The Part-Time Parliament

LESLIE LAMPORT

Digital Equipment Corporation

最近在Paxos岛上的考古学发现揭示了尽管立法者们都很逍遥，但是议会仍然运作。立法者们维护着议会记录的一致的副本，尽管他们经常不在室内而且他们的传信者也总是健忘。Paxos的议会协议提供了一种新的状态机实现去设计一个分布式系统。

关键词：计算机网络交互，分布式系统，操作系统，行政数据处理

最近在TOCS社论办公室的文件柜后发现了这个提案。尽管因为它年龄的问题，但是主编认为这个是值得出版的。这位作者现在正在希腊岛做相关的工作，所以我被要求准备这次出版。

作者似乎是一位只对计算机科学感兴趣的考古学家…（无关紧要的内容）

正文：

1. The Problem

1.1 The Island of Paxos

在这个千年的初期，爱琴海上的Paxos岛是一个蓬勃发展的商业中心。财富导致政治复杂化，Paxos的人用政府的议会取代了民主制度。但是岛上的公民在贸易面前，没人愿意一直待在议会。所以Paxos的议会需要一个好的处理方法，这样即使议员不停地进出议会厅也能够对事务进行处理。

兼职的议会的问题是如何出色的解决对应的问题以面对如今的容错分布式系统，即使有议员处理对应任务但是部分议员不在议会导致失败。Paxos的解决方案可能因此成为计算机科学中的一部分。我在这里的Paxos议会协议的一小段描述，是为了引出分布式系统的讨论。

Paxos文明被外来侵略者所破坏，考古学家最近才开始挖掘这段历史。我们关于Paxos议会的了解因此而模糊。虽然我们知道基本协议，但是我们还缺少很多细节。这些细节很有趣，我会自由的想象议会是怎么做的。

1.2 Requirements

议会的主要任务是决定岛的法律，这个由通过的法律序列定义的。现代的议会会雇佣秘书来记录行为，但是在Paxos没人愿意停留在办公室始终作为一个秘书。每个Paxos议员维护了一份记录着通过的法律序号的账本。比如，议员λ的账本有这么一句：

155: The olive tax is 3 drachmas per ton

如果她相信155号法令被议会通过。账本会由不可消除的墨水书写，并且他们的不会被修改。

议会协议的第一个要求是账本的一致性，也就意味着两份账本不能有冲突的信息。如果议员Φ的账本中有

132: Lamps must use only olive oil

就不能有其他议员在132这条规则上存在不同。然而，其他的议员可能在他的账本上没有132这一条法令，如果他没有学习过已经通过的法令。

只有账本的一致性是不够的，因为可以简单的使账本为空白。还需要一些要求用以保证法令会被最终通过并记录到账本上。在现代议会中，通过法令时会被不同意的部分议员阻扰。Paxos的情况并非如此，议员之间相互信任。Paxos的议员愿意通过被提出的任意提案。然而，他们的周期倾向造成了一个问题。如果一个议员团体通过法令

37: Painting on temple walls is forbidden

并举行宴会，不管另一个议员团体是否在议员办公室，并对他们刚做的决定一无所知，又刚好通过了一个冲突的法令

37: Freedom of artistic expression is guaranteed

此时，一致性就会丢失。

进展不能被保证除非有足够的议员在一段足够长的时间内待在议员办公室。因为Paxos的议员不愿意减少他们外出的活动，这不能够完全确定任何法令一定能被通过。然而，议员们愿意保证，在议员办公室的议员，他们和他们的助手会及时的通知所有的议会相关事宜。这个保证允许Paxos人们设计一个议会协议满足下面进展条件。

如果大部分的议员在办公室并且在足够长的一段时间内没有人进入或离开，办公室中议员提出的任何法令都会被通过，并且每个通过的法令都会被记录到每个在办公室的议员的账本上。

1.3 Assumptions

议会协议的要求只有当议员被提供足够的资源时才能实现。每一个议员收到一本包含法令记录的账本，一支笔和一盒不会消失的墨水。议员可能在离开办公室后忘了他们做了什么，所以他们会写笔记在账本后面以提醒自己一些重要的议会任务。一个列表中的法令实体永远不会被改变，但是笔记可以被划掉。达到这个进展的条件要求议员可以测量时间的流逝，所以有一个简单的沙漏。

议员随身携带着他们的账本，可以一直阅读法令清单和未被划掉的笔记。账本由最好的羊皮纸做成，只写重要的笔记。一个议员可以写其他笔记在纸上，这些纸在他离开办公室后可能丢失。

办公室内的声音很轻，不能做演讲。议员可以通过信差交流，并且有足够的金钱雇佣很多的信差。信差不能修改内容，但是他可能会忘了他已经传过消息，然后再传一遍。就像议员一样，信差也是兼职的。信差可能离开办公室去度假。也可能永远离开，也就是让他传的消息永远不会被送到。

尽管议员和信差可以在任意时间进入和离开，当他们在办公室内，他们会只专注于工作。当他们留在房间内时，信差迅速分发消息给其他议员，并且议员也会最快时间内处理并回复他们收到的信息。

Paxos官方记录显示要求议员和信差完全的诚信和遵守议会协议。大部分学者宣传这些内容，为了显示Paxos的人在道德上更好。不诚实的说，虽然罕见，但确实存在。然而，因为不是在官方文档中提到，所以我们对议会如何应对不诚实的议员和信差的手段知道的不多。这些被揭露的证据在3.3.5节会被提到。

