

Native XML 数据库技术综述^{*})

逯 鹏 吕良双 高庆一

(北京航空航天大学计算机学院 北京100083)

摘 要 XML 成为数据统一描述和表示的标准。如何管理已经出现的大量的 XML 数据成为重要的问题。Native XML 数据库(NXD)是解决 XML 数据管理的重要手段。分析和综述了 NXD 研究现状,以及几个主要研究方向:数据模型、查询语言、存储技术等。分析了 NXD 的基本特性和具有代表性的 NXD 系统,最后指出了目前研究存在的不足以及该学科的发展趋势。

关键词 Native,数据库,XML,数据模型,查询,存储

An Overview of Native XML Database Technology

LU Peng LU Liang-Shuang GAO Qing-Yi

(School of Computer, Beijing University of Aeronautics and Astronautics, Beijing 100083)

Abstract XML has become the standard for data representation and exchange. How to manage large number of XML data is an important problem. Native XML Database Management System (NXD) is an important means of managing XML data. First, this paper presents an overview and analysis of NXD including its data model, query language, storage technology, etc. Second, the paper analyzes the basic features in NXD and presents a representative NXD system. In the end, some shortages and research directions on the NXD subject are pointed out.

Keywords Native, Database, XML, Data model, Query, Storage

1 引言

XML 已经成为数据表示和交换的标准,网络上用 XML 描述的数据正在以指数级增长,如何有效地管理大量的 XML 数据成为重要的问题。普遍的观点是使用数据库这种易于使用、系统化、为人们所熟悉的技术对 XML 进行管理。至于如何使用,则基本上形成两大阵营^[1]: XML-Enabled DBMS (XED)和 Native XML DBMS(NXD)。

XED 派的观点认为,不必考虑底层数据的存储模式,只要能存入和取出 XML 数据,并符合数据库的基本特性,就是 XML 数据库。例如 Oracle Oracle9i, IBM DB2, MS SQL Server 2000 等。从实现角度,基本上是在原有数据库基础上增加了支持 XML 的功能,内部的数据模型没有改变,通过适当的 XML API,对 XML 文档进行存储和查询。

但是,XML 的半结构化特性并不符合传统的(对象)关系模型的结构化特性。强制转换会造成数据信息的丢失和系统性能的减弱,所以 NXD 的观点认为在数据库中应当以 XML 原始的格式存取 XML 文档,数据库内部的数据模型是依据 XML 特性建立的,而不是关系表或者对象。

兴起于20世纪60、70年代的层次数据库和网状数据库,在当时的计算机数据处理技术方面发挥了重要作用。20世纪70年代以后, E. F. Codd 与其他一些人通过严格的数学工作创造了现在无处不在的关系数据库模型。20世纪80年代数据库技术引入了面向对象的数据模型,之后产生了大量的理论和应用成果。目前 XML 的出现,同样对数据库研究提出了新的问题。由于 XML 数据模型既能描述半结构化数据也能够描述结构化数据,因此将现有的数据库技术扩展到 XML 的半结构化数据模型基础之上会给数据库研究领域带来根本的变革^[2]。目前已经有了一些 NXD 产品^[3],关于 NXD 数据模型^[4,5]、查询^[6,7]和存储^[8,9]的研究已成为热点。这些研究的结

果最终将改变数据管理的模式。

本文分析和综述了 NXD 研究现状,以及几个主要研究方向:数据模型、查询、存储技术等。分析了 NXD 数据库系统的基本特性,简要介绍了具有代表性的 NXD 数据库系统。最后指出目前 NXD 研究的发展趋势。

2 NXD 的基本概念

2.1 NXD 的定义

Native XML Database 这个术语首先是于 Tamino^[10] 的市场推广活动中显露出来。除了专门用于存储 XML 文档, NXD 同样需要支持诸如:事务、安全、多用户访问、编程接口和查询语言等传统数据库的特性。XML:DB 组织对 NXD 进行了如下的定义:

1)为 XML 文档定义了一个逻辑模型,并且根据模型对 XML 文档进行存储和检索;

2)把 XML 文档作为其基本的存储单元,就像关系数据库在一个表中把行作为其基本存储单元(在逻辑概念上);

