中国科学 F 辑: 信息科学 2009 年 第 39 卷 第 1 期: 107～124
www.scichina.com info.scichina.com
《中国科学》杂志社
SCIENCE IN CHINA PRESS

LoI: 分布式仿真中数据的相关性评价与过滤方法

周忠 *, 吴威

虚拟现实技术与系统国家重点实验室 (北京航空航天大学), 北京航空航天大学计算机学院, 北京 100083
* E-mail: zz@vrlab.buaa.edu.cn

收稿日期: 2008-03-03; 接受日期: 2008-11-22
国家自然科学基金 (批准号: 60603084)、国家重点基础研究发展规划 (批准号: 2009CB320805) 和国家高技术研究发展计划 (批准号: 2006AA01Z331) 资助项目

摘要 大规模分布式仿真中大量对象在同一个虚拟环境中运动与交互, 会持续产生巨大数量的报文. HLA 是建模与仿真的国际标准, 它规定了基于类与基于值的发布/订购机制, 但这两种机制只确定了报文中的数据与接收方的相关与否. 由于缺乏报文相关性的评价, 接收方需要对所有的相关报文按照相同的重要性进行处理, 限制了仿真规模与效率. 针对数据的相关性缺乏评价问题, 文中提出可根据空间位置对属性和属性值接收的影响对数据的相关性分类, 从而进行评价, 提出了兴趣层次 LoI(layer of interest). 在 LoI 的基础上建立了一种自适应的发布/订购机制, 它可以直接识别并抛弃大多数的不相关报文. RTI 基于 LoI 进行报文更新和反映率的控制可以在保障重要数据的基础上进行 RTI 拥塞控制. BH RTI 中完整实现了 LoI, 简介了实现机制, 最后进行了报文过滤效率和 RTI 拥塞控制实验.

关键词
分布式仿真
HLA
相关性评价
兴趣层次
RTI

1 引言

大规模分布交互仿真中成千上万的对象在同一个虚拟环境中运动与交互, 而每个动态对象都不停地产生最新的状态数据, 为了实现节点之间的对象状态一致性, 系统中必然会持续进行巨大数量的报文交换. 这种数据的急剧增长限制了系统的规模与效率. 兴趣管理通过相关性过滤实现仿真节点只发送及接收与其相关的数据报文, 弱化了节点交互的耦合性, 是大规模分布式仿真的重要技术. 在兴趣管理中, 仿真节点通过位置或类似属性来表达自己的兴趣范围, 从而使不符合需要的数据可以被过滤掉, 只接收感兴趣的数据.

兴趣表达方法主要有公式法、单元表达法和限域法 3 类 [1]. 感兴趣区域 (AOI, area of interest)[2] 是一种典型的公式过滤法, 它定义了发送方和接收方的距离公式, 能够满足公式的对象之间将进行数据的发送/接收, 但连续大量的数学公式计算会使公式法的计算开销很大. NPSNET[3] 使用的空间划分属于单元表达法, 其开销较小, 但过滤效率低. 限域法则在公式法和单元表达法之间进行折中. 20 世纪 90 年代随着 DIS(distributed interactive simulation) 标准的兴起, 大量研究关注于各类兴趣管理方法 [4], Morse[1] 综述了其中多个典型系统并提出了一种兴趣管理的分类方法. 高层体系结构 (HLA, high level architecture) 是建模仿真领域的最新技术标准, 它规定了 RTI(run-time infrastructure) 平台提供包括兴趣管理在内的仿真运行时服务 [5].

本文重点研究能支持 HLA 的兴趣管理技术. HLA 标准规定了两种进行兴趣表达的发布–订购机制 [6,7], 由 RTI 具体实现. 一种是基于类的发布–订购机制, 在对象类/交互类以及对象类的属性层次上过滤不相关数据, 这种机制在声明管理 (DM, declaration management) 和对象管理 (OM, object management) 服务中进行定义. 另一种是基于值的发布–订购机制, 基于限域法实现过滤, 定义在数据分发管理

(DDM, data distribution management) 中. HLA 1.3 标准在 DDM 服务中定义了 3 个基本概念: 路径空间 (RS, routing space)、更新区域 (UR, update region) 和订购区域 (SR, subscription region)[7]. RS 是抽象的多维矩形空间, 兴趣管理在 RS 的维上进行数据过滤. UR 和 SR 都是限域的集合, 发送方声明 RS 中其对象所在的区域为 UR, 接收方声明 RS 中其感兴趣的区域为 SR. 如果接收方订购了发送方的对象类属性, 同时发送方对象的 UR 与接收方的 SR 相交, RTI 将把发送方的对象数据发送到接收方.

虽然 DDM 已成功地应用于美国 STOW97、JSIMS 和 MC'02 等一系列大规模军事仿真, 数据的急剧增长带来了很多麻烦. 其经验总结认为 RTI-NG 如不修改不可能支持大规模分布式仿真 [8], 并指出美军使用的 RTI-s 不应严格遵守 HLA DDM 标准 [9]. 这里主要有 3 个原因. 首先, 现有的发布–订购机制只能根据属性集匹配和区域匹配来判断一个报文是否与订购者相关, 而缺少评价相关性程度的能力, 即使发生拥塞, 所有的报文还是必须按照同等的优先级传输, 即无法通过丢弃报文来降低通信流量. 其次, HLA 只规定了可靠和尽力两种传输类型, 这种传输分类过于粗糙, 极易被开发者不恰当使用. 实际上, 在 STOW 级别的演练中仅仅可靠传输报文都有可能引起拥塞而造成网络的灾难性崩溃, 而美国的历次大规模军事仿真甚至根本没有使用可靠传输 [10]. 再次, 基于类/属性的过滤需要进行大量的属性集匹配. 设一个订购者订购了某对象类的 m 个属性, 更新报文中包含该对象类的另 n 个属性, 那么在报文最终确定与订购者无关之前需要进行时间复杂度为 $O(m*n)$ 的属性集匹配. 可见, 亟需找到一种方法从属性相关性层次上快速丢弃无关报文, 优化匹配效率.

为了解决相关性评价问题, 以提高报文过滤速度和过滤掉尽量多的无关报文, 本文提出了一种相关性评价机制: 兴趣层次 (LoI). LoI 根据空间距离对接收属性和属性值的影响定义了一种相关性分类方法, 并提供了 HLA 的扩展方法. 基于 LoI 提出了一种自适应的发布/订购机制, 用 LoI 来表示发布、订购和更新数据的相关性, 通过 LoI 的比较, 大多数的不相关报文可以直接被发送方或接收方抛弃, 这样在基于类的发布/订购过滤中可以避免大量的属性集匹配计算. RTI 也可以根据对象的 LoI 通过降低属性更新的发送或接收频率来进行拥塞控制. 实验结果表明, LoI 在数据过滤和 RTI 拥塞控制中具有较高的效率. 从 2004 年开始, LoI 技术已经在 BH RTI 中实现并得到应用. 在一次校园网 3 个地点共 50 台机器参与的仿真中, BH RTI 成功支撑了约 50000 个对象规模的分布交互仿真虚拟环境.

本文其余章节组织如下: 第 2 节介绍了相关工作; 第 3 和 4 节提出了兴趣层次 LoI 并介绍了 HLA 的 LoI 扩展方法; 第 5 和 6 节提出了基于 LoI 的自适应发布/订购机制, 并进行算法分析; 第 7 节提出了 RTI 拥塞控制模型; 第 8 节介绍了 LoI 在 BH RTI 中的实现; 第 9 节进行了 LoI 性能实验; 最后进行了总结.

2 相关工作

已有一些工作对分布式仿真中的相关性评价问题进行研究, 根据过滤类型的不同, 这些工作大致可分为基于对象类/属性和基于属性值的方法两类.

在基于对象类/属性的方法中, 对象类或其属性被关联上一些参数来描述发布者–订购者的相关性. 文献 [11, 12] 用 QoS(quality of service) 需求作为参数, RTI 或者底层网络根据这些参数处理报文. 为了提高实时性, McLean 等 [11] 实现了硬实时的 HRT RTI, 并指出更好的办法是让 RTI 根据发布者–订购者对确定 QoS 需求的范围. Zhao 和 Georganas[12] 提出了一种将属性与一些优先级相关联的 HLA 扩展机制, 优先级是基于应用层、操作系统层和网络设施层的 QoS 参数, 如平均速率、最小速率、突发持续时间、丢包率、优先级、安全等级等, 他们还提出可以用这些优先级基于网络 QoS 服务实现 RTI.

基于值的方法主要根据实体间的距离对报文进行相关性划分. SANDS 采用主动网络技术在 IP 组播分发树中的主动路由器节点上安装基于内容的过滤器 [13], 通过组播树的路由节点不再传播不需要的转发报文来提高传输效率, 并且路由节点还可以根据实体间的距离控制接收实体发布更新报文的频率 [14]. Lee 等 [15] 根据实体距离远近定义了感兴趣区域和敏感区域, 提出了一种多阈值的 DR 推算方法, DR 推算的阈值可根据实体之间的距离动态调节. 他们还提

出了根据阈值调节和外推方程选择进行 DR 推算的两个自适应算法. Zhou 等 [16] 定义了一种实体重要性, 该重要性取决于两个要素: 实体可能影响到的其他实体数量和它们之间的距离, 并提出了一种评价仿真实体重要性的作用模型. 按照距离远近把不同类型实体的感兴趣区域 AOI 分为多个层次, 基于该作用模型, 提出了控制实体报文发送频率的更新机制, 从而更有效地利用带宽.

本文工作与这两类方法都有关系, 其不同点主要体现在以下 3 个方面. 首先, 现有工作只关注于研究发布者和订购者之间的相关性, 本文则是对发布者、订购者和数据报文三者进行相关性研究. 事实上, 对同一对发布者–订购者而言, 不同的属性更新报文也可能与订购者有着不同的相关性. 例如, 有两个报文, 一个报文中根本不包含订购者已订购的属性, 另一个则含有某个订购者订购的属性, 虽然它们来自于同一个发送方, 但显然应该被区别对待. 在本文第 5 节我们将得到关于发布者、订购者、数据报文相关性的两个重要推论, 从而揭示了它们之间的关系. 其次, 兴趣层次能很好地支持 HLA 中的发布–订购机制. 由于 HLA 本身规定了发布–订购的机制, 现有的研究很难通过小的改动就扩展到 HLA 系统中, 尤其是一些多级兴趣管理技术. LoI 的 HLA 扩展方法只需要对 HLA 进行很少的添加就可以实现其扩展, 而且能保持盟员代码的兼容性. 本文把基于类和基于值的 HLA 发布–订购机制统一到自适应 LoI 发布–订购机制中, 使传输更好理解, 容易使用. 再次, 基于 LoI 的 RTI 拥塞控制很实用. 拥塞发生时, 所有的发送方和接收方节点一起进行控制, 根据每个对象的 LoI 控制到每个对象的更新/映射频率, 从而最小化网络总体流量. 关于 LoI 的最初一些思想曾发布于 2003 年春季的 Simulation Interoperability Workshop, 本文工作则是之后几年的深入, 经过了理论推算与证明、设计与 BH RTI 实现及其反馈, 大约通过 4 年多的时间完善而成.

