云计算（第三版）

CLOUD COMPUTING Third Edition

第 2 章

Google云计算原理与应用（二）

主编：刘鹏 教授

目录

初步了解Chubby

Chubby是Google设计的提供**粗粒度**锁服务的一个**文件系统**，所有操作都是在文件的基础上完成的。
它基于**松耦合**分布式系统，解决了**分布的一致性问题**，是一种**建议性**的锁。

通过使用Chubby的**锁服务**，用户可以确保数据操作过程中的一致性

Chubby作为一个稳定的**存储系统**存储包括元数据在内的小数据

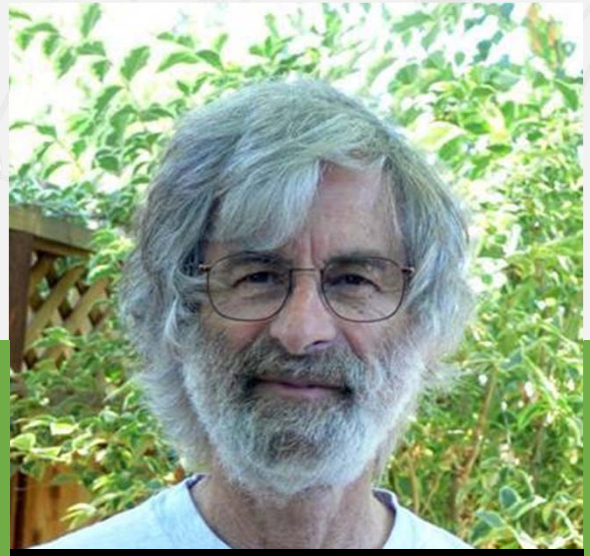
Google内部还使用Chubby进行**名字服务**（Name Server/DNS）

GFS使用Chubby来实现对GFS Master服务器的选取

2.3 分布式锁服务Chubby

- 2.3.1 Paxos算法
- 2.3.2 Chubby系统设计
- 2.3.3 Chubby中的Paxos
- 2.3.4 Chubby文件系统
- 2.3.5 通信协议
- 2.3.6 正确性与性能

产生背景



Leslie Lamport
算法设计者

Paxos

- 基于消息传递的一致性算法
- 解决分布式系统中的一致性问题
- 一致性算法中，该算法最常用、公认最有效

Leslie Lamport 1984年开始研究并行计算和分布式计算；1990年提出Paxos算法。
因为在分布式计算方面的杰出贡献，获得ACM（国际计算机学会）颁发的2013年度图灵奖。

- 产生背景

故障

恶意

Paxos

- 基于消息传递的一致性算法
- 解决分布式系统中的一致性问题
- 一致性算法中，该算法最常用、公认最有效

分布式系统的一致性问题， 就是如何保证系统中初始状态相同的各个节点在执行相同的操作序列时， 看到的指令序列是完全一致的， 并且最终得到完全一致的结果。

一个专门节点

失效

多个节点

- Paxos算法

约束条件

Proposers→acceptors→learners

少数服从多数原则，大多数

集合论：两组多数派至少有一个公共的acceptor

每个acceptor只能接受一个决议

- **系统的约束条件**

p1：每个acceptor只接受它得到的第一个决议。 不完备

p2：一旦某个决议得到通过，之后通过的决议必须和该决议保持一致。

p2a：一旦某个决议v得到通过，之后任何acceptor再批准的决议必须是v。

p2b：一旦某个决议v得到通过，之后任何proposer再提出的决议必须是v。

p2c：如果一个编号为n的提案具有值v，那么存在一个“多数派”，要么它们中没有谁批准过编号小于n的任何提案，要么它们进行的最近一次批准具有值v。

为了保证决议的唯一性，acceptors也要满足一个约束条件：当且仅当 acceptors 没有收到编号大于n的请求时，acceptors 才批准编号为n的提案。

- **一个决议分为两个阶段**

1 准备阶段

S1a. Proposer 发送 Prepare请求：

proposers选择一个提案并将它的编号设为n，编号是全局唯一且可递增的，将它发送给全部acceptors或其中的一个“多数派”。

S1b. Acceptor 应答 Prepare请求：

(1) 如果acceptor收到的提案的编号 > 它已经回复的所有提案的编号，则acceptors将该提案的编号记录下来，作为当前接收到的最大提案编号；然后回复请求，并将自己上次批准的编号和value（如果有）回复给proposers，并不再批准小于n的提案。

(2)如果acceptor收到的提案的编号 < 已经回复的所有提案的编号，则acceptor不回复或回复error。

- 一个决议分为两个阶段

2

批准阶段/选举阶段

S2a：Proposer 发送 Accept请求

经过一段时间后，Proposer 收集到一些 Prepare 回复，有下列几种情况：

(1) 回复数量 > 一半的Acceptor数量，且所有的回复的value都为空，则Porposer发出accept请求，并带上自己指定的value。

(2) 回复数量 > 一半的Acceptor数量，且有的回复value不为空，则Porposer发出accept请求，并带上回复中编号最大的value（作为自己的提案内容）。

(3) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的编号，再转回S1a执行。

- 一个决议分为两个阶段

2

批准阶段/选举阶段

S2b：Acceptor应答请求

(1) 收到的编号N >= 接收到的最大编号值，则回复提交成功，并持久化编号N和value。
(2) 收到的编号N < 接收到的最大编号值，则不回复或者回复error。

S2c：Proposer 统计投票

经过一段时间后，Proposer 收集到一些 Accept 回复提交成功，有几种情况：
(1) 回复数量 > 一半的Acceptor数量，则表示提交value成功。此时，可以发一个广播给所有Proposer，通知它们已commit的value。
(2) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的编号，再转S1a执行。
(3) 未收到回复或收到error回复，则尝试更新生成更大的编号，再转S1a执行。

每个Proposer的Paxos协议是一轮一轮分别进行的

确保编号唯一：

n个proposer，每个编号为i_r (0<=i_r <n)，则proposer的编号值可以取为
$s = m*n + i_r$（ m≥0 ）

以3个proposer P1、P2、P3为例，编号分别为0，1，2 ，开始m=0
P1提交的时候发现了P2已经提交，P2编号为1 > P1的0，因此P1重新计算编号：new P1 = 1*3+0 = 4
P3以编号2提交，发现小于P1的4，因此P3重新编号：new P3 = 1*3+2 = 5