2. The Single-Decree Synod

Paxos议会是从每19年召开的早期祭祀仪式演变而来，以选择一个单独的象征性的法令。数百年来，会议根据常规程序选择了一个个法令，这些程序要求所有的牧师都在场。但是由于商业的蓬勃发展，当会议在仪式中时，牧师开始不停的徘徊于室内外。最后，旧协议失败了，会议在结束时没有选择任何法令。为了避免重复神学的困难，Paxos的宗教领导要求数学家制定一个协议可以选择会议的法令。这个协议的要求和假设实质上是和后来的议会期望一致的，也就是包含一个法令的序列号，一个账本对应的最多有一个法令。会议协议的描述在本节中，议会的协议将在第3节中描述。

数学家们通过多步派生出会议协议。首先，他们证明结果表明协议满足一些约束能够保证一致性并允许进度。一个初步的协议在这些限制下衍生了出来。受限版的初步协议提供了基础协议保证一致性，但不包括进度。完整的会议协议，满足一致性和进度要求，是通过对基础协议的再限制得到的。

数学结果在2.1节描述，协议非正式的描述在2.2-2.4节。对基础协议的正式描述和正确证明在附录中。

2.1 Mathematical Results

会议的法令通过一系列编号的选票，每张选票只能投给一个法令。每个牧师只能选择投或不投这张选票。和选票相关的是叫法定人数的牧师集合。一张选票当且仅当每个法定人数中的牧师都投了这个法令才叫投票成功。正式的来说，选票B由以下四部分组成（除非另有说明，集合就代表了有限集合）

一个法令（被投票的那个）

牧师的非空集（选票法定人数）

牧师集合（投给了这个法令的人）

选票数量

一张选票B被认为成功当且仅当，所以一张成功的选票是法定人数中的每一个都投的。

选票数量是从无限的有序集合中选出的。如果，那么选票被认为落后于选票B。然而，这不代表选票被进行的顺序；后面的选票可以替代更早的。

Paxos的数学家们定义了三个条件在选票集合B上，并且证明了如果选票的集合可以符合这些条件，一致性将被保证，同时程序也是可行的。开始的两个条件很简单，可以被如下的表示简单说明：

B1(*B*) 每个*B*中的选票有一个独特的选票序号

B2(*B*) *B*中任意两个选票的法定人数至少有一个共同的牧师。

第三个条件略微复杂。一份Paxos的手稿包含了如下描述，相当混乱：

B3(*B*) 每个*B*中的选票B，如果任意选票B法定人数中的牧师早就投给了*B*中的更早的选票，那这个B中的法令等于之前的选票中最新的法令。

这段密文的解释在图1所示的手稿中，说明了条件B3(*B*)，五位牧师A, B, Г, Δ, E组成的会议中有5个选票。*B*集合有五张选票，对于每张选票，投票者的集合是法定人数牧师的子集。举个例子，14号选票有法令α，一个法定人数包含三个牧师和包含两个投票者的集合。条件B3(*B*)有形式“对每一个*B*中的选票B:….”，其中“...”是选票B上的条件。图1中这五个选票B的条件如下。

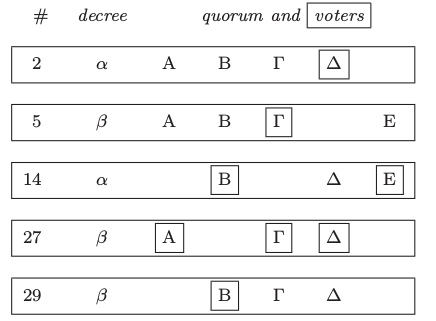


图1. Paxos手稿显示了一个集合*B*，包含了五个选票，满足条件B1(*B*)-B3(*B*)。

2. 2号选票是最早的选票，所以这张选票显然满足条件。

5. 5号选票的四个法定人数成员没人投票给更早的选票，所以这样选票也满足条件。

14. 14号选票的法定人数中只有一个 Δ投票给了更早的选票，也就是2号选票，所以根据条件要求14号选票的法令必须等于2号选票的法令。

27. （这是一个成功的选票）27号选票的法定人数是A, Г和 Δ。牧师A没有投票给更早的选票，Г的唯一一张更早的选票是投给5号选票的，同时 Δ的更早一张选票是给2号的。最近的这些两张选票是5号选票，所以根据条件要求，27号选票的法令必须等于5号选票的法令。

29. 29号选票的法定人数是B, Г和Δ。B投的更早一张选票是14号选票，牧师Г投了5号和27号，Δ投了2号和27号。最近的这四张中更相近的是27号选票，所以根据条件要求，29号选票的法令必须等于27号选票的法令。

为了更好的声明B1(*B*)-B3(*B*)，需要更多的符号。一张投票*v*定义为三部分：一个牧师，一个选票序号，一个法令。它代表了由牧师投给了选票的法令。Paxos居民们也定义了*null*投票，并且，其中并且BLANK不是法令。对于任意的牧师*p*，定义为唯一的null投票*v*当。

Paxos岛的数学家们对于所有投票定义一个总的排序，但是包含定义的手稿的部分丢失了。剩下的部分表明如下，对于任意投票v和v’，如果，那么。并不清楚当时v和v’的相对排序如何定义。

对于任意的选票集合*B*，*B*中的Votes(*B*)定义为包括所有的投票v即对于部分 B ∈ *B*,

。如果*p*是牧师并且*b*是选票序号或者，那么是被定义为由*p*投出的Votes(*B*)中最大的投票v，并且或者如果没有这个投票。因为比任何*p*的真实投票都小，也就意味着是集合中最大的投票。

对于任意的牧师非空集合*Q*，被定义为等于当*p*在*Q*中时的。

条件B1(*B*)-B3(*B*)正式描述被如下：

虽然的定义依赖于投票的顺序，B1(*B*)暗示了是独立于如何选择被排序的选票号相同的投票的。

为了展示这些条件如何实现一致性，Paxos岛民们首先展示了B1(*B*)-B3(*B*)，如果一个*B*中的选票B成功了，那么任意之后的*B*中的选票是和选票B的法令一样的。

引理： 如果满足B1(*B*), B2(*B*), B3(*B*)，则对于任意*B*中的B, B’

引理证明：

对于任意的*B*中的选票B，让作为*B*中晚于选票B的且法令不同的选票集合。

为了证明lemma，如果则是空的足以展示这一点。Paxos的人给出了矛盾的证明。他们假设存在*B*且和，并且获得了一个如下的矛盾：

1. 选择，则

证明：C存在因为非空且有限

2.

