算法 1 将副本存储在不同特征域中的“后续结点”上，这里的“后续”是针对名称码而言的。假设某 P2P 系统有 5 个特征域，数据存储点位于编号为 3 的特征域中，其名称码值为 27，数据需要备份 3 份，则 1、2、4、5 号特征域中的名称码为 28 的结点即是符合要求的“后续结点”，若某特征域中此结点不存在，则顺序寻找名称码为 29 的结点，以此类推，若到达特征域边界则从名称码为 0 的结点起继续查找，直至找到有效结点或遍历整个特征域。

语句“产生一个与 flist 中所有相容特征码均相容的特征码”是算法的关键。flist 一开始存储的是存储点的特征码，之后逐步加入备份点的特征码。由于是避开特征，因此产生的特征码需要与 flist 中所有特征码均不相同。可以采用自增的方法产生该特征码，每次都在 flist 最大的特征码上加一得到新的特征码，当达到最大值时变为 0 继续增加。若遍历完所有特征码值仍找不到符合条件的特征码，则 $fc$ 为空，算法失败返回。

### 2.2 多避开特征

有时我们需要同时考虑多种避开特征，比如需要将副本存放在不同区域的不同类型的结点上。为此，需要在结点标识中融入多个“避开”特征。

设这些特征为 $F1,F2,\cdots,Fh$，对应的特征相关关系为 $R1,R2,\cdots,Rh$，$R_1$ 将结点集合划分为 $M1$ 个特征域，编号为 0－$M1$；$R_2$ 将结点集合划分为 $M2$ 个特征域，编号为 0－$M2$；以此类推，$Rh$ 将结点集合划分为 $Mh$ 个特征域，编号为 0－$Mh$。扩展式 1，变成如下形式：

$$f1_{m1},f1_{m1-1},\cdots,f1_1,f2_{m2},f2_{m2-1},\cdots,f2_1,\cdots,fh_{mh},fh_{mh-1},\cdots,fh_1,a_n,a_{n-1},\cdots,a_1 \quad (2)$$

其中，$f1_{m1},\cdots,f1_1;f2_{m2},\cdots,f2_1;\cdots;fh_{mh},\cdots,fh_1$ 分别为特征 $F1,F2,\cdots,Fh$ 对应的特征码，$mi=|\log_2^{Mi}|$，这些特征码组合在一起形成特征序列。编址方法扩展为：

方法 2：多避开特征编址方法

1）使用方法 1 所述方法产生名称码；

2）关系 $R1,R2,\cdots,Rh$ 划分的特征域中，设结点所属特征域编号分别为 $f1,f2,\cdots,fh$，分别将 $f1-fh$ 转换为特征码：$f1_{m1},\cdots,f1_1;f2_{m2},f2_{m2-1},f2_1;\cdots,fh_{mh},\cdots,fh_1$；

3）按式(2)的形式顺序连接各特征码和名称码，得到完整的二进制标识序列；

4）将二进制的标识序列转换为 P2P 算法所需的形式，得到结点标识符。

相应地，修改备份点选择算法，每次查询时需要产生一个与存储点和已有备份点均相容的特征序列。

**算法 2** 多避开特征备份点选择算法

```
k←0,nlist←NULL,flist←NULL;
ac←nd 名称码值加 1;
将 nd 特征序列放入 flist ;
循环 C:从右到左考察 nd 的第 i 个特征码
  flist[i]←nd 的第 i 个特征码;
循环 C 结束;
循环 A:k < K
  fcl←NULL;
  循环 B:从右到左考察特征码序列的第 i 个特征码
  fc←产生一个与 flist[i]中所有特征码均相容的特征码;
  若 fc 为空,则选择失败,返回;
  将 fc 放入 flist[i];
  fcl←连接 fcl 和 fc;
  循环 B 结束;
  nc=使用 fcl 和 ac 生成结点标识;
  ns←使用 nc 查找结点;
  若 ns 不为空,则将 ns 放入 nlist,k 值加 1;
循环 A 结束;
返回 nlist;
```

