基于树拓扑结构的多联邦系统

林 新 王行仁

(北京航空航天大学 自动化科学与电气工程学院, 北京 100083)

摘 要: 多联邦系统的逻辑结构对系统的开发、运行、以及可扩展性有着重要的影响. 提出了基于树形拓扑结构的多联邦系统,在此系统中联邦之间通过代理(proxy)相互连接,每个联邦只与上级和下级联邦存在直接的逻辑连接关系,避免了数据传输多通道问题和传输回路等问题. 针对实时树拓扑结构分级多联邦系统,提出了利用运行时间框架(RTI, Run-Time Infrastructure)的时间管理机制对系统的时间推进进行同步的策略,通过合理设置时间主控联邦成员和受控联邦成员,源自根联邦的同步信号能够通过树拓扑结构传递到最底层的叶联邦,从而保证了整个系统在时间推进上的同步. 最后分析了系统中同步信号的时间延迟,给出了计算时间延迟的公式. 该公式显示,系统分级越多,同步信号的时间延迟越大.

关 键 词: 高层体系结构; 运行时间框架; 多联邦; 树拓朴; 时间管理

中图分类号: TP 391.9

文献标识码: A **文 章 编 号**: 1001-5965(2006)06-0709-04

Multi-federation system based on tree-topology

Lin Xin Wang Xingren

(School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Abstract: The system logic structure has innumerous effect to the development, execution, and scalability of multi-federation system. A multi-federation system based on tree-topology structure was provided. Federations in the system were connected through proxies. Each federation was connected only with federations directly upper and lower. Thus multi-channels and loops in data transmission were avoided. A strategy was also presented to synchronize time advancement in real time multi-federation systems using RTI (run-time infrastructure) time management. Time-regulation federates and time-constraint federates in multi-federation systems were settled specially to ensure that synchronization signals created in root federation could be transmitted to leaf federations through tree-topology. The synchronization of time advancement of the whole system was realized. The time delay of synchronization signals in system was also analyzed as a result a time-delay formula was given. This formula reveals that more levels in system structure would lead to more delay.

Key words: HLA(high level architecture); RTI(run-time infrastructure); multi-federation; tree-topology; time management

当前仿真应用系统的规模越来越大,仿真任务越来越复杂,演练过程中所仿真的实体也越来越多. 以美军联合演练为例,1997 年的 STOW97 中参演实体数目达到 3 700 个左右,对象达到 8 000 多个,虚拟战场环境约 500 ~ 750 km^2;而 2002 年举行的挑战 2002(MC02)系统则包括了 42 个仿真应用系统,90 多个高层体系结构(HLA, High Level Architecture)联邦成员,实况兵力 13 500 人,所仿真的实体类型有 675 种,仿真的虚拟战场环境约16 万 km^2.

当仿真系统规模比较大,单个联邦执行完成仿真任务有困难的时候,可以将地域分布相对集中、任务相近、或者交互密切的仿真应用根据需要组成多个联邦,并在此基础上对各个联邦进行互

收稿日期: 2005-06-22

作者简介: 林 新(1972-),男,山东茌平人,讲师, lx@buaa.edu.cn.

联,构成多联邦系统协同完成仿真任务. 文献[1-2]对多联邦系统的互联进行了研究,提出了多联邦系统构造的一些原则和基本方法. 通过文献[3-4]中的分析可以看到,多联邦系统的逻辑结构对系统的开发投入、开发周期、运行性能以及系统的可扩展性,有着重要的影响,应当避免系统中出现数据传输回路和数据传输多通道,尽量避免系统中任意 2 个联邦之间存在 2 个或者 2 个以上的逻辑连接通道. 也就是说,在系统的逻辑连接拓扑图中,不应当出现逻辑连接回路,否则将降低多联邦系统的可靠性,甚至导致系统崩溃. 为此,提出了一种基于树形拓扑结构的多联邦系统,并分析了利用运行时间框架(RTI, Run-Time Infrastructure)的时间管理功能实现的系统同步策略.

