

考虑类型信息的异构数据源信息交换方法研究

王 博 郭 波

(国防科学技术大学信息系统与管理学院 长沙 410073)

摘 要 异构信息源模式集成中,除解决模式结构冲突外,还需要考虑模式结构对应的数据类型差异,以保证模式集成和信息交换过程的准确性。本文在考虑不同模型语言类型约束和复合对象类型的基础上形式化描述了类型体系,在公共类型体系内容定义的基础上给出了考虑约束和复合对象类型的中间模型类型体系的构建方法,建立了异构数据模式类型体系与中间模型类型体系的类型映射关系。相比其他类型匹配方法,具有搜索空间小、处理范围广的特点。

关键词 模式集成,信息交换,类型体系,中间模型类型体系

Study on Method of Data Exchange in Heterogeneous Data Sources Considering Type Information

WANG Bo GUO Bo

(College of Information System and Management, National University of Defense Technology, Changsha 410073)

Abstract In order to ensure the veracity of the process of schema integration and information exchange, both the structs conflicts and type differences of data corresponds to schema structs should be considered in the process of schema integration of heterogeneous data source, based on considering the constraints of different model language and compound object type, formalize the type hierarchy system, base on the common type hierarchy system content definition, taking type constraints and compound object type into account, we give the construction method of inter model type hierarchy system, build the mapping relationship between heterogeneous schema type hierarchy system and inter model type hierarchy system, which has a smaller searching space, and a larger scope of type comparing to other type matching algorithm.

Keywords Schema integration, Data exchange, Type hierarchy system, Inter model type hierarchy system

1 引言

在异构信息集成系统中,无论是数据集成还是服务集成,都需要考虑数据和语义信息。其中,信息类型是语义集成的关键。异构信息源的数据建模语言的差异使得数据在异构信息系统之间的交换很难做到精确,其中数据类型的不一致是导致交换异常的主要因素。基本的数据类型包括:整型、字符串、浮点、布尔等,其他数据类型包括复合类型(例如对象类型)。给定两个数据源^[6], $\langle S_a, I_a, Ext_{S_a, I_a} \rangle, \langle S_b, I_b, Ext_{S_b, I_b} \rangle$, 其中 S_a, S_b 表示数据源模式, I_a, I_b 表示模式对应的实例外延(例如 S 可以表示关系数据库系统中一组表结构,而 I 则对应表结构的具体数据), $Ext_{S, I}$ 给出了模式和外延的映射关系。根据模式转换理论^[7], 数据集成过程是一个异构数据模式向公共数据模型(CDM)转换和映射的过程: $Trans(S_a \rightarrow S_b)$, 但该过程只关注转换前后模式中模式结构的对应关系,例如等价关系,包括名称变化、结构添加和删除等,以及结构之间以及结构自身的相互约束关系。在异构信息集成环境中,在延 $S_a \rightarrow S_b$ 路径进行数据交换时,除了考虑 S_a, S_b 的模式结构对应关系,也需要考虑模式之间对应的结构的数据类型之间的关系,否则会发生交换错误。本文首先总结了异构信息系统之间信息交换过程中的类型因素,给出了全局类型体系形式化表示方法,考虑了模式集成核数据交换过程中的类型约束条件,定义了中间类型体系,作为类型变换的中介。

2 数据类型对异构信息系统之间信息交换的影响分析

数据类型异构并不反映异构信息系统的数据模型设计差异,而体现了数据模型对应的模式结构本身以及对应的实例外延语义范围的差别。不同信息系统由于设计和开发独立,使用的平台和工作环境不同,使得其信息类型千差万别。考虑数据类型的异构数据集成信息系统在进行数据交换时需要解决如下五个方面给数据交换带来的问题,这里用 S_x 表示数据交换过程对应的源模式, S_t 对应目标模式, $struct_{x, i}$ 表示模式 S_x 中标号为 i 的模式结构,用 $type(struct_{x, i})$ 表示对应的类型,用 $=$ 表示类型等价。

(1) 类型范围变化^[1]: S_s 和 S_t 对应的 $type(struct_{i, s})$ 具有共同的高层父类型: $type(ancestor(struct_{i, s})) = type(ancestor(struct_{i, t}))$, 但具有不同的类型: $type(struct_{i, s}) \neq type(struct_{i, t})$ 。若 $type(struct_{i, s}) \subseteq type(struct_{i, t})$, 则在转换过程中不会发生类型错误,反之则可能出现溢出等类型错误。例如, long 和 short 同属于 Integer 类型,但若从 long 类型,到 short 类型进行数据交换可能发现问题。