证明：按1和的定义。

3.

证明：按B2(*B*)和假设

4.

证明：按2,3和的定义

5.

证明：按4（表明非空）和的定义

6.

证明：按5和B3(*B*)

7.

证明：按1,6 和 定义可得

8.

证明：按4，因为7和B1(*B*)表明

9.

证明：按7,8 和定义

10.

证明：按定义

11. 矛盾

证明：按1,9,10得

引理证明结束

根据Lemma，很容易得出，如果B1-B3满足，那么任意两个成功的选票会是同一个法令。

定理：如果满足B1(B), B2(B) 和B3(B)，则对于任意*B*中的B, B’

定理证明：

如果，则B1(*B*)表明。如果，则定理满足引理。

定理证明结束：

2.2 The Preliminary Protocol

Paxos人随后证明了这个定理断言，如果会议中有足够的牧师，则有可能产生一个成功的选票当它维持B1-B3。虽然这不能保证进度，但它至少展示了基于选票协议不会造成死锁。

定理2：使*b*作为选票序号，*Q*作为牧师集合，则对于所有的，都有和。如果B1(*B*)，B2(*B*)，B3(*B*)保持，则存在一个选票B’使得和，则保持。

定理证明：

条件基于B1(*B*)，的选择，和关于*b*的假设。条件基于B2(*B*)，的选择和关于*Q*的假设。如果，则使为任意法令，否则让其等于。条件基于B3(*B*)。

定理证明结束：

Paxos岛民们从要求B1(*B*)-B3(*B*)为真，当*B*作为所有进行过的选票集合的情况下派生出了初步协议。协议的定义指明*B*如何更改，却从未被明确计算过。Paxos的人们将B的数量视为只有上帝可以知道，因为它从不被任何凡人所知。

每个选票有牧师定义，他选择序号，法令和法定人数。每一个法定人数中的牧师再决定是否投票。这个为了维护B1(*B*)-B3(*B*)而派生出的规则决定了初始人如何选择选票号，法令和法定人数，以及牧师决定是否投票。

为了保持B1，每一个选票必须收到一个唯一序号。通过记住他以前初始化的选票，一个牧师可以轻松的避免初始化两张同样序号但内容不同的选票（通过账本上的笔记）。为了保持不同牧师用同样的序号初始化选票，可能的选票序号集合在牧师间被划分了。当不清楚这是否被使用时，一个明显的方法将会让选票序号作为一对由序号和牧师组成的数，使用词典顺序，如

因为Γ比Λ早。在任何情况下，每个牧师都有一个无界的选票序号集合给他自己使用。

为了保持B2，一个选票的法定人数被选择包含牧师。原来，只意味着一个简单的多数。后来它被观察到胖牧师更少移动并且花费更多时间在办公室中，所以被用来代表总体重比一般牧师总体重更重的牧师的集合，不仅仅是简单的牧师中的多数。当瘦牧师的团体抱怨说不公平时，实际体重被基于牧师的出席记录代表的象征体重取代。主要的要求试任意两个保持的牧师集合中最少有一个公共的牧师。为了保持B2，牧师初始化选票B选择作为主体。

条件B3要求如果不为空，序号为b的选票和法定人数Q必须有法令。如果为空，则选票可以有任何的法令。为了保持B3(*B*)，初始化一个序号为b法定人数为Q的新选票前，牧师p必须找到。为了做到这点，p必须找到每一个Q中的牧师p的。

回顾是小于b的最大选票序号，由q投起或的投票。牧师p获得q的通过交换信息。因此，协议的最初两步是：

(1) 牧师p选择一个新的选票序号b并发送一个的信息给一些牧师集合。

(2) 一个牧师q收到了消息并回复给p，v是q投起的小于序号b的选票序号，或者当q没有任何选票序号小于b时为。

牧师q必须使用笔记在他的账本上记录他曾经发起的选票序号。

当q发送消息，v等于。但是*B*改变了。因此牧师p选择法令时将使用v作为的值，为了保持B3(*B*)成立，必须在q发送后不改变。为了保持不断变化，q必须不用到b之间的序号投票。通过发送信息，q承诺不投票（为了保证这个承诺，q必须在账本上记录必要的信息）。

选票协议（牧师p的第1步开始）接下来的两步是：

(3)在从Q中大部分牧师收到消息后，牧师p初始化一个新的选票序号为b，法定人数Q，法令d，其中d满足B3。他之后在他的账本上记录这个选票，并发送一个消息给Q中的每一个牧师。

(4)收到消息后，牧师q决定是否给这个选票投票（他可能不会投票如果这会违背他给其他选票发送消息后做的承诺）。如果q决定投票给b，他就会发送消息给p，并且在他的账本上记录。

第3步的执行考虑到在*B*中添加选票B，其中和。第4步中，如果牧师q决定投票，则执行该步被认为是通过添加q至来改变*B*，当B∈*B*。

牧师可以在第4步选择不投票，即使投票也不会违背之前的承诺。实际上，协议中的所有的步骤都是可选的。举个例子，牧师q可以忽略消息而不是执行第2步。不采取行动可以防止进展，但是可能造成不一致因为违反了B1(*B*)-B3(*B*)。由于没有收到消息的唯一结果是不发生动作，消息丢失也不会造成不一致。因此，协议保证了一致性即使牧师离开办公室或者信息遗失。

