

基于蚁群算法的树环结构应用层组播树构建算法

许建真, 何婷婷*, 何 丹, 周 瞳

(南京邮电大学 计算机学院, 南京 210003)

(*通信作者电子邮箱 tingtingher@hotmail.com)

摘 要:作为对 IP 组播技术的改进,应用层组播(ALM)技术具有不受网络架构限制、资源丰富、数据传输速率高等优势。综合考虑节点性能和端到端延迟,提出一种快速且高效的建立应用层组播树的方法:基于蚁群算法的树环结构 ALM 模型(ACOTRM)。目前对 ALM 结构的研究大部分都是仅仅粗略地给出拓扑结构,而对整个构建过程却缺乏完整且清晰的描述。鉴于此,完整地给出了一种树环结构 ALM 树的具体构建过程,主要包括网络节点分簇、簇内成环连接、数据转发数的建立等关键步骤。另外,为每个节点设定特有的优先级,使得整个 ALM 树达到最优化状态。仿真结果表明,该模型得到了较低的平均时延以及较高的平均数据传输率,在增加系统稳定性的同时也提高了其转发效率。

关键词:应用层多播;树环结构;蚁群优化算法;节点性能;优先级

中图分类号: TP393 **文献标志码:** A

Building algorithm for tree-ring application layer multicast based on ant colony algorithm

XU Jianzhen, HE Tingting*, HE Dan, ZHOU Tong

(School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu 210003, China)

Abstract: As an improvement of IP multicast technology, Application Layer Multicast (ALM) has many advantages such as unlimited network architecture, rich resource and high data transfer rate. Considering node performance and end to end delay, a fast and efficient method was proposed to establish application layer multicast tree, it was named Ant Colony Algorithm based Tree-ring Application Layer Multicast Model (ACOTRM). The available studies only gave a topology cursory and had no complete and clear description of the concrete construction process. In view of this, a complete ALM hierarchical tree-ring concrete construction process was put forward including several key steps, such as clustering division, connection in cluster ring, generation of feasible solution and maintenance of the model in survival time. In addition, in order to optimize the ALM state tree, each node was set with a specific priority. The simulation results show that the proposed model provides lower average delay and higher average data delivery ratio, which increases the system stability and forwarding efficiency at the same time.

Key words: Application Layer Multicast (ALM); tree-ring; Ant Colony Optimization (ACO) algorithm; node performance; priority

0 引言

作为对 IP 组播^[1-3]技术的改进,应用层组播(Application Layer Multicast, ALM)^[4-9]技术逐渐成为 Internet 中提供组播服务的主要方案。应用层组播是一种在覆盖网络(overlay network)上实现组播的特殊组播方式,它的组播功能由主机来实现,组播数据的实际传输通过底层网络的单播链接来进行。应用层组播与 IP 组播的最大区别就是组播的复制、转发功能是由主机实现,而不是由路由器实现。与在单播传输模型中一样,路由器只需要支持“尽力传输”功能,而无需支持与组播相关的其他额外功能。因此,可以认为应用层组播中的组播功能是在覆盖网络层次上进行的,此处的覆盖网络是指由节点自组织成的、架构在底层网络之上的逻辑网络。

提高应用层组播树稳定性^[10-11]的一个重要途径是减少节点失效影响范围,在此,树环结构应用层组播树有着比

NICE 协议更稳定且高效的优点。基于此,本文提出一种基于蚁群优化(Ant Colony Optimization, ACO)算法^[12]的树环结构应用层组播(Tree-Ring Application Layer Multicast Based on Ant Colony Optimization, ACOTRM)模型,综合考虑节点性能和端到端延迟,利用一种快速蚁群优化(Fast Ant Colony Optimization, ACO-F)算法快速且有效地实现环状结构的构建。另外,针对 NICE 协议中当 Leader 节点失效后没有对应的处理方法,从而导致网络收敛慢的问题,ACOTRM 根据节点的性能以及端到端延迟,为每个节点设定特有的优先级,然后在分簇中选举出具有最高优先级的节点作为本簇的 Leader,结合合理的分层数,使得整个分层结构达到最优化状态。