3)不需要由任何特殊的底层物理存储模型。它可以建立在关系、层次、或者面向对象的数据库之上。

上述定义表明了 NXD 是为了存储和检索 XML 文档的所有组成部分(包括元素、属性、CDATA、实体、URI 等)而设计的,这是 XED 不能完成的;NXD 存入和取出的数据格式都是 XML;NXD 底层的数据存储格式并不重要。

2.2 NXD 的体系结构

不同的 NXD 的系统体系结构也不尽相同^[10-13],但都基本采纳了传统关系数据库管理系统的成熟特性,并在其中增加了符合 XML 结构特性的部分。根据 XML 数据不同的存储形式,文[1]将体系结构分为:基于文本的 NXD 和基本模型的 NXD。这两者在数据存储层之上的部分没有本质的区别。主要区别在于前者将 XML 文件视为一种文本,强调文本之

^{*} 基金项目:国家863计划计算机软硬件技术主题资助项目(863-306-02-01)。逯 鹏 博士生。

应字段进行对应。

6)支持 XML 文档中的链接属性,包括文档内和跨文档的链接。链接是网络数据的主要特性之一。

7)支持 XML 文档的版本和时间特性。XML 文档很多时候是处于变化之中,不是静态的,应当考虑到文档的更新和版本问题。

8)支持文档集的概念,便于在多个 XML 文档中,进行跨文档和跨库的各种查询操作。

9)能够表达文档中元素的层次和顺序关系。XML 数据的层次和顺序关系是比较严格的,这与关系数据表中表项之间的顺序可以互换是不同的。

10)准确表达元素与属性的附属和语义关系。元素和属性的标记是具有一定语义的,它们之间的附属关系也是这种语义关系的扩展。

依据以上原则改进数据模型需要在简约型和复杂型之间求得一种平衡。数据模型的简约化可以方便数据库的灵活处理,但是会丢失一些数据特性,给数据库的后续处理造成结构上的缺陷,有些功能甚至无法实现;数据模型的复杂化虽然保留了更多的数据特性,但却给以后的存储、查询处理造成了技术上和效率上的问题。

4 查询技术

由于需要存储大量 XML 数据,高效的查询能力对 NXD 是非常重要的。尽管开展了许多的研究工作,相对于已经发展了近30年的关系数据查询技术,NXD 的查询技术还处于起步阶段。

NXD 的查询技术可以分为三类^[21]:1)查询语言之上的部分,如查询用户界面(Lore 系统中已经提出了一个 DataGuide 交互式用户界面)、查询数据视图、数据维护、触发器等;2)查询语言之下的部分,如查询优化技术;3)查询语言本身的设计。根据关系数据查询技术发展的经验,前两种技术是应当根据相关的标准独立于查询语言的表层语法而存在。

目前 NXD 查询的一个重要的特点是它的查询语言的设计与它的数据模型是紧密相关的。实际上每一种 XML 数据模型都隐式地决定了查询语言的结构和语义描述。例如,基于图/树结构描述和基于函数式编程描述的数据模型所关联的查询语言^[22]有着显著的不同。所以,多数的 XML 查询语言都定义了自己的 XML 数据模型。

NXD 查询形式化的理论基础是 XML 代数。它有着两方面的重要功能:一是对查询语言的语义描述;二是支持 NXD 的查询优化。目前多数 XML 代数是基于自身所采用的查询语言而提出的。通用的 NXD 系统的 XML 代数研究具有重要的意义。

4.1 查询语言

已经提出了许多面向半结构化数据和 XML 的查询语言如:StruQL, LoreL, Quilt, XML-QL, XML-GL 等。在这些研究的基础上,W3C 提出了 XML 数据的标准查询语言 XQuery^[5]。XML 数据的查询语言应该符合哪些特性;不同的研究者^[6,16]和 W3C^[7,23]都给出了一些论述。从一般查询语言的角度,NXD 的查询语言应当符合:1)语言是非过程性的,即直接定义需要查询的数据,不必定义数据获取的过程;2)独立于任何已经存在的协议;3)易于实现;4)支持嵌套查询;5)能够被优化;6)语法可以用 XML 描述;7)支持多文档查询;8)输入、输出应当以 XML 的形式。

从查询语言功能的角度,NXD 的查询应当实现:

1)选择和抽取 是传统数据库中最重要的功能。NXD 选

择和抽取的范围包括文档内容、结构和属性。文档内容是指开始标记和结束标记之间的文本。结构包括层次关系;顺序的位置关系;链接关系;文档的标记(标记是结构信息的重要组成部分);数据类型信息等。属性包括与属性的相关的名称和值。