收到多份消息的拷贝可能造成行动重复。除了步骤3，第二次执行动作不会有任何效果。举个例子，在第4步中发送多个消息和只发送一次是一样的。第3步的重复试通过账本中已执行的记录避免的。因此，一致性的条件被保证即使一个同样的消息被多次传递。

步骤1-4描述了初始化选票和投票的完整协议。剩下的是决定选票结果和当法令被选择后的宣布流程。回顾一个选票是成功的当且仅当法定人数中的每一个牧师都投给它。成功的选票的法令是被会议选择的。剩下的协议如下：

(5)如果p从每一个Q（选票号b的法定人数）中的牧师q收到消息，然后他在他的账本上写d（该选票的法令）并发送一个消息给每一个牧师。

(6)收到消息后，牧师在他的账本上记录法令d。

步骤1-6描述了一个独立的选票如何产生。这个初步的协议允许任何牧师在任意时间初始化新的选票。每一个都保持了B1(*B*)-B3(*B*)，所以整个协议也保持了这三个条件。因为一个牧师在他的账本上记录法令仅当法令被成功通过，定理1表示牧师们的账本是一致的。协议没有解决进程的问题。

在第3步中，如果法令d由B3决定，则可能法令早就在一些牧师的账本上被记录了。牧师不需要作为Q中的一员，他可以离开办公室。因此，一致性可能不被保证，如果第3步允许任意更大的自由空间去选择d。

2.3 The Basic Protocol

在初步协议中，一个牧师必须记录(1)他初始化的每一张选票(2)他投的每一个选票和(3)他发送的所有消息。对于忙碌的牧师而言，保持追踪所有的这些信息是很困难的。Paxos初步协议因此限制了基本协议中要求每个牧师p必须获得的内容为仅限于他账本后的以下信息：

*p*尝试初始化的最后一张选票序号，如果没有则为

*p*之前投的选票中序号最高的选票，如果没有则为

*p*发送消息后*b*的最大值，如果没有则为

初步协议的第1-6步描述了一个单独的选票如何由它的创造者*p*进行的。初步协议允许*p*并发生成Ballot的任意序号。在基本协议中，他一次只能生成一个序号即。在*p*初始化该选票后，他会忽略他之前初始化的其他选票相关的消息。牧师*p*保持所有的关于的信息在一张纸上。如果他丢失了这张纸，他就会停止该选票的进行。

初步协议中，每一个*q*回复的消息表示他承诺不投票给选票序号在和*b*之间的选票。在基本协议中，它表示强保证不新投票给任何小于*b*的选票序号。这个不投给其他选票的强保证在基本协议的第4步不可避免，但在初步协议中是可以避免的。然而，因为初步协议总是给q不投票的选择，基本协议不要求他做任何在初步协议中不允许的事。

初步协议的第1-6步转变成基本协议中的如下6步。（所有的包括上面描述的3种信息的内容，都写在一张纸上）

(1) 牧师*p*选择一个新的选票序号*b*，同时*b*大于，设置为*b*，并且发送给其他牧师。

(2) 从*p*收到一个 消息并且 ，牧师*q*设置为*b*并且发送消息给*p*，其中*v*等于 。（当 时被忽略）

(3) 从牧师集合Q中收到大部分牧师的信息后，其中要求，牧师*p*初始化一个序号为*b*新的选票，法定人数Q，法令为*d*，其中*d*满足条件B3。之后他发送消息给Q中的每一个牧师。

(4) 收到满足的消息后，牧师*q*投票给选票*b*，设置为该选票，并且发送消息给p。（如果则被忽略）

(5) 如果*p*从Q中每一个牧师*q*处收到消息，其中，然后他在他他账本上写下该法令*d*，并且发送消息给每一个牧师。

(6) 收到消息后，牧师在他账本上记录法令*d*。

基本协议是初步协议的限制版本，意味着在基本协议中被允许的每一步都在初步协议中被允许。因为初步协议满足一致性条件，基本协议也满足条件。就像初步协议一样，基本协议不要求每一步都被执行，所以没有解决进度的问题。

B1-B3推导基本协议的的进度使满足一致性条件这一点被观察到。然而，一些类似的“观察”古老的智慧结果是错误的，并且怀疑的公民们要求更严格的证明。Paxos的数学家们证明协议满足一致性条件的进度在附录中。

2.4 The Complete Synod Protocol

基本协议保持了一致性，但是不能确保进度因为它只表明牧师可以做什么，不要求牧师做任何事。完整的协议由基本协议中同样的6步组成。为了帮助实现进度，它包括了明显的额外要求，牧师在2-6步要尽可能的快。然而，为了实现进度条件，一些牧师执行步骤1是必要的。完整协议的关键在于决定一个牧师何时应该初始化选票。

绝不初始化选票肯定会影响进度。然而，初始化太多选票同样会影响进度。如果*b*比任意其他的选票序号都大，然后在第2步中牧师*q*收到信息可能引起一个保证即在第4步中对之前任意初始化过的投票都不投。从而，一个新选票的初始化会影响任何之前成功初始化的选票。如果一个新选票是在之前的选票成功之前使用递增的选票序号初始化，则不会有任何进度产生。

实现进度条件需要初始化新的投票，直到成功，但不会太频繁地初始化。为了发展完整的协议，Paxos的人们首先需要直到信使需要多久传递消息以及牧师需要多久回复。他们决定在办公室中的信使总是需要最多4分钟传递信息，并且留在办公室的牧师在该事件中总是需要最多7分钟采取动作。从而，如果*p*和*q*都在办公室中，一些事件造成*p*发给*q*消息，然后*q*回复给*p*，如果双方的信使都没离开办公室，则*p*将会在22分钟内收到该回复。（牧师*p*将会用7分钟以内的时间发送该事件的信息，*q*收到需要4分钟，回复需要7分钟，*p*收到需要4分钟，总22分钟）。