### 2.3 多类型特征

进一步考虑“靠近”需求。有时我们既需要“避开”特征相关的结点也需要“靠近”特征相关的结点，比如需要将副本存放在同一区域的不同类型的结点上。

为描述这种需求，我们调整特征码表示形式，在特征码最高位前增加一个类型位。若某特征是避开特征，则此位为 0，否则为 1。

支持避开和靠近特征的编址方法与方法 2 差别不大，仅需要按特征类型填写类型位即可。在备份点选择算法中，需要相应调整算法 2 中产生相容特征序列以及查找不到结点的处理方法：

```
……
fcl←NULL;
ad←nd 名称码值加 1;
循环 B:从右到左考察特征码序列的第 i 个特征码
  fc←产生一个与 flist[i]中所有特征码均相容的特征码;
  若 fc 为空,则选择失败,返回;
  若 nd 的第 i 个特征码最高位为 1,则 ad 值加 1;
  若 nd 的第 i 个特征码最高位为 0,则将 fc 放入 flist[i];
  fcl←连接 fcl 和 fc;
循环 B 结束;
ac←ad;
nc=使用 fcl 和 ac 生成结点标识;
ns←使用 nc 查找结点;
fsl←ns 特征序列;
若 ns 为空且 fcl 中不存在避开特征码,则选择失败,返回;
若 ns 不为空,则将 ns 放入 nlist,k 值加 1;
……
```

对于靠近特征，其相容特征码为其自身。需要注意的是，由于涉及到在相同安全域中选取后继，因此需要不断增加后续结点的起始值，避免重复选取同安全域中相同的结点；当特征序列中全部是靠近特征时，若遍历存储点所在安全域仍找不到足够的备份点，此时算法执行失败。

### 2.4 非强制特征和特征满意度

前面所说的几个备份点选择算法中都存在一条这样的语句“若 fc 为空，则选择失败，返回”，即算法并不一定总是能执行成功。发生在以下两种情况下：

一种情况是 P2P 系统结点数目较少或分布严重不均，某个安全域中的结点数小于按靠近特征分发副本所需“靠近”的结点数。比如某 P2P 系统的某个区域仅有 3 个结点，而按位置靠近策略，该区域中的数据需要备份到相同区域的 3 个不同的结点上，这显然无法分发。

另一种情况是 $M$ 小于等于 $K$ 的情况，即某特征划分的区域数不满足避开该特征所需的区域数目。如某 P2P 系统的结点分布在仅有的两个地理区域中，这种情况下显然也无法为数据选择 3 个与存储点区域不同且互不相同的地理区域。

为避免此类问题，最直观的策略是排除上述不满足条件的特征，如前所述，这会降低系统性能或可靠性。更好的解决方法是修改上面的算法，将区域特征作为一种非强制特征，优先而并不必须避开或靠近具有该特征的结点。

定义“强制特征”是一个必须满足的特征，对应的特征码称为“强制特征码”。若标识中存在强制特征码，则备份点必须在此特征码上与数据存储点和其他已选备份点相容，否则不能作为备份点。而“非强制特征”则是一个可以不被满足的特征，对应特征码称为“非强制特征码”。即使某结点在此特征码上与数据存储点或其他备份点不相容，此结点也可成为备份点。

若同时存在4个候选结点A、B、C、D，在强制特征码上均与数据存储点相容，但在非强制特征码均存在不相容的情况，这时该如何为数据选择3个备份点？

为此，提出特征优先级和特征满意度的概念。修改特征码表示形式，为其扩充优先级属性，如下所示：

$t, p_u, \cdots, p_1, f_m, \cdots, f_1$

最高位 $t$ 以及 $f_m, \cdots, f_1$ 位的含义与前文相同，前者表示特征类型，后者存储结点所属特征域编号；$p_u, \cdots, p_1$ 用于表示该特征的优先级，数值小于0则表明该特征码是一个强制特征码；数值大于0则表明该特征码是非强制特征码，数值越大则优先级越高。