(2) 数据类型变化^[1]: S_s 和 S_t 的对应结构类型分别为 $type(struct_{i, s})$ 和 $type(struct_{i, t})$, $type(struct_{i, x})$, 若 $type(struct_{i, s}) \neq type(struct_{i, x})$, 例如表示人员信息中的性别属性定义在 String 类型集合{"male", "female"}上,而在 S_t 对应为 boolean 类型,需要进行类型映射: $\{ \langle \text{"male"}, \text{true} \rangle, \langle \text{"female"}, \text{false} \rangle \}$ 才能保证信息交换的准确性。

(3) 类型值约束变化: S_s 和 S_t 的对应结构类型分别为 $\text{type}(\text{struct}_{i,s})$ 和 $\text{type}(\text{struct}_{i,t})$, $\text{Type}(\text{struct}_{i,s}) = \text{Type}(\text{struct}_{i,t})$, 但 $\text{Range}(\text{type}(\text{struct}_{i,s})) \neq \text{Range}(\text{type}(\text{struct}_{i,t}))$ 。例如 $\text{type}(\text{struct}_{i,s}) = \text{type}(\text{struct}_{i,t}) = \text{Integer}$, 在 S_s 中满足约束 $\text{Range}(\text{type}(\text{struct}_{i,s})) \in [0, 100]$, 而 $\text{Range}(\text{type}(\text{struct}_{i,t})) \in [60, 100]$, 当数据从 S_s 到 S_t 进行交换时, S_t 可能因不满足约束发生错误。

(4) 带有语义信息的数据类型: 数据类型表示某种语义信息。例如在 S_s 中, 有表示学生成绩的属性, 其取值范围为 $[0, 5]$, 表示 5 分制; 而在 S_t 中, 对应学生成绩属性的取值范围为 $[0, 100]$, 表示 100 分制。因此转换时, 需要建立两个同类型信息的语义映射关系。

(5) 同类型、异名称情况: 不同信息系统对内涵相同的类型命名或表述方法不同, 同样也需要建立映射关系。

考虑类型信息的异构信息系统进行信息交换时, 主要考虑上述五方面。在集成和数据交换时, 如不考虑这些问题, 则无法做到准确的数据交换。同模式集成一样, 不同本体域的模式集成到同一个本体域下, 不同人设计的独立的模式集成到同一的全局模式 (*GlobalView*) 下, 都需要一个共同的知识体系。类型信息在数据交换过程中也需要一个参考标准, 例如虽然某程序设计语言被不同人使用, 开发出不同的软件系统, 但其内部的数据类型都遵循统一的标准。在信息集成过程中, 同样需要类似的类型标准。参考模式集成的模式匹配方法, 目前主要的方式包括建立可扩展的类型信息存储结构、建立类型匹配表、领域信息抽取^[3]、构建类型的公共本体域^[8]等。Erhard 等通过建立类型映射表或语义映射表达方式, 辅助数据交换避免发生错误^[8], 虽然实现了自动的模式匹配, 但工作量比较大, 在实现多数据模型异构信息系统的集成和信息交换时, 该方法显得比较繁琐。Unity^[5] 系统采用树状结构, 实现了可扩充的全局数据字典作为异构信息系统集成的统一依据, 但该树状结构的全局数据字典只用于描述数据模型的属性名称, 并未涉及类型信息。TSIMMIS^[4] 系统将数据类型作为对象交换的一部分进行存储, 其公共数据模型为对象模型, 并未考虑数据交换过程中的类型约束以及类型变换。目前在模式集成和模式匹配领域中, 类型信息并未作为重点内容考虑, 大部分研究人员主要关注如何通过选取合适公共数据模型以及如何构建公共领域本体来实现异构信息系统集成和数据交换。本文重点解决异构模式集成和异构系统间信息交换过程中的类型匹配问题, 首先建立类型空间, 包括简单类型和复合类型, 并给出类型约束描述, 在此基础上通过构建映射函数来实现类型匹配。