假设只有一个牧师*p*正在初始化选票，并且他在协议的第1步给每个牧师发送信息。如果大部分牧师都在办公室中时，*p*初始化了一个选票，则他可以用22分钟执行步骤3，再用22分钟执行步骤5。如果他不能在这些时间内执行这些步骤，则有其他牧师或信使在*p*初始化选票后离开了办公室，或者有其他牧师之前初始化的更大序号的选票（在*p*成为唯一的初始化选票的牧师之前）。为了处理后一种可能，*p*必须学习关于其他牧师初始化的比更大的选票序号。这个可以由扩展协议完成，要求如果一个牧师*q*从*p*处收到了或者信息，其中，则他回复给*p*的消息中包含。牧师*p*接着会使用更大的数字进行初始化。

仍旧假设*p*是唯一的一个初始化选票的牧师，假设他被要求初始化一个新的选票当且仅当(i)他在之前22分钟内未执行第3或5步(ii)他得知了其他牧师初始化了更大序号的选票。如果办公室的门被*p*和大部分牧师反锁，法令将会在99分钟内被通过并且记录在所有在办公室的牧师的账本上。（对于*p*，需要22分钟开始下一个选票，超过22分钟知道其他牧师初始化了更大序号的选票，然后用55分钟执行1-6步去完成一个成功的选票）。从而，进度条件将被满足如果只有一个没离开办公室并且在初始化选票的牧师。

完整的协议因此包含了选择唯一的牧师的程序，叫做主席，来完成初始化选票。在大多数的政府组成中，选择一个主席都是困难的问题。然而，难度上升的主要原因是政府要求任意时间只能有一位实际的主席。在美国，当有部分人认为Bush应该当总统，而其他人认为Dukakis应该当时就会产生混乱，因为一个人决定会在签署一项法案而另一个则不会。然而，在Paxon Synod中，有多个主席会阻碍进度，不会造成不一致。为了使完整协议满足进度条件，选择主席的方法需要满足以下要求：

如果没有人进入或离开办公室，则T分钟之后，在办公室中的某位牧师会考虑将他自己作为主席。

如果满足该条件，接着完整协议拥有属性：如果大部分牧师在办公室中，并且没有在接下去的T+99分钟内进入或离开办公室，则周期结束后每个在办公室的牧师都将有一条法令写在他的账本上。

Paxos岛民们选择一个牧师作为主席根据谁的名字在办公室中所有牧师名字按照字母表排序的最后一个，虽然我们并不知道实际怎么做的。这个主席选举条件将满足如果一个牧师在办公室中每T-11分钟发送包含自己名字的消息给其他牧师至少一次，并且该牧师考虑他自己作为主席当且仅当他T分钟后没收到更高排名的名字的牧师的回复。

完整协议是通过要求牧师及时执行基础协议的第2-6步演化的，并且添加了选择主席进行初始化选票，并且在适当的时间初始化选票的方法。协议的很多细节我们并不清楚。我描述了简单的选择主席方法并且何时主席应该初始化一个新的选票，但是这显然不是Paxos使用的版本。我给出的规则要求主席保持初始化选票即使法令已经被采纳，从而确保刚进入办公室的牧师们得知被采纳的法令。很明显有更好的方法保证牧师可以在法令被采纳后可以知道。同时，在选择主席时，每一个牧师可能发送他的给其他牧师，允许主席去选择足够大的选票序号作为他的首次尝试。

Paxos岛民认识到任意可以达到进度条件的协议必须涉及到测量时间的流逝。协议给出以上选择President和初始化选票是可以被很简单的实现的，只要精确的算法保证时间设置和超时发生时的执行动作——假设完美准确的计时。最近的分析揭示这样的协议可以与足够准确的计时器一起工作。Paxos岛上的熟练工们可以很轻松的构造合适的计时器。

鉴于Paxos的数学家们的复杂性，他们认为必须找到一个最佳的算法去满足选择President的要求。我们只能希望将来可以发现这个算法。

3 The Multi-Decree Parliament

当议会被建立，从主教协议中派生出一个协议满足它的一致性和进度要求。原始议会协议的推导和属性在第3.1和3.2节中描述。3.3节讨论协议的更多演化。

3.1 The Protocol

不同于仅通过单个法令，Paxon议会必须通过一系列法令。像在主教协议中，一个主席被选中。任何想要法令通过的人会通知分配给法令序号并尝试使其通过的主席。逻辑上来说，议会协议给每一个法令序号使用完整主教协议的一个单独的实例。然而，一个单独的President是从所有的实例中选择，并且他只会执行一次协议开始的两步。

推导议会协议的关键是意见(observation)，在主教协议中，主席直到第3步才会选择法令或者法定人群。一个新被选择的主席*p*可以发送一个立法者集合作为一个消息作为消息给所有的主教协议实例。（这里有一个无限数量的实例——每一个对应一个法令序号）。一个立法者*q*可以在第2步中回复一个消息作为信息给所有的主教协议实例。该消息仅包含有限数量的信息，因为*q*只能在有限数量的实例中投票。

当新主席收到从大部分牧师的回复，他准备执行每一个主教协议实例的第3步。对于有些有限数量的实例（法令数量），在第3步中法令的选择由B3决定。主席立即执行这些法令的第3步，他选择最小序号的法令使其自身仍可以自由选择，并且执行该法令（主教协议实例）的第3步去尝试通过这些法令。