结点编址过程中，优先级高的特征码放置在前面，即结点首先按高优先级特征聚集在一起，之后彼此再按低优先级特征聚集。备份点选取算法中，首先满足强制特征相容需求，之后按优先级先后顺序分别为非强制特征生成相容特征码，若无法为某非强制特征生成特征码，则随机为其分配一个不相容的特征码。

完整的、支持多个“避让”和“靠近”特征、支持特征满意度的结点编址方法和备份点选择算法描述如下。

方法3：特征相关的结点编址方法

准备工作：

1）罗列副本选择过程涉及的各类“避开”、“靠近”需求，明确这些需求对应的特征及其优先级；

2）确定标识序列构成，为序列中的每个特征码填入类型位和优先级字段；

3）依据特征相关关系，将结点划分为若干特征域并编号。

结点编址步骤：

1）使用P2P系统原有的标识符分配方法为结点分配名称码；

2）分别考察标识序列中的每一个特征码，填入对应的结点所属特征域编号；

3）顺序连接各特征码和名称码，得到完整的二进制标识序列；

4）将二进制的标识序列转换为P2P算法所需的形式，得到结点标识符。

**算法3** 需求相关的备份点选择算法

```
k←0,nlist←NULL,flist←NULL;
ac←nd 名称码值加 1;
将 nd 特征序列放入 flist ;
循环 C:从右到左考察 nd 的第 i 个特征码
  flist[i]← nd 的第 i 个特征码;
循环 C 结束;
循环 A:k < K
  fcl←NULL;
  ad←nd 名称码值加 1;
  循环 B:按优先级由高到低的顺序
        考察特征码序列的第 i 个特征码
        fc←产生一个与 flist[i]中所有特征码均相容的特征码;
        若 fc 为空,则
            若第 i 位特征码是强制特征,则选择失败,返回;
            否则,fc←随机选择一个特征码;
        若 nd 的第 i 个特征码最高位为 1,则 ad 值加 1;
        若 nd 的第 i 个特征码最高位为 0,则将 fc 放入 flist[i];
        fcl←连接 fcl 和 fc;
  循环 B 结束;
  ac←ad;
  nc=使用 fcl 和 ac 生成结点标识;
  ns←使用 nc 查找结点;
  fsl←ns 特征序列;
  若 ns 为空且 fcl 中不存在避开特征码,则选择失败,返回;
  若 ns 不为空,则将 ns 放入 nlist,k 值加 1;
循环 A 结束;
返回 nlist;
```

## 3 实验

为直观地看到采用特征相关编址方法后结点的分布形态，我们对Chord算法进行了修改，在p2psim仿真环境上进行了若干次实验。其中结点空间大小为65536，随机产生1000个结点；使用文件名产生数据标识，将数据存储在与其标识相等的结点上，若该结点不存在，则标识加一直至找到存储点；每个数据需要备份3次。

第一次实验中，我们以地理位置作为强制特征，副本分发时需要避开此特征。共有5个地理区域，分别编号为0－4。结点标识序列中前3位为仅包含特征域编号的特征码，后13位为名称码。采用随机方法为结点分配名称码和特征域编号。

得到的结点标识和特征域的关系如图1所示，很明显，结点按特征域聚集在一起。

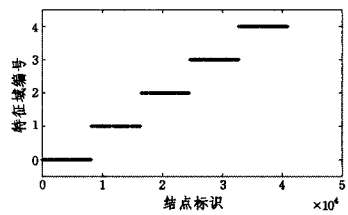


图1 结点标识与其所属特征域的关系

我们再考察数据备份的情况，将数据标识分别为35339、3765、5210的3份数据存储到系统中，每个数据的存储点位置和备份点位置如表1所列。

表1 存储点和备份点位置

| 数据 | 数据分布(名称码,特征域编号,结点标识) | |
|---|---|---|
| | 存储点 | 备份点 |
| 10827 | (2677,10869,1) | (2748,19132,2),(2679,27255,3),(2702,35470,4) |
| 28421 | (3849,28425,3) | (3868,36636,4),(3976,3976,0),(3896,12088,1) |
| 39582 | (6823,39591,4 ) | (6833,6833,0 ),(6900,15092,1),(6896,23280,2) |