3 LoI(兴趣层次)

HLA 标准没有考虑相关性的差异, 现有的发布–订购机制只涉及到发布者和订购者的相关性. 但是, 从相关性评价有关的研究可以看出空间距离对接收的对象属性和属性值都有影响. 首先, 发送者向不同距离的接收方发送的数据具有不同内容和频率. 例如, 你看到一个离你很远的人, 但是只有你们足够近时才可以看到他举起的手, 这样, 手的数据就没有必要被发送. 一般地, 距离发送方较远的接收方会收到频率较低、内容也少一些的报文, 即少数属性值以低频发送到这些接收方. 其次, 接收方通常对发送方的对象只有几类兴趣. 对于具有 n 个属性的对象, 从组合来看, 可能的订购类型有 2^n 种. 但实际上订购类型一般只是几种, 这是因为一些属性的订购往往具有关系, 例如订购者订购了属性 X 坐标, 他几乎一定会订购属性 Y 坐标.

基于以上观点, 我们提出了兴趣层次的概念对相关性进行分类. LoI 是基于接收方对对象的兴趣在抽象维上具有的一些明显的层次特性. 根据接收方对对象的兴趣程度, 我们将相关性分为 6 个层次: NO_LAYER, LAYER_CRITICAL, LAYER_VISION, LAYER_ABOUT, LAYER_COMPONENT 和 LAYER_INSIDE. LoI 代表了对象类属性和属性值的相关性.

NO_LAYER: 对象 A 的 UR 和对象 B 的 SR 没有重合部分时 (图 1(a)), B 不会收到 A 发出的数据, 则 LoI 定义这种相关性为 NO_LAYER. B 对 A 不感兴趣, 所以不会收到 A 发送的任何报文.

LAYER_CRITICAL: HLA 标准为重要数据定义了可靠传输类型, 只要订购者对该对象感兴趣, 这些数据在传输过程中就不能被丢失, 这种相关性称为 LAYER_CRITICAL. 只会出现一次的重要数据或者很长时间才会出现的心跳报文也可以属于这一层次.

LAYER_VISION 定义的是最基本的兴趣程度, 描述了接收 RS 中对象消息的兴趣, 类似于在视野范围内的兴趣. 与传统 AOI 或 DDM 中的 SR 具有相同的订购区域. 当 A 处在 SR 中时 (图 1(b)), LoI 即达到了 LAYER_VISION.

LAYER_ABOUT 作为缺省的 LoI, 是兴趣主体对目标产生兴趣的常规层次, 通常表示兴趣主体能比较清楚地感知到目标. 图 1(c) 的 D_SR 区域为 LAYER_ABOUT 层次的订购区域. 我们使用一个辅助的 EXTEND 变量来定义 LAYER_VISION. 以订购者位于 SR 的中心为前提, 定义 EXTEND

为 LAYER_VISION 和 LAYER_ABOUT 所对应的 range 的比值. EXTEND 缺省值为 1, 1 是其值域下界. 当 LAYER_VISION 在某维上的区间为 $[b_{\text{lower}}, b_{\text{upper}})$ 时, LAYER_ABOUT 所对应的区间为

$$\begin{aligned}&[a_{\text{lower}}, a_{\text{upper}})\\ =\ &[(b_{\text{upper}}+b_{\text{lower}})/2-(b_{\text{upper}}-b_{\text{lower}})\\ &/2/\text{EXTEND},\\ &(b_{\text{upper}}+b_{\text{lower}})/2+(b_{\text{upper}}-b_{\text{lower}})\\ &/2/\text{EXTEND})\\ =\ &[((\text{EXTEND}-1)*b_{\text{upper}}+(\text{EXTEND}+1)\\ &*b_{\text{lower}})/2/\text{EXTEND},\\ &((\text{EXTEND}+1)*b_{\text{upper}}+(\text{EXTEND}-1)\\ &*b_{\text{lower}})/2/\text{EXTEND}).\end{aligned}\tag{1}$$

LAYER_COMPONENT 是兴趣主体对目标产生兴趣的细节层次, 适用于部件级聚合或者碰撞检测. 碰撞检测对虚拟环境的整体性能影响很大的主要原因之一就是匹配计算量很大, 因为盟员需要频繁地对本地对象实例和所有的远程对象实例进行碰撞的计算匹配. 为碰撞检测限定作用范围可以大大限制需要进行计算匹配的实例范围. LAYER_COMPONENT 层次的订购区域可近似用 UR 表示 (图 1(d)), 也可定义 PACE 变量来辅助区域匹配计算. 当 PACE 为 0 时, 不需要进行区域重叠计算.

$$\begin{aligned}[c_{\text{lower}}, c_{\text{upper}}) = [&(b_{\text{upper}}+b_{\text{lower}})/2-\text{PACE},\\ &(b_{\text{upper}}+b_{\text{lower}})/2+\text{PACE}).\end{aligned}\tag{2}$$

LAYER_INSIDE: 在某些情形下, 兴趣主体的中心正好位于目标的更新区域内 (图 1(e)), 这是一种特殊的状态, 在维上是零距离的匹配. 从仿真意义上来说, 此时兴趣主体落在目标的位置上, 距离的量变产生了质变. LAYER_INSIDE 就是 LoI 的质变结果, 表示兴趣主体对目标的内部细节感兴趣的特殊要求. 这种 LoI 可用于子场景的切换、超大对象等特殊聚合实体的仿真.

本节根据距离与兴趣的关系对 LoI 进行了分类, 其中不低于 LAYER_VISION 的 LoI 代表了尽力传输模式的不同兴趣程度. 高的 LoI 应包括了那些处于较低 LoI 的数据要求, 同样, 除了 NO_LAYER 和 LAYER_CRITICAL, 低 LoI 的订购区域覆盖了高 LoI 的订购区域. 和目前的大多数虚拟环境所使用的兴趣技术一样, 在 LoI 中, 地理坐标是一种典型的维的集合, 可以较好地体现各种属性的层次特征.

4 HLA 的兴趣层次扩展

因为 LoI 的基本概念与 HLA 是一致的, 它可以很容易实现 HLA 的扩展. HLA 扩展的原则是与基于 HLA 标准的系统相兼容, 并尽量减少改动. HLA1.3 和 IEEE 1516 标准在区域相关的概念上有一定差异,

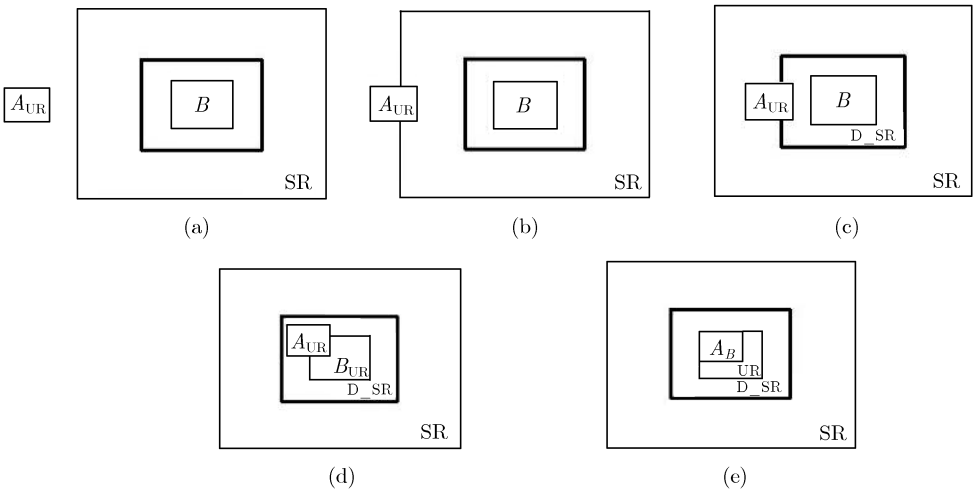

图 1 LoI 定义

(a) NO_LAYER; (b) LAYER_VISION; (c) LAYER_ABOUT; (d) LAYER_COMPONENT; (e) LAYER_INSIDE

在 IEEE 1516 标准中, 一个对象属性可以和一个维集合相关联, 而在 HLA 1.3 中对象属性只能和确定的路径空间相关联. 因此, 在 IEEE 1516 中, 可以很方便地将一个属性和一个维及其 LoI 直接关联. 而在 HLA 1.3 中, 为进行统一, 当几个属性在同一个维上需要不同的 EXTEND 和 PACE 变量值时, 只能进行一些折中. 本节的扩展方法是面向 HLA 1.3 进行的.

LoI 的 HLA 扩展只涉及到 HLA OMT(object model template), 而不需要对 HLA 框架、规则和接口规范进行任何修改. 扩展的方法是在路径空间表中增加 EXTEND 变量和 PACE 变量这两列, 在属性表中增加兴趣层次这一列, 这些列是扩展 OMT 的可选项, 如果一个属性没有与任何 LoI 相关联, 则 LAYER_ABOUT 将作为缺省的 LoI 与该属性相关联, 因此扩展后的 HLA 可以和标准的 HLA 系统保持向后兼容. 扩展的路径空间表和属性表分别如表 1 和 2 所示. 在一个联盟对象模型 FOM(federation object model) 中, 同一时刻只允许有一个路径空间和 LoI 相关联, 作为该 FOM 的主路径空间.

相应地, FED(federation execution data) 文件也需要扩展, 扩展的 FED 文件在初始化时将对象类属性与 LoI 相关联, 提供给 OM 和 DDM 服务使用. 将 HLA FED DIF(data interchange format) BNF(backus-naur form, 巴科斯范式) 中增加如下语法规则进行 LoI 扩展:

1 ⟨⟨LoI⟩⟩::= ⟨NameString⟩;

2 ⟨⟨ExtendValue⟩⟩::= ⟨Float⟩;

3 ⟨⟨PaceValue⟩⟩::= ⟨Float⟩;

4 ⟨⟨Extend⟩⟩::= “(Extend” ⟨⟨ExtendValue⟩⟩“)”;

5 ⟨Pace⟩::= “(Pace” ⟨⟨PaceValue⟩⟩ “)”,

并将以下两条语法规则进行扩展:

1 ⟨Dimension⟩ ::= “(dimension” ⟨⟨Dimension Name⟩⟩ “)”;

2 ⟨Attribute⟩ ::= “(attribute” ⟨⟨Attribute Name⟩⟩ ⟨⟨Transport⟩⟩ ⟨⟨Order⟩⟩ [⟨⟨Space Name⟩⟩] “)”,

扩展后的 BNF 范式如下:

1 ⟨Dimension⟩::=“(dimension”⟨⟨Dimension Name⟩⟩[⟨⟨Extend⟩⟩][⟨⟨Pace⟩⟩]“)”;

2 ⟨Attribute⟩ ::= “(attribute” ⟨⟨Attribute Name⟩⟩ ⟨⟨Transport⟩⟩ ⟨⟨Order⟩⟩ [⟨⟨Space Name⟩⟩] [⟨⟨LoI⟩⟩]“)”,

FED DIF 术语表需要在扩展 HLA FED DIF BNF 定义中增加如下 3 项:

LoI: 对象类属性的 LoI;

ExtendValue: 路径空间中维所关联的 EXTEND 值;

PaceValue: 路径空间中维所关联的 PACE 值.