以下对简单协议的修改引出实际的Paxon协会的协议。

——没有必要经过主教协议去拿已经知道结果的法令的序号。因此，如果一个新被选择的主席*p*在他的账本上有所有的小于或等于序号n的法令，则他发送消息给所有的主教协议实例中序号大于n的法令一个消息。在他的对这个消息的回应中，立法者*q*通知*p*所有的大于n的法令序号早就出现在*q*的账本中（除了发送通常的的信息不在他的账本上），并且他问*p*是否发送给他不在他账本上的所有法令序号小于等于n的法令。

——假设法令125和126是在上周五下午被采用的，法令126被通过并被写在1-2份账本上，但在任何事情发生之前，所有立法者周末都回家了。假设接下去的周一，被选作新的主席并且学习法令126，但是他不知道125因为之前的主席和所有的给它投票立法者都还没回来。他将举行投票通过126，这在账本中留下了空白。分配序号125给一个新法令将导致他在账本中比126先出现。使用这种方法通过乱序的法令可能造成混乱，举例来说，如果有市民因为知道126早就被通过就故意提出新的法令这么做。相反，会尝试通过

125: The ides of February is national olive day

这种对Paxos的任何人都没有任何影响的法令。

议会协议的一致性和进度属性和从主教协议派生出来的对应属性中得出。根据我们的了解，Paxos岛的人从不困扰于写一份精确的议会协议的描述，因为这很容易从主教协议中推导得出。

3.2 Properties of the Protocol

3.2.1 *The Ordering of Decrees* 选票可以使用不同的序号并发发生，选票也被不同的立法者初始化，每一个立法者在他初始化选票时都认为他自己是主席。我们又恰恰不能说应该以什么顺序通过法令，特别是不知道主席是怎么被选出的情况下。然而，这里又可以推导出一个关于法令排序的重要的属性。

一个法令被认为提出是当主席执行到主教协议对应的实例的第3步时选择到它。一个法令被认为通过是当它第一次被写到账本上。在主席能够提出新的法令前，他必须从大部分人那里知道他们都投了哪些法令。任意一个被通过的法令必须被大部分中的至少一个立法者投票。主席不会填补账本上的重要法令的空白，也就是比“olive-day”更高的法令。他也不会提出不按顺序的法令。因此，协议满足如下的*有序法令属性*：

如果法令A和B很重要，并且法令A的通过在B被提出之前，则A的法令序号比B低。

3.2.2 *Behind Closed Doors* 虽然我们不知道选择一个新主席的细节，但是我们恰好知道议会在主席被选中并且没人进入或离开办公室时是如何运作的。在收到一个通过法令的请求后—或者直接从市民或其他立法者处得到—主席分配法令一个序号并且通过它有如下这些消息交换。（序号对应主教协议中的具体步骤）

(3) 主席发送消息给法定人数中的其他立法者。

(4) 每一个法定人数中的立法者发送消息给主席。

(5) 主席发送消息给每一个立法者。

假设一个议会有N个立法者并且法定人数差不多是N/2，这是所有的三种信息并且差不多有3N条信息。此外，如果议会很忙，主席可以组合一个法令的消息和之前一个法令的消息，这样每一个法令只有2N条信息。

3.3 Further Developments

治理这个岛开始变得比Paxos岛民想象的更复杂了。一些问题出现，并且这些问题如果想要解决，就要修改协议。最重要的一些修改如下所述。

3.3.1 *Picking a President* 议会主席本来使用主教协议中用的方法，也就是按照名字排序。因此，当立法者休假回来后，他立即做了主席—即使他并不知道在他不在的时候发生了什么。在还在将他缺失的法令更新到最新状态时，议会的行动变得迟钝了。

这个事件导致了关于更好的选择主席的方法的讨论。一些人认为当一个立法者做了主席后，在他离开办公室之前他都是主席。市民中有影响力的部分人认为应该由在办公室中的最富有的立法者做主席，因为这样他才可以雇用更多的人和仆人帮助他执行作为主席的职责。他们认为一旦一个富有的立法者将他的账本更新完之后，他没有理由不继续做主席。另外的人，认为最突出的市民可以做主席，不管多有钱。足够突出意味着不诚信的可能性更小，虽然没有Paxon岛民有公开管理官方不法行为的可能。不幸的人，辩论结果并不为人所知，没有任何关于最终的选择主席协议的记录存在。

3.3.2 *Long Ledgers* 随着时间的推移，议会通过了越来越多的法令，Paxon的人必须在一个超长的法令清单上找当前仍在执行的法令。一个立法者从一个延长假期中回来后，必须去大量的操作使其的账本更新。终于，立法者们被迫将他们的账本从列表转变为只包含当前法令状态和通过该状态对应的法令序号的法律书籍。

为了知道现在的橄榄税，去“税法”下面寻找；为了知道什么颜色的山羊可以被贩卖，去“商业法”下面寻找。如果一个立法者的账本包含了1298条法令，并且他得知法令1299将橄榄税设置为6德克拉玛每吨，他就修改橄榄税对应的法令并且标记他的账本法令数量为1299条。如果他接着学习到法令1302，他将在账本背后记录下该信息，并等到法令1300和1301后结合1302一并写到法律书中。

为了使立法者能够快速赶上进度而不需要复制整本法律书，立法者们将过去一周的法令清单记载书的背后。他们可以将清单保持在一张纸上，但是将新的法令记录在账本的背后更方便，因为他们一周通常只会通过2到3个法令。

3.3.3 *Bureaucrats* 因为Paxos的繁荣，立法者们变得非常繁忙。议会开始不能处理所有的政府事项的细节，所以一个官僚主义被建立了。相对以前的通过一条法令去规定那些奶酪可以出售，现在议会通过一条法令指定一个奶酪检查员去做这些决定。

很明显可以发现选择官员不像一开始看起来的那么简单。议会通过法令使做第一次的奶酪检查员。几个月后，商人们抱怨说太严格了并且拒绝了很好的奶酪。议会于是决定通过法令替换他