1 模型构建

1.1 模型结构

模型的整个拓扑结构分为 $m(m = 0, 1, 2, \dots)$ 层,各层分

收稿日期:2013-06-09;修回日期:2013-08-30。 基金项目:江苏省研究生科研创新计划项目(CXLX13_469)。

作者简介:许建真(1966-),男,安徽宿州人,教授,博士,主要研究方向:计算机通信与网间互联;何婷婷(1991-),女,安徽宿州人,硕士研究生,主要研究方向:计算机通信与网间互联、光网络优化;何丹(1988-),男,安徽宿州人,硕士研究生,主要研究方向:计算机通信与网间互联;周瞳(1990-),男,江苏盐城人,硕士研究生,主要研究方向:计算机通信与网间互联。

别表示为 L_0, L_1, \dots, L_{m-1} , 并将各层节点组织成环状结构。最高层 L_0 层为 Root 节点, 是整个体系结构的管理和控制中心。下层域中的环节点由上层环中某个特定的节点进行控制, 这样每个上层环中节点都分散控制管理着自己的下层域中节点, 从而减轻了 Root 节点的控制管理负担。该拓扑结构的最高层是 Root 节点, 紧接着的高层是一个由 Leader 节点组成的中心环, 这些 Leader 节点是通过计算节点的性能和优先级从普通节点中选出来的。图 1 是一个 3 层的树环结构应用层组播拓扑结构。由于随着树的深度增加, 网络中各节点的负担以及传输延时都将会增大。并且相对于以往的部分研究^[5], 本文提出的模型更为灵活, 各个节点控制管理负担比较小。因此, 适当增加每层节点数以减小树的深度, 从而平衡网络负担与传输时延。本文规定每一层中的节点总数在 2^{2i} 和 $2^{2(i+1)}$ 之间 (其中 i 表示层数), 通过实验的反复论证, 最后的仿真结果说明此规定能够达到比较短的时延以及较高的数据传输率。

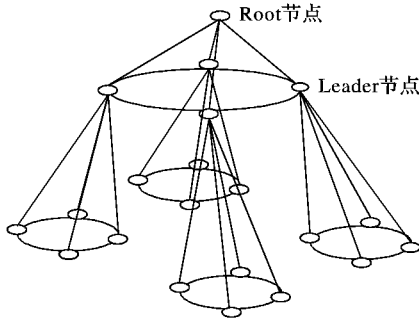


图 1 ACOHRM 模型结构示意图

1.2 基于蚁群算法的动态构建算法

由于应用层组播中的节点通常是普通的主机, 不能进行线速转发, 所以复制转发工作消耗的时间较大。因此, 在计算应用层组播延迟时不能忽略节点的转发处理时间, 本文将中间节点的转发处理时间和源节点的发送时间统称为“发送延迟”, 记为 sd ; 而把端到端延迟即“传输延迟”, 记为 td 。

D_{ij} 表示节点 i 和节点 j 的距离, 本文所有提到的距离均不是指物理意义上的距离, 而是综合考虑节点性能以及端到端延迟的一种矢量距离。

$$D_{ij} = \lambda_1 \times (sd_i + sd_j) / 2 + \lambda_2 \times td_{ij} \quad (1)$$

其中: sd_i 表示节点 i 的发送时延; td_{ij} 表示节点 i 和节点 j 之间的传输时延; $\lambda_1, \lambda_2 \in [0, 1]$ 。

本文提出的 ACOTRM 模型的构建过程分为聚类的划分、簇内成环连接和树环结构数据转发树的建立等几个关键步骤, 详细介绍如下。

1.2.1 基于蚁群聚类的节点分簇算法

蚁群优化 (ACO) 算法是由意大利学者 Dorigo 等^[12] 根据蚂蚁群体在觅食过程中所体现出的智能行为首先提出的一种仿生寻优算法。本文利用一种基于信息素的蚁群聚类方法对网络中的节点进行聚类划分, 主要思想是: 蚂蚁在觅食过程中会在其经过的路径上释放信息素, 并能够感知信息素及其强度; 经过蚂蚁越多的路径其信息素越强, 同时信息素自身也会随着时间的流逝而挥发; 蚂蚁倾向于信息素强度高的方向移动, 某一路径上走过的蚂蚁越多, 后来的蚂蚁选择该路径的概率就越大。

假设 $X = \{x_{i1}, x_{i2}, \dots, x_{in}, i = 1, 2, \dots, N\}$ 表示本网络的全部节点列表; τ_{ij} 表示两节点 i, j 间的信息素值; a_{ij} 表示节点 i 和节点 j 之间的路径; D_{ij} 已在上文中定义, 称为节点 i 和节点 j 之