2)重构或转换 文^[7]中认为查询必须能够转换(transformation)XML 的结构并且能够建立新的结构。NXD 中涉及的操作包括:分类排序(sort)操作;分组(Group)操作;聚集(Aggregate)操作,例如 min,max,sum,count,avg 等。

3)组合 应当能够从一个给定的文档或多 XML 文档中组合(combination)出需要的信息^[7]。

4)更新 该操作从数据维护的角度来说是每一种数据库系统的要求。由于 XML 文档的特殊结构,需要将原有的 delete, rename, insert, replace 等操作重新进行定义,使其符合 XML 数据的图/树结构。

以上功能应当独立于 XML DTD/Schema,并且能够利用已有的 DTD/Schema 进行操作。

4.2 研究方向

虽然提出了很多的面向 NXD 的查询语言,但这些语言多数是针对特定的系统提出的,语法繁杂,查询表达能力有限。理论上还没有给出全面的查询表达能力的形式化描述。

为了更好地对 XML 查询语言进行查询优化、查询计算,必须建立一个具有良好表达能力的代数系统,有相应的代数作为形式化理论的基础,这是一个重要的问题。在查询优化中可以利用代数提供的查询等价式对语句进行变换、重写,以改善和提高查询执行的性能、效率。目前缺少通用的面向 NXD 表达的查询代数系统,但针对特定系统已经提出一些有关半结构化数据的查询代数,如 YAT 集成系统中对 YATL 提出的代数、Lore 数据库系统针对 LoreL 提出的代数。

此外 NXD 查询系统的图形用户接口也很有意义,Lore 系统的 DataGuide 是这方面研究的代表。

5 存储技术

对 NXD 的数据存储来说,挑战性的问题是:要有足够的灵活性,能够存储任何的 XML 组成部分,以及 XML 文档的物理结构;而且当存储具有规则结构的 XML 数据时,和常规的数据存储一样有效^[24]。目前,哪一种存储策略更适用于 NXD 是一个有争论的问题,文^[9]对其中的几种存储方式进行了比较。目前主流的 NXD 存储主要有两种:基于已有数据库系统和基于 NXD 存储管理器。

5.1 基于已有数据库系统

这种方式的底层数据库类型采用已有的关系或面向对象数据库,例如著名的 eXist^[14],Ozone^[25],Tamino^[10]。以底层是关系型数据库的系统为例,这种方式,将一个图/树结构的 XML 文档划分成一个或者多个关系表,使用边、节点或者该文档的 ID 作为关键字。由于是基于文档的物理结构,因此不会造成数据的丢失,并且不依赖 XML DTD/Schema。XML 查询输入被转换为 SQL 查询,由底层关系数据库系统处理,查询结果由上层 NXD 的查询处理系统转换为 XML 格式。

它的最大优点是易于实现,不需要过多的底层数据处理。代价是增加了 XML 查询输入到 SQL 的转换,以及结果转换的系统开销。

5.2 基于 NXD 存储管理器

NXD 数据存储管理器致力于将上层的数据概念模型转换为底层物理数据存储模型,进行有效的 Native XML 数据管理。这方面的代表是美国斯坦福大学的 Lore 系统,德国曼海姆大学的 Natix 系统^[8]。

以 Natix 系统为例,存储管理器(SM:Storage Manager)将上层 XML 文档概念模型(树型结构)转换为物理存储模型,再通过 SM 中的记录管理器将物理树型结构转换成一系列记录,一个或多个记录组成页,磁盘读写的基本单位是页。底层的磁盘空间、缓冲区以及其它与物理存储相关问题也由 SM 负责管理。

在 SM 方式下,NXD 查询由 Natix 系统本身负责查询解析、查询优化、查询代码生成、查询执行等查询处理工作。优点是:不需要作查询转换,而且可以利用 XML 特点,优化查询性能。

5.3 研究方向

基于已有数据库系统的存储管理,由于增加了系统开销,系统性能的提高受到了一定的限制。但是,在一定的应用背景下,例如跨 XML 文档与关系表的查询,这种方式是有一定意义的。系统存储设计的重点在于如何将 XML 文档划分成结构合理的一个或多个关系表。文档划分方法有很多种,主流方法有基于索引的方法和基于路径的方法。

