第4卷 第5期
1993年10月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 4, No. 5
Oct. 1993

协议自动验证的可靠策略

温 涛　刘积仁　李华天

(东北工学院计算机系,沈阳 110006)

RELIABLE STRATEGY FOR AUTOMATED PROTOCOL VALIDATION

Wen Tao, Liu Jiren and Li Huatian

(*Department of Computer Science, Northeast University of Tech., Shenyang* 110006)

Abstract After Analyzing existing protocol validation techniques, we propose a FSM simplified method and an error-first search algorithm. Based on this strategy, an automated protocol validation system (APVS) in which protocols are specified in ESTELLE is developed and discussed.

摘要 本文在分析现有验证技术的基础上,提出了实体自动机化简方法和错误优先的搜索算法(EFSA),同时介绍了根据上述策略建立的以ESTELLE为描述模型的协议自动验证系统APVS.

§0. 引 言

随着协议形式化描述技术的日益成熟,网络通信协议的开发已逐渐从非形式化描述、手工实现向形式化描述及自动实现过渡.一个研究协议的形式描述、协议的验证、协议的自动实现和一致性测试的新领域——协议工程[1,3]已经形成.协议验证作为协议工程研究的一个方面,十多年来人们在此方面作了大量的工作,取得了一些较为满意的进展,虽然没有形成一致公认的方法,但它在协议开发的生命周期中所起的作用是举足轻重的.有效的验证方法和手段能使协议设计的正确性得以证明,减少后期实现的错误,达到降低协议开发费用的目的.

协议验证是对协议设计正确性的证明.它包括两个方面的内容,即:协议句法特性的验证(Validation)和协议功能特性的验证(Verification).[2,3]

Validation:是验证协议形式描述是否满足对所有协议来说都应具有的几个基本特性,也称句法特性(syntactic properties)或一般特性(general properties).

Verification:是来验证协议通信实体之间的交互动作是否满足服务描述或功能特性

本文1991年1月28日收到,1991年5月3日定稿.本课题得到国家教委博士点基金资助.作者**温涛**,讲师,1987年硕士毕业于西北工业大学,主要研究领域为计算机网络协议,CASE.**刘积仁**,教授,1987年博士毕业于东北工学院,主要研究领域为计算机网络,CASE.**李华天**,教授,主要研究领域为计算机网络协议工程学,计算机软件工具环境.

(functional properties),它基于协议的描述及其下层提供的服务.

本文所讨论的内容仅限于 Validation 的范围. 协议的 Validation 验证包含如下内容:

1. 完备性(completeness):协议必须能处理可能出现的任何情况.

2. 无死锁(deadlock freeness):每个全局状态都有它的后继状态.

3. 无活锁(livelock freeness):所有的全局状态回路都应提供有意义的通信操作.

4. 无溢出(overflow freeness):不允许协议放置更多的信息至信道以致信道溢出.

5. 终止性(termination):协议将达到一个期望的状态.

协议验证的方法依赖于协议描述所使用的模型. 目前所采用的协议形式描述技术可分为三大类:[3]1. 状态转换模型(state transition model):如 FSA,formal programmar,Petri net 等;2. 程序设计模型(programming model):抽象程序,抽象数据和时态逻辑等;3. 混合模型(hybrid model):是前两者的结合,如 ESTELLE,LOTOS 等.

根据原描述采用的模型不同,常用的分析途径有两类:一类是可达性分析,另一类是演绎推理. 可达性分析是通过对协议状态机模型的状态遍历来发现协议的设计错误;演绎推理是从协议的程序结构和数据空间来分析协议逻辑上的正确性. 本文使用并行组合自动机模型,提出了实体自动机化简方法和错误优先的搜索算法,在可达性分析的基础上交叉使用多种搜索方式,提高了对协议全局状态搜索的覆盖面,有效地解决了状态空间爆炸问题.[6,7,16]