3 通用数据类型体系形式化描述

首先给出几个概念。

定义 1 类型匹配函数空间 Φ_f , 具有 n 个数据源 $DS_{[1 \sim n]}$ 的数据转换系统的转换函数空间 Φ_f 是所有定义在 $\text{any-}Type_{[1 \sim n]} \times \text{DSTyp}_{[1 \sim n]}$ 上的类型转换函数 f_{exch} 的集合。

下面给出用于在数据交换和模式集成过程中做标准参考的类型体系定义。

定义 2 类型体系 $\langle \text{Types}_x, \text{Ext}_x, =_x, <_x, \cap_x, \text{Types-Constraint}_x, \{ \text{type}_x \otimes, \dots, \otimes \text{type}_x \}, \text{Mapping}_x \rangle$, 其中 Types_x 为模式 S_x 对应的数据建模语言采用的类型集合; Ext_x 是以类型为参数的实例函数, 返回值为该类型对应的所有的数据值, 表示为 $t \in \text{Types}_x \rightarrow \text{Ext}_x(t) \subseteq \text{Ext}_x(\text{allType}_x)$, allType_x 代表某个数据源 DS_x 数据模型语言定义的所有数

据类型, $\text{Ext}_x(\text{allType}_x) \subseteq C$, 其中 $C = \text{Ext}(\text{allType}_x \times \text{DSType}_x)$, $x \in [1, n]$, 表示所有可能数据类型实例;

$=_x$ 定义了一个等价关系, 对于 $t, t' \in \text{Types}_x$ 有 $t =_x t' \Leftrightarrow \text{Ext}_x(t) = \text{Ext}_x(t')$, 表示两个类型具有等价关系, 可以自由交换信息;

$<_x$ 定义了一个偏序关系, 对于 $t, t' \in \text{Types}_x$, 有 $t <_x t' \Leftrightarrow \text{Ext}_x(t) \subset \text{Ext}_x(t')$, 偏序定义了两个类型之间的包含关系;

$\cap_x$ 定义了一个不交关系, 对于 $t, t' \in \text{Types}_x$, 有 $t \cap_x t' \Leftrightarrow \text{Ext}_x(t) \cap \text{Ext}_x(t') = \phi$, 不交类型的子类型同样不交, 即若 $t \cap_x t'$, 则对 $\forall t_{\text{sub}} \subseteq t, t'_{\text{sub}} \subseteq t'$, 都有 $t_{\text{sub}} \cap_x t'_{\text{sub}}$;

TypesConstraint_x 定义了类型的约束空间, 表示为 $\{ \text{type}_x \times \text{Constraint}_x \}$, 例如整型类型取值范围、字符串长度等。

由于在面向对象等高级数据模型中包含高层元素, 这里定义为复合类型元素: $\{ \text{type}_x \otimes, \dots, \otimes \text{type}_x \}$ 。例如对象属性“Person”可以理解为一个复合类型:

```
Struct Person
{
    Boolean sex;
    String DataofBirth;
    Int Height;
    ...
}
```

此时, $\text{Person} \in \text{Type}_x$, 对应的 $\{ \text{type}_x \otimes, \dots, \otimes \text{type}_x \}$ 表示为 $\{ \text{boolean} \otimes \text{String} \otimes \text{Int} \otimes \dots \}$; Mapping_x 定义了一个映射表 $\langle t_a, t_b, \{ \langle s_a^1, s_b^1 \rangle, \dots, \langle s_a^n, s_b^n \rangle \} \rangle$, 其中 $t_a, t_b \in \text{Types}_x$, $s_a^1, \dots, s_a^n \subseteq \text{Ext}(t_a)$, 且 $s_a^1 \cap s_a^2, \dots, \cap s_a^n = \phi$, 类似地 $s_b^1, \dots, s_b^n \subseteq \text{Ext}(t_b)$, 且 $s_b^1 \cap s_b^2, \dots, \cap s_b^n = \phi$ 。 $\text{Mapping}_x(\text{type}_a, \text{type}_b)$ 指定了由 type_a 到 type_b 的映射关系表, 重载 Mapping_x , 定义如下的函数:

$$\text{Mapping}_x(t_a, t_b, v_a) = \text{First}(s_b^i) \mid \langle t_a, t_b, \text{map} \rangle \in \text{Mapping}_x \wedge \langle s_a^i, s_b^i \rangle \in \text{map} \wedge v_a \in s_a^i$$

$$\text{Mapping}_x^{-1}(t_a, t_b, v_a) = \text{First}(s_b^i) \mid \langle t_b, t_a, \text{map} \rangle \in \text{Mapping}_x \wedge \langle s_b^i, s_a^i \rangle \in \text{map} \wedge v_a \in s_a^i$$

