计算机网络 网络协议 安全性 (15)

67-72

计算机科学Vol.20№.1 1993

计算机网络协议的安全性与活性验证

李腊元 （430063 武汉水运工程学院）

TP393

摘 要

In this paper, protocol verification for computer networks and its main methods are described. Key properties of protocols, which include safety and liveness, are formally defined using Z notation. Taking alternating bit protocol as an example, verification problem of safety and liveness is discussed.

计算机通信协议，或简称协议，是管理网络与分布式系统中各种元素(实体与进程)之间交互与通信的一套规则。协议是并行执行的，可能借助不可靠的传送介质进行通信，因此设计正确可靠的协议，对网络与分布式系统具有特别重要的意义。传统的非形式化技术已在许多实际协议的设计和实现中获得了很大的成功，但也导致了许多设计错误。当前，协议设计及实现的形式化技术研究已成为一个非常重要的课题[3,4]，出现了一个新的领域—协议工程，它主要包括：协议及服务的形式描述，协议验证和证实，形式描述的自动生成，协议转换，性能分析，自动实现和一致性测试。本文将就其中的一个重要作业阶段——协议验证展开讨论。

一、协议验证及其主要方法

验证是指证明所描述的对象是否满足其规格说明。对于一个协议层而言，服务和协议实体是两类主要的被说明对象。协议验证必须解决两个方面的问题：其一是协议设计验证，它包括分析本层各实体之间可能要进行的交互活动，抽取规范所要求提供的各种功能并通过下一层服务的通信活动，验证这类组合操作是否满足协议设计要求和服务规范。其二是协议实现验证，主要包括每个协议实体的实现是否满足其抽象协议规范。目前通常所说的“协议验证”实际上是指上述的第一个验证问题，即协议设计验证。此外，本层协议的设计验证还与下一层协议的性能密切相关，协议将利用下一层服务的功能，必须满足其服务规范。若协议不满足服务规范，那么问题可能出在协议本身，也可能出在下一层提供的服务。一般而言，协议设计验证所涉及的协议性能应包括：①公共性质与特定性质；②安全性与活性。公共性质指所有协议所具有的共同性质。特定性质指要求提供某些特定的服务规范。安全性指确保那些不希望发生的事情（或有害的事情）不要发生（如死锁等）；活性是指确保那些有益的事情终究必将发生（如有效转移等）。

协议验证有两种形式：静态分析和动态分析。前者以规格说明文本为基础，不涉及描述的执行过程，以静态方式对协议的上下文无关语法、作用域规则、类型一致性及其它的语义进行检查，一般可由相应的形式描述编译器完成。动态分析主要针对被描述系统的“执行”特性，检测出静态分析检查不出来的某些协议错误，因此它比静态分析更加复杂。动态分析方法又可分成逻辑分析和仿真。

1．逻辑分析。主要包括以下三种形式：1）可达性分析：其基本思想是对协议实体

或进程之间的交互活动进行详尽检查和分析，主要适用于有限状态机模型，也可适用于通过设置"断点"，程序协议模型的可达性分析，经适当改进后还可适用于Petri网协议模型。对于状态机协议模型分析，先要设置一个初始状态，由此出发，经转移后进入新的状态，重复此过程，穷举所有可能的转移活动，直至没有新状态产生为止。主要分析工具有状态转移矩阵，可达图和可达树等。验证活动一般可通过有限状态机分析器，借助一定的算法或工具自动完成。该方法的主要问题之一是对于较大型复杂的协议会引起状态空间爆炸；其二是当检测到错误后缺乏再验证的自适应能力。为克服此局限性，现已建议了若干改进方法，概括起来有五种：闭合覆盖法、局部验证法、修改的可达性分析、分而治之法以及部分状态爆炸法。其基本出发点是试图尽量减小验证的时间和空间复杂度，及时检测出协议的主要设计错误，提高验证效率。2）象征性执行。主要适用于程序协议模型，其基本思想是兼有程序验证和可达性分析的特征，既要求程序变量和交互参数的断言证明，又要求对各种可能的执行分支进行分析。3）程序正确性证明。主要适用于程序协议模型，一般可采用顺序程序或并行程序正确性证明等经典技术。关于泛函、谓词演算和某些公理性模型等的证明问题尚有待进一步探讨。