如果强调系统的性能,那么基于存储管理器的存储管理方式是好的选择。研究的重点在于数据概念模型到物理模型的转换,Natix 系统中提出了基于‘节点代理’的转换方法,取得了比较好的效果;另一方面的重点在于底层物理存储单元的管理,这方面的研究目标和传统数据库系统有相似之处,例如减少数据访问和搜索的 I/O 操作、提供高效的数据压缩、在全文索引和结构索引的辅助下高效的访问、树结构的平衡等。

完整的 NXD 存储管理应当具有哪些功能,有许多不同的意见^[9,16],一般认为应当满足以下几个方面:1)符合传统数据库系统存储管理的一般要求,包括并发控制、版本控制等;2)XML 文件的全部数据内容应当在存储管理范围之内;3)能够处理大数据量的文档;4)能够满足 XML 文档内容和结构的动态更新需求;5)具有标准的 API 函数,提供与其它应用程序的接口。

6 典型系统

在 NXD 系统中,Lore 系统^[13](如图2所示)是具有代表性的一种。Lore 系统大致可以分为三个部分:外部接口、查询实现、数据引擎。外部接口包括一个简单的文本接口,可以为系统开发者所用;一个图形化接口,可以作为普通用户浏览查询结果的界面;以及基于 Lore 提供的 API 所开发的应用程序。查询实现部分包括查询解析:负责将查询输入转换成一棵解析树;查询处理:负责将 LoreL 转换成 OQL 的形式;查询计

划生成:负责根据转换后的查询输入生成查询计划并传送给查询优化模块;查询优化:负责对生成的查询计划作转换,优化查询行为并将查询计划交给数据引擎部分。底层的数据引擎包括对象管理器:负责将 OEM 数据概念模型转换成底层文件物理存储模型;查询算子:主要负责查询计划的执行;工具集:包括索引管理器、DataGuide 管理等;外部数据管理模块:基于 Lore 的查询系统,动态检索查询过程中出现的外部数据,内外部数据对于用户来说是透明的。

Lore 系统是一个研究型的 NXD 系统,它的各项功能也在不断完善之中。

7 其它问题及发展趋势

作为一种 DBMS,NXD 系统还要满足如数据一致性、完整性、独立性、数据抽象、视图、控制数据冗余、数据访问控制、数据恢复、数据安全等 DBMS 特性。在满足这些特性的过程中还需要考虑 XML 文件的半结构化特性,相关的研究也正在开展之中。NXD 自身的特性需求问题在文[2,16]中有一些相关的论述,但只是初步的、概括性的,需要进行进一步的探讨。

对于 NXD 的发展趋势可以从市场、技术、理论三个方面进行概括。

从市场的观点来看首先 XML 数据在网络上不断增长,造成了 NXD 很大的市场潜力;其次,传统数据库厂商多数是基于关系型数据,虽然都在积极引进 XML 特性,但是由于遗留产品的惰性,不可能在很短的时间内作重大的技术转换工作,这给 NXD 系统的发展留下了空间;最后,具体的应用领域如内容管理系统、异构集成系统、文档管理系统等都为 NXD 走向专用领域提供了机遇。

从技术观点来看,由于底层数据模型的改变,DBMS 上层的很多技术已经不适应新的 NXD,产生了大量的技术问题。除已经论述的问题外,NXD 中的高效索引、事务管理、多用户的并发性、触发器、数据库安全性、二次开发接口、统一的数据库开发标准、不同数据库产品的兼容性都是亟待解决的问题。

从理论观点来看,20世纪70年代关系模型的提出带来了数据库领域大量的理论研究成果。NXD 的理论研究有相似之处。理论研究可以有两种视角,一种是基于 NXD 的长远发展的视角,将 NXD 系统概念化,引入新的理论研究工具和方法等;另一种是基于解答 NXD 实现中出现的重要技术问题的短期视角,利用已有的理论工具,如形式化理论、自动机理论、模型论等,为技术问题的解决提供工具和方法上的支持。