其中 $\text{First}()$ 返回了按照某种顺序排序的有序集合中的第一个元素, 重载的 Mapping_x 函数给出了类型匹配的方法。下面举例说明。

给定某关系数据模式的两个表示学籍的数据模式 S_1, S_2 , 类型集合为 Type_{rel} , 都包含成绩 *grade* 属性, S_1 中 *grade* 类型定义为 t_1 , S_2 中 *grade* 类型定义为 t_2 , $t_1 = t_2 = \text{Float} \in \text{Type}_{\text{rel}}$ 。 S_1 中成绩为五分制, S_2 中成绩为 100 分制, 则对应的映射关系 $\text{Mapping}_{\text{rel}}$ 表示为

$$\text{Mapping}_{\text{rel}}(t_1, t_2) = \{ \langle t_1, t_2, \{ \langle 1, [0, 20] \rangle, \langle 2, [20, 40] \rangle, \langle 3, [40, 60] \rangle, \langle 4, [60, 80] \rangle, \langle 5, [80, 100] \rangle \} \} \}$$

通用类型体系给出一种标准化描述某数据模型(包括高层数据模型以及中间数据模型^[2])类型体系的框架, 通过提取信息源模式的类型信息并映射到类型体系中, 可以在异构信息源模式之间建立类型匹配和转换的桥梁, 避免数据传输过程中的类型匹配异常。下面利用类型体系, 通过定义中间公共类型体系, 给出异构模式间的类型转换和匹配方法。

4 异构模式间类型转换和匹配

数据集成的对象通常都采用不同数据建模语言, 即便采用相同模型语言, 其对应的数据模式由于设计因素也不尽相同。当源模式和目标模式采用建模语言不同(例如 ER 模型到 XML 模型的数据交换)或两个模式之间没有共同的类型体系时, 需要建立一个公共类型体系作为交换信息的参考标

准。同公共数据模型类似,各种高层的数据建模语言对应的类型体系在公共类型体系基础上转换为中间模型类型体系以实现交互,因此实现异构模式类型转换和匹配需要解决两个问题:

- (1)公共类型体系的构建;
- (2)中间模型类型体系的构建。

4.1 公共类型体系

定义 3(公共类型体系) 公共类型体系是定义在类型体系形式化框架下的类型体系,其作用是异构类型体系建立公共类型标准,并在公共类型体系中通过 $Mapping()$ 函数给出其与各异构类型体系的映射关系,这里简称为 TH_c :

$$TH_c = \langle Types_c, Ext_c, =_c, <_c, \cap_c, TypesConstraint_c, \{type_c \otimes, \dots, \otimes type_c\}, Mapping_c \rangle$$

TH_c 是可扩充的,在数据交换和类型匹配过程中不断丰富。下面给出公共类型体系个元素的概念范围:

$$Types_c = \{anyType_c, string_c, char_c, integer_c, short_c, fshort_c, float_c, real_c, boolean_c\}$$

$anyType_c$ 表示最高层的任意类型,用于避免匹配过程中的类型失配; $block_c$ 用于表示复合类型元素,例如图像、视频、音频或自定义的对象。

$$Ext_c = \{boolean_c \rightarrow \{true, false\}, integer_c \rightarrow \{-2^{31}, \dots, 2^{31} - 1\}, \dots\};$$

$$=_c = \{\};$$

$$<_c = \{short_c < integer_c, integer_c < anyType_c, real_c < float_c, float_c < anyType_c, char_c < String_c, string_c < anyType_c, boolean_c < anyType_c, block_c < anyType_c\}$$

$$\cap_c = \{integer_c \cap boolean_c, integer_c \cap float_c, \dots\};$$

$TypesConstraint_c = \{natural\}$, $natural$ 表示自然约束,在出事情况下,类型约束为其本身约束,在不涉及到具体模式时可以忽略。

$Mapping_c() = \{\}$,在公共类型向其他异构类型体系映射时需要定义该函数。

4.2 中间模型类型体系构建过程

中间模型类型体系是避免异构模式之间数据交换类型错误的保障,其构建过程如下。

(1)初始状态,存在一个定义 3 给出的 TH_c (内容如上),以及若干被集成模式 $\{S_{i,x}\}$,称为被选模式。

(2)从被集成的各异构模式或者数据交换过程涉及到的异构模式中取模式 $S_{i,x}$,获取其类型体系 TH_x ,同 TH_c 进行融合,包括类型扩充、类型映射关系建立等,形成中间模型类型体系 TH_{int} 。