1375: is the new cheese inspector

但是没有更多的注意议会怎么做的，所以他不知道立即得到该法令。在和同时检查奶酪并做出矛盾的决定时，市场有一段混乱的时期。

为了避免这些混乱，Paxos岛民必须保证在任意时间一个位置可以有多个官员担任。为了做到这个，主席将法令提出的日期和时间都作为法令的一部分记录了账本。一个使成为奶酪检查员的法令读起来应该是这样的

2716: 8:30 15 Jan 72 — is a cheese inspector for 3 months

这表明他的任职期从1月15日8点30分或者上一任退任后开始。他的任职周期将在3月15日8点30分结束，除非他要求主席通过他辞职的法令

2834: 9:15 3 Mar 72 — resigns as cheese inspector/

一个官员被短期任命，所以他可以被快速替代——比如说他离开了岛。议会将会通过一条法令在官员做的很好时扩展官员的任期。

如果官员现在有担任职务，他需要告知多少时间来决定。机械表在Paxos不为人知，但是Paxos人可以通过太阳或者其他星辰确定精确到15分钟。如果的任期从8:30开始，他不会开始检查奶酪直到他的检测结果表明现在是8:45分。

如果高序号的法令总是被晚提交，那么任命官员的工作是很简单的方法。但是如果议会通过法令

2854: 9:45 9 Apr 78 — is wine taster for 2 months

2855: 9:29 9 Apr 78 — is wine taster for 1 month

分别由不同的认为各自是主席的立法者在9:30到9:35分之间提交。类似的乱序的提交时间是很简单就可以避免的，因为议会协议满足如下特性。

如果两条法令由不同的主席通过，则在学习其他人的提交的法令之后提交他法令的主席被通过。

为了使该条件满足，假设选票序号b是成功的法令D的序号，选票序号b’是成功的法令D’的序号，并且。使q作为同时投给这两个选票的立法者。D’的选票开始于一个消息。如果消息发送者不知道D，则n比D的序号小，并且q的回复必须表明他投给了D。

3.3.4 *Learning the Law* 除了要求通过法令外，普通市民需要查询现在的法律。Paxos岛民一开始认为市民可以检查任意立法者的账本，但是接下去的事件表明需要一个更复杂的方法。数百年以来，只能卖白山羊已经成为了法律。一个名叫的农场主要求通过法令

77: The sale of black goats is permitted

接着要他的牧羊人出售一些黑色山羊给一个叫的商人。作为一个安分的市民，询问立法者这样的交易是否合法。但是不在办公室并且在他的账本上没有76号法令。他建议说这个交易在当前法律中不合法，所以拒绝买这些山羊。

该事件导致下面的关于法律查询的单调性条件公式。

如果一人查询先于另一个查询，则第二个查询不能展示比第一次查询时更早的法律状态。

如果一个市民得知了特定的法令被通过，则根据该条件，获得新知识的过程是隐含的查询。就像我们将看到的，这些年来单调性条件的解释改变了。

一开始，单调性条件通过给每一次查询新建一个法令实现。如果想要知道当前的橄榄税，他就得去议会通过类似如下的法令

87: Citizen is reading the law

他将接着读取任意的至少通过86条法令的完整的账本来知道当前的橄榄税。如果市民接着查询橄榄税，他的查询法令会在87号法令通过后提出，所以法令顺序属性（3.2.1节）保证它收到一个大于87的法令序号。因此，不能比更早知道橄榄税。这个方法在“优先”被解释为查询A先于查询B当且仅当A结束早于B的开始时间时满足了单调性条件。

给每一次查询都通过一次法令很快就显得太笨重了。Paxos岛民们意识到如果他们通过改变“优先”的解释来减弱单调性条件，那么就会更简单。他们决定对于一件事优先于另一件，第一件事不仅要更早发生，而且必须对第二件事造成因果影响。更弱的单调性条件避免了第一次的农场主和商人的事，因为在的隐含的查询结束和的查询开始之间有一条因果链。

更弱的单调性条件通过在所有的商业交易和查询中使用法令序号来满足。举例来说，农场主，有很多包括非白色的山羊，要求议会通过法令

277: The sale of brown goats is permitted

当出售他的棕色山羊给时，他通知商人因为法令277，所以该售卖是合法的。接着询问立法者法令277是否之后该售卖是否合法。如果的账本未包含法令277，他会等到法令更新或者建议去询问其他人。如果的账本更新到了法令298，他就会告诉在当前法令298时，该售卖是合法的。商人将会记住序号298，并在下一次交易或者查询时使用。

Paxos人满足了单调性条件，但是普通市民不喜欢记住法令序号。再一次，Paxos人通过重新解释单调性条件解决问题——这一次，通过改变法律状态的含义。他们拆分法律在分离的区域，并且每个区域有一个被选中的立法者作为专家。当前每个区域的法律状态由专家的账本决定。举例来说，假设法令1517改变了关税法，法令1518改变了税法。税法将会首先改变如果税法专家比关税法专家首先学到相关法令，产生通过数字顺序发布法令而不能获得的法律状态。

为了避免当前状态的冲突定义，Paxos人要求对于任意区域，最多同时有一个专家。这个要去通过和选择官员（3.3.3节）一样选择专家的方法来满足。如果每一个查询只参与法律的一个区域，单调性则通过直接查询对应区域的专家被实现。因为学习法律是通过一个隐含的查询结果，Paxos人要求法令修改最多只在法律的一个区域，并且法律通过的通知只能从该区域的专家处传出。

多区域查询也不难被处理。当商人询问进口金羊毛的关税是否高于本地一次采购的销售税，税法和关税法专家必须合作给出答案。举例来说，税法专家可以通过首先询问关税法专家当前金羊毛的关税来回答，所以在他收到答复前，他可以不修改账本。