表 1 LoI 扩展后的路径空间表

Routing Space	Dimension	Dimension type	···	EXTEND	PACE
PositionSpace	pos_x	Float	···	4	50.0
	pos_y	Float	···	4	50.0

表 2 LoI 扩展后的属性表

Object	Attribute	Data type	···	Routing Space	LoI
DveTank	DESTROYED_LEVEL	enum	···	PositionSpace	LAYER_CRITICAL
	Position	vector type	···	PositionSpace	LAYER_VISION
	Color	enum	···	PositionSpace	LAYER_VISION
	Direction	vector type	···	PositionSpace	LAYER_ABOUT
	Velocity	vector type	···	PositionSpace	LAYER_ABOUT
	Acceleration	vector type	···	PositionSpace	LAYER_ABOUT
	Pedrail	vector type	···	PositionSpace	LEYER_COMPONENT
	···	···	···	···	N/A

5 基于 LoI 的自适应发布–订购机制

早期研究用 “对象” 或者 “实体” 代表虚拟对象, HLA 引入了 “对象类” 和 “对象实例” 的概念, 对象类是指一种类型的对象, 对象实例则是指一个具体的对象. 以下部分采用对象类和对象实例两个概念.

为提高数据过滤效率, 必须在发送消息之前确定发布者、订购者和发送数据之间的相关性. 一个实用的自适应发布–订购机制需要能在动态变化的发布者–订购者环境中评价消息和发布者、订购者的相关性. 这使系统能够对每条通信链路的数据传输进行管理, 从而优化网络利用率. 目前的发布–订购机制主要在对象类层次起作用, 而本方法还实现了对象实例层次的发布–订购. 我们对发布/订购的概念进行一下扩展: 如果订购者需要某远程对象实例的数据, 那么我们称该订购者订购了该对象实例. 发布者也同理.

首先, 定义 1 给出 LoI 的数据类型.

定义 1 (LoI 数据类型) LoI 是一个枚举类型. enum LoI {NO_LAYER=0, LAYER_CRITICAL, LAYER_VISION, LAYER_ABOUT, LAYER_COMPONENT, LAYER_INSIDE}.

本机制由 LoI 变量计算公式、理论推导和过滤算法组成. 我们首先定义了 5 个 LoI 变量, 它们分别给出了发布者–订购者–数据报文 3 者之间的不同相关性, 然后通过证明得到了两个重要的推论. 这两个推论揭示了发布–订购过程中 LoI 变量之间的关系. 基于这两个推论我们得到了发送和接收报文的新型算法. 这 5 个 LoI 变量包括:

1) 发布者对对象类的 LoI;

2) 发布者对对象实例的 LoI;

3) 属性更新的 LoI;

4) 订购者对对象类的 LoI;

5) 订购者对对象实例的 LoI.

5.1 基本符号定义

为了更好地描述发布–订购服务中对象类、对象实例、属性集和区域之间的关系, 我们定义了一些基本符号, 包括对象类句柄、属性句柄和对象实例句柄. 对象类和对象实例用 32 位句柄唯一表示, 属性由包含它的对象类句柄和 32 位属性句柄唯一表示. 属性集是指对象类的发布–订购过程中一系列属性句柄的集合.

设 i 为对象类的句柄, o 为对象实例句柄, m, j, k, l 为属性集的大小即集合的模. m, j, k, l 是 FOM 中定义的类属性的子集的大小, 分别代表对象类的发布属性集、属性更新属性集、对象类的订购属性集和对象实例的订购属性集的大小. 例如, 模为 m 的属性集表示集合中有 m 个属性句柄元素.

令函数 loi(int i, int hd) 表示句柄为 i 的对象类的句柄为 hd 的属性关联的 LoI; 令 $P_m^{(i)}$ 为发布者对对象类 i 的大小为 m 的属性集的 LoI; 令 $p_m^{(i,o)}$ 为发布者对对象类 i 的对象实例 o 的大小为 m 的属性集的 LoI; 令 $\eta_j^{(i,o)}$ 为发布者的对象类 i 的对象实例 o 的大小为 j 的属性集的属性更新的 LoI; 令 $S_k^{(i)}$ 为订购者对对象类 i 的大小为 k 的属性集的 LoI. 对特定发布者而言, 设该类的各订购者的 $S_k^{(i)}$ 的最大值为 $S_{k,\max}^{(i)}$; 令 $s_l^{(i,o)}$ 为订购者对对象类 i 的对象实例 o 的大小为 l 的属性集的 LoI, 设该实例的各订购者的 $s_l^{(i,o)}$ 最大值为 $s_{l,\max}^{(i,o)}$.

DDM 服务还需要区域相关的定义. 从算法完整性考虑, 一些前面已给出的符号此处再次列出.

令更新区域为 UR, 订购者的更新区域为 UR′; 令订购区域为 SR; 令 LAYER_ABOUT 对应的区域为 D_SR, LAYER_COMPONENT 对应的区域为 P_SR, UR′ 也表示 LAYER_INSIDE 的订购区域.

当 SR 在某维上的区间为 $[b_{\text{lower}}, b_{\text{upper}})$ 时, D_SR 对应的区间为 $[a_{\text{lower}}, a_{\text{upper}})$, P_SR 对应的区间为 $[c_{\text{lower}}, c_{\text{upper}})$. 相关计算参见 (1) 和 (2) 式.

图 2 描述了发布–订购服务中的属性集. 属性集 $\Phi_{i,m}$ 和 $\Phi_{i,k}$ 用于对象类层次的发布–订购服务. 使用 DDM 时, 订购者只会请求对象实例的 $\Phi_{i,k}$ 的子集 $\Phi_{i,l}$. 本机制中注册对象实例时不需要提供属性集.

5.2 发布者对对象类的兴趣层次

当发布者订购了对象类 i 的属性集 $\Phi_{i,m}$, 兴趣层次 $P_m^{(i)}$ 表示发布者发布该对象类的细节程度.

定义 2 (发布者对对象类的兴趣层次) 如果发布者发布了对象类 i 的属性集 $\Phi_{i,m}$, 那么 $P_m^{(i)} = \text{loi}(i,h), \text{where } h \in \Phi_{i,m} \text{ and } \forall x \in \Phi_{i,m},\ \text{loi}(i,x) \leqslant \text{loi}(i,h)$.

根据定义 2 有如下性质:

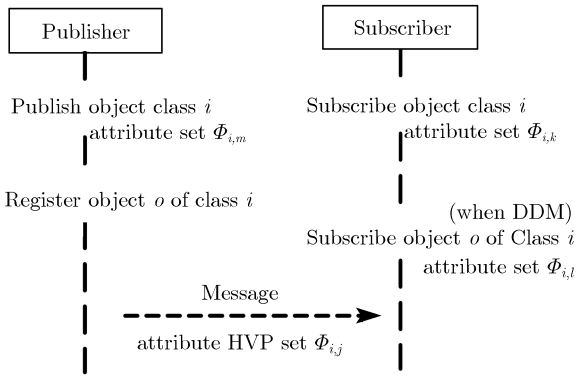

图 2 发布–订购服务中的属性集

仅当 $\forall x \in \Phi_{i,m}, \text{layer}(i,x)=$ LAYER_CRITICAL 时, $P_m^{(i)}=$LAYER_CRITICAL. $\Phi_{i,m} = \Phi$ 表示该发布者没有发布对象类 i, 因此在这种情形下 $P_m^{(i)}=$ NO_LAYER.

5.3 发布者对本地对象实例的兴趣层次

发布者注册了属性集为 $\Phi_{i,m}$ 的本地对象实例 o, 并不表示会向所有订购者发送每次属性更新报文. 由于值没有改变的属性不需要发送, 对象的属性并不是每次都需要全部更新. 因此不同报文常常携带不同属性的更新数据, 发送方应该把报文发送给真正对其数据感兴趣的订购者, 而对于没有订购者感兴趣的数据最好在发送方就直接过滤掉. 由于 DDM 服务进一步约束了接收方的范围, 本地对象实例的发布在有 DDM 和没有 DDM 时会有所区别. 兴趣层次 $p_m^{(i,o)}$ 表示发布者对本地对象实例 o 所需要实际发布的细节程度, 它和发布者对对象类的兴趣层次 $P_m^{(i)}$ 以及实际订购者的兴趣层次 $S_k^{(i)}$ 或 $s_l^{(i,o)}$ 有关.

定义 3 (发布者对本地对象实例的兴趣层次) 如果发布者发布了对象类 i 的对象实例 o, 那么

$$p_m^{(i,o)} = \begin{cases} \min(P_m^{(i)}, \mathrm{S}_{\max}^{(i)}), & \text{不存在UR}, \\ \min(P_m^{(i)}, s_{\max}^{(i,o)}), & \text{存在UR}. \end{cases}$$

定义 3 中, 在发布对象实例 o 时是否使用了 DDM 服务是由该对象实例是否与 UR 区域关联判定.

5.4 属性更新/映射报文的兴趣层次

在发布–订购过程中, 对象实例的数据在发送方被称为属性更新 (或者属性值更新), 在接收方为属性映射 (或者属性值映射). 属性更新由 HVP(handle-value pair, 句柄–值对) 构成, HVP 中包括属性句柄和属性值. 当本地对象实例中的属性值改变时, 发布者即向订购者发送属性更新报文. 令 $\Phi_{i,j}$ 为对象类 i 的实例 o 的属性更新大小为 j 的属性集, 由于只有已发布的属性才能进行更新, 有

$$\Phi_{i,j} \subseteq \Phi_{i,m}. \tag{3}$$

兴趣层次 $\eta_j^{(i,o)}$ 表示属性更新的基础或重要程度的相关性.

定义 4 (属性更新/映射的兴趣层次) 对象类 i 的实例 o 的属性更新的兴趣层次 $\eta_j^{(i,o)}$ 为

$$\eta_j^{(i,o)} = \text{loi}(i,h), \text{ where } h \in \Phi_{i,j},$$
$$\text{and } \forall x \in \Phi_{i,j}, \text{loi}(i,x) \geqslant \text{loi}(i,h).$$

根据定义 4 有如下性质:

当 $\exists x \in \Phi_{i,j}, \text{loi}(i,x)=$ LAYER_CRITICAL 时, $\eta_j^{(i,o)}=$ LAYER_CRITICAL.

发布者计算出报文的 $\eta_j^{(i,o)}$ 并把它标记到报文上, 订购者接收到该报文时, 就可以从中提取出 LoI.

5.5 订购者对对象类的兴趣层次

当订购者订购了对象类 i 的大小为 k 的属性集 $\Phi_{i,k}$, 用兴趣层次 $S_k^{(i)}$ 表示订购者所订购对象类的细节程度. 对于对象类 i 的某个发布者而言, $\cup\Phi_{i,k}$ 为所有订购对象类 i 的属性集的集合, $\cup\Phi_{i,k}$ 不包括发布者自己订购的属性集, 因为发布者不会向自己发送数据. 由于只有已订购属性的更新才能接收, 故有

无 DDM 时接收 $\Phi_{i,j}$ 的属性更新

$$\Leftrightarrow \Phi_{i,j} \cap \Phi_{i,k} \neq \Phi; \tag{4}$$

$$\Phi_{i,j} \subseteq \cup\Phi_{i,k}. \tag{5}$$

定义 5 (订购者对对象类的兴趣层次) 订购者订购对象类 i 的属性集 $\Phi_{i,k}$, 则兴趣层次 $S_k^{(i)}$ 为

$$S_k^{(i)} = \text{loi}(i,h), \text{ where } h \in \Phi_{i,k},$$
$$\text{and } \forall x \in \Phi_{i,k}, \text{loi } (i,x) \leqslant \text{loi}(i,h).$$

根据定义 5 有如下性质:

当 $\forall x \in \Phi_{i,k}, \text{loi}(i,x)=$ LAYER_CRITICAL 时, $S_k^{(i)}=$ LAYER_CRITICAL; $\Phi_{i,k} = \Phi$ 表示订购者没有订购该对象类, 因此这种情形下 $S_k^{(i)}=$NO_LAYER.