间的矢量距离; 设簇的半径为 r 。

$$\tau_{ij}(t) = \begin{cases} 1, & D_{ij} \leq r \\ 0, & D_{ij} > r \end{cases} \quad (2)$$

则对象 X_i 合并到 X_j 的概率为

$$p_{ij}^k = \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}]^\beta}{\sum_{l \in S} [\tau_{il}(t)]^\alpha \times [\eta_{il}]^\beta} \quad (3)$$

其中: $S = \{X_s | D_{ij} \leq r, s = 1, 2, \dots, N\}$; η_{ij} 表示两节点 i, j 间的路径能见度, 本文取 $\eta_{ij} = Q/D_{ij}$, 其中 Q 为常数; α, β 分别表示路径轨迹的相对重要性和路径能见度的相对重要性。

1.2.2 簇内成环连接

通过对传统的蚁群优化算法学习, 可知参加寻径的蚂蚁通过留在链路上的信息素交互来选择新的路由从而达到寻径的目的。各路径上的信息量作更新如式 (4) 所示。

$$\tau_{ij}(t+1) = (1-\rho) \times \tau_{ij}(t) + \rho \times \Delta\tau_{ij} \quad (4)$$

其中: τ_{ij} 表示两节点 i, j 间的信息素值; $\rho \in [0, 1]$, 表示信息量 $\tau_{ij}(t)$ 随时间的推移而衰减的程度; $\Delta\tau_{ij}$ 表示信息增量。

虽然传统的蚁群优化算法机制使得蚁群算法具有很强的发现较好解的能力, 但是, 它也有一些缺陷。比如迭代次数多、收敛速度慢以及过早收敛于非全局最优解仍是制约 ACO 算法广泛应用的主要瓶颈。因此, 许多改进的 ACO 算法被提出。一般来说, 改进方法主要从两方面入手: 1) 信息素更新机制的改进, 比如每次让信息量最大的路径以较大的概率被选中, 以充分利用学习机制, 强化最优信息的反馈^[13]; 2) 搜索机制的改进, 对随机搜索过程的控制策略进行调整, 以减少迭代次数, 加速收敛。

本文综合考虑信息素机制和搜索机制的改进, 提出一种快速的蚁群优化算法 ACO-F 来求解簇内以及簇间节点成环问题。首先采用一种定向信息素^[14] 来提高短路径上的信息素, 以此来增大短路径的入选概率, 并且有效地减少了迭代等计算, 从而达到算法的快速收敛。定向信息素分布在各个节点, 本文用向量 τ_{ij} 来表示节点 i 到节点 j 的信息素值。不同于 ACO 算法, 本文的定向信息素的计算并没有挥发和叠加的过程, 而是用式 (5) 来进行定向信息素的更新:

$$\tau_{ij}(t+n) = \begin{cases} Q'/L_k, & Q'/L_k > \tau_{ij}(t) \\ \tau_{ij}(t), & \text{其他} \end{cases} \quad (5)$$

其中: $\tau_{ij}(t+n)$ 表示经过 n 个单位时间更新后的信息素, Q' 为常数, L_k 为从时刻 0 开始周游到现在经过的总距离。

另外, 传统蚁群算法在求解过程中存在容易陷入局部最优解的问题, 蚁群算法中蚁群的转移是由各条路径上留下的信息量浓度和城市间距离来引导的, 蚁群运动的路径总是趋向于信息量最强的路径, 这样信息素强的路径有可能多次被蚂蚁经过并且每次都需要进行计算, 这就造成了过多的重复计算, 从而增加了问题解的复杂度^[15]。因此, 本文用式 (6) 来计算蚂蚁 k 从节点 i 转移到节点 j 的概率:

$$p_{ij}^k = \begin{cases} 1, & \delta_{ij} = 1 \\ \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}]^\beta \times [\tau_{ij}(t)]^\gamma}{\sum_{l \in U_k} [\tau_{il}(t)]^\alpha \times [\eta_{il}]^\beta \times [\tau_{il}(t)]^\gamma}, & j \in U_k \text{ 且 } \delta_{ij} \neq 1 \\ 0, & \text{其他} \end{cases} \quad (6)$$

$$\delta_{ij} = \begin{cases} 1, & v_{ij} \geq \phi \\ 0, & \text{其他} \end{cases} \quad (7)$$

