

# 用有穷状态自动机 描述计算机网络的传输层协议

◆ 中国人民银行深圳市中心支行 刘里鹏

## 一、有穷状态自动机

实际的计算机是相当复杂的,很难直接对它建立一个容易处理的数学模型,于是人们采用计算模型的理想计算机,来开展对计算机特性的研究。有穷状态自动机是比较简单的一种计算模型。

### 1. 有穷状态自动机模型

有穷状态自动机是一个存储容量极其有限的计算机模型,是接收语言和生成语言的装置。在处理语言的过程中,装置的状态发生变化。它接收输入,输入是一个字符串,并且被传送到输入带上。它没有输出,只给出是否接收输入的字符串。装置每隔一定时间自动从输入带上读入一个符号,然后进入一个新的状态。新的状态的生成只与当前状态和刚刚读到的符号有关。在读一个字符后读头在输入带上向右移动一格,准备读下一个字符。重复这样的动作,读一个字符,读头向右移动,并且改变装置的状态。

这样的装置,我们称为有穷状态自动机。

### 2. 对有穷状态自动机的理解

有穷状态自动机是一个理想的模型,它最大的特点是具有有限的存储容量。一般来讲,在现实中的一切都是有容量限制的,极限情况下,不能超过宇宙的限制。

在现实生活中,有穷状态自动机模型可以说无处不在。例如,商场或宾馆的自动门的控制器、电梯门的控制器,以及家用电器的控制器等,它们的核心部件控制器都可以用有穷状态自动机模型来描

述,有穷状态自动机模型的用途还是比较广泛的。本文试图将有穷状态自动机用于描述计算机网络的传输层协议。

### 3. 有穷状态自动机的形式定义

对有穷状态自动机有了初步的理解之后,可以对它进行形式定义。进行形式定义有两方面的理由:

(1)形式定义是精确无误的。形式定义是精确、严谨的,深入到每一个细节,可以消除有穷状态自动机中任何不明确的疑点。

(2)提供一种表示方法。良好的表示方法有助于人们的思考和清楚地表达人们的思想。

形式定义将一台有穷状态自动机描述成5个部分的表:状态集、输入字母表、动作规则、初始状态和接受状态集。用数学语言表示,5个元素的表叫做5元素组。

有穷自动机是一个5元组 $(Q, \Sigma, \delta, q_0, F)$ ,其中

(1) $Q$ 是一个有穷集合,叫做状态集合。

(2) $\Sigma$ 是一个有穷集合,叫做字母表。

(3) $\delta: Q \times \Sigma \rightarrow Q$ 是一个转移函数。可以用 $f: Q \times \Sigma \rightarrow Q$ 表示, $Q \times \Sigma$ 意味着函数的定义域,是状态与字母表元素有序对集合(笛卡儿积)。 $Q$ 为函数的值域。

(4) $q_0 \in Q$ 是起始状态。

(5) $F \subseteq Q$ 是接受状态集合。接受状态,也可称为终结状态。

有了这个形式定义,就可以使用定义的数学记号来描述具体的有穷自动机了。

#### 4. 如何设计有穷状态自动机

《计算机理论导引》(见参考文献1)中给出了一种设计有穷状态自动机的方法,即把你自己的位置放在要设计的机器的位置上,体会如何去实现机器的任务,用“读者即自动机”的方法去设计有穷状态自动机。

笔者认为,设计有穷状态自动机的关键是状态转换函数 $\delta$ 。因为,在实际应用中,我们关心的是具体实例的状态的变化,关心的是状态发生转换的条件和下一个状态是什么。使用形式定义的好处是,可以精确地定义状态的边界情况,去掉不会发生的状态,以及避免同一个条件产生两个或多个状态等情况。

### 二、计算机网络的传输层协议

#### 1. 网络协议

网络的本质是什么?网络的本质就是协议。协议定义了网络上各种计算机和设备之间的相互通信、数据管理、数据交换的一整套规则。通过这些规则,网络上的计算机才有了彼此通信的“共同语言”。

任何一种协议都需要解决三方面的问题:

##### (1) 协议的语法(如何讲)问题

协议定义了如何进行通信的问题,即对通信双方采用的数据格式、编码等进行定义。例如,报文中内容的组织形式、报文中内容的顺序、形式等。

##### (2) 协议的语义(讲什么)问题

协议应解决在什么层次上定义了通信,其内容是什么,即对发出请求、执行的动作,以及对方的应答做出解释。例如,对于报文,它由哪些部分组成,哪些部分用于控制数据,哪些部分用于真正的通信内容。