对发布者而言, 表征其发布的对象类 i 的实际订购情形的 $S_{\max}^{(i)}$ 的计算方法为

$$S_{\max}^{(i)} = \max\{\text{除发布者自身外的各订购者} S_k^{(i)}\}. \tag{6}$$

设 $S_k^{(i)}(\cup\Phi_{i,k})$ 是以 $\cup\Phi_{i,k}$ 为属性集的 $S_k^{(i)}$.

定理 1 $S^{(i)}_{\max} = S^{(i)}_k(\cup\Phi_{i,k})$.

证明 根据定义 5, $\forall x \in \Phi_{i,k}, \mathrm{loi}(i,x) \leqslant \mathrm{loi}(i,h)$, $S^{(i)}_k = \mathrm{loi}(i,h), h \in \Phi_{i,k}$.

因为 $\cup\Phi_{i,k}$ 是除发布者自身外的 $\Phi_{i,k}$ 的合集, $h \in \cup\Phi_{i,k}$, 所以 $\mathrm{loi}(i,h) \leqslant S^{(i)}_k(\cup\Phi_{i,k})$, 并且 $\exists h' \in \cup\Phi_{i,k}, \mathrm{loi}(i,h') = S^{(i)}_k(\cup\Phi_{i,k})$. 那么 $S^{(i)}_k \leqslant S^{(i)}_k(\cup\Phi_{i,k})$ 并且 $\exists S^{(i)}_{k'}, S^{(i)}_{k'} = S^{(i)}_k(\cup\Phi_{i,k})$, 根据公式 (6) 对 $S^{(i)}_{\max}$ 的定义, $S^{(i)}_{\max} = S^{(i)}_k(\cup\Phi_{i,k})$.

5.6 订购者对远程对象实例的兴趣层次

订购者发现远程对象实例 o 时, 即订购了与对象类订购中相同的属性集 $\Phi_{i,k}$, 这类似于 5.3 小节中的本地对象实例的兴趣层次. 与之相同, 发布者对对象类 i 的属性集 $\Phi_{i,m}$ 不适用于具体某对象实例的订购.

令 $\Phi_{i,l}$ 为订购者订购对象类的实例 o 的大小为 l 的属性集, 那么 $\Phi_{i,l}$ 应该为 $\Phi_{i,k}$ 与 $\Phi_{i,m}$ 的子集. 由于只有该对象类已被发布的属性才会被接收, 有

$$\Phi_{i,l} \subseteq \Phi_{i,m}. \tag{7}$$

只有订购者已订购的属性才能被接收, 因此有

$$\Phi_{i,l} \subseteq \Phi_{i,k}. \tag{8}$$

订购者接收属性集为 $\Phi_{i,j}$ 的属性更新的条件是 $\Phi_{i,j}$ 含有该订购者订购的一个或多个属性, 所以有

$$\text{接收}\Phi_{i,j}\text{的属性更新} \Leftrightarrow \Phi_{i,j} \cap \Phi_{i,l} \neq \phi. \tag{9}$$

设 $\cup\Phi_{i,l}$ 为 $\Phi_{i,l}$ 的并集. 发布者自身没有订购的 $\Phi_{i,l}$, 因此此处不需要排除发布者的 $\Phi_{i,l}$. 至少存在一个订购者时, 属性更新数据才需要被发出, 有

$$\Phi_{i,j} \subseteq \cup\Phi_{i,l}. \tag{10}$$

定义 6 (订购者对远程对象实例的兴趣层次) 当订购者订购了远程对象实例 o 的属性集 $\Phi_{i,l}$ 时, 订购者对远程对象实例 o 的兴趣层次 $s^{(i,o)}_l$ 为

$$s^{(i,o)}_l = \mathrm{loi}(i,h), \text{ where } h \in \Phi_{i,l} \text{ and } \forall x \in \Phi_{i,l}, \mathrm{loi}(i,x) \leqslant \mathrm{loi}(i,h).$$

定理 2 $s^{(i,o)}_l \leqslant S^{(i)}_k$.

证明 根据定义 5, $S^{(i)}_k = \mathrm{loi}(i,h), h \in \Phi_{i,k}$, 并且 $\forall x \in \Phi_{i,k}$, $\mathrm{loi}(i,x) \leqslant \mathrm{loi}(i,h)$. 由公式 (8) $\Phi_{i,l} \subseteq \Phi_{i,k}$, 得 $\forall x \in \Phi_{i,l}$, $\mathrm{loi}(i,x) \leqslant S^{(i)}_k$. 根据定义 6, $s^{(i,o)}_l = \mathrm{loi}(i,h')$, $h' \in \Phi_{i,l}$, 因此我们得出 $s^{(i,o)}_l = \mathrm{loi}(i,h') \leqslant S^{(i)}_k$, 即 $s^{(i,o)}_l \leqslant S^{(i)}_k$.

对该对象实例而言, 表征其实际订购情形的 $s^{(i,o)}_{\max}$ 的定义为

$$s^{(i,o)}_{l,\max} = \max\{\text{各订购者的 } s^{(i,o)}_l\}. \tag{11}$$

设 $s^{(i,o)}_l(\cup\Phi_{i,l})$ 是以 $\cup\Phi_{i,l}$ 为属性集的 $s^{(i,o)}_l$.

定理 3 $s^{(i,o)}_{\max} = s^{(i,o)}_l(\cup\Phi_{i,l})$.

证明 同定理 1 的证明, 略.

根据定义 6 可以得出当对象实例 o 不与任何区域绑定时, $s^{(i,o)}_l = S^{(i)}_k$.

由于 DDM 中定义的是区域匹配的关系, 实际计算过程中需要根据区域相交来得到属性集 $\Phi_{i,l}$. 定义 6 不能直接用来计算 $s^{(i,o)}_l$, 需要其他表述. 以 $\cap$ 表示区域的重叠关系操作符, $\subseteq$ 表示区域的包含关系, ϕ 表示区域不具有重叠关系. 根据 LoI 定义, 给出区域相关的符号定义: 设 $R1$=UR$\cap$SR$\rightarrow \neg\phi$; 设 $R2$=UR$\cap$D_SR$\rightarrow \neg\phi$; 设 $R3$=UR$\cap$P_SR$\rightarrow \neg\phi$; 设 $R4$=UR$\subseteq$UR$'$.

定义 7 (订购者对远程对象实例的兴趣层次, 重述) 订购者带区域订购了对象实例 o, 那么

$$s^{(i,o)}_l = \begin{cases} S^{(i)}_k, & \text{不存在UR or 不存在SR,} \\ \text{NO\_LAYER}, & \neg R1, \\ \text{LAYER\_CRITICAL}, & R1 \text{ and } S^{(i)}_k = \text{LAYER\_CRITICAL}, \\ \text{LAYER\_VISION}, & R1 \text{ and } \neg R2 \text{ and } S^{(i)}_k \geqslant \text{LAYER\_VISION}, \\ \text{LAYER\_ABOUT}, & R2 \text{ and } \neg R3 \text{ and } S^{(i)}_k \geqslant \text{LAYER\_ABOUT}, \\ \text{LAYER\_COMPONENT}, & R3 \text{ and } \neg R4 \text{ and } S^{(i)}_k \geqslant \text{LAYER\_COMPONENT}, \\ \text{LAYER\_INSIDE}, & R4 \text{ and } S^{(i)}_k = \text{LAYER\_INSIDE}. \end{cases}$$

从兼容性和初始的扩展应尽可能少考虑, 本文的扩展不考虑 LAYER_COMPONENT 和 LAYER_INSIDE 的区域匹配计算, 因此定义 7 简化为如下形式.

定义 8 (订购者对远程对象实例的兴趣层次, 简化定义) 简化的 $s_l^{(i,o)}$ 计算方法为

$$s_l^{(i,o)}=\begin{cases} S_k^{(i)}, & (\text{不存在UR or 不存在SR}) \text{ or } (R2 \text{ and } S_k^{(i)} \geqslant \text{LAYER\_ABOUT}),\\ \text{NO\_LAYER}, & \neg R1,\\ \text{LAYER\_CRITICAL}, & R1 \text{ and } S_k^{(i)}=\text{LAYER\_CRITICAL},\\ \text{LAYER\_VISION}, & R1 \text{ and } \neg R2 \text{ and } S_k^{(i)} \geqslant \text{LAYER\_VISION}. \end{cases}$$

5.7 基于发布–订购 LoI 的更新/映射定理

基于上述有关定义和定理, 可以得到如下定理.

定理 4 (更新准则) $p_m^{(i,o)}$ 是发布者发送属性更新的 $\eta_j^{(i,o)}$ 的上确界.

证明 当 $p_m^{(i,o)}$=NO_LAYER 时, 该对象实例不发送属性更新, $\eta_j^{(i,o)}$ 不存在.

当 $p_m^{\{(i,o)\}} \geqslant$LAYER_CRITICAL 时, $\eta_j^{(i,o)}$ 存在, 此时根据定义 4, $\exists h \in \Phi_{i,j}$, $\forall x \in \Phi_{i,j}$, $\text{loi}(i,h) \leqslant \text{loi}(i,x)$, 并且根据定义 2, $\exists h' \in \Phi_{i,m}$, $\forall x \in \Phi_{i,m}$, $\text{loi}(i,x) \leqslant \text{loi}(i,h')$. 根据 (3) 式 $\Phi_{i,j} \subseteq \Phi_{i,m}$ 并且 $\forall x \in \Phi_{i,j}, \text{loi}(i,x) \leqslant \text{loi}(i,h')$. 因此, $\text{loi}(i,h) \leqslant \text{loi}(i,h')$, i.e. $\eta_j^{(i,o)} \leqslant P_m^{(i)}$. 同理, 由公式 (5) 和定理 1, 得 $\eta_j^{(i,o)} \leqslant S_{\max}^{(i)}$, 同理, 由公式 (10) 和定理 3, 得 $\eta_j^{(i,o)} \leqslant s_{\max}^{(i,o)}$, 然后根据定义 3, 得 $\eta_j^{(i,o)} \leqslant p_m^{(i,o)}$.

当只有一个订购者存在, 并且 $m=k=l=1$ 时, 显见, $\text{loi}(i,h')=\text{loi}(i,h)=\text{loi}(i,x), \eta_j^{(i,o)}=P_m^{(i)}$. 又根据定义 5、定理 1、定义 6 和定理 3, 分别有

$$\eta_j^{(i,o)}=S_k^{(i)}=S_{\max}^{(i)},$$
$$\eta_j^{(i,o)}=s_l^{(i,o)}=s_{\max}^{(i,o)},$$

因此 $\eta_j^{(i,o)}=p_m^{(i,o)}$.