(3)从被选模式中提取模式 $S_{j,y}$,获取其类型体系 TH_y ,同 TH_c 和 TH_{int} 融合,得到中间模型类型体系。

(4)循环(3),直至融合所有的被选模式的类型体系,得到最终的中间模型类型体系。

对于新加入模式 S_{new} ,可以重复(2)、(3)过程,得到中间模型类型体系 $TH_{new,c}$ 。

记被融合高层类型体系为 TH_x ,已存在的由类似过程生成的中间模型类型体系为 TH_m ,参考公共体系为 TH_c ,融合后的中间模型类型体系为 TH_x 。下面给出迭代过程 C 语言伪码:

```
Merge( $TH_x, TH_m, TH_c, CM_x$ ) {
    //初始化过程:包括类型和关系的合并
```

```
Typesx = Typesx ∪ Typesm;
Extx = Extx ∪ Extm;
∩x = ∩x ∪ ∩m;
<x = <x ∪ <m;
=x = =x ∪ =m
Mappingx = Mappingx ∪ Mappingm ∪ CMx;
//第一次循环用来重定义约束
for( $t_x \in Types_x$ ) {
    for( $t_m \in Types_m$ ) {
        if( $t_x = t_m$ )
            TypesConstraintx = TypesConstraintx ∪ TypesConstraintm;
    }
}
//第二次循环用来进行类型压缩,通过比较和搜索确定类型最精确表示
for( $t_x \in Types_x$ ) {
    tx = anyTypec; //初始赋值去公共类型体系中的最大类型元素
    for( $t_c \in Types_c$ ) {
        if(Ext( $t_c$ ) ⊆ Ext( $t_x$ ) ∧ Ext( $t_c$ ) ⊆ Ext( $t_x$ )) tx = tc;
    }
    if(Ext( $t_x$ ) ⊆ Ext( $t_x$ )) { //对产生的新类型,扩展公共类型体系
        t'c = newType( $t_x$ );
        Typesc = Typesc ∪ {t'c};
        <c = <c ∪ {t'c < tx};
        tx = t'c;
    }
    for( $t_c \in Types_c$ ) {
        if(Ext( $t_c$ ) ∩ Ext(t'c)) ∩c = ∩c ∪ {tc ∩ t'c};
    }
}
=x = =x ∪ {tx = tx};
}
//根据更新的子类型,更新源模式中的复合类型
Update({typex ⊗, ..., ⊗ typex}) to {typec ⊗, ..., ⊗ typec};
{typec ⊗, ..., ⊗ typec} = {typec ⊗, ..., ⊗ typec} ∪ {typec ⊗, ..., ⊗ typec};
return <Typesx, Extx, =x, <x, ∩x, TypesConstraintx, {typec ⊗, ..., ⊗ typec}, Mappingx>;
```

上述伪码中 CM_x 为 TH_c 和 TH_x 中同名 $Type$ 的 $Mapping$ 关系,函数 $newType()$ 给出了新生成类型定义。

通过定义不同模型语言模式的类型体系与中间类型体系的映射关系,可以决定异构信息系统之间数据交换的约束关系。例如具有 $=_x$ 关系的模式结构对应的数据外延可以自由进行数据交换,举例说明:由等价映射关系可发现在 SQLServer2000 中 $smallint_{sql}$ 类型和 XML 中的 $short_{xml}$ 都映射到 TH_c 中的 $short_c$,因此二者中具有此类型的结构对应的数据外延,可以在仅有类型约束的前提下自由交换。对于 $<_x$ 关系,由偏序关系的左操作数向右操作数进行的对应结构数据外延的信息交换是自由的,而反之则会出现类型错误,可能会出现数据溢出情况。

相比其他类型匹配方法,本文给出的方法更适合多模式的异构数据源之间的信息交互。采用领域本体的方式很难做到完全准确,且实现自动地类型匹配和数据交换比较困难。构建全局类型字典,能够保证数据交换过程中类型的准确性,但大部分依赖手工操作。通过类型映射表方式可以利用程序实现自动的类型匹配,但当数据源模式的数量和类型增加的时候,其工作量比较繁琐且考虑类型约束信息时搜索空间成指数增长,而且无法处理复合类型。本文扩充了类型层次体系,引入了复合类型和类型约束,通过构建算法自动实现约束和类型的映射,能够缩小匹配过程中的搜索空间。