其中: U_k 表示蚂蚁点在本次周游中在当前位置允许选择的节点列表; τ_{ij} 表示两节点 i, j 间的信息素值; η_{ij} 表示两节点 i, j 间的路径能见度, 本文取 $\eta_{ij} = Q/D_{ij}$, 其中 Q 为常数; α, β 分别表示路径轨迹的相对重要性和路径能见度的相对重要性; 如上文定义, τ_{ij} 表示节点 i 到节点 j 的矢量信息素值; v_{ij} 表示边 a_{ij} 被访问的次数; 本文设 $\phi = \lambda m \varphi$, $\lambda \in [0, 1]$, m 为蚂蚁个数, φ 为迭代次数。

下面给出本文利用快速蚁群算法 ACO-F 求得簇内最优环的具体过程:

```
ACO-F (X)                                /* X 为簇标号 */
{
  for p = 1 to n                          /* 遍历所有节点, 最后回到初始节点 */
  for q = 1 to m
    /* 蚁群的一步转移(假设有 m 只蚂蚁随机的放在节点集中) */
    {
      if p < n then                       /* 还没有遍历完所有节点 */
      {
        按式(6)完成下一个节点选择, 并更新允许访问的节点列表
        按式(4)和式(5)进行信息素的更新
      }
    }
  }
  if 满足结束条件 then
    输出多次迭代后的最优解
  else  $\varphi = \varphi + 1$                       /* 迭代次数加 1 */
}
```

1.2.3 树环结构数据转发树的建立

如前文所述, ACOTRM 模型将组播中节点进行分层组织结构, 在每层分成不同的簇中采用环结构。该拓扑结构的最高层是 Root 节点, 上层环中的节点由下层各簇节点中选出簇头节点 Leader 组成。这样每个上层环中节点都分散控制管理着自己的下层域中节点, 从而减轻了 Root 节点的控制管理负担。

本文采用一种基于优先级的选择算法来进行各簇中 Leader 节点的选择。采用一种动态分析的方法根据各节点的平均效率、节点性能、端到端时延以及在系统的存在时间等, 计算节点相应的优先级。各层节点根据自己的优先级动态调整自己所在的层位置, 让拥有大量资源和平均时延小的节点在上面的层中。这样性能好的节点都保持在上面的层次, 有利于信息和数据的分发。下面给出节点优先级的计算方法:

$$pri(i) = P_effc(\rho_i) + \zeta D_i + \theta t_i / T \quad (8)$$

其中: $pri(i)$ 表示节点 i 的优先级, ρ_i 为一段时间 Δt 内节点 i 的平均效率, $P_effc(\rho)$ 表示随 ρ 的变化簇中节点优先级的变化趋势, D_i 表示节点 i 到本簇内所有节点的平均矢量距离, t_i 为节点 i 在系统的存在时间, T 为系统稳定时间, 参数 $\zeta, \theta \in [0, 1]$ 。

$$D_i = \frac{1}{N_i} \sum_{j \in C_i} \lambda_1 \times (sd_i + sd_j) / 2 + \lambda_2 \times td_{ij} \quad (9)$$

其中: C_i 表示节点 i 所在的簇; N_i 为簇 C_i 内的节点总数; sd_i , td_{ij} , λ_1, λ_2 均如前文式(1)所定义。

$$P_effc(\rho_i) = \frac{\rho}{1 + \rho} \exp\left(\frac{\xi \rho^2}{1 - \rho}\right); \xi \in [0, 1] \quad (10)$$

选举出来的 Leader 节点构成节点集, 利用上文提出的 ACO-F 算法完成 Leader 节点的成环连接工作, 选取 Leader 节点集中优先级最高的节点作为 Root 节点, 层次环状组播树的初步构建完成。稳定状态时, 每个簇的 Leader 通过 heart-beat 信息定期检查本簇中节点的优先级, 根据接收的节点的更新信息动态选举出本簇中优先级最高的节点作为该簇的新 Leader,

判断该簇是否需要更换 Leader; 如果不更换, 则原 Leader 的状态保持不变; 如果更换, 则该分组的原 Leader 设置作为新 Leader 的节点的信息表中的“Leader = 1, Activity ++”, 然后原 Leader 把更新信息通知其在上一层簇中的 Leader 以及本簇的所有节点。原 Leader 在其上一层簇中的 Leader 设置原 Leader 信息表中的“Leader = 0, Activity --”。