参考文献

- 1 Ronald Bourret XML and DataBase. <http://www.rpbouret.com/xml/XMLAndDatabases.htm> Jan. 2003
- 2 Suciu D. An Overview of Semistructured Data. SIGACT News, 1998,29(4)
- 3 Bourret R. XML Database Products. <http://www.rpbouret.com/xml/XMLDatabase Prods.htm> April 2003
- 4 Goldman R,McHugh J,Widom J. From Semistructured Data to XML:Migrating the Lore Data Model and Query Language. In: Proc. 2nd Int. Workshop on the Web and Databases (WebDB'99), Philadelphia, Pennsylvania 1999
- 5 <http://www.w3.org/TR.2003-05-11>
- 6 Bonifati A,Ceri S. Comparative Analysis of Five XML Query Languages. SIGMOD Record,2000,29(1)
- 7 XML Query requirements. <http://www.w3.org/TR/xmlquery-req/15-02-2001>
- 8 Carl-Christian K,Guido M. Efficient Storage of XML Data. In: Proc. IEEE Conf. on Data Engineering,2000

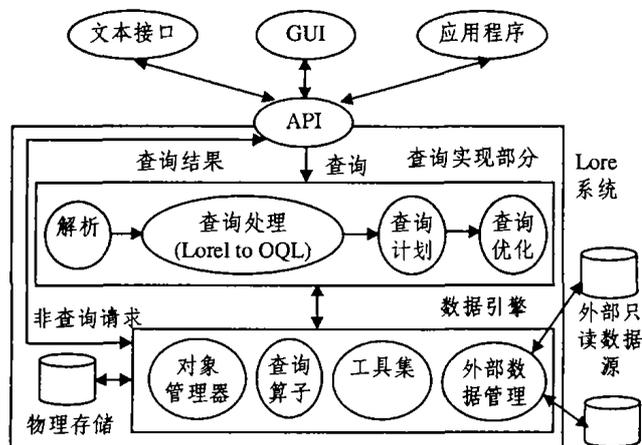


图2 Lore 系统结构图

9 Tian F, DeWitt D, Chen J, Zhang C. The Design and Performance Evaluation of Alternative XML Storage Strategies: [Technical Report]. CS Dept., University of Wisconsin, 2000

10 Schoning H. Tamino—A DBMS designed for XML. In: Proc. ICDE Conf., 2001

11 Jagadish H V, et al. Timber: A native XML database. [Technical report]. University of Michigan, April 2002

12 <http://exist.sourceforge.net/>

13 McHugh J, et al. Lore: A Database Management System for Semistructured Data. SIGMOD Record, 1997, 26(3)

14 Salminen A, Tompa F W. System Desiderata for XML Databases. In: Proc. of the 27th VLDB Conf. Roma, Italy, 2001

15 Abiteboul S. Querying semi-structured data. In: Proc. of the Int. Conf. on Database Theory, Greece, 1997

16 Calvanese D, De Giacomo G, Lenzerini M. Representing and Reasoning on XML Documents: A Description Logic Approach. J. Logic and Computation, 1999, 9(3)

17 Nestorov S, Abiteboul S, Motwani R. Extracting schema from semistructured data. In: Proc. of ACM SIGMOD Conf. on Management of Data. Seattle, WA, 1998

18 Papakonstantinou Y, Garcia-Molina H, Widom J. Object exchange

across heterogeneous information sources. In: Proc. of the Eleventh Intl. Conf. on Data Engineering, Taipei, Taiwan, March 1995

19 Fernandez M, et al. An algebra for XML query. In Foundations of Software Technology and Theoretical Computer Science, (FSTTCS), Delhi, India, 2000

20 Murata M. DTD Transformation by Patterns and Contextual Conditions. In: Proc. SGML/XML/97 Conf. 1997

21 Buneman P, et al. Beyond XML query languages. In: Proc. of the Query Language Workshop (QL'98), 1998

22 Wallace M, Ranciman C. Haskell and XML: Generic Combinators or Type-Based Translation. In: Proc. of the 4th ACM SIGPLAN Conf. on Functional Programming (ICFP. 99)

23 David Maier Department of Computer Science and Engineering Oregon Graduate Institute. Database Desiderata for an XML Query Language. <http://www.w3.org/TandS/QL/QL98/pp/maier.html> May 2003

24 Suciu D. On Database Theory and XML. SIGMOD Record, 2001, 30(3)

25 <http://ozone-db.org/ozone-main.html>

(上接第49页)