逻辑分析除上述三种方法外，还有代数验证、映象、定理证明、Petri网以及时态逻辑验证等技术。

2．仿真。这种方法往往需要假设某种模拟模型，有选择地对某些可执行的分支进行验证，其所得到的结果往往是近似和不确定的。与逻辑分析相比，模拟或仿真验证可弥补逻辑分析的不足，较适于复杂的协议。

二、活性与安全性

上面所论及的协议验证方法主要适用于协议的安全性验证，即对协议的可达性、终止性、死锁、活锁和完全性进行检查；而对于活性性质，现有的大多数形式描述语言，包括ISO和CCITT提出的三种标准形式描述语言，都不能直接处理活性性质。研究表明时态逻辑可以用于活性的验证[1,2]。为了进一步理解协议的安全性与活性，我们应用Z表示法来定义和描述之。

1．安全性

```
State Safety Invariants,
Behav Safety Invariants:
System→P property
---------------------------------------
∀s: System. State Safety Invariants(s)
  =∩{st: Reachable(s)·sprops(st)}
∀s: System; P: property.
    P∈ Behav Safety Invariants(s)⇐⇒
    ∀b: s; t: Time· P ∈ bprops (pre-
fix(b, t))
```

一个安全性状态是指从不发生有害的事情。某些安全性可从系统的状态得到（即，存在于一个系统所有可达状态的性质），其它的安全性则要通过考察历史状况才能获得（即，存在于每个行为之前的某些性质）。

2．活性

```
Liveness Invariants:
System→P property
---------------------------------------
∀s: System; P: property· P ∈
    Liveness Invariants(s)⇐⇒
∀b: s· P ∈ bprops(b)
∃b: s; t: Time· P∉
    bprops(prefix (b,t))
```

一个活性状态是指使某些"好"事情能有效地发生。形式一点说，活性性质是一种行为性质，对于每个完整的行为它必须为真，否则为假。活性性质实际上对于无限和有限系统均适用。关于这一点目前国际上并没有统一认识，有的认为它只适用于无限系统，实

际上包含了无限推理过程，因而不适用于实际系统。也有的认为活性性质可为协议的某些特性提供高级的规范说明，减少实现错误。总的看来，活性验证比安全性验证更为困难，目前可用于活性验证的形式化技术，比可用于安全性验证的形式化技术少得多，而且活性验证的自动实现也比安全性验证困难得多。此外，目前尚不清楚，活性验证的形式化技术与其它形式描述技术之间的关系，以及如何组合使用才能获得最佳效果。

三、交替位协议的安全性验证

交替位协议是一种简单的数据链路协议。下面我们采用标号有限状态机对该协议进行形式描述，并讨论其安全性验证问题。所谓标号有限状态机是一个四元组〈S,i,E,T〉。其中S表示一个状态集，在本文的讨论中约定S是有限的。i表示初始状态，$i \in S$。E为表示原子事件的标号集，它可用来表示两类事件：通信事件和内部事件。前者主要是指输入、输出以及与外部环境的交互等。这些交互活动通常发生在端口。通信事件可由p? m或p! m来表征。其中p表示发生事件的端口名，m表示一份报文的值，符号? 和! 分别代表输入和输出。这与通信顺序进程的记法相类似。内部事件将由符号τ表示。T表示标号转移集，它是$S \times E \times S$的一个子集。每个转移都是一个三元组。它具体包括一个$s \in S$，一个标号$e \in E$和一个状态$s' \in S$。在标号有限状态机中，状态转移可由$s \xrightarrow{e} s'$表示，即状态s在事件e的激发下，可转移成新的状态s'。转移系统可采用图形表示。一个转移系统可包含在一个方框之内，方框边界上的椭圆表示端口，椭圆中的字符表示端口名，圆圈表示状态，有向弧表示转移，事件由有向弧旁边的字符表示，带有黑点的状态表示系统的初始状态。上述数据链路协议的设计一般要考虑在不可靠的介质上如何提供可靠的数据链路服务的问题。要想在不太可靠的传输介质上，提供可靠的传输服务，可采取某些措施，例如接收器接收到正确的信息后必须向发送器回送信息。对于可能出现的差错必须采用某些校验措施，如适当的编码技术，象CRC等。为检测报文丢失，可为每份待发报文按顺序编号。接收器接收到正确报文后，应按相应的编号回送信息。发送器在预定时间内，接收不到最后编号的回答信息时，则应按超时机制重发。在不太可靠的介质上传送报文时，回答信息也有可能被丢失，因此接收器在预定时间内收不到下一份报文时，也应重发上一次的回答信息。当收到的报文具有相同的顺序号时，发送器应考虑将当前状态增加到下一个预定的顺序号。类似地，接收器也应执行相应的操作。交替位协议最初由美国学者W.C. Lynch在一篇题为"在半双工电话线上实现可靠的全双工传输"的论文中提出。他认为发送器在发送一份报文之前，如果要求其发送的上一份报文得到接收器的回答，那么只需要两个值来编排报文的顺序号。其中一个值（1比特）用于验证位，另一个值为代替变化位，该位主要可用于报文编号。时隔一年，英国K.A. Barlet等人在他们题为"关于在半双工电话线上实现可靠的全双工传输的一点注记"的文章中提出，只需一位（在0和1之间交替）信息，就足以表达报文的顺序号。自此以后，"交替位"协议才正式得名，并被人们广泛引用，尤其是在协议的形式描述和验证时，作为例子用得很多。交替位协议的标号有限状态机描述如图1所示。