故 $p_m^{(i,o)}$ 是 $\{\eta_j^{(i,o)}\}$ 的上确界.

根据更新准则可以得到如下推论:

推论 1 (更新准则, 重述) 发布者仅能发送 $\eta_j^{(i,o)} \leqslant p_m^{(i,o)}$ 的属性更新.

定理 5 (映射准则) $s_l^{(i,o)}$ 是订购者接收属性更新的 $\{\eta_j^{(i,o)}\}$ 的上确界.

证明 当 $s_l^{(i,o)}$=NO_LAYER, 不接收该对象实例的属性更新, $\eta_j^{(i,o)}$ 不存在.

当 $s_l^{(i,o)} \geqslant$LAYER_CRITICAL 时, $\eta_j^{(i,o)}$ 存在, 此时根据定义 4, $\exists h \in \Phi_{i,j}$, $\forall x \in \Phi_{i,j}$, $\text{loi}(i,x) \geqslant \text{loi}(i,h)$. 根据公式 (9), 接收 $\Phi_{i,j}$ 的属性更新 $\Leftrightarrow \Phi_{i,j} \cap \Phi_{i,l} \neq \emptyset$. 令 $\varphi=\Phi_{i,j} \cap \Phi_{i,l}$, 则 $\varphi \neq \emptyset$. $\exists y \in \varphi$ 满足 $\forall x \in \varphi, \text{loi}(i,x) \geqslant \text{loi}(i,y)$. 因为 $\varphi \subseteq \Phi_{i,j}, \text{loi}(i,y) \geqslant \eta_j^{(i,o)}$. 根据定义 6, $\forall x \in \Phi_{i,l}, s_l^{(i,o)} \geqslant \text{loi}(i,x)$. 因为 $\varphi \subseteq \Phi_{i,l}, s_l^{(i,o)} \geqslant \text{loi}(i,y)$. 所以 $s_l^{(i,o)} \geqslant \eta_j^{(i,o)}$.

显见, 当 $j=l=1$ 时, $\text{loi}(i,y)=s_l^{(i,o)}=\eta_j^{(i,o)}$.

故 $s_l^{(i,o)}$ 是 $\{\eta_j^{(i,o)}\}$ 的上确界.

根据映射准则可以得到如下推论:

推论 2 (映射准则, 重述) 订购者仅能接收 $\eta_j^{(i,o)} \leqslant s_l^{(i,o)}$ 的属性更新.

5.8 发送/接收报文的算法

前面介绍了发布者、订购者和数据报文的 LoI, 这样 HLA 中基于类和基于值的发布订购机制都可以统一, 并且得到了发布者、订购者和数据这三者 LoI 关系的两个重要推论. 这两个推论可以帮助简化属性集匹配, 加快报文过滤速度. 在新的发布–订购机制中, 发布者带 $P_m^{(i)}$ 发布对象类 i, 带 $p_m^{(i,o)}$ 发布本地对象实例 o, 订购者带 $S_k^{(i)}$ 订购对象类 i, 带 $s_l^{(i,o)}$ 订购远程对象实例 o. 4 个 LoI 表达了发布–订购中双方的动态细节感知的相关性. 报文的 $\eta_j^{(i,o)}$ 也表示了数据报文的一种感知基础程度的相关性.

从推论 1 和 2 可以得到动态发布–订购过程中关于发送和接收数据报文的两个自适应算法, 如算法 1 和 2 所示:

算法 1 (发送数据算法) LoI_UAV

```
FOR each attribute update of local object instance o
    int l = η_j^(i,o), compute according to Definition 4.
    IF (l ⩽ p_m^(i,o))
      //attach l to the update packet;
      update.loi=l;
      multicast the update packet to the subscriber group;
END FOR
```

算法 2 (接收数据算法)　LoI_RAV

```
FOR each attribute reflect of remote object instance o
    int l=reflect.loi; //get η_j^(i,o)
    IF(l ⩽ s_l^(i,o))
      //the reflect packet is wanted by the subscriber
      accept the reflect packet;
      callback the corresponding user function;
END FOR
```

现在发送和接收算法变得非常简单. 本机制通过评价发布者、订购者和数据报文三者之间的相关性, 大大加快了相关性过滤的速度, 同时还发布–订购的实际需求. 鉴于 LoI 对相关性的分类并不精确到属性集, 算法 3 给出了精确的属性集匹配算法, 可以根据应用需要进行.

算法 3 (接收数据增强算法)　PreciseLoI_RAV

```
FOR each attribute reflect {HVP_j{⟨attr_i, value_i⟩}, loi}
of remote object instance o
    int l=reflect.loi; //get η_j^(i,o)
    IF(l ⩽ s_l^(i,o))
      //perform precise matching
      WHILE(int h = 1; h <= j; h++)
        IF attr_h ∉ subscription attribute set {c_i, ⟨attr_i⟩}
          remove ⟨attr_h, value_h⟩ from HVP_j {⟨attr_i,
          value_i⟩};
      IF sizeof HVP_j {⟨attr_i, value_i⟩}== = 0
        break;
      //the reflect packet is wanted by the subscriber
      accept the reflect's valid data HVP_j{⟨attr_i, value_i⟩};
      callback the corresponding user function;
END FOR
```

6　算法分析

分布交互仿真中报文过滤的计算开销主要在于属性集的匹配和基于值的兴趣过滤, 对应的两个算法的效率在很大程度上决定了数据分发的效率和 RTI 的性能.

属性集匹配是发布/订购的基础, 不同的发布/订购机制会包括更新报文 HVP 集和发布属性集的匹配、发布属性集和订购属性集的匹配、HVP 集和订购属性集的匹配中的部分或全部. 属性集匹配以两个属性集中存在相同的属性为匹配成功. 发布方发送报文到订购方的过程中往往需要多次属性集匹配, 通常包括发布端的匹配和订购端的匹配. 不同的实现过程还可能有一些动态属性集或属性集的分解.

设属性更新 HVP 集的大小为 m, 订购属性集的大小为 n, 则进行属性集匹配的时间复杂度为 $O(t) = O(m * n)$, 最差的情形下, 需要对每个元素进行遍历. 可以先对属性集进行句柄的插入排序, 则进行匹配的时间复杂度可以降低到 $O(n)$, 但是插入排序需要额外的计算. 采用其他的数据结构, 如 Hash 表等, 可以在一定程度上提高数据和发布属性集、订购属性集的匹配效率, 但是根据实现机制的不同, 可能涉及到发布表或订购表在网络上的传输, 实现的复杂度将大大增加.

基于 LoI 的自适应发布–订购机制将属性集的排序、匹配简化为值的比较, 即 $\eta_j^{(i,o)}$ 分别和发布方 $p_m^{(i,o)}$、订购方 $s_l^{(i,o)}$ 的比较, 它可以在不需要进行插入排序或遍历的情形下快速完成匹配, 其时间复杂度降低到 $O(t) = O(1)$ 加上 $\eta_j^{(i,o)}$ 的计算, 远远低于现有的属性集匹配方法. 虽然看起来这种方法会比精确的属性集匹配过滤效率低一些, 但是在实际应用中, 几乎不可能存在发布/订购属性集的各种排列组合的可能, 只会是有限种组合. 因此, 结合应用需求合理进行属性和 LoI 的关联, 在实际运行中可以达到相同的过滤效果. 当然, 也可以在 LoI 匹配的基础上进一步进行接收方的属性集精确匹配.

基于 LoI 的发布–订购机制的示例如图 3, 其中订购者订购了 “person” 类. 外面的标有 “SR” 的矩形为 LAYER_VISION 的范围, 里面的标有 “D_SR” 的矩形为 LAYER_ABOUT 的范围. 根据第 4 节的方法对 HLA 进行了扩展, 包括定义了表 1 和 2. 在图 3 中, D_SR 区域中的 person 发送包括位置、旋转角度、颜色和姿势等丰富属性的报文给订购方, 订购方可以很准确地看到它们. 另一方面, 订购者只能得到在 D_SR 外部区域中 person 的少量数据, 主要是位置. 这样, 大部分不相关报文可以不需要属性集匹配就直接丢弃, 不同 LoI 的对象实例也会被区别对待.

7　基于 LoI 的 RTI 拥塞控制模型

大规模分布式仿真带来的数据急剧增长, 限制

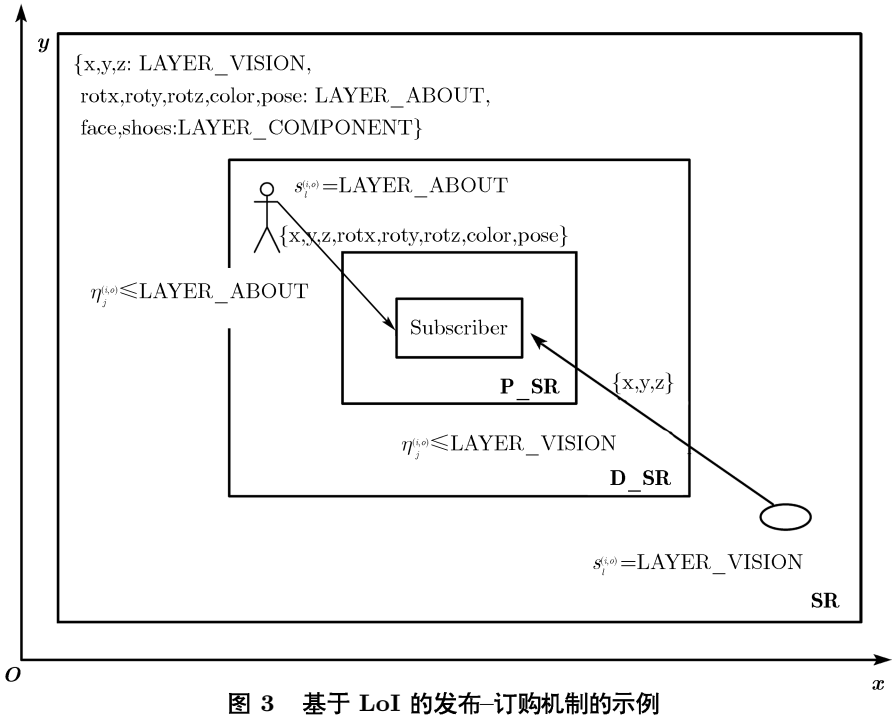

图 3 基于 LoI 的发布–订购机制的示例

了仿真的规模和性能. 由于以前缺乏有效的相关性评价方法, 国际上没有可以实用的 RTI 拥塞控制方法. LoI 的划分使 RTI 可以在有限的资源条件下评估数据报文对 HLA 应用程序的相关性程度, 进行取舍, 在尽可能不影响应用程序的仿真逼真性、正确性和一致性的基础上进行拥塞控制.

要缓解拥塞状况, 必须尽可能地丢弃不相关报文. 我们根据对象实例的 LoI 控制其更新频率, 在此基础上实现了 RTI 拥塞控制. 拥塞检测和拥塞控制算法的早期做法曾发表于文献 [17], 近期我们仍然在研究改进. 此处我们进一步研究 RTI 拥塞控制的理论模型, 如图 4 所示.