【例子1】

在server1、server2和server3中选一个Master

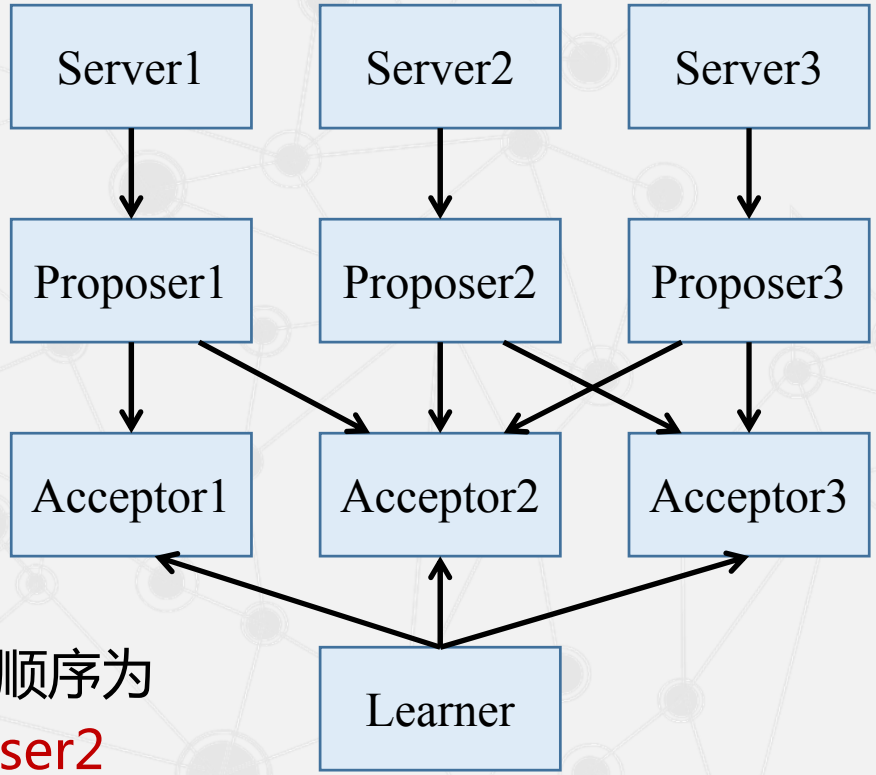
已知：

Proposer1的编号为2；

Proposer2的编号为1；

Proposer3的编号为3；

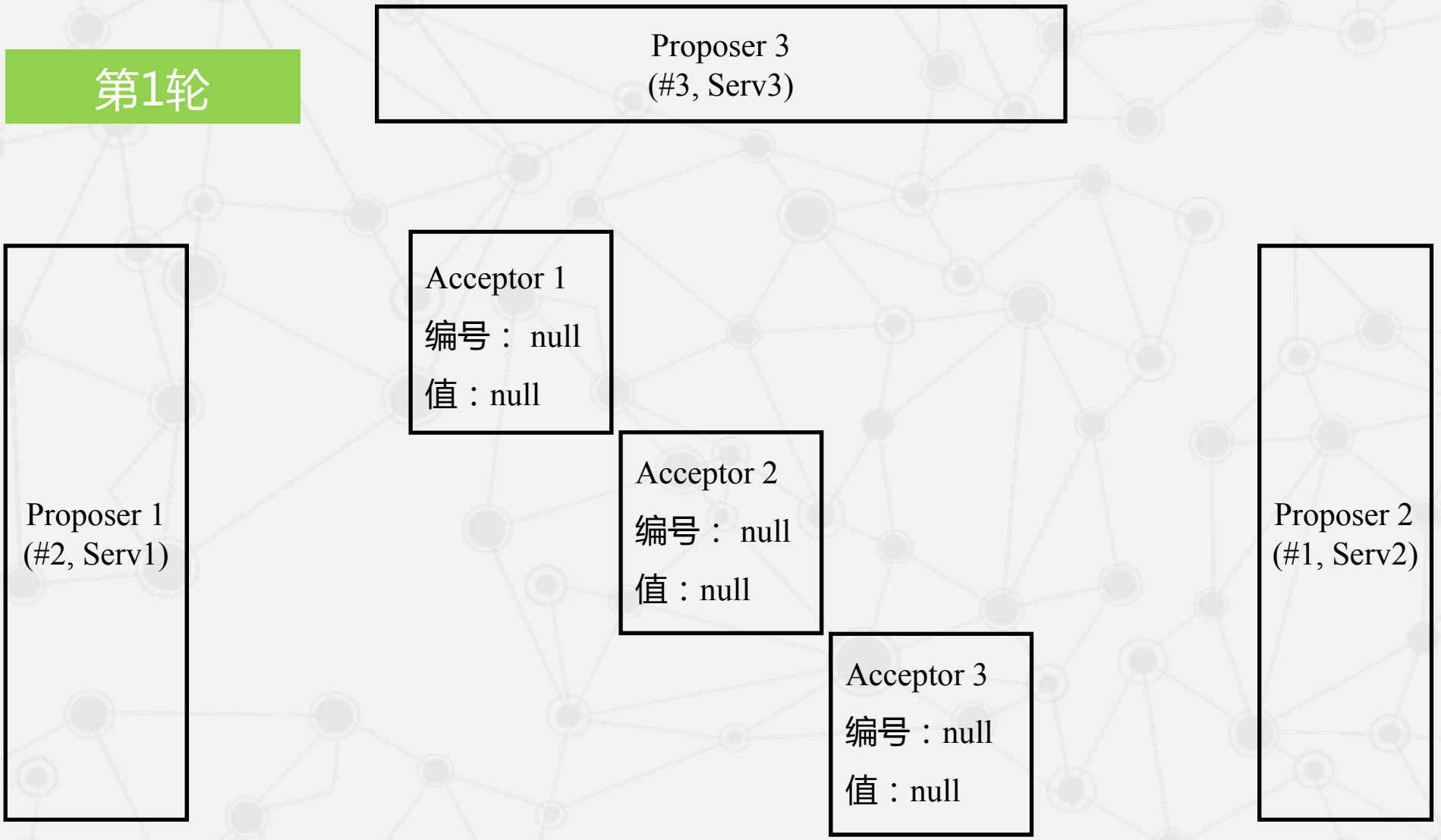
Proposer向Acceptor提交决议的顺序为Proposer1、Proposer3、Proposer2



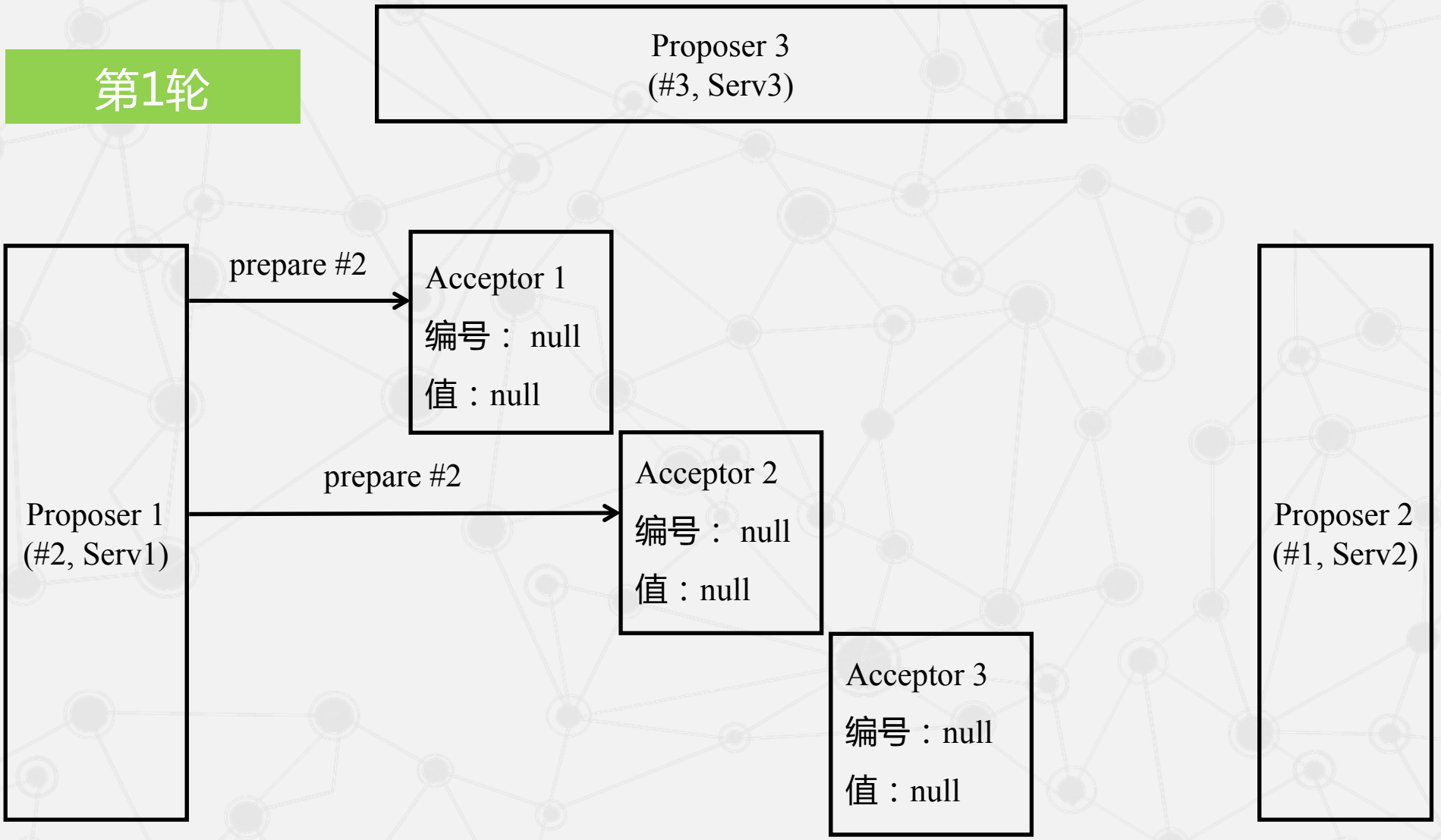
试分析：

最终选取出的Master是哪台服务器？写出分析过程。

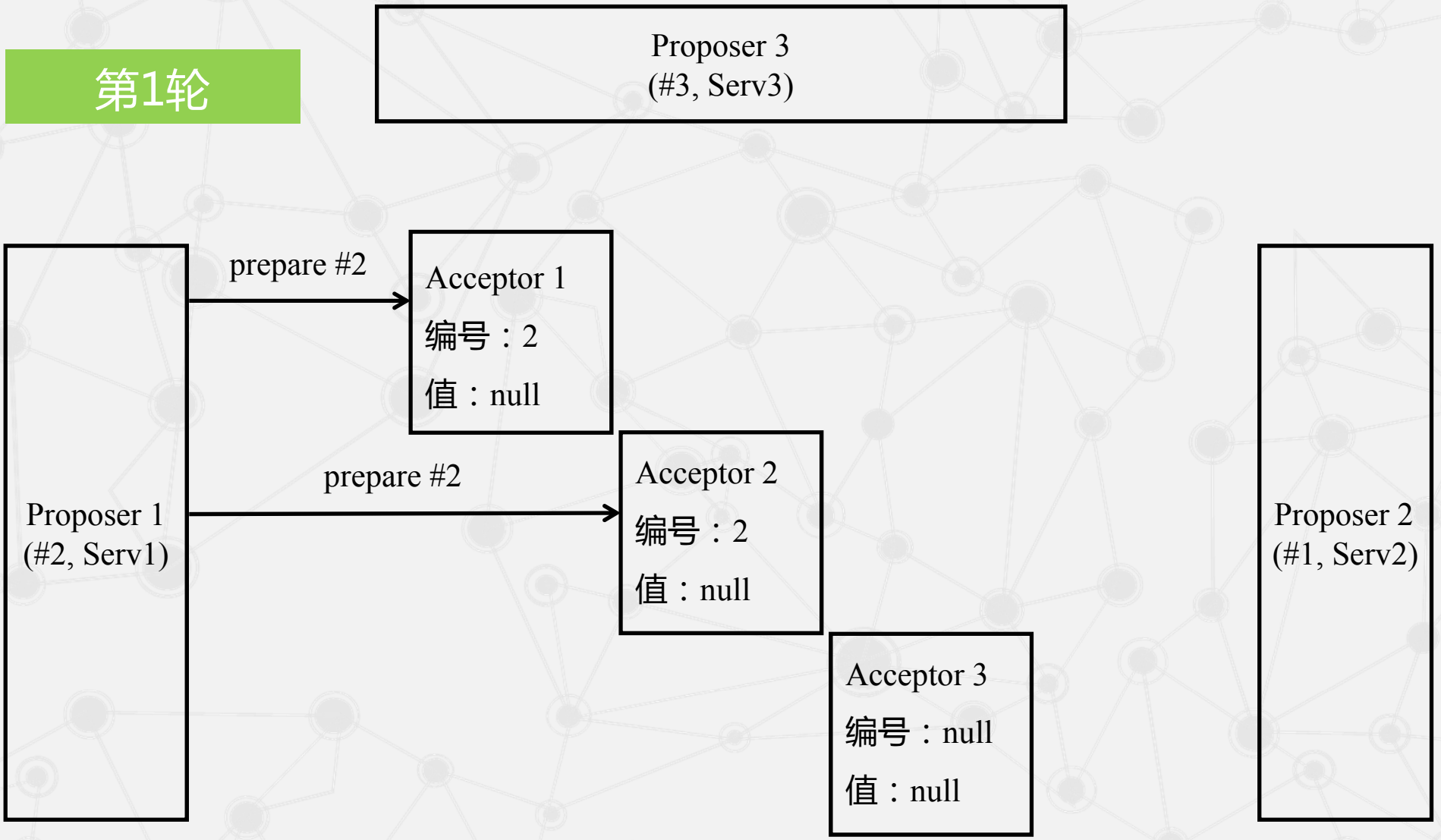
第1轮


Proposer 3
(#3, Serv3)
Acceptor 1
编号： null
值：null
Acceptor 2
编号： null
值：null
Acceptor 3
编号：null
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#1, Serv2)