标识为28421的数据的存储点位于3号特征域，备份点位于4、0、1号特征域，如图2所示，可以很明显地看出其按地理位置分布的特性。

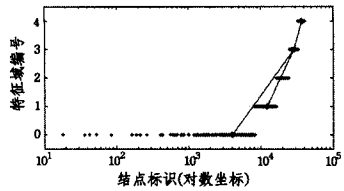


图 2　数据存储点和备份点的分布

第二次实验中，我们以地理区域和安全域作为强制特征，数据副本需要分发到相同安全域中不同地理位置的结点上。共有 5 个地理区域，分别编号为 0－4。4 个安全域，编号为 0－3。结点标识序列中前 2 位为安全域编号，接下来的 3 位为地理域编号，后 11 位为名称码。采用随机方法为结点分配名称码和特征域编号。

得到的结点分布状况和标识为 39107 数据的存储情况如图 3 所示。标识为 39107 的数据被存储在标识为 39120 的结点上，其副本被分发到标识为 41224、32989、35036 的结点上，存储点和备份点的类型域编号分别为 2、2、2、2，地理域编号分别为 3、4、0、1，满足在同一安全域但在不同地理位置的要求，同时还具有很明显的规律，这是由备份点选择算法决定的。

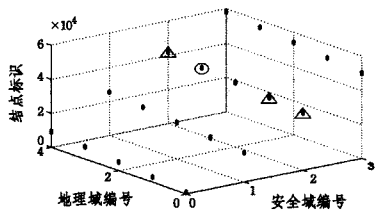


图 3　结点标识和特征域的关系

通过上述实验我们得出结论，特征相关的结点编址导致结点按特征聚集，特征相关的副本分发算法能够满足避开、靠近需求。

## 4　总结与讨论

本文提出特征相关的编址方法和副本分发算法。在结点标识中融入结点所具有的特征信息，依据此信息可在副本分发过程识别结点特性，避开或靠近指定类型的结点，从而提高系统可用性和数据分发效率。

### 4.1　对地址空间的影响

由于需要在标识中放置特征信息，因此采用本文提出的方法在某些情况下会减少地址空间所能容纳的结点数目。可扩充地址空间解决此问题，也可采用更合理的类型划分和名称码产生方法，降低特征码对地址空间的影响。一方面，在满足需要的前提下，应精简特征码数量和长度。对于大部分应用来说，两个特征码就已足够。特征域的划分也不能过于零散，特征域数目越多则需要的特征码位数也越多。另外，当同时使用靠近和避开两类特征，或使用非强制特征时，特征类型以及特征优先级位进一步增加了特征码长度。而这种增加是可通过其他方法避免的，比如使用配置文件存储特征信息甚至直接写到算法中，这并不影响系统的可用性。另一方面，可改善名称码产生方法。由于特征码的存在，不同特征域中的名称码是可重复的，因此每个特征域单独分配名称码可提高地址空间利用率。

### 4.2　对系统性能的影响

特征码的设置不会降低系统性能，合适的特征划分反而能有效提高系统路由效率和数据分布效率。比如使用地理位置作为特征，这将导致地理位置相同的结点具有更近的标识距离；而在大部分路由算法中，标识距离更近的结点间消息交换的次数也更多，这种情况下，地理位置相关的结点分布能够充分利用位置相近的结点间更好的网络状况，提高路由的效率。同样的道理，按地理位置分布也能提高数据副本分发过程中的传输效率。

