一种基于 RAQ 的具有拓扑意识的覆盖网络①

Topology-Aware Overlay Network Based on RAQ

李兰英　刘威廷　(哈尔滨理工大学 计算机科学与技术学院 黑龙江 哈尔滨 150080)

摘　要： 本文提出一种基于 RAQ(Range Queriable)的 P2P 多维覆盖网络 T-RAQ，RAQ 是一种结构化 P2P 覆盖网，在 RAQ 中，节点可在多维空间上进行精确匹配查询和范围查询；同时，其路由开销并不依赖于查询空间的维数，但是 RAQ 并不具备拓扑意识。针对此问题，本文改进了最初的路由算法并使得路由具有拓扑意识，节点加入，节点离开以及节点路由表的构造都考虑到了底层物理网络，从而使得覆盖网与物理网的尽量匹配。

关键词： 覆盖网 拓扑意识 RAQ

1　介绍

P2P 覆盖网络拓扑的几何性质一直是研究的中心和热点，许多研究成果不断涌现，出现了一系列的基于 DHT(distributed hash table)的结构化对等网络设计，其中最具代表性的是：CAN[1]，Chord[2]，Pastry[3]以及 Tapestry[4]等几种拓扑结构，与无结构 P2P 系统(如 Napster ,Gnutella 等)相比，这几种结构在系统扩展性、资源定位速度上都有了很大的提高。在这几种拓扑结构中，通过路由表可以在有限跳数内定位资源，同时在算法上也有很多相似的地方，但有一个重要的区别就是：它们是否考虑到了底层物理网络，即是否具有拓扑意识。

Chord 没有考虑到底层物理网络，虽然其协议和覆盖网维护是轻量级的，但是覆盖网上的一跳在物理层上则可能是任意长的距离。

CAN 采用多维空间拓扑结构，空间被动态分配给其网络节点，每个节点负责一块，每个数据对象被映射到一个点，由负责该点所在区域的节点来保存。CAN 中会设置一些界标，每一个节点都会测量其与界标的相对距离，并且测量其与每个界标的往返时间(RTT，Round-Trip Time)，并按升序(或降序)对这些值进行排序。根据这些值，拓扑上相近的节点更可能有相同的排序，所以覆盖网上的邻居节点在物理网络上也能更相近。

Pastry 中的每个节点保存有叶集、路由表、邻居集。其中邻居集保存有在网络物理层与当前节点邻近的节点，以便在节点加入和离开的时候考虑到物理网络。

RAQ 中节点可在多维空间上进行精确查询和范围查询，但是 RAQ 并不具有拓扑意识，文章第二部分将会简单介绍 RAQ 的实现。

以上提及的几种覆盖网模型有的考虑到了物理网络，而有的则没有考虑到。因为使用底层网络信息来构造覆盖网能提高数据传输性能，所以我们应当关注那些具有拓扑意识的覆盖网络。

基于 RAQ[5]，本文提出一种新的多维覆盖网络，称之为 T-RAQ(Topology-aware RAQ)，其具有拓扑意识，并改进了路由算法。

本文结构组织如下：第 1 部分简要介绍拓扑意识问题，第 2 部分介绍 RAQ，第 3 部分详述 T-RAQ 的设计，第 4 部分介绍实验与结论，第 5 部分总结。

2　RAQ简介

在这一部分，介绍 RAQ 的基础设计和基本数据结构，因为 T-RAQ 的设计是基于 RAQ 的。

2.1 空间分割

分割树是 RAQ 的主要数据结构，其将空间分割成 n 个区域，一个区域对应一个节点。假设 r 是分割树的根并代表整个查询空间，树中每个中间节点将其区域分割成两个更小的区域。虽然只有分割树的树叶才

① 收稿时间:2008-11-10

代表了真实的网络节点，但是树中每个节点还是对应于搜索空间的一个区域。每一个网络节点(即分割树的一个树叶)都会有一个 PE(Plane Equation)，用以指定在整个空间中它所拥有的区域。每个 PE 由一些标签组成，如节点 x 的 PE 可表示如下：$X_{PE}=((p_1,d_1),(p_2,d_2),\ldots,(p_{r(x)},d_{r(x)}))$。其中，r(x)代表节点 x 与根节点的距离，$p_i$ 表示将第 i 个区域分割成两个区域的平面等式，d_i 决定了 pi 的一侧(左边还是右边)。RAQ 中的每个叶节点存储了它的 PE 以及其连接节点的 PE，利用这些信息，任意节点 x 可以局部的确认一个查询请求是属于 x 的左儿子或右儿子还是和 x 相连的其它节点。在图 1 中，节点 c 的 PE 是：[(x=4,–),(y=2,–),(x=2,+),(y=1,–)]。其中"+"和"–"分别表示平面的右边还是左边。图 2 表示了一个二维搜索空间，分割树即是根据此二维搜索空间建立的。