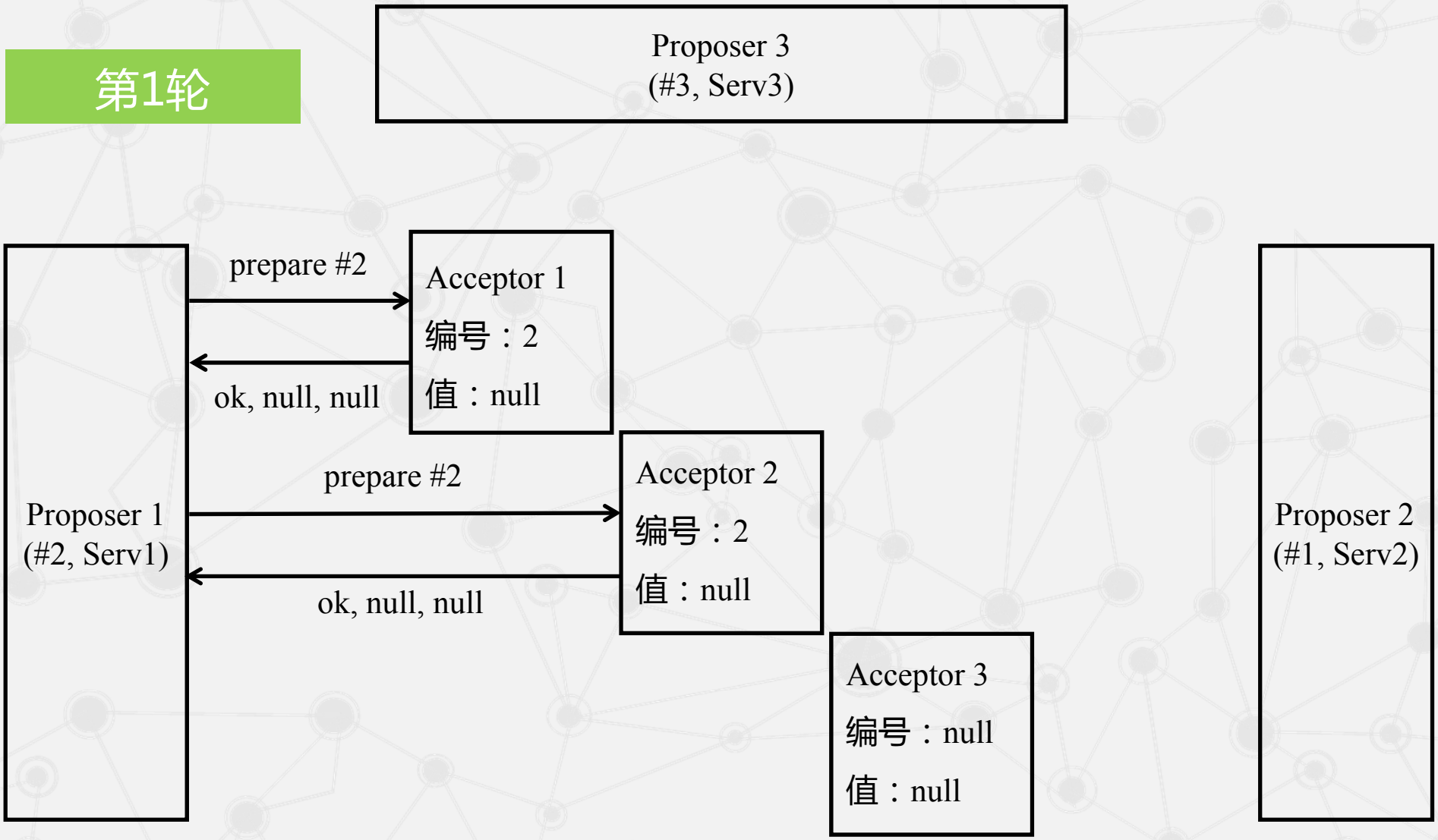
第1轮

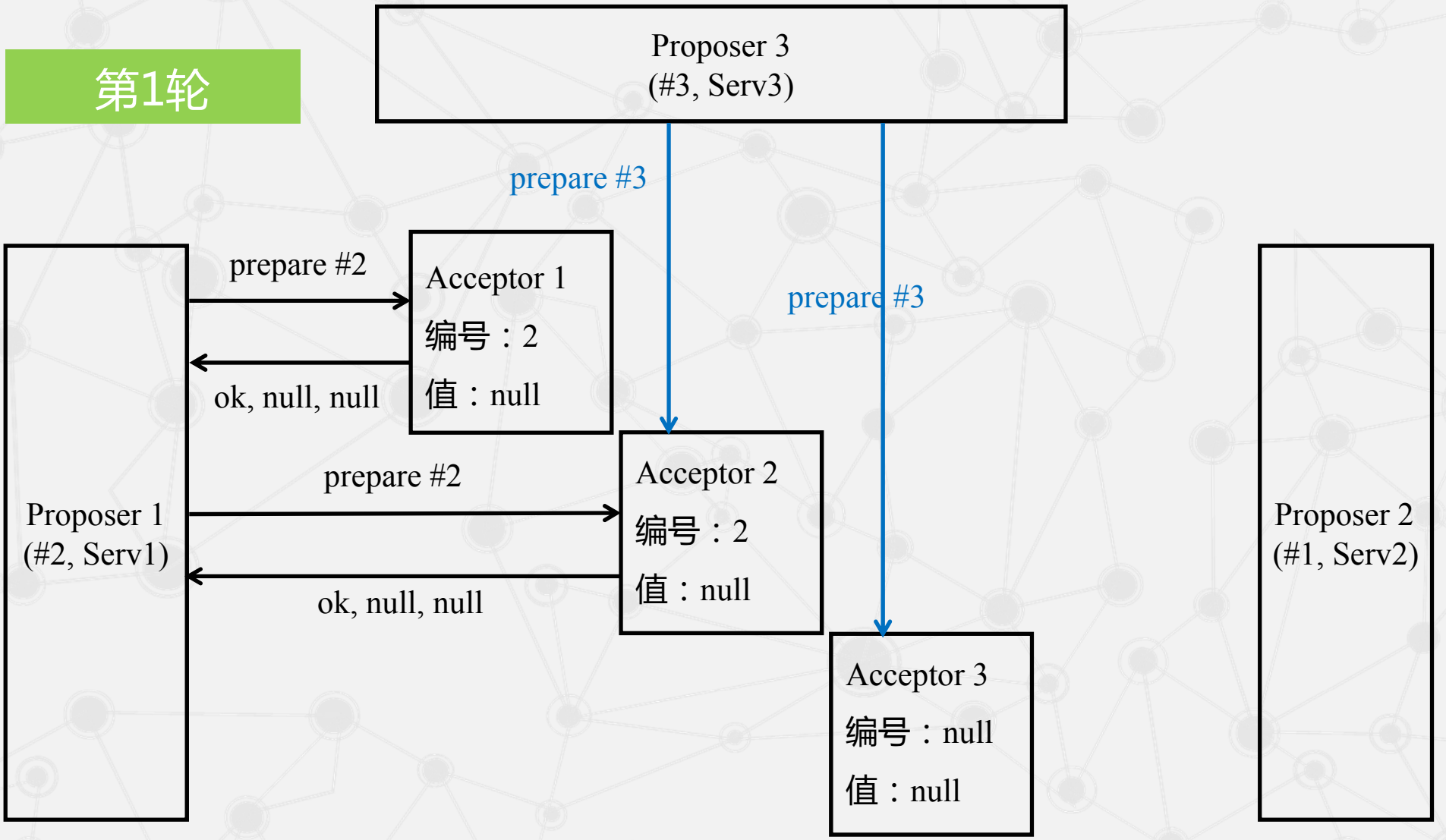

Proposer 3
(#3, Serv3)
prepare #2
Acceptor 1
编号： null
值：null
Proposer 1
(#2, Serv1)
prepare #2
Acceptor 2
编号： null
值：null
Acceptor 3
编号：null
值：null
Proposer 2
(#1, Serv2)