### 4.3　特征相关的数据编址和存储点选择

本文解决的是副本分发过程中的避开、靠近问题，选择备份点的依据是候选备份点特征与数据存储点和已选备份点间的关系。

事实上，本文提出的结点编址算法和副本分发方法可以很容易扩展到数据存储点的选择上。即采用类似的编制方法为数据生成标识，使数据标识能够反映数据的某种特征。存储数据时，依据数据的特征为数据选择存储点和备份点。

### 4.4　特征域交叉问题

特征域交叉指的是使用某特征相关关系划分结点集合时，得到的若干特征域之间交集不为空的情况。交集中的结点同时位于一个或多个同类特征域中，比如某结点同时隶属于亚洲和欧洲两个地理域中。

第 2 节所述特征码结构只能存储一个特征域编号，因此要想支持特征域交叉现象，需要增加特征码长度，使之能够同时容纳多个特征域编号。在副本分发算法中，判断两特征码是否相容的条件修改为，对于避开特征，只有当特征码中的所有编号都不相等时才被认为相容；对于靠近特征，若存在一个编号相等，则两特征码相容。

特征域交叉的情况越严重，则需要的特征码长度也越大，相应地，结点管理策略也越复杂。除非必须，否则应限制特征域交叉现象。

## 参 考 文 献

[1] Tian J, Dai Y F. A survey of durable peer to peer storage techniques[J]. Journal of Software, 2007, 18(6)

[2] Wells C. The oceanstore archive: Goals, structures, and self-repair[R]. UC Berkeley Masters Report. May 2002

[3] Kubiatowicz J, Wells C, Zhao B, et al. OceanStore: an architecture for global-scale persistent storage[C]// Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems. 2000: 190-201

[4] Rhea S, Eaton P, Geels D, et al. Pond: the OceanStore prototype [C]// Proc. of FAST, 2003. 2003

[5] Bhagwan R, Tati K, Cheng Y C, et al. Total Recall: System Support for Automated Availability Management[C]// Proc. of the First ACM/Usenix Symposium on Networked Systems Design and Implementation (NSDI). 2004

[6] Amazon S3[OL]. http://aws.amazon.com/en/s3/

(下转第 183 页)

对 CMAR 和 MASK 进行对比，可以发现，在网络背景流量不大的情况下，CMAR 和 MASK 性能相近，MASK 甚至会优于 CMAR，这是因为 MASK 密码学运算负荷比 CMAR 低；但是随着网络背景流量增加，CMAR 的性能明显比 MASK 要好，这得益于 CMAR 的分簇多路径相对于 MASK 简单的节点不相交多路径方式，无论是多路径的条数还是路径的分散度、报文经过路径的随机性均有很大优势，因此网络负载更加均衡。

**结束语** 本文首先基于双线性配对技术，描述了在分簇结构的网络中邻居节点之间密钥协商以及簇内路由表项建立的过程。在此基础之上，本文设计了基于网络分簇和多路径的匿名通信路由协议(CMAR)，该协议不但能够达到全面的匿名通信的目标，减小匿名通信所带来的密码学运算代价，而且充分利用了簇内和簇间存在多路径的优势，增强了网络通信能力和抗报文的追踪性能力。通过理论分析和仿真实验可以发现，CMAR 协议密码学运算负荷较小且通信性能良好。

## 参 考 文 献


[1] Kong J, Hong X. ANODR: Anonymous on Demand Routing with Untraceable routes for Mobile Ad-hoc Networks[C]// Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing (MOBIHOC'03). 2003:291-302

[2] Zhang Y, Liu W. MASK: Anonymous On-demand Routing in Mobile Ad Hoc Networks[J]. IEEE Transactions on Wireless Communications, 2006, 5(9):2376-2385

[3] Pongaliu K, Xiao L. Maintaining Source Privacy under Eavesdropping and Node Compromise Attacks[C]// Proceedings of the 30th IEEE International Conference on Computer Communications(INFOCOM'11). 2008:1656-1664

[4] Liu J, Hong X, Kong J, et al. A Hierarchical anonymous Routing Scheme for Mobile Ad-Hoc Networks[C]// Proceedings of the 2006 IEEE conference on Military communications (MILCOM'06). 2006:2310-2316