图1中左边的方框为发送器，中间的方框表示信道，右边的方框表示接收器。假设报文的类型只有一种，为了验证该协议的性质，我们不必对不同的报文去加以区别。若报文值有几个则可得到不同的状态。图1中，信道已作了一些简化，每条信道只包含描述报文收发的一个状态。在同一个转移中，已包含了两种通信事件。作了这样的简化后，状态数将大大地减少了。由于信道上不能驻留报文，因此我们把这种简化叫作空介质抽

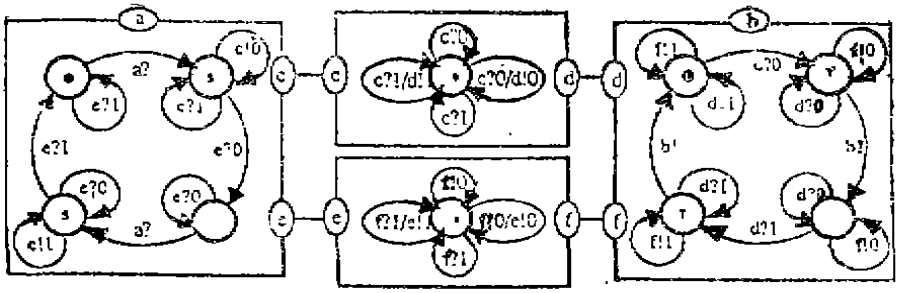

图1　交替位协议的形式描述

象。另一种简化途径则是将介质短路，即对输入和输出端口不加区别，这将不会给某些介质性质(如不可靠性、随机报文损失等)的分析带来影响。它们也可直接包含在协议实体的描述中。当发送器进入标有s的状态时，一份报文正在a端口被接收，并准备连同其顺序号一起再发送出去。当接收器进入标有r的状态时，一份报文已被接收，并准备回答信息。图1所示的交替位协议可通过并行组合过程或算法生成相应的可达图。构造并行组合的过程可作如下归纳：1)找出并行组合的事件和转移，并遵循以下算法：从初始全局状态出发，对于每个全局状态都检查其可能包含在组合中的事件，并检查可能出现的转移。2)对于每个新的全局状态将重复以上过程，直至无进一步的转移发生为止。借助以上算法，图1所示协议的可达图如图2所示。在图2中，每个状态中循环转移的标号τ代表报文丢失、重发和回答信息。为了使可达图具有可读性，与报文传输和应答相对应的内部事件，均用符号τ和所包含的端口名(cd和fe)标示，端口名后面的数字表示报文的顺序号。从图2所示的可达图分析可知，此交替位协议的标号有限状态机描述不存在没有定义的事件，且无死锁等，即该协议具有安全性。