§1. 并行组合自动机模型

一个协议可能由若干个交互通信的通信实体组成,它有别于单一的顺序系统,必须用并行模型来描述. 参于通信的任一通信实体都可用一个有限状态机来描述,其每个状态是协议实体内一组维持一定时间的变量值的集合,一个事件是从其它实体来的外部刺激,如收到一个 PDU,或者是实体内部的一个动作或发出一个 PDU;另外一个事件应是内部超时;一个事件还可以是一个外部事件和内部事件的组合,如接收到一个 PDU 或内部时钟超时后立即发出一个 PDU. 事件的发生引起某些变量的变化,并导致有限状态机状态的变迁. 通信实体 i 的有限状态机 i—FSM 是一个确定的有限状态机,可形式地定义为一个五元组:[6,7,16]

$$i\text{—}FSM=<S_i,E_i,T_i,s_{i0},F_i>$$

其中 $S_i=<s_{i0},s_{i1},s_{i2},\cdots,s_{im}>$是实体 i 的有限状态集.

$E_i=<e_{i1},e_{i2},\cdots,e_{in}>$是与实体 i 相关的事件集. 其中 $e_{ij}=+X_j/-Y_j$,$+X_j$ 表示接收一个 PDU 或 $X_j=$timeout;$-Y_j$ 表示发送一个 PDU;当 $X_j=$SPONTANEOUS,说明实体在此状态可自激. $Y_j=$NOUTPUT,说明实体在 X_j 激发下无输出.

$T_i\subset S_i\times E_i\times S_i$ 且 $T_{ij}(S_{ij},e_{ij})=S_{ik}$,　$T_{ij}\in T_i,s_{ij}\in S_i,e_{ij}\in E_i,1\leqslant k\leqslant m$

S_{i0}是实体自动机的起始状态.

$F_i\subset S_i$ 是 i—FSM 的终态集.

而整个协议机 PM 是各协议实体机的并行组合.

$$PM=\|_{i=1}^{l}i\text{—}FSM=<S,C,E,T,S_0>$$

其中 $S=S_1\times S_2\times S_3\times\cdots\times S_l$,l 是协议实体个数. C 是诸协议实体机(FSM)之间的信道状态集. E 是系统事件集. $T=\bigcup_{i=1}^{l}T_i$ 是协议机状态转换规则集. $S_0=<S_{10},S_{20},\cdots,S_{l0}>$是协议机

的初始状态.

在系统的实现过程中,我们用一个1×1的方阵来表示协议的每一个全局状态.例如,一个协议包含A、B、C、D四个通信实体,则它的每一个全局状态可表示如下.

$$\begin{bmatrix} A & A\to B & A\to C & A\to D \\ state & channel & channel & channel \\ B\to A & B & B\to C & B\to D \\ channel & state & channel & channel \\ C\to A & C\to B & C & C\to D \\ channel & channel & state & channel \\ D\to A & D\to B & D\to C & D \\ channel & channel & channel & state \end{bmatrix}$$

其中,对角线上的元素表示通信实体的状态,非对角线上的元素表示每对通信实体间的FIFO半双工信道.

并定义协议机的初始状态如下.其中,各通信实体间的信道中空.

$$\begin{bmatrix} s_{10} & null & null & null \\ null & s_{20} & null & null \\ null & null & s_{30} & null \\ null & null & null & s_{40} \end{bmatrix}$$

采用此模型可以实现对通信实体间所有可能交互动作的自动分析.从协议的初始状态开始,在外部事件或内部超时的驱动下,按照一定的状态转换规则来探索协议的所有状态,并对所扩展的状态进行分析,来发现原描述的错误,达到对协议一般特性的验证.PM可以形象化刻划一个协议的动态行为和过程,便于理解、自动实现和模拟仿真.

可达性分析能够很好地验证协议的一般特性,并能被自动实现.然而,面临的主要困难是状态空间爆炸这个瓶颈问题.这就促使人们寻求一系列的可靠策略来解决这个瓶颈问题.

§2. 协议验证的可靠策略

解决状态空间爆炸不应就事论事,应将解决策略贯穿于协议开发的始终,采用多级策略逐步解决,使问题得到弱化.对此许多研究者做了大量的工作.