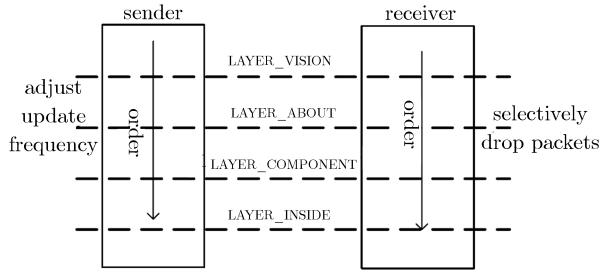

图 4 基于 LoI 的 RTI 拥塞控制模型

首先, 每个对象实例已经关联了 LoI, 拥塞发生时主机节点根据发布/订购需求减少实际发送/接收的报文数量. 在发送方, 根据本地对象实例的 LoI 调节更新报文的发送频率, LoI 较低的对象实例的更新频率要低于 LoI 较高的对象实例. 此处的更新频率是指经过数据过滤后发送方的实际发出报文频率, 因此, 尽管发送方在仿真循环中仍然进行发送, 单位时间内实际发出的报文数量会减少. 同理, RTI 选择性地丢弃不重要的报文, 只向接收方提供必需的报文, 这样, 接收方只需处理更少数量的报文. 由于保证了更新频率及重要报文的发送和接收, 系统在有多数报文被丢弃的情形下仍可以正确运行.

下面给出模型中的几个定义, 实际应用过程中, 用户可以根据具体应用需求调整这些参数. 为了得到对象实例的实时更新频率, 需要知道最后几个报文的总发送时间. 令 τ_s 为数据更新频率的统计周期, 则根据每次发送 τ_s 个报文所用的时间可计算出报文的连续更新频率. 令 τ 为实现拥塞控制算法进行数据过滤的周期.

定义 9 (更新频率) 对象实例的更新频率定义: $v = \tau_s/\Delta t = \tau_s/(t - t_0)$, 其中 t_0 为统计周期的开始

时间, t 为统计周期的结束时间.

用户应根据应用需求为每个 LoI 设定标准更新频率, 在标准更新频率下接收对象实例的映射报文应是可以接受的. 由于在一次仿真中不同盟员可能要求不同配置, 而分布式 RTI 可以为每个 RTI 维护特定的盟员 LoI 设置, 因此在这方面具有一定的优势.

定义 10 (兴趣层次的标准更新频率) 令标准更新频率函数为 $f(l)$, 其中 l 是对象实例的 LoI.

可以得出, 在发送方, $f(l) = f(p_m^{(i,o)})$, 在接收方, $f(l) = f(s_l^{(i,o)})$.

假设仿真中有 n 个对象实例, 分别为对象类 $C_1, C_2, \ldots, C_m$ 的实例. 对象类 C_i 的对象实例数量为 $n_i, i \in [1, m]$, 且 $\sum_{i=1}^{m} n_i = n$. 在一次仿真中, 有 N_0 个实例的 LoI 为 LAYER_CRITICAL, N_1 个实例的 LoI 为 LAYER_VISION, N_2 个实例的 LoI 为 LAYER_ABOUT, N_3 个实例的 LoI 为 LAYER_COMPONENT, N_4 个实例的 LoI 为 LAYER_INSIDE, 并且有 $\sum_{i=0}^{4} N_i = n$.

令 $U(I_j)$ 为保证其他主机正确运行的基础上对象实例 I_j 应达到的更新频率. 如果没有更新报文的相关性评价, 主机节点产生的总数据流量为 $\sum_{j=1}^{n} U(I_j) = n * U(I_j) = n * U_0$, U_0 为仿真所需要的更新速率, 且有 $U_0 >= f(l)$.

而基于 LoI 的发送方产生的总数据流量为

$$\begin{aligned}\sum_{j=1}^{n} f(l_j) &= \sum_{j=1}^{n} f(p_m^{(i,o)}) \\ &= N_0U_0 + \sum_{j=1}^{N_1} f(\text{LAYER\_VISION}) \\ &\quad + \sum_{j=1}^{N_2} f(\text{LAYER\_ABOUT}) \\ &\quad + \sum_{j=1}^{N_3} f(\text{LAYER\_COMPONENT}) \\ &\quad + \sum_{j=1}^{N_4} f(\text{LAYER\_INSIDE}) \\ &= N_0U_0 + N_1 f(\text{LAYER\_VISION}) \\ &\quad + N_2 f(\text{LAYER\_ABOUT}) \\ &\quad + N_3 f(\text{LAYER\_COMPONENT}) \\ &\quad + N_4 f(\text{LAYER\_INSIDE}).\end{aligned}$$

假设在接收方有 n' 个对象实例, 并且每个 LoI 的实例个数分别为 $N_0', N_1', N_2', N_3', N_4'$, $\sum_{i=0}^{4} N_i' = n'$, 则主机节点接收的总数据流量为 $\sum_{j=1}^{n'} U(I_j) = n'U(I_j) = n'U_0$.

而基于 LoI 的接收方接收的总数据流量为

$$\begin{aligned}\sum_{j=1}^{n'} f(l_j) &= \sum_{j=1}^{n'} f(s_l^{(i,o)}) \\ &= N_0'U_0 + \sum_{j=1}^{N_1'} f(\text{LAYER\_VISION}) \\ &\quad + \sum_{j=1}^{N_2'} f(\text{LAYER\_ABOUT}) \\ &\quad + \sum_{j=1}^{N_3'} f(\text{LAYER\_COMPONENT}) \\ &\quad + \sum_{j=1}^{N_4'} f(\text{LAYER\_INSIDE}) \\ &= N_0'U_0 + N_1' f(\text{LAYER\_VISION}) \\ &\quad + N_2' f(\text{LAYER\_ABOUT}) \\ &\quad + N_3' f(\text{LAYER\_COMPONENT}) \\ &\quad + N_4' f(\text{LAYER\_INSIDE}).\end{aligned}$$

可以看出, 对象实例占用的网络带宽被划分为 4 个 LoI 报文组加上关键数据占用的带宽, 含关键数据的属性更新/映射报文不能被丢弃. 这样可以通过调节更新频率使 LoI 高的对象实例占用较多的网络带宽. 最终用户可根据需要进行配置, 增强了模型的灵活性. 通过这种方法, 我们就可以调节每个对象实例的更新/映射报文的数量, 从而既控制网络总流量, 又能有所取舍.

8 LoI 在 RTI 中的实现

我们自主研制了 RTI 软件, 并将自适应发布–订购机制和 RTI 拥塞控制在其中进行了实现. 本节简要介绍其实现方法.

RTI 是 HLA 系统的软件基础设施, 从功能的角度上类似于仿真应用的分布式操作系统. 盟员基于 LRC(本地 RTI 组件) 链接库提供的符合 HLA 标准的编程接口进行开发, 通过 RTI 进行互操作. 我们从 2001 年开始研制了一种采用分布式结构的 BH RTI,

如图 5 所示. 到 BH RTI 2.2 为止, 一个联盟中可以同时有多个对等的 RTI 运行, 而没有 CRC(中心 RTI 组件), 从 BH RTI 2.3 开始, BH RTI 增加了具有丰富监测功能的 RTI 管理监测中心 CentralServer. 每个 RTI 节点均可为多个盟员提供服务, 并基于组播只维护那些本节点相关的数据, 同时单个的 RTI 节点也可以作为一个集中式 RTI 运行. BH RTI 于 2006 年 8 月开始在 http://www.hlarti.com 上发布.

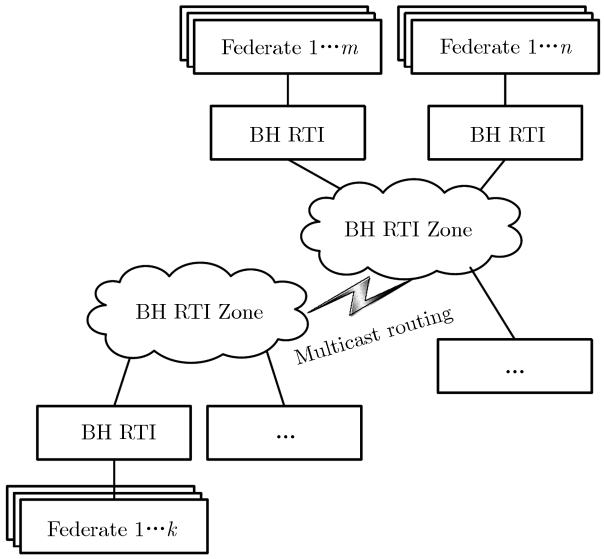

图 5 BH RTI 的分布式运行结构

除管理监测中心, BH RTI 主要包括 2 个程序模块: LRC 和 ritexec(图 6). LRC 根据 HLA 接口规范为应用程序提供了接口函数库, rtiexec 实现了 HLA 规定的联盟管理、对象管理、声明管理、所有权管理、时间管理、数据分发管理等 6 类管理服务和 MOM(management object model, 管理对象模型). 基于 LoI 的自适应发布–订购机制把对象管理、声明管理和数据分发管理统一到了发布–订购机制中, 所以, 在图 6 中的 ObjectManager 和 DataManager 两个模块上实现了对象管理、声明管理和数据分发管理服务.

8.1 基于 LoI 的自适应发布–订购过程

LoI 扩展和自适应发布–订购机制已经实现到 BH RTI 中. 图 7(a) 是发布对象类、注册对象实例和发送属性更新的过程, 图 7(b) 是订购对象类、发现对象实例以及映射属性更新的过程. 这两个图显示了使用 LoI 后具体的发布–订购过程, 包括 4 个进程: HLA 应用程序 (盟员)、LRC、BH RTI 以及其他 BH RTI 节点. 图中的 RTI 是指 BH RTI 的主服务程序 ritexec, BH RTI 通过各自的 LRC 为所有的 HLA 应用程序提供服务, 其中, 每个 LRC 负责一个盟员. 整个联盟中, RTI 之间通过组播进行通信. 简单地说, BH RTI 收集本地对象实例的 LoI 并发布给其他 BH RTI, 发现订购的远程对象实例并且将其更新发送给需要的盟员.

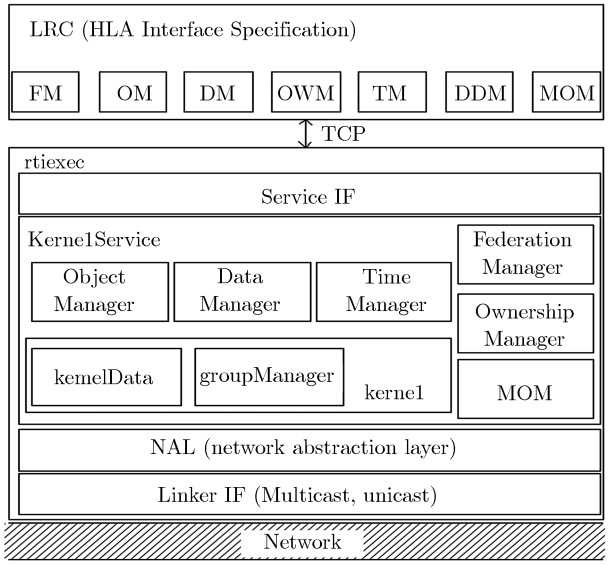

图 6 BH RTI 主要模块划分

8.2 RTI 拥塞控制

RTI 拥塞控制在 BH RTI 的 NAL(network abstraction layer, 网络抽象层) 模块实现, 用来控制携带属性更新/映射的 PDU(protocol data unit, 协议数据单元) 报文的流量, 如图 8 所示. 根据拥塞控制模型, 当拥塞发生时, NAL 提取数据报文所属对象实例的句柄并执行本地 LoI 过滤. 相应地, 收到数据报文时, NAL 提取远程对象实例句柄并且执行远程 LoI 过滤. 本地和远程过滤均采用第 7 节描述的过滤机制来控制报文更新/映射的频率. 早期的 LoI 过滤算法详见文献 [17], 但我们近年来有了新的想法 [18], 已初步实现于 BH RTI, 正在试图发展一种更完善的算法.