小结 模式集成和数据无障碍交换是目前数据集成领域的研究热点。尤其在面向服务的集成环境下,局部模式类型多,包括普通数据模型、服务模型等,其变化频繁,信息类型对模式集成和信息交换的影响同模式集成中的语义集成一样是不可避免的问题。本文结合异构信息源模式,在考虑类型约束条件和复合数据类型的基础上形式化描述了类型体系,在公共

类型体系的基础上给出了构建中间模型类型体系的方法。中间件方法是目前异构信息集成的主流思想。通过定义普通类型体系与公共类型体系的映射关系,可以保证异构信息源在模式集成和信息交换过程中在数据类型方面的准确性。

参考文献

- 1 Smith A C, McBrien P. Inter Model Data Exchange of Type Information via a Common Type Hierarchy. DISWeb06, Universitaires de Namur, Luxembourg, 2006. 307~321
- 2 Boyd M, McBrien P. Comparing and Transforming Between Data Models via an Intermediate Hypergraph Data Model. Computer Science, 2005(3730): 69~109
- 3 Chen S K, Lo M L, Wu K L, et al. A practical approach to extracting DTD-conforming XML documents from heterogeneous data

sources. Information Sciences, 2006, 176 (7): 820~844

- 4 Garcia-Molina H, Papakonstantinou Y, Quass D, et al. The tsimms approach to mediation: Data models and languages. J Intell Inf Syst, 1997, 8 (2): 117~132
- 5 Lawrence R, Barker K. Unity-A Database Integration Tool. Department of Computer Science, University of Manitoba, 2000
- 6 McBrien P, Poullovassilis A. Schema Evolution in Heterogeneous Database Architectures. In: A Schema Transformation Approach. In: CAiSE02, Springer Verlag LNCS, Toronto Canada, 2002. 484~499
- 7 McBrien P, Poullovassilis A. A Uniform Approach to Inter-Model Transformations. In: CAiSE'99, Springer, Heidelberg Germany, 1999. 333~348
- 8 Rahm E, Bernstein P A. A survey of approaches to automatic schema matching. VLDB J, 2001, 10 (4): 334~350

(上接第 55 页)

$$\pi_0 = (1/L)(a/(1-\eta)^{b+1}) \quad (14)$$

$$\pi_1 = (1/L)(a/(1-\eta)^{b+1}) \quad (15)$$

从而可得

$$a = \exp \left\{ \frac{\ln \pi_1 - \ln \pi_0}{\ln(1-\eta) - \ln(1-\eta)} - \ln(\pi_0 L) \right\} \quad (16)$$

$$b = \frac{\ln \pi_1 - \ln \pi_0}{\ln(1-\eta) - \ln(1-\eta)} - 1 \quad (17)$$

式(12)、(13)与(16)、(17)一起,即为我们的定价公式。其中式(12)为服务定价公式,式(13)为资源定价公式。根据式(12)可知,各用户应用的价格与它所经路由上带宽资源的使用情况有关:如果某应用在其路由上带宽资源的占用率较低,则价格将被设置为较低值,这将刺激用户的消费;如果某应用在其路由上带宽资源的占用率过高(容易拥塞),则价格将被设置为较高值,这将抑制用户的消费。

3 基于市场模型的宽带分配

系统基于市场模型计算得到带宽的市场价格后,就可以利用该价格作为带宽分配的一个参数进行带宽协商、分配和动态调整等工作。其方法是由带宽管理程序实时监测新用户在当前价格下的带宽需求和带宽实际使用情况,并根据用户对价格的承受力及实时性要求的不同,进行带宽资源协商及分配,从而使网络带宽资源平衡使用和建立端到端的适时通道。基于市场模型的带宽分配的基本过程如下:

(1)各用户提出带宽需求 $\langle x_1, \dots, x_N \rangle$ 和各自可承受的价格上限 $\langle W_1, \dots, W_N \rangle$ 。

(2)系统分析用户的带宽需求和网络可用带宽的情况 $\langle X_1, \dots, X_M \rangle$,并按式(12)、(13)计算价格 $\langle P_1, \dots, P_N \rangle$,设所有满足 $P_i \leq W_i$ (即愿意接受其服务价格)的用户集为 U 。