1 树拓扑结构的分级多联邦系统

根据图论[5]的定义,树形拓扑中不存在回路,任意 2 个节点之间的连接都是单通路. 因此树拓扑结构的分级多联邦系统避免了数据传输多通道问题和传输回路问题. 当系统中联邦比较多时,树形拓扑多联邦系统结构清晰,数据传输路径明确.

图 1 为树拓扑结构的示意图,参与仿真演练的联邦分为 3 级:最顶层为联邦 F_1;中间层有 2 个联邦 F_{21} 和 F_{22};处于最底层的有 5 个联邦,F_{311},F_{312},F_{313} 与 F_{21} 连接,F_{321},F_{322} 与 F_{22} 连接. 其中,在功能上处于最顶层的联邦,如 F_1,定义为根联邦;处于末端的、没有下级联邦的联邦,如 F_{311},F_{312},F_{322} 等定义为叶联邦;在拓扑图中处于根联邦和叶联邦之间的位置,既有上级联邦又有下级联邦的联邦,定义为分支联邦. 图中联邦之间的连接线表示联邦之间的逻辑连接,同时也表示连接不同联邦的联邦代理(proxy). 对每个 proxy 进行了编号,如连接 F_{21} 和 F_{311} 的 P_{311},连接 F_{322} 与 F_{22} 的 P_{322}. 根据图论的相关知识[5]可知,假设树形拓扑的分级联邦系统中联邦的数目为 n,则连接联邦的 proxy 数目为$(n-1)$.

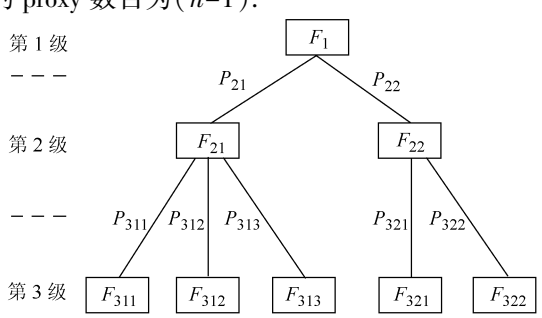


图 1 树拓扑结构的分级多联邦系统

在采用树拓扑结构的分级多联邦系统中,由于每个联邦只与上级和下级联邦存在直接的逻辑连接关系,所以只有直接上下级连接的联邦之间才存在数据交互;没有直接连接的联邦之间的数据交互需要通过其共同的上级联邦作为中转;如果 2 个需要数据交互的联邦之间没有共同的直接上级,则需要通过更上一级的联邦进行中转,在某些情况下,联邦之间的交互数据需要通过根联邦进行中转. 例如,图 1 中 F_{311} 与 F_{322} 发生数据交互时,由于 2 个联邦之间没有共同的直接上级联邦,因此 2 个联邦之间的数据交互需要先传至各自的上级联邦 F_{21},F_{22},再经过 F_{21},F_{22} 的共同上级联邦 F_1(也是根联邦)进行中转. 此过程中的数据传输通道为 F_{311}—P_{311}—F_{21}—P_{21}—F_1—P_{22}—F_{22}—P_{322}—F_{322}.

2 系统的时间同步

在 HLA 系统中,可以利用 RTI 的时间管理对单个联邦进行时间同步;同样,在分级多联邦系统中,也可以利用时间管理对整个仿真系统进行时间同步.

如图 2 所示,假定根联邦 F_1 中存在一个时间主控联邦成员(time regulating federate)clock ,用于控制联邦 F_1 中的时间推进,其它联邦成员都设置为时间受控(time constrained);联邦代理 P_{21} 在联邦 F_1 一侧设置为时间受控,而在联邦 F_{21} 一侧设置为时间主控,F_{21} 中的其它联邦成员也设置为时间受控;联邦代理 P_{311} 在联邦 F_{21} 一侧设置为时间受控,在联邦 F_{311} 一侧设置为时间主控. 系统中其它的联邦代理和联邦也以类似的方式设置,即 proxy 加入上级联邦时设置为时间受控,加入下级联邦时设置为时间主控,联邦中其它联邦成员都设置为时间受控.