2 模型维护

2.1 节点加入

节点申请加入时, 首先向 L_0 层 Root 节点发送申请请求信息, 此申请请求信息包含节点所在网络状态以及节点自身物理位置等。Root 把收到的申请信息与自己保存的节点集信息表对照, 在 Root 节点中记录着 L_1 层节点的数量, 如果当 L_1 层节点数量未达到 Root 设定的最大节点数目时, 则申请加入的节点加入 L_1 层的节点集中。Root 节点会记录新加入的节点的信息, 并给节点发 ACK 确认信息。新加入 L_1 层中的节点将获得该层环中其他节点的信息, 并存储这些信息。Root 节点也将使其他 L_1 层中的节点相应更新自己的邻接 Leader 节点列表。但当 L_1 层中的节点数量达到 Root 设定的最大节点数目时, Root 节点将把申请加入的节点分配给 L_1 层中合适的 Leader 节点, 该 Leader 节点将把新加入的节点放在自己的下层域的环中。当节点加入某 Leader 节点的下层环时, 将更新自身的节点列表, 该表记录包含同一环中的邻居节点信息, 还记录了下层环的节点列表和上层 Leader 节点列表。然后新加入的节点将向自身节点列表的节点发送更新请求信息。Leader 节点收到信息后将记录新加入节点的位置, 其他收到更新请求的节点将根据具体的情况更新相应的列表。如果 Leader 节点的下层环的节点数目都达到初始设定的数值时, 中心环中的节点将把新加入的节点交给自己所属的下层环中的合适节点, 由它来分配新加入节点的位置。它会根据需要, 将新加入节点放在合适的环中。

2.2 节点退出

当 L_1 层环中的 Leader 节点正常退出时, 它将向 Root 节点和自己下属层环中的节点发送退出请求信息, Root 节点收到信息后将更新自己的节点记录, 该 Leader 节点下属层环收到信息后, 将根据 $pri(i)$ 公式从环中选择一个节点成为 Leader 节点, 并同时更新相应节点的节点列表。当其他环的节点正常退出时, 上层环的节点将把它从自己的列表信息中删除, 并通知退出节点所在环的其他节点更新自己的列表信息。

由于网络的实际环境, 系统的节点可能发生非正常退出。所以系统中的节点会周期性地发送心跳消息 (Heartbeat Message) 给上层所属节点。如果一个上层节点在几个心跳周期内都没有收到下层节点发来的心跳消息, 那么将把这个节点的状态视为离开, 并将这个下层节点在列表中的表项删除, 并通知其他相应的节点更新自己的节点列表。

系统每隔一段时间 Δt 便对组播树中的各个簇利用 ACO-F 算法进行环内节点的遍历, 改变各节点在环中的位置, 对环进行最优化, 以减小整个组播树的延迟。

3 实验结果及分析

本文用 Matlab 仿真软件进行仿真分析。本文假设初始加入的节点数在为 100, 然后在 300 s 内均匀增加到 1000。本文的参数取值为: 路径轨迹的相对重要性 $\alpha = 1$, 路径能见度

的相对重要 $\beta = 5, \rho = 0.2, Q = 100$, 迭代次数 $\varphi = 180$ 。

图2给出了ACOTRM算法与NICE算法平均时延的对比分析。这里的时延即本文定义的矢量距离, 平均时延是指从源端到组播树中节点的矢量距离之和与经过路径中节点数之比。由图2可以看出, ACOTRM模型比NICE模型有更小的平均延迟, 这是由于ACOTRM模型由上层环中节点对下层环中节点进行控制传输, 当上层节点不能及时传输数据到下层节点时, 下层环中收到数据的节点会及时传输数据给自己的邻居环中节点。当上层节点失效时, 下层环中节点会根据优先级快速选择一个上层节点出来, 而NICE中上层节点失效对下层节点影响比较大, 要进行拓扑结构重组。另外, 随着节点数的增大, ACOTRM模型下的时延并没有明显增大, 这是因为该模型每隔 Δt 的时间便进行基于ACO-F算法的最优环调整, 从而减少了延迟时间。