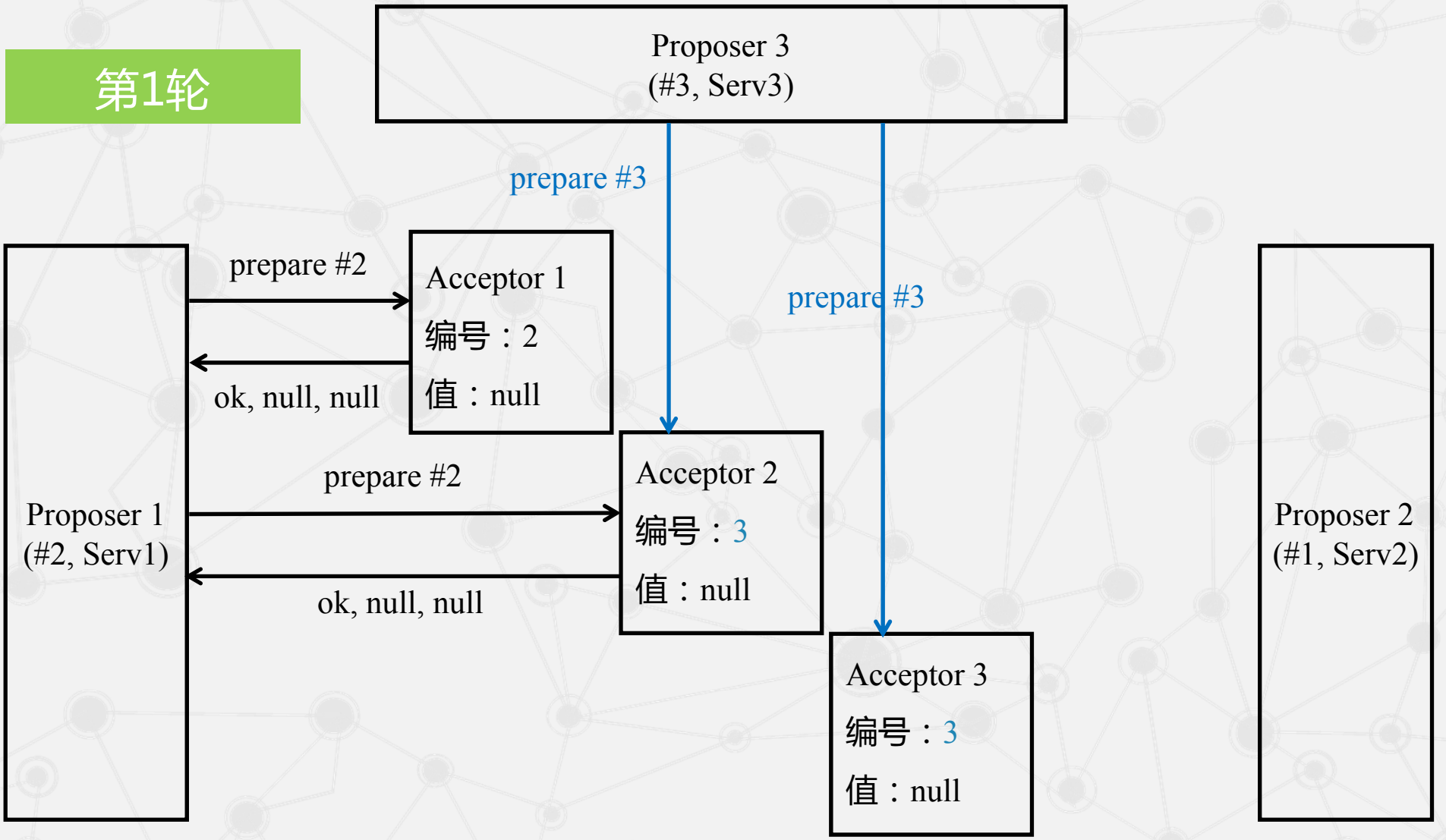
第1轮

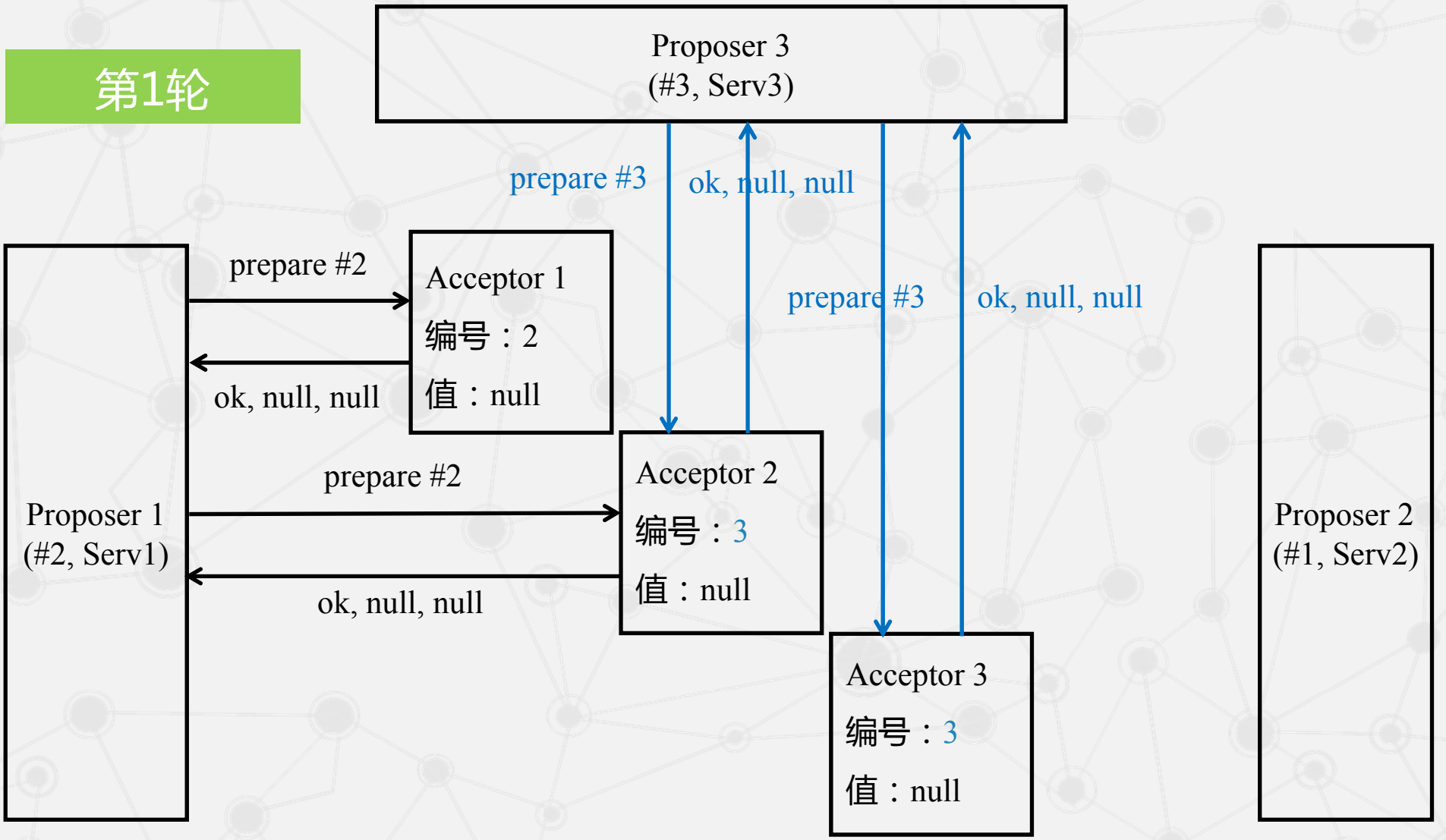

Proposer 3
(#3, Serv3)
prepare #2
Acceptor 1
编号：2
值：null
Proposer 1
(#2, Serv1)
prepare #2
Acceptor 2
编号：2
值：null
Acceptor 3
编号：null
值：null
Proposer 2
(#1, Serv2)