##### (3) 协议的定时(讲话次序)问题

定时协议定义了什么时候进行通信、先讲什么后讲什么、讲话的速度等。例如,采用同步传输还是异步传输。

总之,协议必须在解决好语义、语法和定时这3部分问题之后,才算比较完整地构成了数据通信的“语言”。语义、语法和定时,称为网络协议的3要素。

#### 2. 网络的传输层协议

国际标准化组织(ISO)的OSI参考模型,将网络分成7层。其中传输层恰好位于中间层,传输层的上层是会话层,传输层的下层是网络层。传输层的基本功能是从上层接收数据,必要时将数据分成较小的单元传给下层,并保证到达对方的各段信息正确无误。从某种意义上讲,传输层使会话层不受硬件技术变化的影响。

传输层是整个协议层次结构的核心,其任务是从源端机到目标机提供可靠的数据传输,并使之与具体的网络无关。如果没有传输层,整个分层协议的概念就失去意义。

在一般情况下,网络传输层协议可以存在于主机的操作系统中,是操作系统的一部分,或者是一组运行在用户本地的进程,或者在主板的网络适配器(网卡)上。

传输层的最终目标是为传送服务用户提供有效、可靠的传输服务。依据ISO/OSI的思想,传输层充分利用网络层提供的服务,来达到这一目的。在传输层中完成这一任务的硬件或软件,被称为传输层实体。

### 三、用有穷状态自动机描述传输层协议

实际的网络协议,以及实现这些协议的程序通常是十分复杂的,因此,人们进行了许多深入的研究,找到一些形式化的、数学的方法来描述和验证这些协议,来加深对协议的理解。

定义:

协议机:通信过程中,协议的发送方和接收方可以统称为协议机,在通信时的每一时刻,协议机总是处在一个特定的状态,这时的状态是由所有变量值和一些机构组成的。当某些事件发生时,会引起状态转移。

对于一个协议机,当发送一帧、接收一帧、计时器超时或产生一个中断等事件时,都可以引起状态的转移。如果对协议机有了完整的描述,就可以把所有状态作为节点,所有状态转换作为方向弧,画出有向状态图。有一个状态被称为初始状态,这个状态对应于协议机的开始运行时的状态。从初始状态通过一个或多个状态转换,协议机可以到达某些

或全部状态。

### 1. 事件说明

在一个典型的应用中,如一台 Server 和多台 Client,影响协议机状态转换的因素有原语、处理分组和定时器。

(1) 原语:面向连接的传输层服务,实际是通过传输原语来完成的。

原语	TPDU 发送的信息	含义
LISTEN	(无)	进入阻塞状态,等待某个过程试图建立连接
CONNECT	CONNECTION REQ	建立一个连接
SEND	DATA	发送信息
RECEIVE	(无)	进入阻塞状态,直到一个 DATA TPDU 到达
DISCONNECT	DISCONNECTION REQ	释放连接

侦听(LISTEN)原语:侦听客户机的请求,若没有客户服务请求出现,则阻塞服务器,直到一客户服务请求的出现。

连接(CONNECT)原语:当客户试图与服务器对话时,便执行连接原语。

客户的连接请求 TPDU(传输协议数据单元, transport protocol data unit)被传送到服务器,将检查服务器是否阻塞在侦听状态,若是,则唤醒服务器并向发出请连接的客户回送一个连接,当 TPDU 到达后,客户被唤醒,连接便建立起来。

发送(SEND)原语:用于发送数据。

接收(RECEIVE)原语:用于接收数据。

建立连接的任何一方可以执行一条 RECEIVE 原语,阻塞本机、等待对方执行 SEND 原语。当 TPDU 到达后,接收方解除阻塞,处理接收的 TPDU,并发送应答信息。要求已经建立连接的双方在通信时保持收发的协调一致。

释放(DISCONNECT)连接:当一个连接不再需要时,必须断开连接释放占用的资源。释放连接可以有两种方式:对称与非对称释放,本文不再具体阐述。

### (2) 进入分组

分组	含义
CALL REQUEST	发送建立连接的请求
CALL ACCEPTED	对 CALL REQUEST 的响应
CLEAR REQUEST	发送释放连接的请求
CLEAR CONFIRMATION	对 CLEAR REQUEST 的响应
DATA	用于传输数据
CREDIT	用于管理滑动窗口的控制分组

### (3) 定时器

定时器超时,会影导致协议机的状态的转换。

归纳起来:

原语:

E1:LISTEN

E2:CONNECT

E3:DISCONNECT

E4:SEND

E5:RECEIVE

分组:

E6:Call - req

E7:Call - acc

E8:Clear - req

E9:Clear - conf

E10:Data

E11:Credit,信用

定时器:

E12:TimeOut,超时

### 2. 状态说明

定义协议机的状态,主要有以下几种:

Q1(空闲/休眠):当用户调用 CONNECT 原语时,网络层便向接收方发送一个 CALL REQUEST 分组,而用户本身将进入空闲/休眠状态。

Q2(等待):当 CALL REQUEST 分组到达接收方后,接收方的传输实体被中断,检查是否有本地用户正在对指定的地址进行侦听,若没有,则将 CALL REQUEST 分组防入等待队列并启动定时时钟。若有,则发送一个 CALL ACCEPTED 分组,远端的用户被唤醒(既开始提出连接请求的用户)。