[5] Zhang R, Zhang Y, Fang Y. AOS: An Anonymous Overlay System for Mobile. Ad hoc Networks[J]. Wireless Networks, 2011, 17(4):843-859

[6] Nguyen A, Milosavljevic N, Fang Q, et al. Landmark Selection and Greedy Landmark-descent Routing for Sensor Networks[C]// Proceedings of the 26th IEEE Conference on Computer Communications(INFOCOM' 07). 2007:661-669

[7] Shao M, Hu W. Cross-layer Enhanced Source Location Privacy in Sensor Networks[C]// Proceedings of IEEE Conference on Sensor, Mesh & Ad Hoc Communication Networks (SECON' 09). 2009:1-9

[8] Wang H D, Sheng B, Li Q. Privacy-aware Routing in Sensor Networks[J]. Computer Networks, 2009, 53(9):1512-1529

[9] Wang H, Sheng B, Tan C, et al. WM-ECC: an elliptic curve cryptography suite on sensor motes [R]. Technical Report WM-CS-2007-11. CS Department, College of William and Mary, 2007:1-14

[10] Oliveira L, Aranha D, Gouvêa C, et al. TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks [J]. Computer Communications, 2011, 34(3):485-493

[11] Szczechowiak P, Kargl A, Scott M, et al. On the application of pairing based cryptography to wireless sensor networks[C]// Proceedings of the 2nd ACM Conference on Wireless Network Security(WISEC'09). 2009:1-12

[12] Ganesan P, Venugopalan R, Peddabachagari P, et al. Analyzing and modeling encryption overhead for sensor network nodes[C]// Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications(WSNA'03). 2003:151-159


(上接第 168 页)


[7] http://www.datacenterknowledge.com/archives/2012/03/14/estimate-amazon-cloud-backed-by-450000-servers

[8] Zhu X S. *Research on Semantic Peer-to-peer Overlay Route Model*[J]. Computer Engineering, 2008, 43(13):110-112

[9] Zhang Y J, Gu J H, Wang X Z. A Hierarchical P2P Semantic Overlay Network Architecture Based on Topic and Physical Proximity[J]. Journal of Electronics & Information Technology, 2008, 30(8)

[10] Wang C Z, Yang N, Chen H W. Improving Lookup Performance Based on Kademlia[C]// Proc. of the Second International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC 2010). Hubei: 2010:446-449

[11] Lian Q, Chen W, Zhang Z. On the Impact of Replica Placement to the Reliability of Distributed Brick Storage Systems[C]// Proceedings 25th IEEE International Conference on Distributed Computing Systems, 2005(ICDCS 2005). 2005:187-196

[12] Chand R, Cosnard M, Liquori L. Powerful resource discovery for Arigatoni overlay network [J]. Future Generation Computer Systems, 2008, 24(1):31-38

[13] Stevens T, Wauters T, Develder C, et al. Analysis of an anycast based overlay system for scalable service discovery and execution[J]. Computer Networks, 2010, 54(1):97-111

[14] Wu W M, Wu Y J, Zhao W Y. Chord-based Semantic Web Service Discovery[J]. Acta Electronica Sinica, 2007, 35(B12):152-155

[15] Zhang Y, Huang H, Yang D, et al. Bring QoS to P2P-based semantic service discovery for the Universal Network[J]. Personal and Ubiquitous Computing, 2009, 13(7):471-477

[16] Di Stefano A, Morana G, Zito D. A P2P strategy for QoS discovery and SLA negotiation in Grid environment[J]. Future Generation Computer Systems, 2009, 25(8):862-875

[17] Zhou J, Dou W. A QoS-Aware Service Selection Approach on P2P Network for Dynamic Cross-Organizational Workflow Development[C]// Proc. of the International Conference on Web Information Systems and Mining (WISM 2009). Shanghai: Springer-Verlag Berlin, 2009:289-298