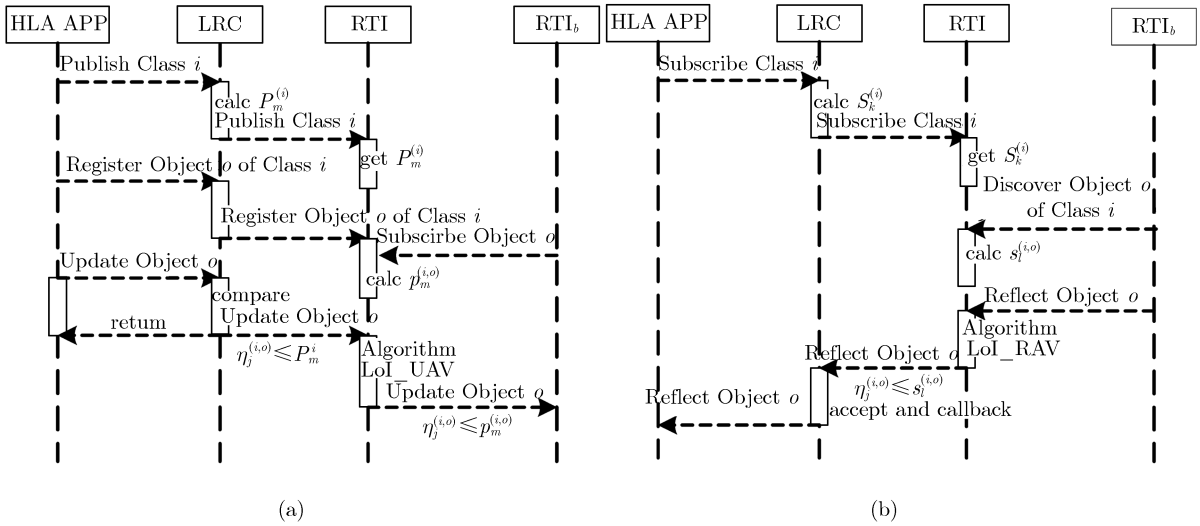

图 7　BH RTI 中基于 LoI 的自适应发布–订购过程

(a) 发布过程; (b) 订购过程

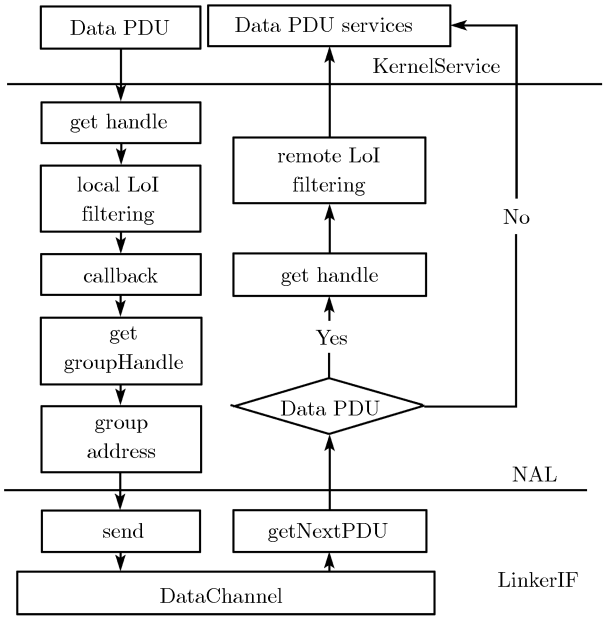

图 8　BH RTI 中拥塞控制的实现

9　性能实验

本节研究了自适应发布–订购机制、RTI 拥塞控制的效率和大规模分布交互仿真的一些具体实践.

9.1　基于 LoI 的自适应发布–订购机制

下面是一个对发布–订购机制的过滤效率进行研究的实例. 假设一个对象拥有 35 个属性, 各属性所对应的 LoI 如表 3 所示.

表 3　各 LoI 的对象属性数量

LoI	属性数量
LAYER_CRITICAL	0
LAYER_VISION	5
LAYER_ABOUT	10
LAYER_COMPONENT	10
LAYER_INSIDE	10

订购者将会根据他们各自的 $S_k^{(i)}$(订购者对对象类的 LoI) 接收到不同属性集的属性更新. 属性更新的接收如表 4 所示.

报文产生时, 需要根据发布–订购表对报文的属性集和订购属性集进行匹配. 表 5 对基于 LoI 的机制和现有标准机制进行了匹配次数对比. 从结果可以看出, 基于 LoI 的机制具有很高的匹配效率, 标准的发布–订购机制的匹配效率在不同情形下变化很大.

表 4　盟员订购者接收的属性

LoI $S_k^{(i)}$	属性数量
LAYER_CRITICAL	0
LAYER_VISION	5
LAYER_ABOUT	5+10 = 15
LAYER_COMPONENT	5+10+10 =25
LAYER_INSIDE	5+10+10+10 = 35

发布–订购机制的匹配效率是影响 RTI 单路时延的主要因素之一. 我们对一些常见的 RTI 进行了时延对比测试. 实验为采用华为 Quidway S3050 交换机的局域网, 具体的配置如表 6 所示.

图 9(a)和(b)分别为传统的集中式 RTI(如 DMSO RTI) 和分布式 RTI 实验配置. 时延测试程序采用 RTI 性能测试平台 DMSO RTI Benchmark 1.3.

RTI 时延测试结果如图 10 所示, 其中 BH RTI 2.2(集中式)、DMSO RTI 1.3NGv6 和 pRTI 1516v2.3 采用图 9(a) 的配置, BH RTI 2.2 采用图 9(b) 的配置. 集中式的 BH RTI 是指只有一个 RTI 提供服务, 所有盟员都连接到这个 RTI 上. 实验结果表明, BH RTI 2.2 具有相对较小的时延并且分布式 BH RTI 2.2 的时延最小. 集中式 BH RTI 2.2 的时延在低负载时相对较小, 并且随着负载的增加呈增大趋势. 由于负载对发布–订购效率没有影响, 所以时延的增加应该是 RTI 实现过程中其他因素导致的. 实验表明 LoI 可以在很大程度上提高报文传输性能.

表 5 发布–订购组合的匹配次数

LoI $\eta_j^{(i,o)}$	报文中属性个数	LoI $S_k^{(i)}$	基于 LoI 的机制	现有的机制
LAYER_VISION	5	LAYER_VISION	1	1
LAYER_VISION	5	LAYER_ABOUT	1	5
LAYER_VISION	5	LAYER_COMPONENT	1	5
LAYER_VISION	5	LAYER_INSIDE	1	5
LAYER_ABOUT	10	LAYER_VISION	1	1
LAYER_ABOUT	10	LAYER_ABOUT	1	1
LAYER_ABOUT	10	LAYER_COMPONENT	1	10
LAYER_ABOUT	10	LAYER_INSIDE	1	10
LAYER_COMPONENT	10	LAYER_VISION	1	1
LAYER_COMPONENT	10	LAYER_ABOUT	1	1
LAYER_COMPONENT	10	LAYER_COMPONENT	1	1
LAYER_COMPONENT	10	LAYER_INSIDE	1	10
LAYER_INSIDE	10	LAYER_VISION	1	1
LAYER_INSIDE	10	LAYER_ABOUT	1	1
LAYER_INSIDE	10	LAYER_COMPONENT	1	1
LAYER_INSIDE	10	LAYER_INSIDE	1	1

表 6 时延测试的主机配置

主机 Id	CPU/GHz	内存/MByte	操作系统	网卡
A_1	P4 2.8	512	WinXP	10/100M
A_2	P4 3.0	512	WinXP	10/100M
A_3	P4 2.4	768	WinXP	10/100M

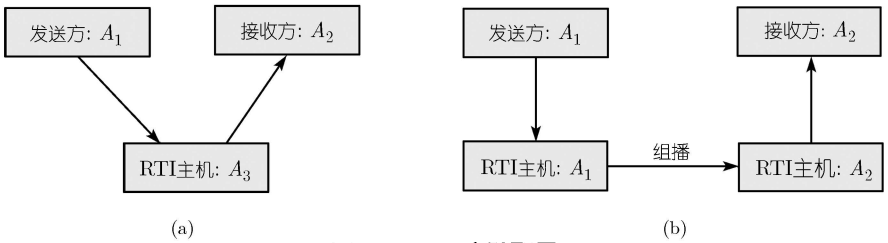

图 9 RTI 实验配置

(a) 集中式 RTI 实验配置; (b) 分布式 RTI 实验配置

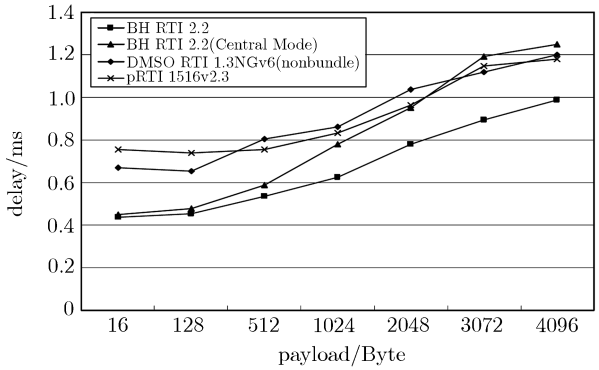

图 10　RTI 的时延对比测试

9.2　RTI 拥塞控制

我们也通过几个实验来测试 RTI 拥塞控制. 实验中使用了 4 台计算机, 配置如表 7 所示. 发送方 S_1 和 S_2 分别模拟了 500 个对象实例, 接收方 R_1 和 R_2 接收这 1000 个对象实例的所有更新数据. 为了单独测试接收方的拥塞控制, 在发送方禁用了拥塞控制, 从而可以看出接收方对数据流量的过滤能力. 所有的初始接收的报文被称为 “原始数据”, 经过拥塞控制后的报文为 “拥塞控制后数据”. 这样从接收方 R_1 和 R_2 中可以观察到拥塞控制的效率.

表 7　RTI 拥塞控制实验的主机配置

主机 Id	CPU	RAM	操作系统
S_1	P4 3.2G	1 G	WinXP
S_2	P4 3.2G	1 G	WinXP
R_1	P4 3.0G	512M	WinXP
R_2	P4 3.0G	512M	WinXP

实验环境为采用华为 Quidway S3050 交换机的百兆以太网. 根据 R_1 和 R_2 的系统需求, $f(l)$ 的具体设置如下:

报文大小: 220 Byte, 包括 10 个属性值, 每个属性值为 8 Byte.