1.在协议的设计阶段就应采用相应的策略.如:限制使用多值参数,规定信道的最大尺寸和错误类型,以保证协议全局状态的可枚举性.当然在此仅能对协议设计者提出建议性的意见.

2.在协议的验证之前对协议的描述进行等价变换,来减少全局状态数.

① Chow,C. H. 1985年提出了一种分解技术.[8]认为一定协议类可以分解成多个部件(或多个阶段),这些部件可以被分别验证,来保证原协议设计的正确性.

② Lam,S. S. 1984年提出了一种投影技术.[9]即为每一个要验证的协议构造一个映象协议.新的映象协议中实体状态、信息和事件是通过聚集原协议相应实体状态、信息和事件而得到的,使映象协议小于原协议.

这些技术仍有其局限性,并不能为其它模型直接采用,也不能彻底解决状态空间爆炸问题,还需要和有效的搜索算法结合.

3.有效的协议验证系统,应采用能在有限时、空下,提供对协议全局状态最大覆盖的搜

索策略.本文将现有的协议验证(PV)搜索算法分成如下四类来进行分析.

① 耗尽搜索 ES(Exhaustive Search)

```
type1()
{while(W is nonempty)
    {q=the element in W;
        if (q is error state)
          report error();
        else
        {for EACH successor state s of q
            if(s is not in A or W)
              add s to W;
        }
        delete q from W;
        add q to A;
    }
}
```

耗尽搜索适用于分析小、中等大小的协议,对于稍大的协议面临空间爆炸,无法保证协议的主要部分被分析.

② 部分搜索 PS(Partial Search)[11-13]

部分搜索的目的是力图在给定内存 M 和限定的时间 T 下,优化搜索的覆盖面,实际搜索的状态仅为协议状态全集 R 的一个子集 CLOSED.部分搜索与耗尽搜索仅差一点,即将耗尽搜索中的 EACH 改成 SOME 即可.

③ 深度优先搜索 DFS(Depth First Search)[13]

深度优先搜索算法见文献[13].它也能得到状态空间的全覆盖,要比耗尽搜索节省空间,但这是以时间耗费为代价的.

④ 随机漫步搜索 RWS(Random Walk Search)[14,15]

随机漫步搜索算法详见文献[14,15].此算法的后继选择可以是随机的,或是选择可能致错的状态.理论上讲,此搜索算法可以覆盖整个状态空间,但所用的时间将趋于无穷,而且算法每运行一次只能发现一个错误状态.

可以清楚地看到:如果 S(一个状态所占用的空间)和 R(状态全集)较小时,ES 是最好的选择,它可以给出一个协议设计正确性的完全证明.如果 S 接近于 M(用户空间)时,只能选择第四类搜索算法.ES 和 RWS 不适合于问题求解.在时间可忍受的情况下,可采用 DFS 算法,它较 ES 算法节省空间.目前,人们的注意力主要集中在 PS 算法上,但笔者发现 PS 算法也有不足之处.1)当一个给定的协议属中、小协议时,采用 PS 算法则得到一个不完全覆盖搜索;2)无论协议的大小,剪枝(pruning)不能保证剪去的部分不含有错误,即是给出不含错误的概率也是困难的;3)启发信息是完全针对当前扩展状态,是静态的.

分析现有的 PV 搜索算法,发现 PV 的搜索算法和 AI 中面向问题求解的搜索算法有如下几方面的不同:

1.目标不同:AI 的搜索目标是要在有限的时、空下,尽快地求得问题的解;而 PV 的搜索算法是在有限的时、空下,提供对协议状态空间的最大覆盖(coveragy),尽可能地发现错

误.

2. 在下步扩展状态方面:AI 搜索是扩展距问题求解目标最近的状态;而 PV 搜索是要扩展容易致错的状态.

3. 在全扩展和部分扩展方面:AI 采用剪枝来减小搜索波的宽度;PV 剪枝有可能正好剪掉了一棵含有设计错误的状态子树.