另外,假设除了初始化以外,各个联邦代理在上级联邦获得 RTI 的时间推进允许之前,不在下

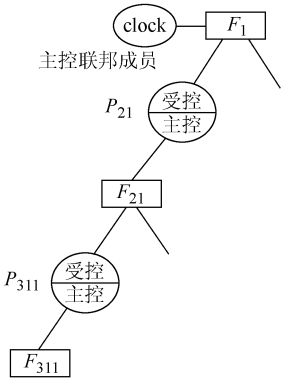


图 2 利用时间管理进行时间同步

级联邦中向 RTI 提出时间推进请求.

当系统以相同的仿真时间启动后,系统中所有联邦成员对 RTI 提出时间推进的申请都将被阻塞,只有联邦成员 clock 的请求被允许. 假设 clock 联邦成员的当前时间为 t_0,Lookahead[6] 值为 t_L,并且联邦代理 P_{21} 申请时间推进到 t_x,且 $t_x \leqslant t_0 + t_L$,则 P_{21} 的时间推进被允许;然后 P_{21} 向 F_{21} 中的 RTI 提出时间推进请求,当请求允许后,同时 F_{21} 中 RTI 将根据各受控联邦成员所申请的目的时刻、P_{21} 的当前时刻和 Lookahead 值,判断是否允许 F_{21} 中被阻塞的受控联邦成员进行时间推进. 同理,联邦代理 P_{311} 和联邦 F_{311} 也将进行类似的过程.

图 3 显示了仿真系统中时间同步的过程. 图中 RTI1 为联邦 F_1 所使用的 RTI,RTI21 为联邦 F_{21} 所使用的 RTI,RTI311 为 F_{311} 所使用的 RTI,并将 proxy 请求时间推进分为 2 种方式. 首先讨论推进方式 1,如图 3 中左侧部分所示. 在初始状态下,proxy 向上级联邦成员的 RTI 提出时间推进申请,如 P_{21} 向 RTI1 提出申请(图中标注为 a_1)、P_{311} 向 RTI21 提出申请(图中标注为 a_2). 由于联邦 F_1 中的 clock 联邦成员还没有进行时间推进,因此 P_{21} 和 P_{311} 的时间推进都将被阻塞.

当实时仿真系统启动后,系统开始同步进行时间推进. 首先联邦成员 clock 中的实时钟触发应用程序,向 RTI1 提出时间推进请求,RTI 响应并调用回调函数,允许其时间推进. 此过程所示的 3 个步骤在图中分别标注为 0,1,2.

接下来,RTI1 判断 P_{21} 满足了时间推进的条件,则 P_{21} 原来被阻塞的时间推进请求被允许(图中标注为步骤 3);然后 proxy 向下级联邦 F_{21} 的 RTI(即 RTI21)提出时间请求推进申请(步骤 4),由于 P_{21} 在联邦 F_{21} 中是时间主控联邦成员,因此 RTI21 立刻允许其时间推进(步骤 5);P_{21} 处理完此处理周期的任务,在最后向 RTI1 提出时间推进申请,并且在联邦成员 clock 时间推进之前,再次被阻塞.

当 P_{21} 时间推进后,RTI21 判断联邦代理 P_{311} 是否满足时间推进的条件,如果满足,则允许其时间推进(步骤 7);然后 P_{311} 向联邦 F_{311} 中的 RTI(即 RTI311)提出时间推进申请(步骤 8),由于 P_{311} 在联邦 F_{311} 中为时间主控联邦成员,因此其时间推进被立刻允许(步骤 9);P_{311} 接着向 RTI21 提出下一个时间推进申请,并在 P_{21} 时间推进之前被阻塞. 当 P_{311} 时间推进后,联邦 F_{311} 中如果有其他时间推进被阻塞的联邦成员满足了推进条件,RTI311 将允许其时间推进.

上述各个联邦时间同步过程中,联邦代理在上级联邦中时间推进被允许后,首先在下级联邦中请求时间推进,被允许后,才再次向上级联邦的 RTI 请求时间推进. 如果下级联邦出现错误,RTI 不能返回推进允许指令,则联邦代理也不再向上级提出时间推进申请.