S_1, S_2: 没有拥塞控制. 具有 LoI 为 LAYER_ABOUT 的 500 个对象.

R_1: 400 个 LoI 为 LAYER_ABOUT 的对象和 600 个 LoI 为 LAYER_VISION 的对象; f(LAYER_ABOUT)=20, f (LAYER_VISION)=5.

R_2: 400 个 LoI 为 LAYER_ABOUT 的对象和 600 个 LoI 为 LAYER_VISION 的对象; f(LAYER_ABOUT) = 10, f(LAYER_VISION) = 2.

带宽占用状况如图 11 所示. 可以看出 R_1 和 R_2 的总带宽使用均超过了 70 Mbit/s. 进行拥塞控制后, 带宽使用分别减少到 20 Mbit/s 和 10 Mbit/s. 从 R_1 和 R_2 选出两个对象实例来检验更新的平滑性. 从图 12 可看出每个对象实例都保持了稳定的更新频率. LoI 为 LAYER_ABOUT 的对象实例的更新频率等于 LAYER_VISION, LAYER_ABOUT 和 LAYER_CRITICAL 的更新频率之和, 其中 LAYER_CRITICAL 没有在本实验中出现. 在每个兴趣层次中, LoI 为 LAYER_ABOUT 的对象实例, 其在 LAYER_VISION 和 LAYER_ABOUT 的 LoI 层次的更新频率也都达到应用处理的要求, LoI 为 LAYER_VISION 的对象实例的更新频率也是连续的. LoI 层次下稳定的连续更新保证了每个属性进行更新的实时性.

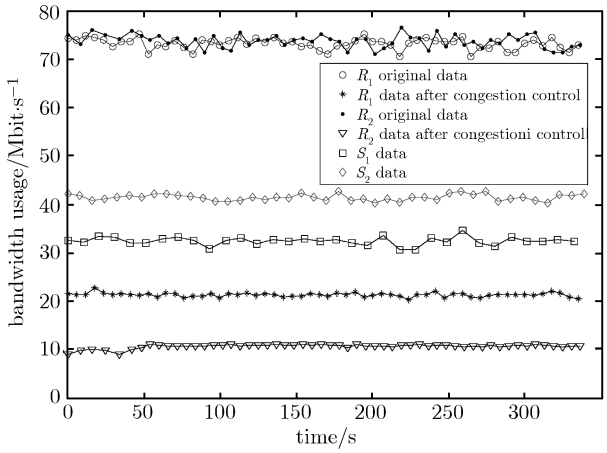

图 11　带宽占用状况

9.3　大规模分布交互仿真中的应用实践

BH RTI已经应用到一些大规模仿真演习中. 在一个战争游戏模拟环境中, 用 50台计算机在混合型 DDM 的基础上仿真了 47200个对象实例. 其网络拓扑如图 13 所示. 3 个子网分布在北京航空航天大学校园的如心楼一层和三层以及逸夫楼的四层, 并且通过光纤和路由器连接. 用Extreme Summit 24和两台Switch Summit 48交换机, 通过 PIM 组播路由协议支持子网间的组播路由. 采用 “ping” 命令测试两个子网间的报文平均时延, 如图 13 显示. 所有的计算机配置为P4 1.5 GHz到P4 2.8 GHz, 内存为 512 MByte. 由于性能所限, 一台计算机只能渲染 200 到 300 个坦克或者飞机的三维模型, 所以大部分仿真盟员采用了二维输出显示, 观察者盟员采用三维输出显示.

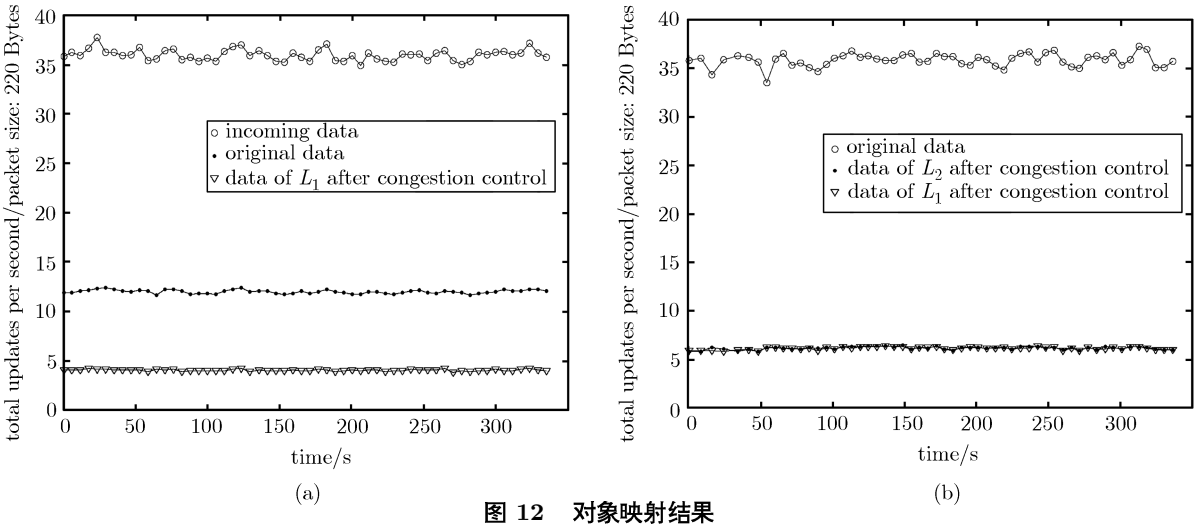

图 12 对象映射结果

(a) Host R_1's Object in LAYER_VISION; (b) Host R_2's Object in LAYER_ABOUT

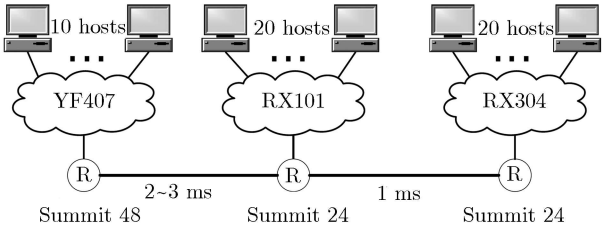

图 13 网络拓扑

每个主机启动 1 个 BH RTI 并且模拟 800 到 1400 个坦克或直升机的对象实例, 不包括动态的子弹/射击实体. 整个战场环境分为 30×30 个区域, 每个区域与一个组播地址相关联. 本次仿真中 EXTEND 设为 2, 设 LAYER_VISION 的对象实例每秒钟更新/映射 5 次, 更高层次的对象实例每秒更新/映射 20 次, 雷达和观察者的 RTI 对于所有的对象实例都是每秒映射 5 次. 这些较低频率的更新/映射能够满足仿真需要. 在组播和 LoI 技术的支持下, 我们成功运行了该大规模分布式仿真. 图 14 为仿真场景显示.

图 14 基于 BH RTI 的大规模分布交互仿真三维场景

10 总结

本文面向相关性评价问题, 提出了一种 LoI 技术. 论文的主要工作包括:

1) 提出了兴趣层次 LoI 的概念, 根据空间距离对接收属性和属性值的影响进行了数据的相关性分类. LoI 的 HLA 扩展仅需要很小的修改, 且维护了盟员代码的兼容性.

2) 在 LoI 基础上进行理论定义与推算, 得到了两个重要推论, 并提出了自适应发布–订购机制. 该机制可以在精确的发布–订购匹配之前直接丢弃大部分的无关报文.

3) 提出了 RTI 拥塞控制模型, 它基于 LoI 控制对象实例的更新频率可更有效地利用网络带宽.

LoI 相关技术支持基于 HLA 的系统, 本文还介绍了 LoI 实现过程中的一些重要问题. 我们从计算、实验等方面验证了 LoI 技术. 匹配次数的计算和 RTI 时延测试表明 LoI 技术能够大大加速数据的过滤. RTI 拥塞控制实验表明它能够根据要求降低网络流量. 此外, 还介绍了基于 BH RTI 支持了近 50000 个对象实例参与的仿真演习, 介绍了其网络拓扑、配置和设置.

参考文献

1 Morse K L, Bic L, Dillencourt M. Interest management in large-scale virtual environments. Presence-Teleop Virt, 2000, 9(1): 52–68

2 Abrams H A. Extensible interest management for scalable persistent distributed virtual environments. Dissertation for the Doctoral Degree. Monterey: Naval Postgraduate School, 1999. 17–26
3 Zyda M, Brutzman D, Darken R, et al. NPSNET-large-scale virtual environment technology testbed. In: Proceedings of the International Conference on Artificial Reality and Tele-Existence. Tokyo, 1997. 18–26
4 IEEE. Standard for Information Technology-Protocols for Distributed Interactive Simulation Applications. IEEE Std 1278.1. 1995
5 IEEE. Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)–Framework and rules. IEEE Std 1516-2000. 2000
6 IEEE. Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)–Federate Interface Specification. IEEE Std 1516.1-2000. 2000
7 U.S. Department of Defense. High Level Architecture (HLA)–Federate Interface Specification Version 1.3, 1998
8 Torpey M, Wilbert D, Helfinstine B, et al. Experiences and lessons learned using RTI-NG in a large-scale, platform-level federation. In: Proceedings of the Spring Simulation Interoperability Workshop. Orlando, 2001. 00F-SIW-031
9 Helfinstine B, Wilbert D, Torpey M, et al. Experiences with data distribution management in large-scale federations. In: Proceedings of the Fall Simulation Interoperability Workshop. Orlando, 2001. 01F-SIW-032
10 Hyett M, Wuerfel R. Connectionless mode and user defined DDM in RTI-NG V6. In: Proceedings of the Spring Simulation Interoperability Workshop. Orlando, 2003. 03S-SIW-102
11 McLean T, Fujimoto R, Fitzgibbons B. Middleware for real-time distributed simulations. Concurr Comp-Pract E, 2004, 16(15): 1483–1501
12 Zhao H, Georganas N D. HLA real-time extension. Concurr Comp-Pract E, 2004, 16(15): 1503–1525
13 Zabele S, Stanzione T, Kurose J, et al. Improving distributed simulation performance using active networks (invited paper). In: Proceedings of the World Multi Conference. 2000
14 Zabele S, Dorsch M, Keaton M, et al. Dynamic interest filtering for optimal state update messaging. In: Proceedings of the I/ITSEC (Interservice/Industry Training, Simulation & Education Conference). Orlando, 2001
15 Cai W T, Lee F B S, Chen L. An auto-adaptive dead reckoning algorithm for distributed interactive simulation. In: Proceedings of the 13th Workshop on PADS (Parallel and Distributed Simulation). Washington: IEEE, 1999. 82–89
16 Zhou S P, Turner S J, Cai W T, et al. A utility model for timely state update in distributed wargame simulations. In: Proceedings of the 18th workshop on PADS (Parallel and Distributed Simulation). New York: ACM, 2004. 105–111
17 Zhou Z, Zhao Q P. Research on RTI congestion control based on the layer of interest. J Softw(in Chinese), 2004, 15(1): 120–130
18 Zhou Z, Zhao Q P. LoI-based flow control on low-bandwidth federates. In: Proceedings of the 2nd International Conference for E-Learning and Games, Edutainment 2007. LNCS, Vol 4469. Hongkong: Springer, 2007. 904–915