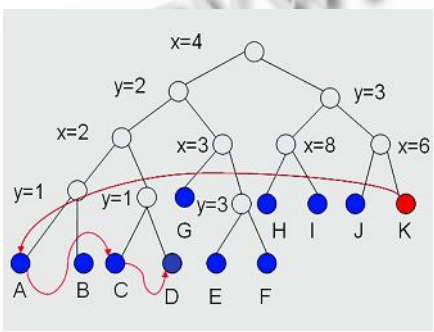

图 1　从 PE 为[(x=4,+),(y=3,+),(x=6,+)]的节点路由一个查询到目标点(2.5,1.5)

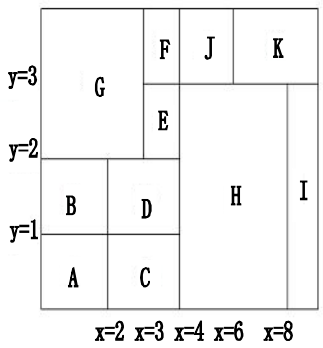

图 2　二维搜索空间

2.2 RAQ 中的网络连接

网络中的每个节点都与其它一些节点相连接，每个连接都有目标节点的一些地址信息，这些信息可以是目标节点的 IP 地址，也可以是其 PE。RAQ 中的连接规则是基于分割树的。考虑节点 x 与其 PE：$X_{PE}=((p_1,d_1),(p_2,d_2),\ldots,(p_{r(x)},d_{r(x)}))$，x 的连接规则意味着 x 必与一些节点相连，我们可以推测出这些节点的 PE：$[((p_1,\bar{d}_1))],[((p_1,d_1),(p_2,\bar{d}_2))],\ldots,[((p_1,d_1),(p_2,d_2),\ldots,(p_{r(x)},\bar{d}_{r(x)}))]$，其中 $\bar{d}_i$ 表示 d_i 的对应面。我们也很容易推断出在 RAQ 中每个节点都与 O(logn)个节点相连接。

2.3 RAQ 查询路由

当网络中的一个节点接到一个点的查询，它必须将这个查询路由到负责这个点的区域，也即负责这个区域的网络节点。假如节点 z 接到查询 Q，如果目标点和节点 z 的 PE 完全匹配，即查询点刚好就落在节点 z 负责的区域，则路由完成。否则，节点 z 将查询请求发送到和其相连的节点 y，当然，节点 y 的 PE 必须与目标点更加匹配。这个过程会一直继续，直到查询达到目标节点。

3　T-RAQ的设计

在这一部分，文章将论述如何修改 RAQ 以构造一个具有拓扑意识的覆盖网。在选择节点连接的时候，既考虑到 RAQ 的数据结构，又考虑到底层拓扑网络。同时，在 RAQ 数据结构的基础上，需要在路由表中加入更多的节点指针，所以会加入一个新的路由表。接下来将会详细阐述节点的加入、离开过程以及 T-RAQ 覆盖网的维护。

3.1 T-RAQ 的路由表

每个节点的路由信息由一个路由表和一个跳表组成，路由表中的每个条目包括一个节点的 PE 和 IP 地址。以下分别介绍路由表和跳表。

路由表有 O(logn)行，2^t列，其中 t 是一个配置参数，这里我们取其值为 2。路由表第 r 行 n 列的条目表示与本地节点 PE 的前 r 个标签相同的节点。第 r 行的所有条目都按照节点 PE 的第 r+1 个标签值进行了排序。图 2 描述了一个路由表，这个路由表和 Tapestry、PRR 中所使用的路由表很相似。

跳表中存放的是与当前节点共享其 PE 的一半的节点，图 3 描述了节点 c 的路由表和跳表，其 PE 为(x=4,–),(y=2,–),(x=2,+),(y=1,–)。例如在图中第三行节点 A 和节点 B 和当前节点 C 都有 PE 标签(x=4,–),(y=2,–)，A 和 B 有 PE 标签(x=4,–),(y=2,–),(x=2,–),再按照(x=4,–),(y=2,–)，(x=2,–)之后的一个标签排序就得到第三行的路由表。