4. 在终止性方面:AI 搜索在求得一个最优解或局部最优解或无解情况下终止算法;而 PV 搜索应在探索了协议全部全局状态或最大程度利用系统提供的空间情况下终止搜索.

5. 搜索方向:PV 是面向错误的;AI 搜索是面向问题求解的.

因此,在选择 PV 的搜索策略时,不能简单地挪用 AI 的搜索算法.

§ 3. FSM 化简和错误优先的搜索算法

基于描述实体自动机(FSM)的文法形式:

<When From Action To Output>即在 From 状态的实体自动机,当 When 条件满足时,将执行动作 Action,转移到 To 状态,且输出 Output. 本文提出了实体自动机化简方法,并构造了实体自动机化简器,来减少每个实体自动机的状态数. 在实体机模型中存在大量的自激无输出的规则,即有大量的 e_{ij}=SPONTANEOUS/NOUTPUT 事件及相应的内部状态,而这些事件的发生和实体状态的跃迁在逻辑上并不影响其它与其有通信关系的实体,故 FSM 化简器在证明了这些内部状态无死锁和活锁的情况下,可以自动删除和拼接原规则,来减少实体机的状态数和规则数,并保证一般特性不受影响. 例如,对一给定的通信实体机描述中的一段如图1,经 FSM 化简器化简后,可得到图2中的结果.

```
+<SN2UNITDATA indication INITIAL ipsprd011 REASSEMBLING NOUTPUT>
+<SN1UNITDATA indication REASSEMBLING ipsprd012 Int037 NOUTPUT>
+<SPONTANEOUS Int037 ipsprd012 Int038 NOUTPUT>
+<SPONTANEOUS Int037 ipsprd012 Int039 NOUTPUT>
+<SPONTANEOUS Int039 ipsprd012 Int041 NOUTPUT>
+<SPONTANEOUS Int041 ipsprd012 Int042 NOUTPUT>
+<SPONTANEOUS Int041 ipsprd012 Int043 NOUTPUT>
+<SPONTANEOUS Int043 ipsprd012 Int044 SN1UNITDATA request>
+<SPONTANEOUS Int043 ipsprd012 Int044 SN2UNITDATA request>
+<SPONTANEOUS Int044 ipsprd012 Int042 NOUTPUT>
+<SPONTANEOUS Int042 ipsprd012 Int040 NOUTPUT>
+<SPONTANEOUS Int039 ipsprd012 Int040 NOUTPUT>
+<SPONTANEOUS Int040 ipsprd012 Int038 NOUTPUT>
+<SPONTANEOUS Int038 ipsprd012 REASSEMBLING NOUTPUT>
+<SN2UNITDATA indication REASSEMBLING ipsprd013 Int045 NOUTPUT>
```

图1 化简前的一段 FSM 描述

```
+<SN2UNITDATA indication INITIAL ipsprd011 REASSEMBLING NOUTPUT>
+<SN1UNITDATA indication REASSEMBLING ipsprd012 Int037 NOUTPUT>
+<SPONTANEOUS Int037 ipsprd012 REASSEMBLING NOUTPUT>
+<SPONTANEOUS Int037 ipsprd012 Int044 SN1UNITDATA request>
```

+<SPONTANEOUS Int037 ipsprd012 Int044 SN2UNITDATA request>
+<SPONTANEOUS Int044 ipsprd012 REASSEMBLING NOUTPUT>
+<SN2UNITDATA indication REASSEMBLING ipsprd013 Int045 NOUTPUT>
(Error First Search)

图2　化简后的结果

在分析已有的PV搜索算法的基础上,本文提出了一种改进的错误优先搜索算法(Error First Search Algorithm),来克服上述PV搜索算法的不足.

本算法在速度上等同于耗尽搜索算法和部分搜索算法,实际运行时,生成一个全局状态的时间为$O(10^{-2})$秒.在存储空间上与部分搜索算法相同.但本算法有如下的优点:

```
Error First Search Algorithm()
  {while(W is nonmepty)
   {q=the element of highest priority from W;
    if(q is error state)
      report error();
    else
    { for some new successor state s of q
      {memory fired transition number;
       if(s is not in A or W)
       add s to w;
      }
    }
    if(q is full expended)
      {delete q from W;
      add q to A;
    }
  }
```

1. 在验证时,对于中、小协议,如果M>S×R时,本算法成为错误优先的耗尽搜索算法,它能较耗尽搜索更早地发现错误.