在图 3 右侧推进方式 2 中,对时间推进次序进行了改变,联邦代理在上级联邦中被允许时间推进后,立刻在上级联邦中提出下一次时间推进申请,然后在下级联邦中提出时间推进申请.

3　系统中时间同步的信号延迟

系统中利用 RTI 提供的时间管理服务进行时间同步,实际上属于软件同步,必然受到网络传输

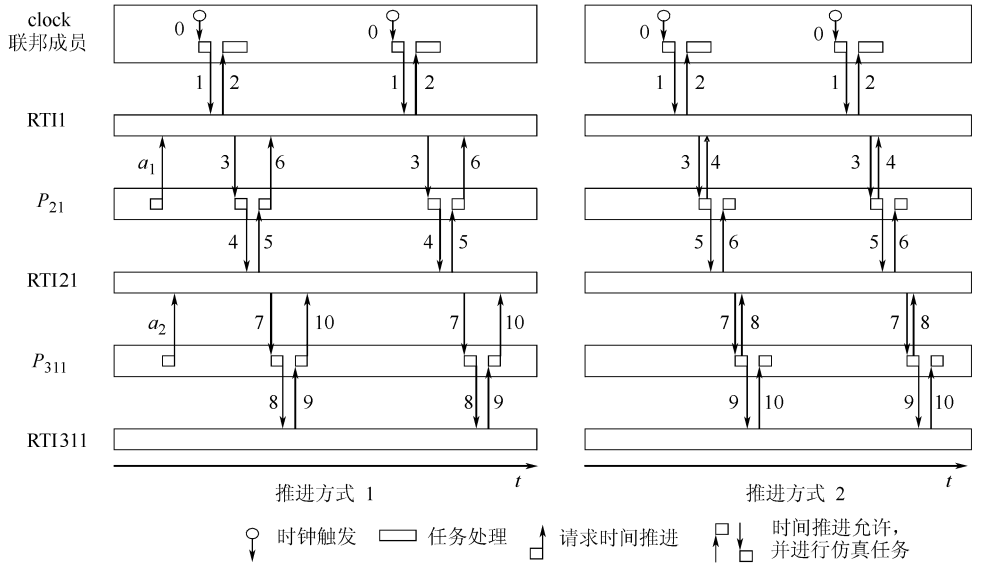


图 3　时间推进同步过程

延迟等因素的影响. 时间推进的传递过程也看作交互数据的传递过程,那么从联邦成员 clock 请求时间推进开始,到联邦 F_{311} 中某个联邦成员(除了联邦代理 P_{311} 以外)f_x 被允许时间推进为止,其间的最大时间延迟为

$$t_{\text{delay}} = t_{\text{f1}} + t_{\text{p1}} + t_{\text{f2}} + t_{\text{p2}} + t_{\text{f3}} = t_{\text{f3}} + \sum_{i=1}^{2} (t_{\text{fi}} + t_{\text{pi}}) \quad (1)$$

其中,t_{f1} 为联邦成员 clock 提出时间推进开始(步骤 1),到 P_{21} 收到 RTI1 允许时间推进(步骤 3)为止,其间时间同步信号传输的延迟时间;t_{p1} 为 P_{21} 接收到 RTI1 的时间推进允许后,到向 RTI21 提出时间推进申请为止的时间间隔,也就是图 3 中步骤 3 到步骤 4 之间的时间间隔;t_{f2} 为 P_{21} 向 RTI21 提出时间推进申请开始,到 P_{311} 收到 RTI21 允许时间推进为止之间的时间间隔,也就是步骤 4 到步骤 7 之间的时间间隔;t_{p2} 为 P_{311} 接收到 RTI21 的时间推进允许开始,到向 RTI311 提出时间推进为止的时间间隔,也就是步骤 7 到步骤 8 之间的时间间隔;t_{f3} 为 P_{311} 向 RTI311 提出时间推进申请开始,到联邦成员 f_x 接收到允许时间推进指令为止的时间间隔.