(x=4,+)	H	I	J	K
(x=4,-),(y=2,+)	G	E	F	*
(x=4,-),(y=2,+),(x=2,-)	*	A	B	*
(x=4,-),(y=2,-),(x=2,+),(y=1,+)	*	D	*	*

(a)节点 c 的路由表

(x=4,-)(y=2,-)	A	B	D

(b)节点 c 的跳表

图 3　表中没有描述相关的 IP 地址，如果没有节点的 PE 适合路由表某个位置的条目，则此条目设置为“*”。

3.2 查询路由

在路由的每一步，通常当前节点将查询发送到至少比当前节点在 PE 上多匹配一个标签的节点，这里的匹配采用前缀匹配。如果没有这样的节点，则将查询发送到 PE 和目标点更相近的节点，并且此节点和当前节点与目标点都能前缀匹配相同的标签数。如果这样的节点也不存在，则查询就落在了本地节点，因为这就是与目标点最相近的节点了。注意，在向第 r 行某个节点发送查询的时候，会试图寻找一个 PE 的第 r+1 个标签也和目标点匹配的节点。

3.3 有拓扑意识的邻居选择

这一部分文章将讨论 T-RAQ 的拓扑意识问题。T-RAQ 寻求和底层物理网络的拓扑匹配以使得路由表更加有效。因此，当有许多节点都满足将其放入路由表某个位置的条件时(具体的说就是这些节点都有所需要的前缀)，具有拓扑意识的邻居选择则会选择物理网络最相近的节点，并将此节点的信息写入本地节点路由表。这里所说的物理网络上相近是基于一种相近机制的，相近机制的选择依赖于覆盖网所期望的质量(比如低延迟，高带宽)。T-RAQ 网络所用的相近机制是基于 RTT(round trip time)探测技术的。

这种邻居选择方法最早被提出来是在 PRR 和 Pastry 中。在 T-RAQ 中，系统期望的路由距离在刚开始的时候会比较小，但是随着每次路由成功，节点 PE 需要更长的前缀匹配，导致满足这一条件的节点越来越少，这样可供路由选择的节点也就少了，因此每路由一跳所经过的距离也会逐渐增加。

3.4 节点加入

假设新节点 x 要加入 T-RAQ 网络，它在搜索空间中选定一个完全随机的点 p，然后通过某种方式连接到一个现存节点 e。这里所谓“某种方式”可以是 IP 多播，也可以是 Web 网站提供众所周知的节点，不管那种方式，通常 e 与 x 在物理上会比较接近。

节点 e 路由加入请求，其路由目标点则为 p。x 从节点 e 那里得到其路由表的第一行和 PE 的第一个标签，然后 e 继续向前路由加入请求到达第二个节点，x 再从这个节点那里得到其路由表的第二行和 PE 的第二个标签，按照这种方式继续下去，就可以得到节点 x 的路由表和 PE 了。

新节点的加入引起了覆盖网的变化，所以还需要更新网络中其余节点的路由表，这样才能保证路由表在有新节点加入之后仍然是最有效的。一旦 x 初始化完了它的路由表，它将向其路由表中每一行的每一个条目所对应的节点发送那一行的路由信息，这样有两个目的：一是告知系统自己的加入，二是传播此新节点的信息。接到一行路由信息的那些节点将会比较一下新来的路由信息是不是比本身路由表相对位置的路由信息更好，然后对其路由表做适当的调整。这个过程保证了路由表中总是放着相近的节点。

3.5 节点离开

在 RAQ 中，当节点 x 建立一个连接到节点 e 时，节点 e 会保存节点 x 的地址信息，或者称之为到节点 x 的 dlink。当 e 要离开网络的时候，它通过它所保存的 dlink 向每一个与它相关联的节点发送一条消息，接到这条消息的节点随即将其路由表中相应的条目做一个标记为失效。而在 T-RAQ 中，当节点路由表的某个条目失效时，它会查询其某个邻居的路由表，如果发现有适合自己路由表失效条目处的路由条目，则将此条目写入自己的路由表。但是，节点没有为失效条目找到任何适合的条目时，就需要触发覆盖网的路由表维护程序来解决这个问题。

由于节点的离开或失效，覆盖网系统中节点的路由表变化会很大，如果不采取相应的措施，将会导致系统路由效率的下降。所以，系统中每一个节点都会运行一个路由表维护任务(一般每 20 分钟运行一次)。这个维护任务对本地节点路由表的每一行做以下一些操作：它首先随机选择这一行的一个条目，向和它相

连的某个节点请求相应的那一行的一份副本。然后将这两行条目做对位比较，如果有不相同的两个条目，则本地节点探测和这两个条目所对应的节点的距离，最后将距离较近的节点条目放入自己的路由表。