2. 错误优先搜索算法克服了部分搜索算法由于剪枝而带来的不足,并且将部分扩展状态仍保留在OPEN中,使启发信息趋于动态化.

3. 系统实现时,使EFSA在三种不同启发信息的引导下,对协议全局状态空间进行三次搜索,增大了对全局状态空间的覆盖面,从而较大程度上证明了协议的正确性.

§4. 协议自动验证系统APVS的实现

协议自动验证系统APVS的总体结构如图3所示.在APVS中选择任一分析方式均能发现死锁、非定义接受、溢出以及活锁等错误,并证明协议的终止性.APVS是在SUN-Workstation上用C语言实现的.在APVS实现时,EFSA使用了如下两方面的启发信息:

1. 在决定扩展哪一状态时,使用的启发信息:

1)死锁错误:检查所有空队列,目标实体正处在接受状态的空队列数记为N_1,不处在接受状态的空队列数记为N_2,该状态可激发规则数记为N_3,求出N_1、N_2和$1/N_3$的加权和,值最大者优先扩展.

2)非定义接受:满足下面两个条件的非空队列数记为U_1:a)队列非空;b)目标实体正处在接受状态.可激发的规则数记为N_3,则具有U_1+k/N_3(其中k是正常数)最大值的状态优先扩展.

3)溢出错误:具有最长信息队列的状态优先扩展.

2. 在决定激发哪一个转换规则时使用的启发信息:

1)死锁错误:接受操作总是优先考虑,即"R">"S";在"R"操作中从最短队列中抽取信

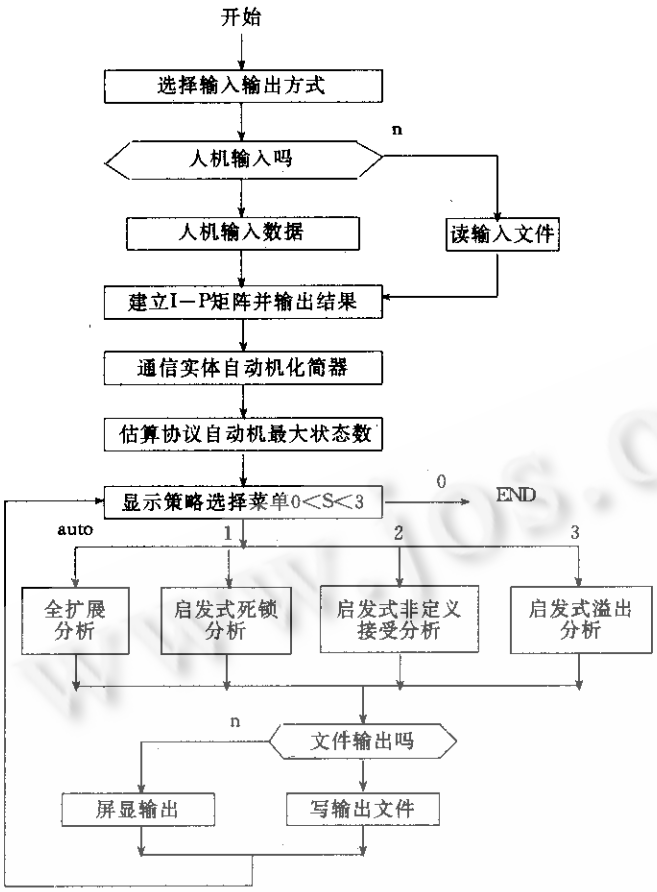

图3 协议自动验证系统总体结构

发现错误的能力.

息的“R”操作优先考虑.

2)非定义接受:选从最长队列中接受信息“R”操作,或向空队列发信息的“S”操作,对于其它情况,“R”操作优先考虑.