假设 P_{21} 的实时服务程序的执行周期为 T_{P21},联邦成员 clock 到 P_{21} 的网络延迟为 t_{net1},考虑到其间还有其它的软件、硬件延迟,则 $t_{\text{f1}} > T_{P21} + t_{\text{net1}}$;类似地,有 $t_{\text{f2}} > T_{P311} + t_{\text{net2}}$,$t_{\text{f3}} > T_{fx} + t_{\text{net3}}$. 其中,$T_{P311}$ 为 P_{311} 的实时服务程序的执行周期;t_{net2} 为 P_{21} 到 P_{311} 的网络传输延迟;T_{fx} 为联邦成员 f_x 的实时服务程序的执行周期;t_{net3} 为 P_{311} 到联邦成员 f_x 的网络传输延迟. 则从 clock 到 f_x 的时间延迟满足式(2).

$$t_{\text{delay}} > T_{P21} + t_{\text{net1}} + T_{P311} + t_{\text{net2}} + T_{fx} + t_{\text{net3}} + t_{\text{p1}} + t_{\text{p2}} > T_{P21} + T_{P311} + T_{fx} + t_{\text{net1}} + t_{\text{net2}} + t_{\text{net3}} \quad (2)$$

式(2)表明,从联邦成员 clock 到联邦成员 f_x 之间的最大时间延迟至少大于所经历的联邦代理、联邦成员 f_x 的实时钟周期、网络传输延迟之和.

类似地,对于存在 k 层分级的多联邦系统来说,假设其按照相同的方式进行时间同步,则从根联邦的时间主控联邦成员提出时间推进申请开始,到第 k 层联邦的联邦成员被允许时间推进,其间的时间延迟为

$$t_{\text{delay}} = t_{\text{fk}} + \sum_{i=1}^{k-1} (t_{\text{fi}} + t_{\text{pi}}) \quad (3)$$

其中,$t_{\text{fi}}(i=1,2,\cdots,k-1)$ 为系统的 clock 联邦成员(当 $i=1$)或者第 $i-1$($i>1$)个 proxy 向下一级联邦提出时间推进开始,到下一级 proxy(即第 i 个 proxy)的实时服务程序接收到数据为止,同步信号的时间延迟;$t_{\text{pi}}(i=1,2,\cdots,k-1)$ 为第 i 个 proxy 处理交互数据所需要的时间;t_{fk} 为最后一级 proxy 调用局部 RTI 组件 LRC(Local RTI Component)功能服务发送数据的时刻开始,到最终将同步信号传送到联邦 F_k 中的联邦成员 f_k 的实时服务程序时所延迟的时间.

从上面的分析可以看出,基于树拓扑结构的分级多联邦中,分级层数越多,时间同步信号的延迟就越大,对于实时系统来说,系统的实时性就越弱. 由于不同的系统对实时性的要求不同,比如雷达系统的仿真步长一般为秒级,而飞行仿真系统中仿真步长可达到 5 ms,两者相差 200 倍,因此在设计实时系统时需要根据系统类型的不同,结合系统所使用设备的性能指标,利用公式(3)求出时间同步信号的最大延迟,以检验系统是否满足实时性要求.

4 结束语

本文提出了基于树拓扑结构的多联邦系统,在此系统中每个联邦只与上级和下级联邦存在直接的逻辑连接关系,所以只有直接上下级连接的联邦之间才存在数据交互,因此系统中任意 2 个节点之间的数据传输通道只有一个,而且不存在传输回路. 在多联邦系统设计中采用树拓扑结构,使得联邦之间的数据传输简单、明确,对于缩短设计周期,提高系统可靠性具有重要意义.

文中还提出了使用 RTI 时间管理,对基于树拓扑结构的多联邦系统进行时间同步的方法,并分析了时间同步信号的最大延迟,所得到的公式可以用来估算树拓扑结构多联邦系统分级的级数. 显然,分级越多,最大时间同步信号的延迟就越大. 在实时系统中,分级的级数只能限制在满足实时性要求的范围之内.

(下转第 733 页)