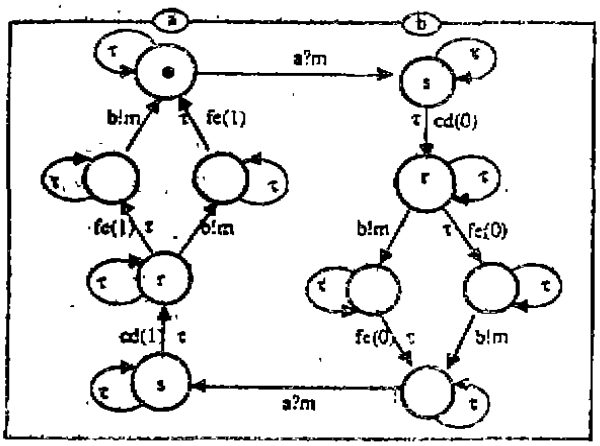

图2　可达图

四、交替位协议的活性验证

以上我们已对交替位协议的安全性进行了验证，但若要验证其活性，上述的方法已不能胜任了。下面我们讨论应用时态逻辑验证协议活性的问题。时态逻辑是一种模态逻辑，它可借助模态算子来区别不同的状态。这些模态算子可使将某个命题说成是真的。例如：现在(在当前的状态)；终止(最终将抵达的某个状态)；今后(在每个后续的状态)等等。时态逻辑有两个主要变量：转移(分支)时间和线性时间。它们之间的基本差别在于：线性时间时态逻辑是独立地考虑每个执行顺序，而转移时间时态逻辑则是同时处理所有的计算顺序。这样，在线性时间变量中，系统将由一套顺序来表示，即线性时间逻辑有一类面向跟踪的语义；转移时间则包含某些辅助信息，如非确定性选择的相关次序，其语义模型是面向树的。线性和转移时间时态逻辑各有其特点，其优劣已争论了近十年之久，但终究尚无统一明确的结论。例如，这两种时间时态逻辑之间究竟有些什么关系，哪一种逻辑最适于协议验证等等。在本

文的讨论中，我们将使用一种所谓计算树逻辑(CTL)，它以命题式转移时间时态逻辑为基础。这些命题将被看成是一些语句，其命题真假与并行组合的全局状态空间里的各状态有关。按照CTL，前述图2中有两个原子命题s，它在两个全局状态里为真；r，它在两个全局状态里为真。其命题含义是：s=一份报文在a端口被接收，并准备连同其顺序号一起被传送。r=发送器发来的一份报文已被接收，并准备回送应答信息。一般而言，此时协议所要求的性质应是：在s之后为真，r将终究变成真。CTL公式可按可达状态的相关结构来表示。每个结构均是一个四元组，即M=(S, i, P, T)。其中S表示状态集，而每个状态又是一个原子命题集。i表示初态，i∈S。p表示对每个状态而言，真原子命题的一个赋值。T表示描述可能转移S的一个二元关系。若将这种结构定义与前述转移系统进行比较，我们不难发现一个结构其实与标号有限状态机中的一张转移图相类似，但结构中的转移无标号，其状态则带有原子命题的标号。CTL公式可用于表达协议规范。我们可利用普通逻辑算子¬和∧，由原子命题组合更复杂的CTL公式。此外，CTL也可提供某些模态算子，其中包含一些常用的通道算子。值得注意的是这些CTL公式只表达通道的某些性质，而不表示状态的性质。

如果f在每个状态中，沿某通道为真，则Gf表示该通道为真；如果在某个状态中，f沿某通道为真，则Ff表示该通道为真；如果在某通道上初始状态的后续状态中，f为真，则Xf表示该通道为真；如果在某通道上g最终为真，且当在此之前f恒为真，则f∪g表示该通道为真。上述算子对于线性和转移时间模态逻辑都是适用的。为了充分利用它们，我们有以下两个状态算子：如果在自s出发的所有通道上，f为真，则Af表示在状态s中为真；如果在自s出发的某个通道上f为真，则Ef表示在状态s中为真。以上两个算子尤其适用于转移时间模态逻辑。这样一来，我们可得到以下算子：AFf表示在每条通道上f均成立，即f必然发生；EFf表示在某通道上有一使f成立的状态，即f可能为真；EGf表示在某通道上f成立；AGf表示在所有通道上的各个状态中f均成立，即f恒成立。这些算子的某些语义如图3所示，其中•=p，◦=¬p。