(3)系统用价格进行市场供需调控和资源分配:

如果 $\forall j \in [1, M] \sum_{(i|A_{ij}=1, i \in U)} x_i \leq X_j$,则让这些用户全部得到资源;否则,与路由上经过各满足 $\sum_{(i|A_{ij}=1, i \in U)} x_i \leq X_j$ 的链路 j 的实时性要求低的各应用进行协商,降低其带宽量,直至带宽可以容纳所有愿意接受该价格和新的(较低)带宽量的用户,然后进行资源分配和调度。

(4)系统监测在当前价格的作用下用户带宽需求和各链路资源变化信息,转(2)。

其中,价格杠杆的动态调整过程始终以用户满意度提高和系统效率最优为出发点,以系统均衡为目标,然后收集资源供求的最新反馈信息,变动现有的价格体系,实施价格的杠杆调节。而根据用户的带宽需求和系统的可用资源情况,制订合理的价格,根据用户需求假定,价格高时用户将降低其需求,价格低时用户将增大其需求。这样,通过资源的协商和价

格杠杆的调整,可以使相对集中的用户需求调控到相对空闲时段,用户满意度得到提高,系统效率趋于最佳,从而使供求双方最终达到均衡状态。

4 基于市场模型的带宽分配方法的有效性证明

我们利用数理经济学原理来证明本文提出的带宽分配方法的有效性。由于我们的带宽分配算法的核心在于第2节给出的定价策略,这里只需证明在该定价策略作用下网络系统存在均衡解即可。

引理 u_i 在 $x_i \in [\lambda_i, +\infty]$ 上是关于 x_i 可微、单调增的凹函数。

证明:由式(4),我们有 $\partial u_i / \partial x_i = k_i / (x_i - \lambda_i)$

$$\partial^2 u_i / \partial^2 x_i = -k_i / (x_i - \lambda_i)^2$$

由于 $k_i > 0$, 故 $\partial u_i / \partial x_i > 0$, $\partial^2 u_i / \partial^2 x_i < 0$, 所以 u_i 在 $x_i \in [\lambda_i, +\infty]$ 上是关于 x_i 可微、单调增的凹函数。

定理 在 L_j 一定且满足用户需求假定的条件下,系统存在均衡解。

证明:由引理, $\forall i \in [1, k]$, u_i 在 $x_i \in [\lambda_i, +\infty]$ 上是关于 x_i 连续、单调增的凹函数,而式(8)的约束条件构成的可行解区域是凸集。根据一般均衡理论^[1],只要用户需求假定成立,则在第2节给出的价格杠杆作用下,必存在资源的最优分配 $\langle X_1, \dots, X_N, Y_1, \dots, Y_M \rangle$ 满足均衡条件 I ~ III, 即系统存在均衡解 $\langle X_1, \dots, X_N, Y_1, \dots, Y_M \rangle$ 。

小结 本文基于市场模型,根据用户提出的带宽资源需求和系统可用资源的状况,按用户满意度最佳和系统效率最佳的原则制订价格,同时给出利用价格实现带宽分配和管理的方法,并分析其有效性。该方法可用于网络拥塞控制和资源管理等。

参考文献

- 1 操龙兵,戴汝为. 集智慧之大成的信息系统——Internet [J]. 模式识别与人工智能, 2001, 14(1): 1~8
- 2 Gupta A, Stahl D, Whinston A. Priority Pricing of Integrated Services Networks. In: Mc Knight W, Bailey J, eds. Internet Economics, Cambridge, MA: MIT Press, 1997
- 3 Mac Kie-Mason J K, Varian H R. Pricing the Internet. In: Proc and Int'l Conf Telecom Syst Model Anal. Nashville, TN, Mar. 1994
- 4 Shenker S, Clark D, Estrin D, et al. Pricing in computer networks: Reshaping the research agenda. Computer Communications Review, 1996, 26(2): 19~43
- 5 Daigle J N. Queuing Theory for Telecommunications. Readings, MA: Addison Wesley, 1992
- 6 Arrow J, Intriligator M D. Handbook of Mathematical Economics. Amsterdam: North Holland, 1982
- 7 富川,青木. イーサネソクテムのQos制御の機能設計と試験. 研究所技術二エース, 富士通研究所, Mar. 1995