4 实验结果与分析

最后通过实验结果来衡量 T-RAQ 中具有拓扑意识的邻居选择方法的性能，将 T-RAQ 运行在一个网络仿真器上，就可以获得实验结果了。由于 T-RAQ 的路由机制并不依赖于搜索空间的维数，所以 T-RAQ 的维数 d 被设置为 2。

实验中，网络拓扑产生模块采用美国乔治大学 GT-ITM 软件包，原因是 GT-ITM 作为著名的开源代码的网络拓扑产生系统，可以产生多种模型的网络拓扑，例如：Transit-Stub(TS)模型、随机模型等。TS 模型能够较好地模拟 Internet 网络拓扑结构，存在着多个不同网络距离(时延)级别的域，类似于实际网络中存在的多个局域网、城域网的混杂联网模式。

下面测试节点加入时，为了维持系统的拓扑意识需要探测的节点数，分别选择了 1000 个节点到 4000 个节点的覆盖网，每一个覆盖网都测试出最大探测节点数和最小探测节点数，并计算出平均探测节点数。图 4 显示了实验结果，可以看到节点加入所需探测的节点数并不随覆盖网规模的扩大而增加，在维持系统的拓扑意识的同时，显示了 T-RAQ 网络良好的性能。

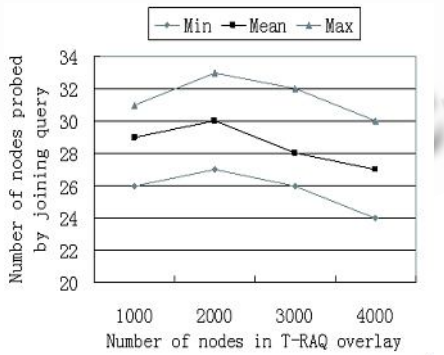

图 4 AQ 节点加入的探测节点数

接下来，路由 200 个查询(选择随机节点，使用随机的目标点)。图 5 分别显示了在 RAQ 中和在 T-RAQ 中的平均路由跳数以及距离比。在这里，距离比指的是路由一个查询所经过的总距离与源节点到目的节点的实际距离之间的比值，这个比值越小证明系统性能越好。从图中可见，T-RAQ 的距离比明显小于 RAQ，这是因为 T-RAQ 的构建具备拓扑意识，消息在覆盖网的路由线路更接近于物理网络的真实连接。

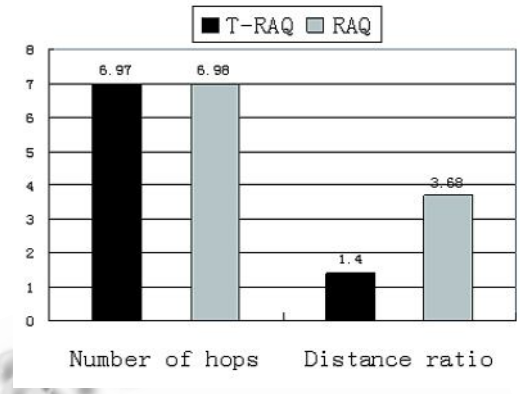

图 5 GT-ITM 拓扑中的平均路由跳数和距离比

5 结论

本文提出了一种多维空间拓扑意识覆盖网络，并通过实验分析了其性能。提出了优化的节点加入方法以及节点失效处理方法，这些方法都考虑到了拓扑匹配问题；同时，在路由表的建立与邻居选择上，也尽量考虑到了底层物理网络。最后，通过仿真实验证明了具有拓扑意识的 T-RAQ 覆盖网在性能上较之 RAQ 有很大提高。

参考文献

1 Ratnasamy S, Francis P, Handley M, Karp R. A scalable content-addressable network. Proceedings of Sigcomm 2001.San Diego,CA,USA,August 2001:161－172.

2 Stoica I, Morris R, Karger D, Kaashoek M, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for internet applications. Proceedings of Sigcomm 2001. San Deigo, CA, USA, August 2001:149－160.

3 Castro M, Druschel P, Hu Y, Rowstron A. Exploiting network proximity in distributed hash tables. Proceedings of FuDiCo 2002, Bertinoro, Italy, June 2002.

4 Zhao B, Kubiatowicz J, Joseph A. Tapestry: An infrastructure for fault-tolerant wide-area location and routing.Technical report, UC Berkeley, April 2001.

5 Nazerzadeh H, Ghodsi M. RAQ: A range queriable distributed data structure. SOFSEM 2005.February 2005.