3)溢出错误:“S”优先考虑.向最长队列发信的“S”操作优先级最高.如没有“S”操作,从最短队列中抽取信息的“R”操作优先考虑.

为了证明 APVS 发现错误和对协议全局状态的覆盖能力,我们在 Encryption 协议中设置了几个错误,[17]分别用1,2和3种搜索方式来发现错误状态,并以文件形式给出初始状态到错误状态的路径.表 1 给出了对 Encryption 协议的验证结果.从表 1 的数据中可看出 EFSA 所用启发信息的有效性;三种搜索方式的结合使用扩大了对全局状态的覆盖面,在一定的时、空限制下,提高了系统

表1 Encryption 协议的验证结果

错误类型	搜索方式	信道模型	最大信道尺寸	搜索道路数	探索的可区分状态数	全扩展状态数
死锁	1	FIFO	2	1226	472	260
	2	FIFO	2	3003	876	464
	3	FIFO	2	3050	1285	562
非定接受	1	FIFO	2	1502	528	302
	2	FIFO	2	3006	932	471
	3	FIFO	2	3001	1350	463
信道溢出	1	FIFO	2	3004	1438	82
	2	FIFO	2	3003	947	436
	3	FIFO	2	3001	2043	102

结语:本文在分析现有协议验证策略的基础上,根据东北工学院的协议开发环境,提出了面向ESTELLE描述协议的多级验证策略,通过化简协议实体自动机,将原自动机的规则数和状态数减少了一个数量级,为解决状态空间爆炸提供了一定程度的保证.多目标搜索方式的结合使用增大了对状态空间的覆盖,通过对 Encryption 协议和 NEUT－TP4传送层协议的验证,运行结果表明 APVS 所用策略是可靠的和有效的.

参考文献

1 Piatkowski, T. F., An Engineering Discipline for Distributed Protocol System, Proc. IFIP Workshop on Protocol Testing Towards Proof? 1981.

2 Sunshine, C. A, Formal Techniques for Protocol Specification and Verification, IEEE Computer Magazine, 12(9).

3 Liu, M. T., Protocol Engineering, Computer Advance, 1989.

4 West, C. H., Generalized Technique for Communication Protocol Validation, IBM Research and Development, 22 (4).

5 West, C. H., An Automated Technique of Communication Protocol Validation, IEEE Trans. Comm. COM－26 (8).

6 ISO/DIS/9704 ESTELLE, A Formal Description Technique Based on an Extended State Transition Model, 1987.

7 Bjorn Pehrson, Protocol Verification for OSI Computer Network and ISDN System, 18(3), 1990.

8 Chow, C. H, Gouda, M. G. and Lam, S. S., On Constructing Multi Phase Communication Protocols, Proc. IPIP Int. Workshop on PSTV, 4th.

9 Lam, S. S. et al., Protocol Verification Via Projections, IEEE Trans. Software Engineering SE－10(4).

10 Sabnani, K. K. and Lapone, A. M., PAV－Protocol Analyzer and Verifier, Proc. IFIP Int Workshop on PSTV 6th.

11 Holzmann, G. G., Tracing Protocols, AT&T Technical Journal, Vol. 64, No. 10.

12 Maxemchuch, N. f and Sanani, K. K., Probilistic Verification of Communication Protocols, Proc. Int Workshop on PSTV 7th, 1987.

13 Holzmann, G. J, A Theory for Protocol Verification, IEEE Trans. on Comput. Vol. C－31, No. 8, 1982.

14 West, C. H., Protocol Verification by Random State Exploration, Proc. Int Workshop on PSTV 6th, 1986.

15 West, C. H., Protocol Validation in Complex Systems, Proc. of SIGCOMM'89.

16 Blumer, T. P. and Sidhu, D. P., Mechanical Verification and Automatic Implementation of Communication Protocols, IEEE Trans. on Software SE－12, No. 8, 1986.

17 Chung, A and Sidhu, D. P. and Blumer, T. P., Automated Validation of Protocol Using EDS, Proc. IFIP Int Workshop on PSTV 8th, 1988.