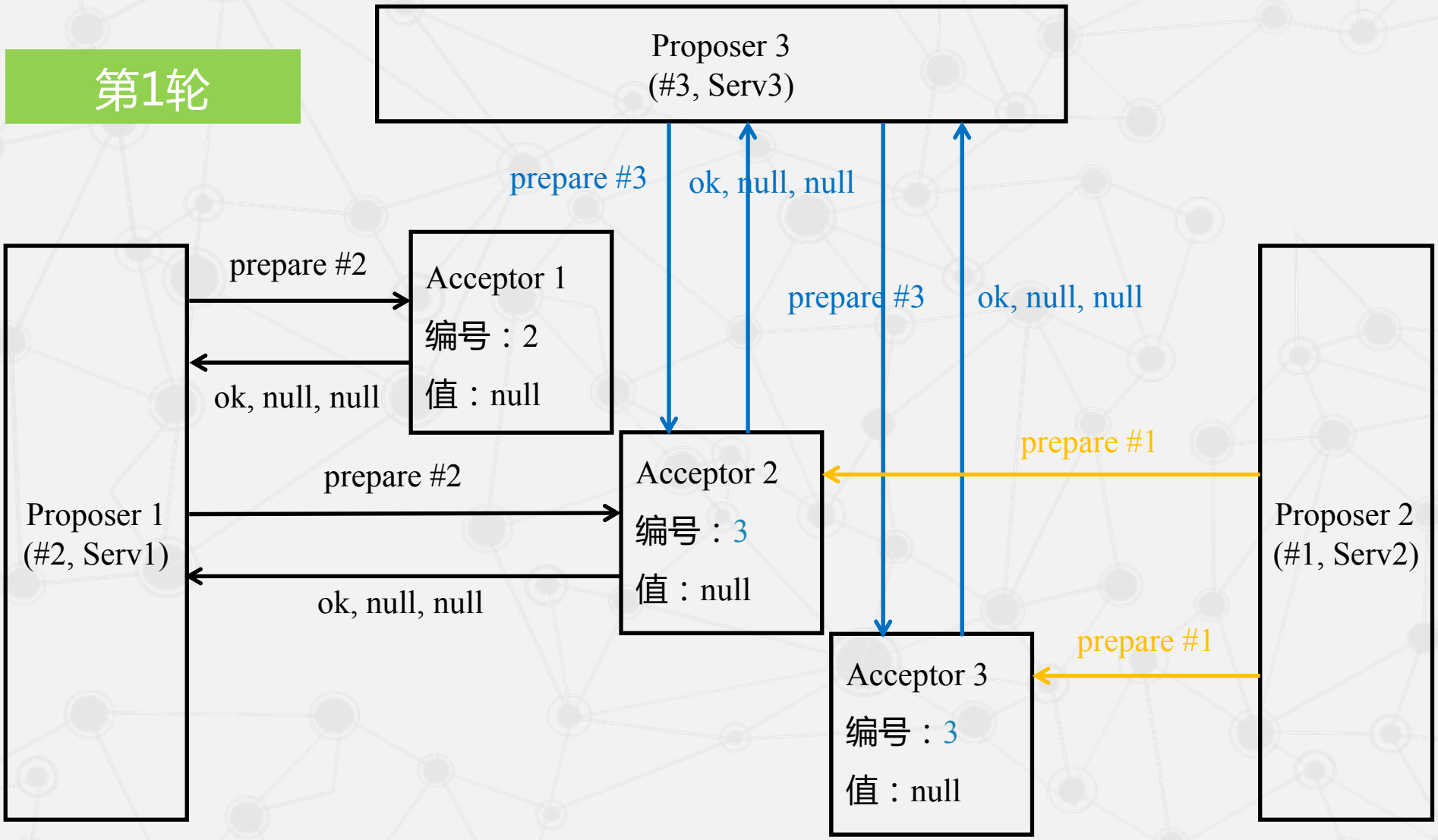
第1轮

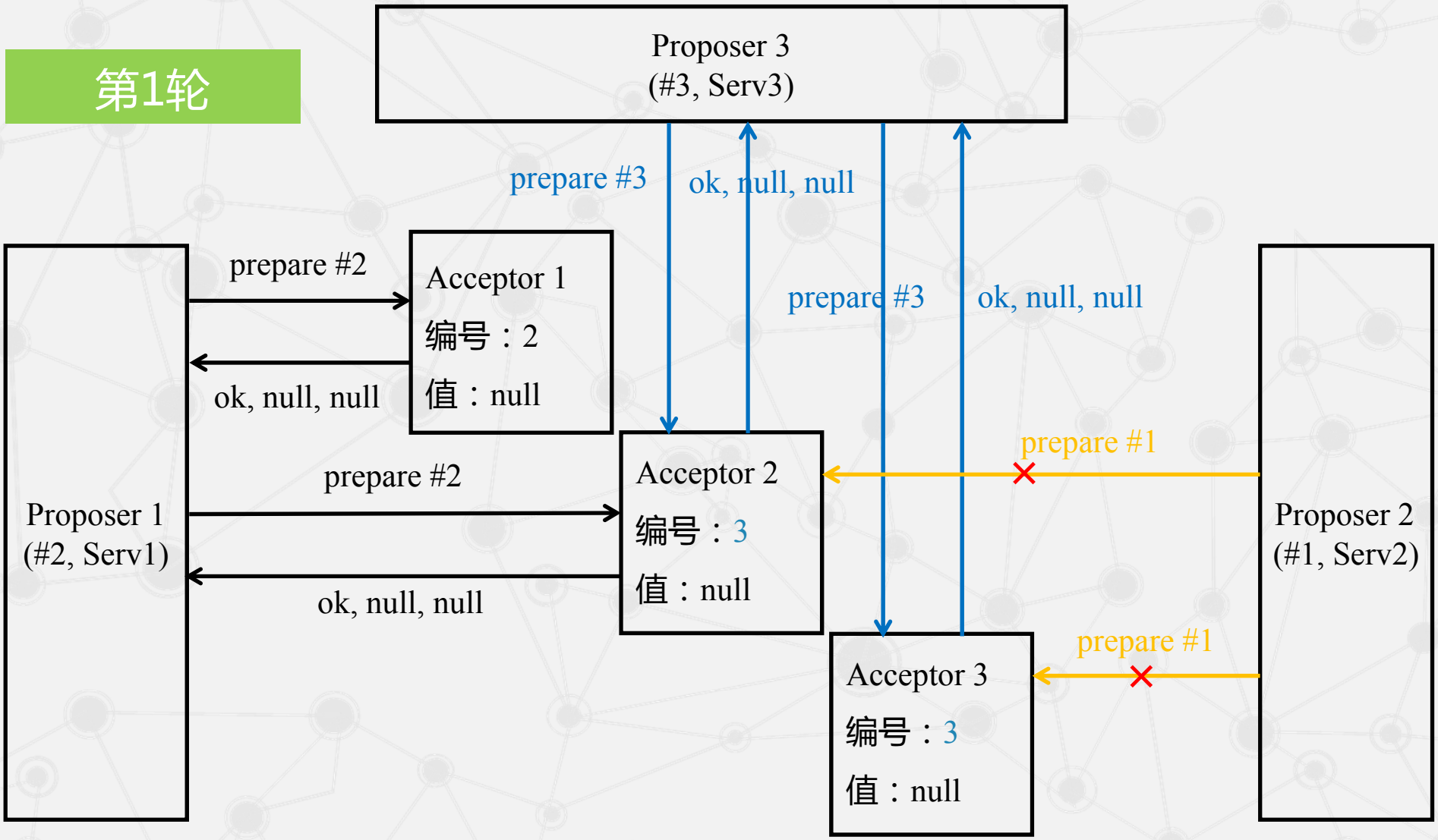

Proposer 3
(#3, Serv3)
Proposer 1
(#2, Serv1)
prepare #2
ok, null, null
Acceptor 1
编号：2
值：null
prepare #2
ok, null, null
Acceptor 2
编号：2
值：null
Acceptor 3
编号：null
值：null
Proposer 2
(#1, Serv2)


第1轮
Proposer 3
(#3, Serv3)
prepare #3
prepare #3
prepare #2
Acceptor 1
编号：2
值：null
ok, null, null
Proposer 1
(#2, Serv1)
prepare #2
Acceptor 2
编号：2
值：null
ok, null, null
Acceptor 3
编号：null
值：null
Proposer 2
(#1, Serv2)


第1轮
Proposer 3
(#3, Serv3)
prepare #3
prepare #3
prepare #2
Acceptor 1
编号：2
值：null
ok, null, null
Proposer 1
(#2, Serv1)
prepare #2
Acceptor 2
编号：3
值：null
ok, null, null
Acceptor 3
编号：3
值：null
Proposer 2
(#1, Serv2)


第1轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
prepare #2
ok, null, null
Acceptor 1
编号：2
值：null
Proposer 1
(#2, Serv1)
prepare #2
ok, null, null
Acceptor 2
编号：3
值：null
Acceptor 3
编号：3
值：null
Proposer 2
(#1, Serv2)


第1轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
prepare #2
ok, null, null
Acceptor 1
编号：2
值：null
prepare #2
ok, null, null
Acceptor 2
编号：3
值：null
prepare #1
prepare #1
Acceptor 3
编号：3
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#1, Serv2)

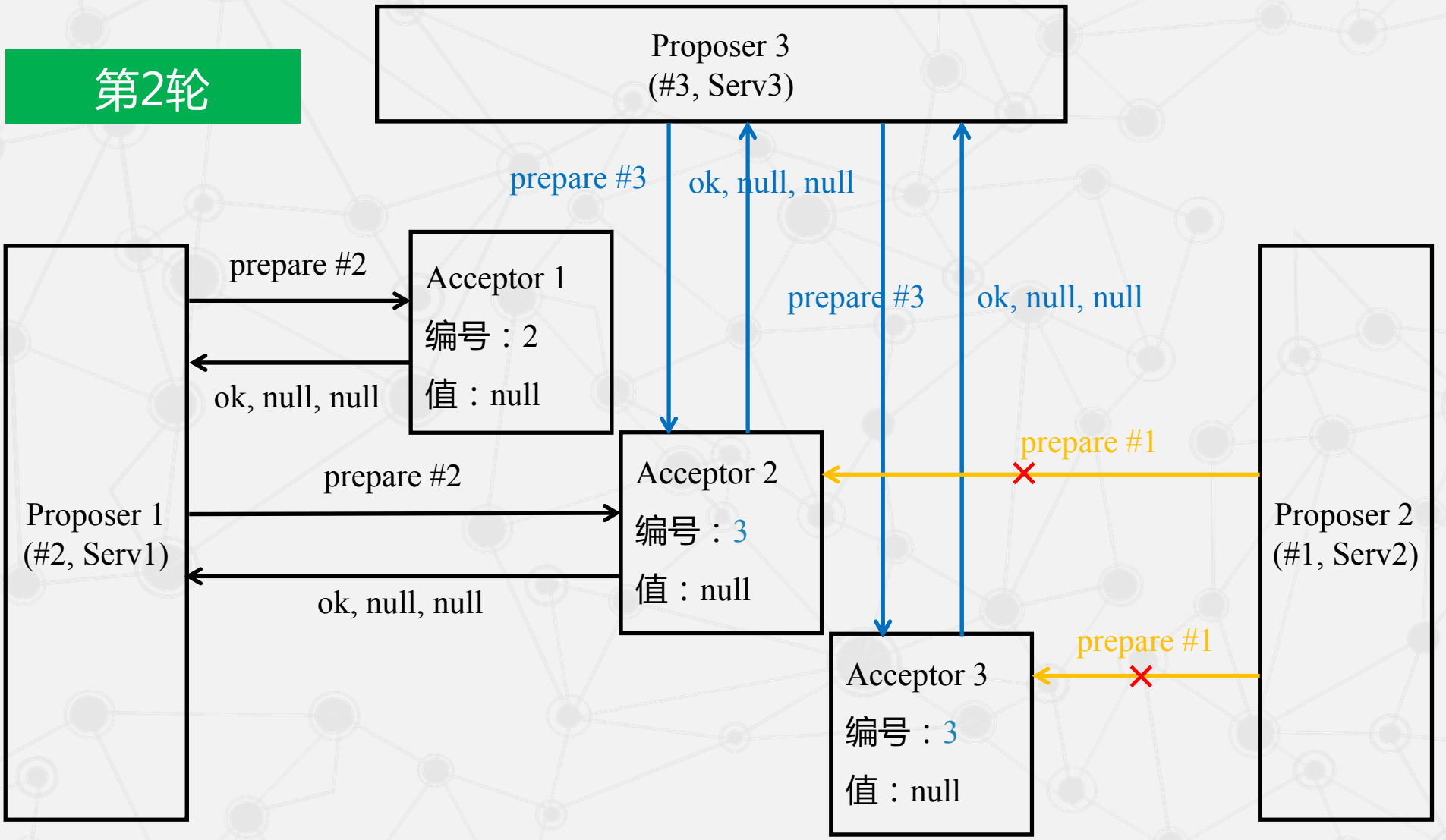

第1轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
prepare #2
Acceptor 1
编号：2
值：null
ok, null, null
prepare #1
prepare #2
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
ok, null, null
Proposer 2
(#1, Serv2)
prepare #1
Acceptor 3
编号：3
值：null