在第一个实验里,比较这些算法的时延特性,仿真蓝牙网络中,主设备与每个从设备建立一个 ON:OFF 的 TCP 连接,每个设备都有业务源发生器,它们产生信息文件长度为10个 TCP 数据包, TCP 数据包的大小为512字节。图3给出了在不同 ON/OFF 的比率的情况下,各种轮询算法建立 TCP 连接的系统吞吐量,从图上可以看出,ERR 和 DTPR 两种算法比其它算法给出了很高的吞吐量,在 ON/OFF 低比率情况(1:1),DTPR 算法比 ERR 算法更好,而使用 ERR 算法在从设备之间引起了不公平性。PRR 算法表现出最低的性能。

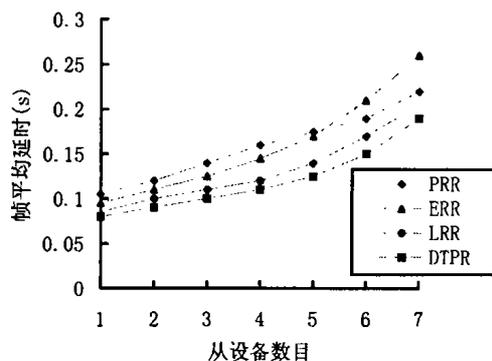


图4 在 ON/OFF 比率为 1:1 情况下, DTPR 算法与 PRR, ERR, LRR 算法帧平均时延的比较

图4给出了端到端的传输时延,结果表明, DTPR 算法比其它算法有更低的时延。并对本文提出的算法在公平性方面也做了进一步的证实,从表2可知,所有建立连接上的数据流平分带宽,也就是说, DTPR 算法对所有的数据流是公平的。

表2 双令牌随机轮询算法系统的吞吐量 (Mb/s)

从设备 连接数 \ 吞吐量	1	2	3	4	5	6	总吞 吐量
1	0.60						0.60
2	0.30	0.31					0.61
3	0.20	0.19	0.21				0.60
4	0.16	0.14	0.15	0.16			0.61
5	0.12	0.14	0.12	0.13	0.11		0.62
6	0.11	0.10	0.09	0.10	0.12	0.11	0.63

结束语 本文针对蓝牙网络中调度算法存在的问题,提出了一种改进算法——双令牌随机轮询算法,此算法解决了蓝牙网络中从设备之间的不公平性,避免了空轮询造成时隙的浪费,增加了对业务量多的从设备的调度。对现已提出的实际轮询算法与本文提出的算法进行了仿真,结果表明,该算法提高了系统的性能,可支持更高性能蓝牙产品的开发。

参考文献

1 Haartsen J. The Bluetooth radio system [J]. IEEE Personal Communications, 2000, 7(1): 28~36

2 Johansson N, Korner U, Johansson P. Performance evaluation of scheduling algorithms for Bluetooth [A]. In: Proc. of BC'99 IFIP TC 6 Fifth Intl. Conf. on Broadband Communications [C], Hong Kong: Kluwer Academic Press, 1999. 139~150

3 Zilrbes S, Stahl W, Matheus K, Haartsen J. Radio network performance of Bluetooth [A]. IEEE ICC 2000 [C], New Orleans: IEEE Computer Society Press, 2000. 1563~1567

4 Garg S, Kalia M, Shorey R. MAC scheduling policies and SAR policies for Bluetooth: a master driven TDD pico-cellular wireless system [A]. MoMuc 99[C], San Diego: Academic Press, 384~386

5 Kalia M, Bansal D, Shorey R. Data scheduling and SAR for Bluetooth MAC [A]. IEEE VTC 2000 [C], Tokyo: Princeton University Press, 716~720

6 Capone A, Kapoor R, Gerla M. Efficient Polling Schemes for Bluetooth Picocells [A]. IEEE ICC 01 [C], MIT Press, 2001. 1990~1994

7 Liron Har-Shai, Ronen Kofman, Gil Zussman, Adrian Segall. Inter-piconet scheduling in Bluetooth scatternet [A]. In: Proc. of the OPNETWORK 2002 Conference [C], Washington: ICC Press, 2002. 1~11

8 Lee Y Z, Kapoor R, Gerla M. An Efficient and Fair Polling Scheme for Bluetooth. MILCOM 2002, Berlin: Springer, 2002. 1~7

9 The Network Simulator ns-2 [EB/OL]. <http://www.isi.edu/nsnam/ns/>. 2002-12-25

10 BlueHoc, IBM Bluetooth Simulator [EB/OL]. <http://oss.software.ibm.com/developerworks/opensource/bluehoc/>. 2001-02-25