Q3(排队):CALL REQUEST 已经到达,但还未执行 LISTEN 原语。

Q4(建立):在时限内有一个用户执行了 LISTEN 原语,则建立了一个连接。

Q5(发送):用户正在等待发送数据分组许可。

Q6(接收):已经执行了 RECEIVE 原语。

Q7(断开):本地执行了 DISCONNECT 原语。

### 3. 传输层协议有限自动机的定义

把传输层的有限自动机的形式化定义为 4 元组模型  $(Q, \Sigma, \delta, q_0)$ , 其中

Q: 是协议机可能进入的状态集合,  $\{Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7\}$ 。

$\Sigma$ : 是引起协议机状态变化的事件集合,  $\{E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12\}$ 。

$\delta: Q \times \Sigma$ , 是一个转移函数。 $Q \times \Sigma$  意味着转移函数的定义域, 是状态与引起协议机状态变化的事件的有序对集合(笛卡儿积)。 $Q$  为转移函数的值域。

$q_0$ : 是协议机的初始状态,  $Q1$ 。

#### 4. 状态转移矩阵

矩阵中每一项可以包括 3 个内容: 断言、动作和新状态。

断言有:

P1: 连接的表空间满

P2: Call - req 进行

P3: LISTEN 进行

P4: Call - req 进行

P5: Credit 可用

动作有:

A1(发送 Call - acc)

A2(等待 Call - acc)

A3(发送 Call - req)

A4(启动计时器)

A5(发送 Clear - conf)

A6(发送 Clear - req)

A7(发送消息)

A8(等待信用)

A9(发送信用)

A10(设置 Cir - req - received 标志)

A11(记录信用)

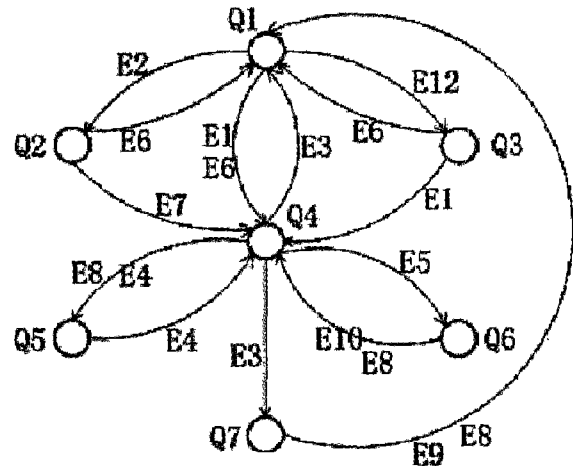
A12(接收消息)

[说明]:

(1)  $\neg$ : 表示断言的否定, 如  $\neg P1$  是  $P1$  的否定。

(2)  $-$ : 表示没有执行的动作。

#### 5. 有限自动机的状态图



#### 四、几点体会

1. 图灵机比有穷状态自动机有更多的通用性, 是否可用图灵机来描述传输协议呢?

2. 有限状态自动机的状态图, 有助于直观理解状态的变化。

3. 有限状态自动机的形式化定义, 可以使状态的变化更加严谨。

4. 有限状态自动机的关节点, 在状态转换函数。HNFC

#### 参考文献

1. (美) Michael Sipser 著(麻省理工学院), 计算理论导引, 张立昂, 王捍贫, 黄雄 译北京: 机械工业出版社, 2000. 2

2. (美) Harry R. Lewis, Christos H. Papadimitriou 著, 计算理论基础(第 2 版), 张立昂, 刘田 译, 北京: 清化大学出版社, 2000. 7

3. 李增智, 计算机网络原理, 陕西: 西安交通大学出版社, 1998. 3

4. Andrew S. Tanenbaum(美), 计算机网络, 熊桂喜, 王小虎 译, 李学农 审, 北京: 清化大学出版社, 2000. 2

状态	Q1(空闲)	Q2(等待)	Q3(排队)	Q4(建立)	Q5(发送)	Q6(接收)	Q7(断开)
事件							
E1(LISTEN)	P1: - / 空闲 P2: A1 / 建立 P2: A2 / 空闲		- / 建立				
E2(CONNECT)	P1: - / 空闲 P1: A3 / 等待						
E3(DISCONNECT)				P4: A5 / 空闲 P4: A6 / 断开			
E4(SEND)				P5: A7 / 建立 P5: A8 / 发送			
E5(RECEIVE)				A9 / 接收者			
E6(Call - req)	P3: A1 / 建立 P1: A4 / 排队						
E7(Call - acc)		- / 建立					
E8(Clear - req)		- / 空闲		A10 / 建立	A10 / 建立	A10 / 建立	- / 空闲
E9(Clear - conf)							- / 空闲
E10(Data)						A12 / 建立	
E11(Credit)				A11 / 建立	A7 / 建立		
E12(TimeOut)			- / 空闲				