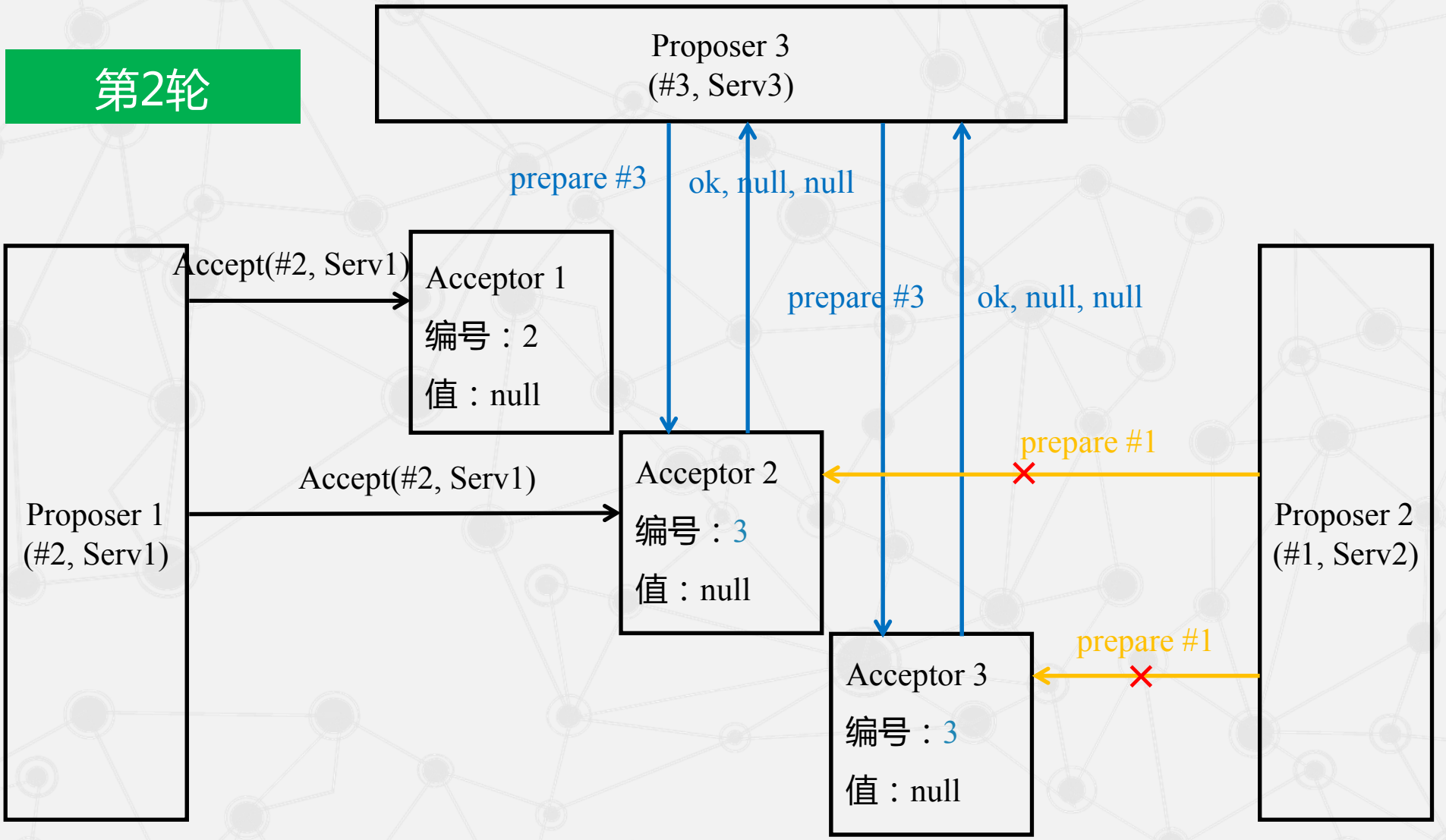
第2轮

各个Proposer开始统计收到的结果：

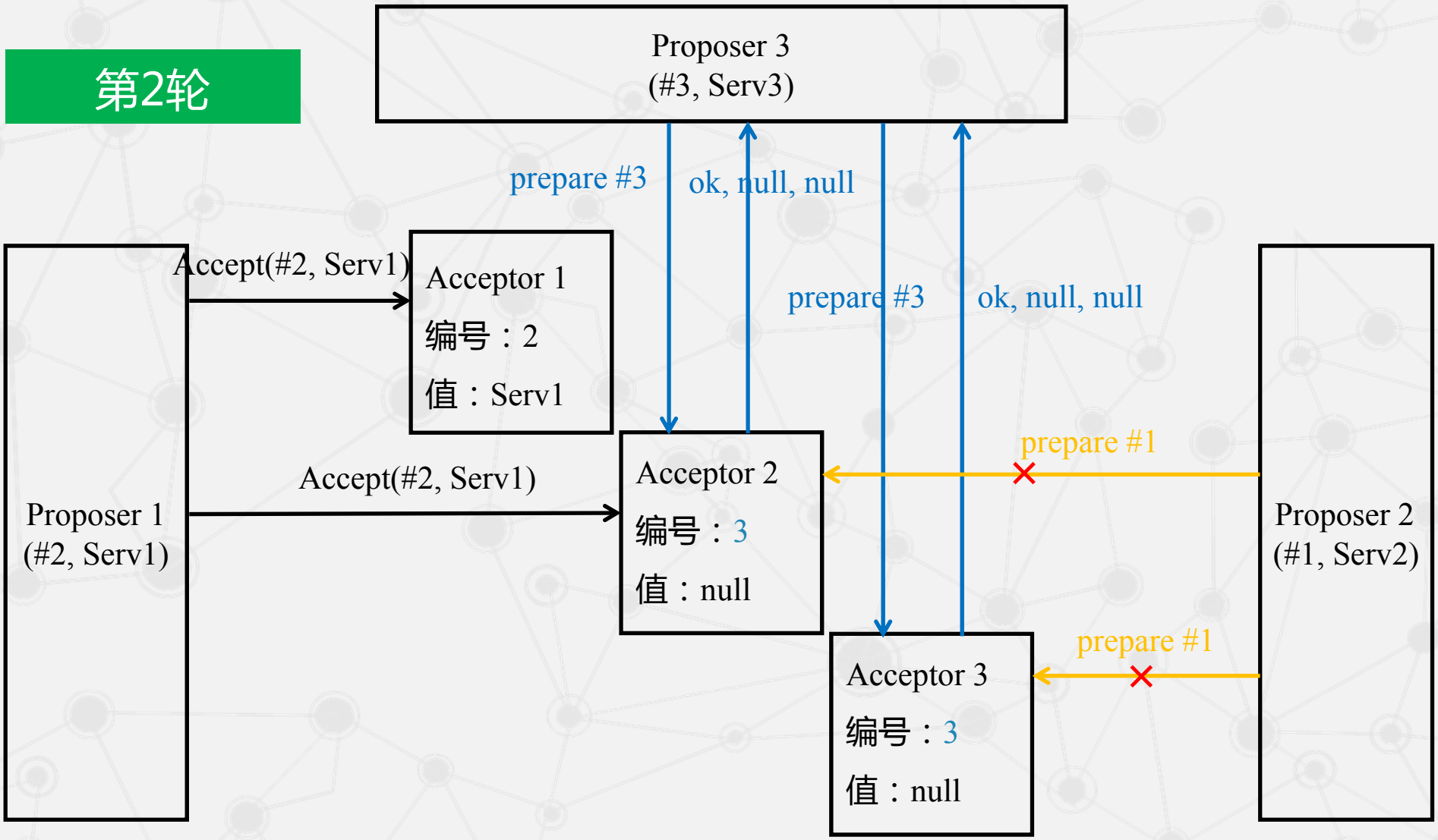
一些Proposer开始进入accept阶段

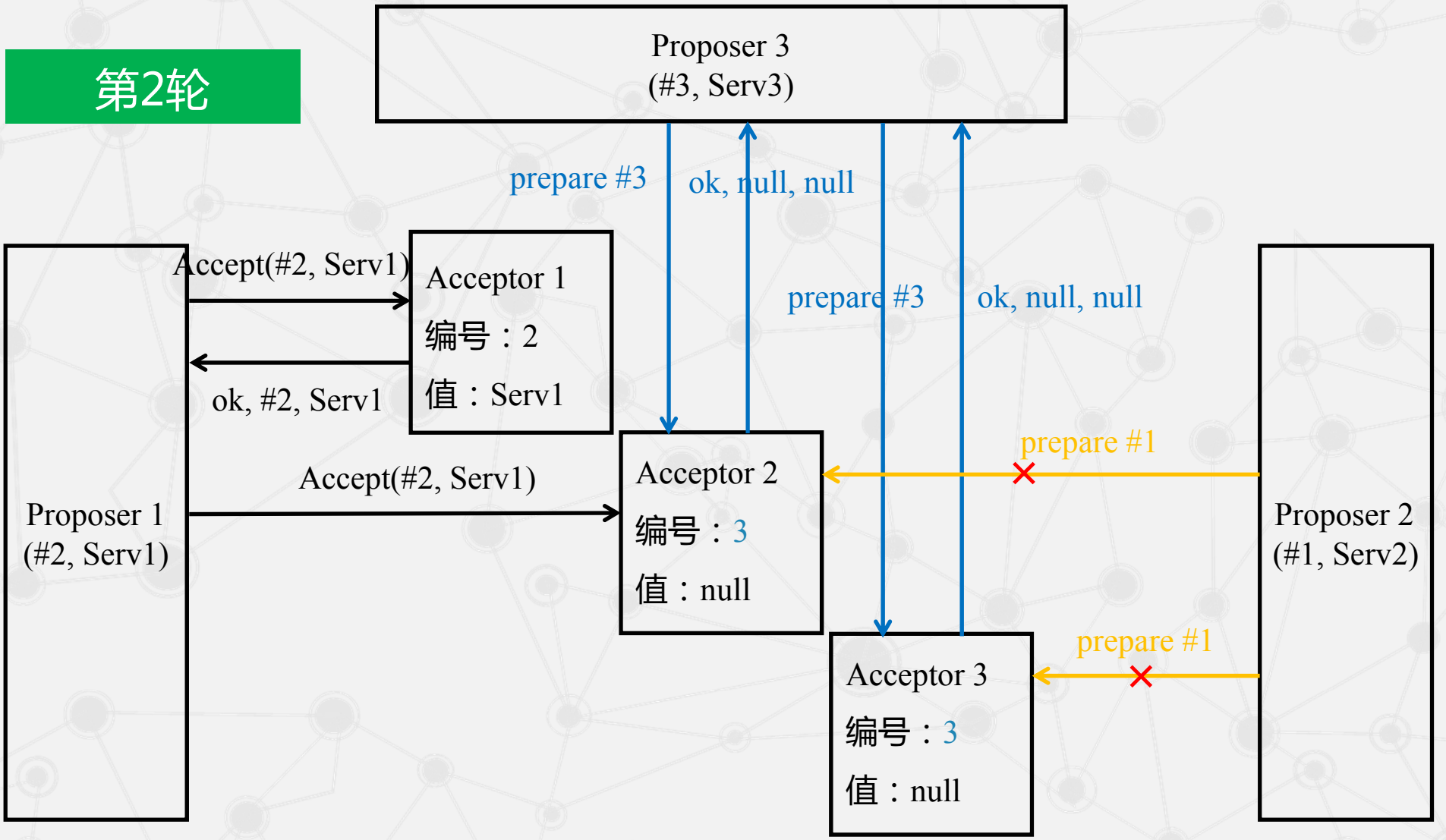
一些Proposer重新提交

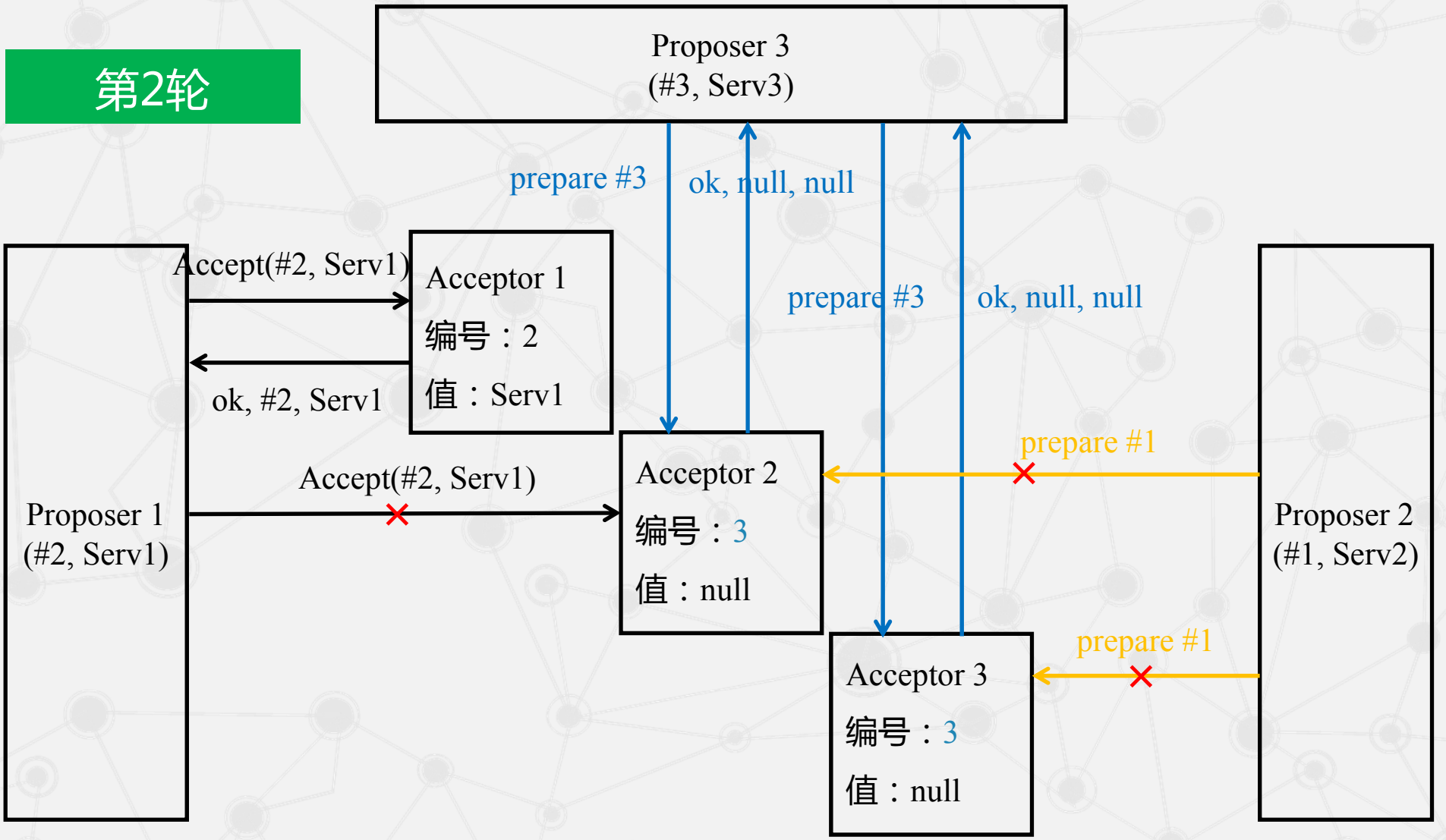

第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
prepare #2
Acceptor 1
编号：2
值：null
ok, null, null
prepare #1
prepare #2