图3 某些简单正确性性质的CTL表示

利用上述算子，我们可根据时态逻辑公式来表示某些性质，例如

无死锁：AG¬deadlock，

有死锁：EF dealock，

一份发送的报文将被接收：AG(sent→AF received)。

为了确定在结构M中，一个公式是否为真，必须在每个状态中对它进行评判。从具有长度为1的分公式的第一级开始，然后进入具有长度为2的分公式的第二级，依此类推，逐级检查每个状态，看在这些状态中，各分公式是否均为真。此过程一般可自动实现。

现在我们应用CTL来规范和验证前述的交替位协议的有关活性。按照CTL的观点，图2中有两个原子命题s和r。前者表示来自数据链路用户的报文已被接收，并准备发送给接收器；r表示发送器发出的报文已被接收，并准备回送应答信息。我们希望证明发送一份报文和接收一份报文应严格交替；接收到的报文应与发送的报文相同。报文收发之间的交替可通过以下命题表达：

p=如果r为假，那么s在r再次为真之前必须为真。

根据上述过程，证明s和r的交替是不难的。证明接收到的报文与发送的报文相同，则需要对所提交的报文加以区别，这一类与前述

图2所示的情形有所不同。这是因为此时交替位协议中的交替位机制对用户而言已不是透明的了。由于CTL是一种命题式逻辑（只提供原子命题，而不包含变量之上的量词），因此在证明过程中，对于每一份可能的报文需要重复证明。为了简单起见，假设报文也只包含一位信息：0或1，并对图1所示的交替位协议模型作适当的扩充。为此，需引入以下四个新的原子命题：发送的报文为0：$m_s=0$；发送的报文为1：$m_s=1$；接收的报文为0：$m_r=0$；接收的报文为1：$m_r=1$。这样，我们则可得到以下CTL公式，它们可用来表示发送报文(s)与接收报文(r)严格交替，接收报文(m_r)与发送报文(m_s)相同等协议活性性质：

$$AG(r\rightarrow A[r\cup(\neg r\wedge A[\neg r\cup s])]),$$

$$AG(s\wedge m_s=0\rightarrow A[s\cup(\neg s\wedge A[\neg s\cup r\wedge m_r=0])]),$$

$$AG(s\wedge m_s=1\rightarrow A[s\cup(\neg s\wedge A[\neg s\cup r\wedge m_r=1])]).$$

其含义可描述如下：

1．如果r为真，那么在它变成假之前一直为真；一旦变假则在s为真之前一直为假。

2．如果s为真，且发送报文为0，那么s在它变成假之前将一直为真；一旦变假，则在r为真和接收的报文为0之前，s一直为假。

3．如果s为真，且发送报文为1，那么s在它变成假之前将一直为真；一旦变假，则在r为真和接收报文为1之前，s一直为假。

对于网络与分布式系统的通信协议，形式化技术是协议设计和实现工作的基础。ISO和CCITT已相继推出了三种标准形式描述技术：Estelle、LOTOS和SDL，在OSI协议开发工作中已显示出广阔的应用前景。与此同时，在协议工程方面已出现了若干值得注意的技术热点，诸如面向对象的形式描述技术（它必将对协议的形式化技术及其标准化发生新的影响）；不同形式描述风格间的变换；图形化或友善的协议人机界面（有助于现有标准形式描述技术更加易于理解，更快地得到推广应用）；计算机辅助协议开发系统研究；以及基于智能、知识等的综合协议开发系统，或自动工具与自动化系统等。在这种背景下，如何有效地将协议验证与其它作业阶段有机地结合起来，对其安全性与活性同时实现自动验证也将是值得进一步探讨的课题。

参考文献

[1] S.S. Owicki and L. Lamport, Proving Liveness Properties of Concurrent Programs, ACM TOPLAS 4(3), 1982

[2] E.M. Clarke et. al., Automatic Verification of Finite-State Concurrent Systems Using Temporal Logi Specification, ACM TOPLAS 8(2), 1986

[3] Li Layuan, A Formal Technique for Communication Protocol Specification, Proc. IEEE INFOCOM, 1989.

[4] 李腊元等，计算机局部网络，湖北科技出版社，1987.