这个方法很好，直到需要一次性对多个区域进行大修改的情况发生。Paxos人随后意识到维护单调性的必要要求不是法令只影响一个区域，而是每个区域影响的有同样的专家。一会可以使用首先制定一个单独的立法者变成所有这些区域的专家的法令来改变多个法律区域。此外，同样的区域可以有多个专家，在这段时间内该区域的法律不允许被修改。就在所得税法到期前，议会将指定多个税法专家去处理季节性的税法查询潮。

3.3.5 *Dishonest Legislators and Honest Mistakes* 尽管官方不这么认为，但是Paxos历史上肯定有一些不诚实的立法者存在。一旦被抓到，就可能被放逐。通过发送矛盾的信息，一个恶意的立法者可能造成不同的立法者的账本不一致。不一致也可能是由于诚实的立法者或信使的失误。

当不一致发生时，他们可以使用通过法令的方法来矫正。举个例子，当前橄榄税的不统一意见可以通过一个新的定义特定的税的法令来消除。矫正不一致账本的难题是没人意识到不一致。

不诚实的立法者或者立法者的失误的存在可以由议会建立以来几年后开始出现在账本中的冗余法令推断出。举例来说，该法令

2605: The olive tax is 9 drachmas per ton

被通过了，即使法令2155早就设定了橄榄税为9德克拉玛每吨，并且没有干预的法令改变它。议会明显地每六个月循环通过它的法律，这样即使立法者们的账本一开始不一致，所有的立法者都将会在六个月内同意当前的法律。这被认为通过使用这些冗余法令，Paxos人使得他们的议会自稳定。（自稳定是Dijkstra的创造的术语）

尚不清楚议会如何在立法者可以随意去留的情况下定义自稳定性。Paxos人不会满意在一致性被保证前要求所有立法者同时在办公室中的定义。然而，实现一致性要求如果一个立法者有一个其他人没有的法令序号，则第二个立法者会将其写入自己的账本。

不幸的是，我们还是不知道它们是如何提出或实现自稳定特性。Paxos的数学家们无疑定位到了问题，但是他们的工作成果未被找到。我希望未来对Paxos的考古发现可以给发掘自稳定性的手稿一个高的优先级。

3.3.6 *Choosing New Legislators* 首先，议会会员是世袭的，由父母传给儿女。当老政治家退休，他将他的账本给他的儿子，并不断持续下去。其他的立法者也和一样。

就像老家庭移民出去而新家庭移民进来一样，这个系统必须改变。Paxos人决定通过法令来添加和移除议会的成员。这个提出了一个循环的问题：议会成员由通过的法令决定，但是通过法令需要知道哪些人构成主要团体，反过来也就是依赖于谁是议会成员。这个循环最后被打破了，方法是让通过法令n的议会成员由法令n-3的法律指定。主席可以不尝试通过法令3255直到他知道所有的法令通过法令3252。在实践中，在通过法令

3252: is now a legislator

主席将立即通过像“橄榄日”法令的法令3253和3254。

使用该方法修改议会组成是很危险的，所以必须小心执行。一致性和进度条件将一直保持。然而，进度条件保证进度除非大部分人在办公室中；不保证大部分人一直在那。事实上，选择立法者的机制导致了在Paxos的议会系统的倒台。因为一个抄写员的错误，一条应该是尊重沉船中溺水死亡的船员的法令变成了宣布他们为议会仅有的成员。它的通过阻止了所有新的法令通过，包括应该修正该错误的法令。政府停止了运作。一个叫的人利用这个机会制止了政变，建立了军事独裁的政府并取代了几个世纪的程序化的政府。Paxos在一系列的独裁者腐败之后逐渐衰弱，并且抵抗不了敌人的入侵，最终走向了灭亡。

4 Relevance to Computer Science

4.1 The State Machine Approach

虽然Paxos的议会在几个世纪前被摧毁了，但它的协议仍然是有用的。举例来说，考虑一个简单的被用作命名服务的分布式数据库系统。数据库的状态包括将值赋给名称。数据库的拷贝由多个服务维护。一个客户端程序的请求可以发送到任何一个服务器，去读取或修改名称对应的值。现在有两种读请求：一个慢读，返回当前名称对应的值，一种快读，更快但可能不反应数据库最近的变化。

这是数据库系统和议会的可视化的对应关系：

Parliament Distributed Database

legislator ↔ server

citizen ↔ client program

current law ↔ database state

command: read(name, client) update(name, val, client)

response: (client, value of name) (client, “ok”)

new state: same as current state same as current state except value of name changed to val

一个客户端的改变值的请求表现为通过一个法令。慢读调用通过法令，就像3.3.4节描述的。快读表现为读取服务的当前版本的数据库。议会协议提供了数据库系统的分布式容错的实现。

分布式数据库的实现方法是一个状态机方法的实例，在[Lamport 1978]中第一次提出。在这个方法中，首先定义一个状态机，包括了一些状态的集合，命令的集合，返回的集合，和分配一个回复/状态键值对（包含了一个回复和一个状态的键值对）给每一个命令/状态对的方法。直观上，状态机通过产生一个回复和修改它的状态来执行命令；命令和机器的当前状态决定了它的回复和它新的状态。

**附录**

原文与译文

|  |  |
| --- | --- |
| 原文 | 译文 |
| Synod Protocol | 主教协议 |
| Basic Protocol | 基础协议 |
| Parliament Protocol | 议会协议 |
| Preliminary Protocol | 初步协议 |
| Progress | 进度 |
| observation | 意见 |
| President | 主席 |
| Priest | 牧师 |
| Ballot | 选票 |
| Decree | 法令 |
| Legislator | 立法者 |
| Quorum | 法定人数 |
| drachmas | 德克拉玛 |
| State Machine | 状态机 |