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
ok, null, null
Proposer 2
(#1, Serv2)
prepare #1
Acceptor 3
编号：3
值：null

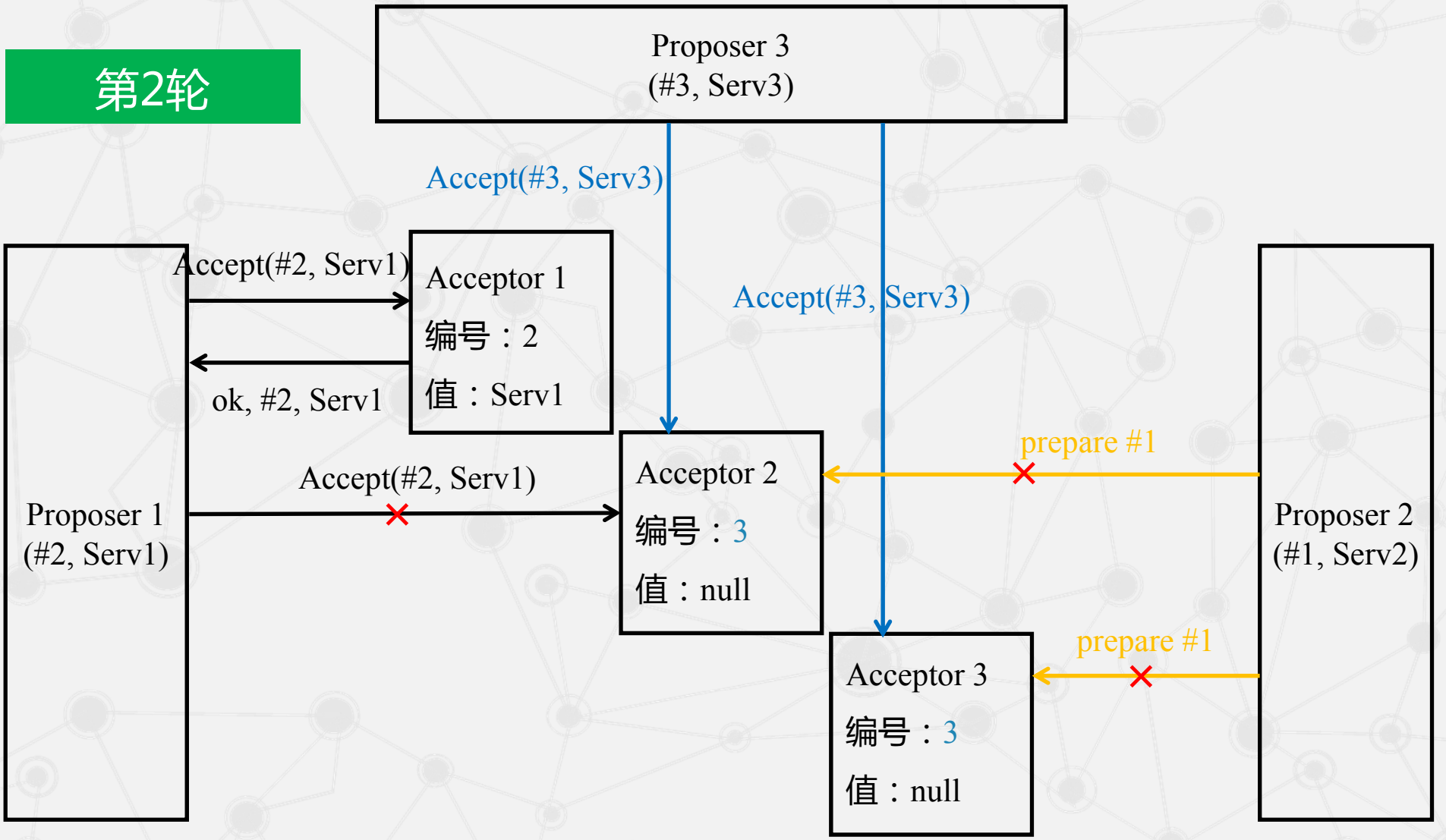

第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
Accept(#2, Serv1)
Acceptor 1
编号：2
值：null
prepare #1
Accept(#2, Serv1)
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#1, Serv2)
prepare #1
Acceptor 3
编号：3
值：null