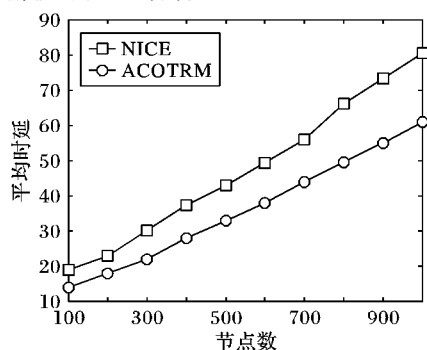


图2 平均时延的比较

图3给出了ACOTRM和NICE控制开销的对比。由于ACOTRM中引入了节点的优先级控制, Root为每个新加入的节点产生初始化列表, Root和各个分簇的Leader还要根据每个簇内节点的转移次数和该节点所包含的资源量计算该节点的优先级; 并且, 各个节点的变动信息要在各个分簇的Leader和Root之间进行交互。因此, ACOTRM的控制开销要比NICE大。但另一方面, ACOTRM中节点只需与邻居节点和父节点以及子节点阶段性地交换消息, 减少了节点间控制消息的数量, 相对于NICE簇中的每个节点都需要与簇内的其他所有节点阶段性地交换消息, 也在另外一方面减小了控制开销, 使总控制开销只是比NICE稍大, 但通过这种方式却增加了系统的稳定性以及提高了其转发效率。

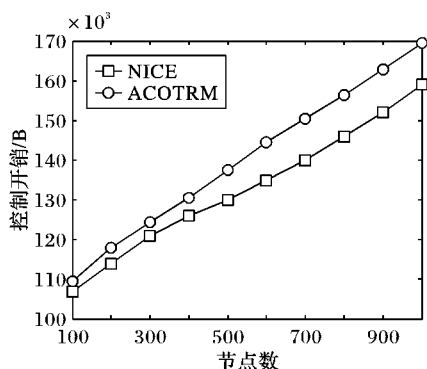


图3 控制开销的对比

图4给出了节点的失效或退出对ACOTRM和NICE性能的影响。实验改变突然离开多播组的节点数目, 并分别进行计算。由图4可以看出, ACOTRM的性能明显优于NICE, 这是因为在NICE结构中, 节点的入度数和出度数都比较高, 当一个节点失效或离开时, 导致大量的节点暂时地脱离数据传输路径而接收不到数据, 特别是当组播树内节点很多时, 数据

丢失比较严重, 因此, 使得数据传输率明显下降。而ACOTRM由上层环中节点对下层环中节点进行控制传输, 当上层节点不能及时传输数据到下层节点时, 下层环中收到数据的节点会及时传输数据给自己的邻居环中节点。当上层节点失效时, 下层环中节点会根据优先级快速选择一个上层节点出来, 快速重构组播树, 因此即使在大量节点失效或退出时, 也不会造成过大的数据丢失, 从而保证了整个组播树的稳定性。

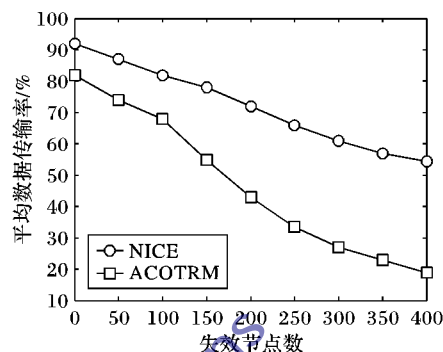


图4 节点的失效和离开对平均数据传输影响的对比

4 结语

本文利用一种快速蚁群优化算法ACO-F, 综合考虑节点性能和端到端延迟, 快速且有效地实现树环结构的构建。另外, 针对NICE协议中当Leader节点失效后没有对应的处理方法, 而导致网络收敛慢的问题, ACOTRM根据节点的性能, 为每个节点设定特有的优先级, 然后在分簇中选举出具有最高优先级的节点作为本簇的Leader, 结合合理的分层数, 使得整个分层结构达到最优化状态。仿真结果表明, 该模型得到了较低的平均时延以及较高的平均数据传输率, 在增加系统稳定性的同时也提高了其转发效率, 对以后应用层组播树的高效建立与维护工作提供了理论参考。

参考文献:

- [1] CAO J J, SU J S, XIE J, *et al.* DBMISD-D: a distributed algorithm for constructing and maintaining high stability application-layer multicast tree [J]. *International Journal of Computer Systems Science & Engineering*, 2011, 26(5): 339 - 352.
- [2] 朱艺华, 周标, 李燕君. 支持 WiMAX 节点灵活休眠的两阶段可靠多播策略[J]. *电子学报*, 2012, 40(8): 1552 - 1557.
- [3] RAHIMI M R, BAIS A, SARSHAR N. On fair and optimal multi-source IP-multicast [J]. *Computer Networks*, 2012, 56(4): 1503 - 1524.
- [4] XU J, ZHOU Y, QUAN Z. Bidimensional model based on region saturation for application layer multicast [C]// *ICCS 2008: Proceedings of the 11th IEEE Singapore International Conference on Communication Systems*. Piscataway: IEEE, 2008: 304 - 308.
- [5] 许建真, 梁克会, 张福炎. 基于区分服务的树环结构应用层多播模[J]. *通信学报*, 2009, 30(10): 128 - 134.
- [6] 邝祝芳, 陈志刚, 刘慧. 一种认知无线 Mesh 网络中负载均衡的组播路由算法[J]. *计算机学报*, 2013, 36(3): 521 - 531.
- [7] 刘永广, 张剑, 姚若河. 基于最大干扰网络编码的应用层多播算法[J]. *计算机应用*, 2011, 31(7): 1959 - 1961.
- [8] 许建真, 张倩, 商锦波. 基于总线环混合结构的可靠组播协议[J]. *计算机应用*, 2011, 31(1): 219 - 222.
- [9] 许建真, 许强. 基于中心环旋转木马的应用层组播模型[J]. *计算机应用*, 2009, 29(2): 484 - 486.

4.3 发送模块时序图

如图 12 所示,协议解析模块输入连续 4 帧的 8 位解析数据 rx_data,在输入控制时钟 rx_int 的控制下,依次将 4 帧数据输入缓存 tx_data1 ~ tx_data4,信号 neg_rx_int 控制时钟 rx_int

的最后一个下降沿,标志接收数据的完成与数据发送的开始。在信号 clk_bps 控制下,4 帧 8 位并行数据分别按串口协议加上头尾校验位后打包,依次转换成串行数据发送出去。图 12 中 rs232_tx_r 即为打包后发送的串口命令。

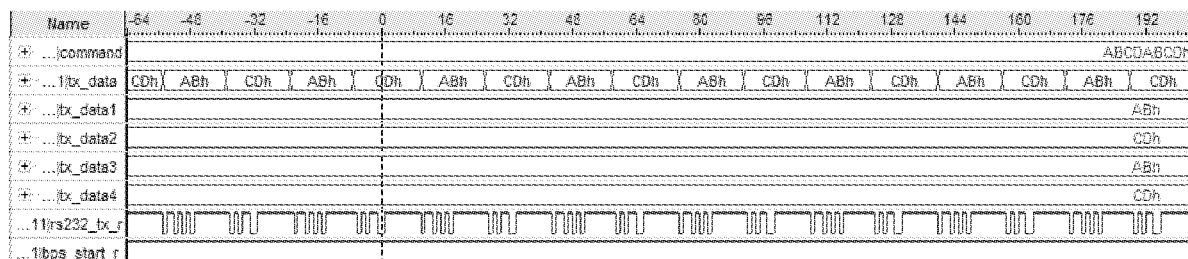


图 12 发送模块时序图

5 结语

本文在构建小卫星模拟智能观测系统基础上提出的多路串行通信控制系统,在功能上解决了星载计算机对多路相机载荷管理与控制的问题;提出了多路串行通信协议,便于提高通信两端信息交互效率。系统在设计上具有相对独立性,同时也可作为子模块集成到整个小卫星模拟系统中,为小卫星观测系统智能化提供性能验证支持。另外,系统在设计思路上采用 Host_Target 设计模式,软硬结合交叉开发,各子系统具有功能关联与研发独立性,灵活性高,为系统进一步的完善与模块功能的扩展打下基础。

参考文献:

- [1] 蔡亚梅, 宁勇, 汪立萍, 等. 美国空间快响小卫星载荷技术现状与分析[J]. 航天电子对抗, 2012, 28(6): 28-31.
- [2] TIAN S Q, YIN Z B, YAN J. Design and implementation of a low-cost fault-tolerant on-board computer for micro-satellite [C]// Proceedings of 7th International Conference on Software Testing, Verification and Validation. Washington, DC: IEEE Computer Society, 2012: 129-134.
- [3] 王海涛, 杨聪伟. 卫星综合电子系统研究与展望 [C]// 中国第二届卫星导航年会电子文集. 上海, 中国卫星导航学术年会组委会, 2011: 1-6.
- [4] 徐伟, 谭超, 王绍举, 等. 基于 TSC695F 的高可靠航天相机控制器设计[J]. 长春理工大学学报: 自然科学版, 2011, 34(3): 28-32.
- [5] LI J F, CHAI M G. Design of 1553B avionics bus interface chip based on FPGA [C]// Proceedings of International Conference on Electronics, Communications and Control. Washington, DC: IEEE Computer Society, 2011: 3642-3645.
- [6] 张建东, 吴勇, 高晓光, 等. 基于 DSPN 的综合航电总线系统的性能评[J]. 西北工业大学学报, 2005, 23(4): 244-248.
- [7] 刘萍先, 曹清华, 赵筱媛, 等. 基于 RS232/485 协议的多机通信系统的设计[J]. 南昌工程学院学报, 2008, 27(6): 16-19.
- [8] 阮颀, 黄培中, 卫炎. 多路异步串行通信系统在光纤陀螺组合中的设计与实现[J]. 电子技术应用, 2004(8): 67-69.
- [9] 喻少林, 韩波, 李平, 等. 基于 FPGA 的飞控计算机多路串行通信设计[J]. 计算机工程, 2011, 37(20): 242-245.
- [10] 刘国栋. 基于 ARM 的多路串行和以太网通信技术的研究与应用[D]. 北京: 北京交通大学, 2007.
- [11] ZHAO H M, ZHENG X F, LIU W Y. Intelligent traffic control system based on DSP and Nios II [C]// Proceedings of International Asia Conference on Informatics in Control, Automation and Robotics. Washington, DC: IEEE Computer Society, 2009: 90-94.
- [12] 文治洪, 胡文东, 李晓京, 等. 基于 PL2303 的 USB 接口设计[J]. 电子设计工程, 2010, 18(1): 32-34.
- [13] LIM J Y, KIM G, O I-S, et al. A CSIC implementation with POC-SAG decoder and microcontroller for paging applications [C]// ASP-DAC '95: Proceedings of the 1995 Asia and South Pacific Design Automation Conference. New York: ACM, 1995: 107-112.
- [14] MAHAT N F. Design of a 9-bit UART module based on Verilog HDL [C]// Proceedings of 10th IEEE International Conference on Semiconductor Electronics. Washington, DC: IEEE Computer Society, 2012: 570-573.
- [15] 廖文婷. 基于 FPGA 的嵌入式同步控制系统设计[D]. 成都: 西南交通大学, 2010.
- [16] FANG Y D, CHEN X J. Design and simulation of UART serial communication module based on VHDL [C]// Proceedings of the Third International Workshop on Intelligent Systems and Applications. Washington, DC: IEEE Computer Society, 2011: 1-4.
- [17] MACHADO L, SCHIRRU R. The ant-Q algorithm applied to the nuclear reload problem [J]. Annals of Nuclear Energy, 2002, 29(12): 1455-1470.
- [18] MENG X P, SHEN Z Y, YUE Y, et al. An improvement to the coordination method of ant colony algorithm [C]// Proceedings of the 2012 International Conference on Computer Distributed Control and Intelligent Environmental Monitoring. Piscataway: IEEE, 2012: 114-117.
- [19] TSENG S-P, TSAI C-W, CHIANG M-C, et al. A fast ant colony optimization for traveling salesman problem [C]// Proceedings of the 2010 IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 2010: 1-6.
- [10] TIAN Y, SHEN H, NG K-W. Improving reliability for application-layer multicast overlays [J]. IEEE Transactions on Parallel and Distributed Systems, 2010, 21(8): 1103-1116.
- [11] KOBAYASHI M, NAKAYAMA H, ANSARI N, et al. Robust and efficient stream delivery for application layer multicasting in heterogeneous networks [J]. IEEE Transactions on Multimedia, 2009, 11(1): 166-176.
- [12] DORIGO M, MANIEZZO V, COLORNI A. Ant system: optimization by a colony of cooperating agents [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 1996, 26(1): 29-41.

(上接第 3452 页)