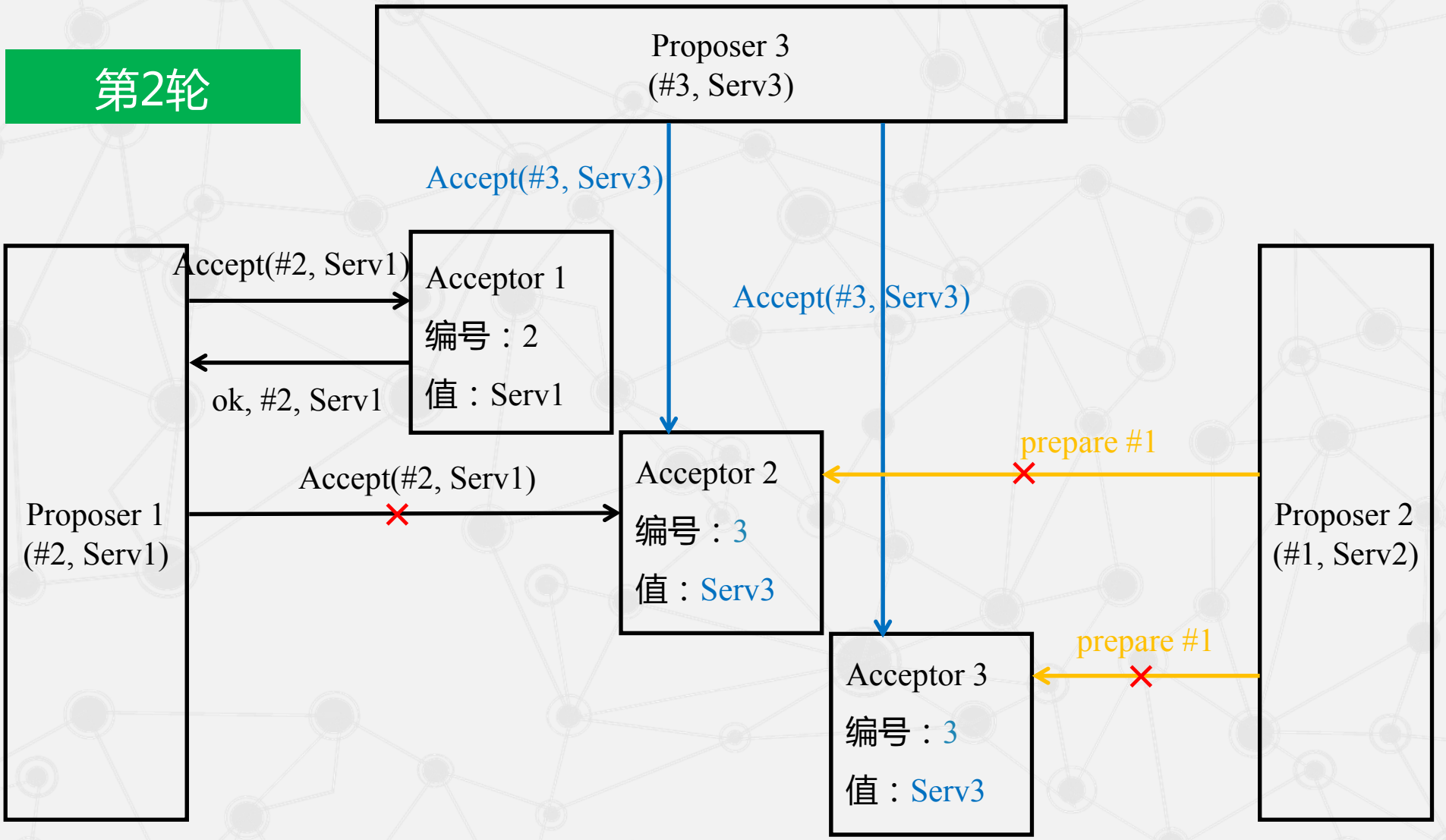
第2轮

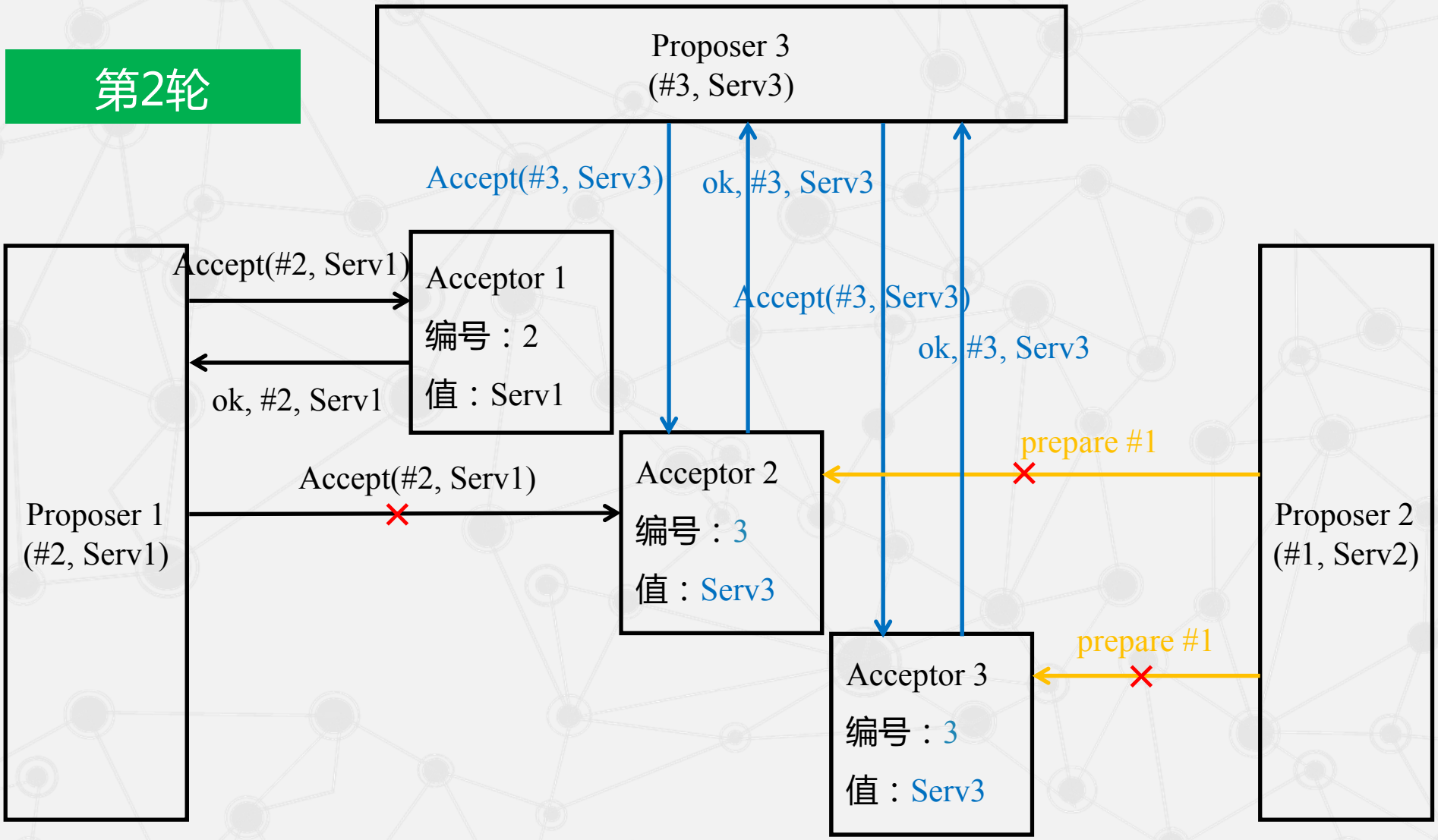

Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
prepare #1
Accept(#2, Serv1)
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#1, Serv2)
prepare #1
Acceptor 3
编号：3
值：null


第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
prepare #1
Accept(#2, Serv1)
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#1, Serv2)
prepare #1
Acceptor 3
编号：3
值：null


第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
prepare #1
Accept(#2, Serv1)
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#1, Serv2)
prepare #1
Acceptor 3
编号：3
值：null


第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#2, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：3
值：null
prepare #1
Proposer 2
(#1, Serv2)
Acceptor 3
编号：3
值：null
prepare #1


第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#2, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：3
值：Serv3
prepare #1
Proposer 2
(#1, Serv2)
Acceptor 3
编号：3
值：Serv3
prepare #1


第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
ok, #3, Serv3
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#3, Serv3)
ok, #3, Serv3
prepare #1
Proposer 1
(#2, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：3
值：Serv3
Proposer 2
(#1, Serv2)
prepare #1
Acceptor 3
编号：3
值：Serv3

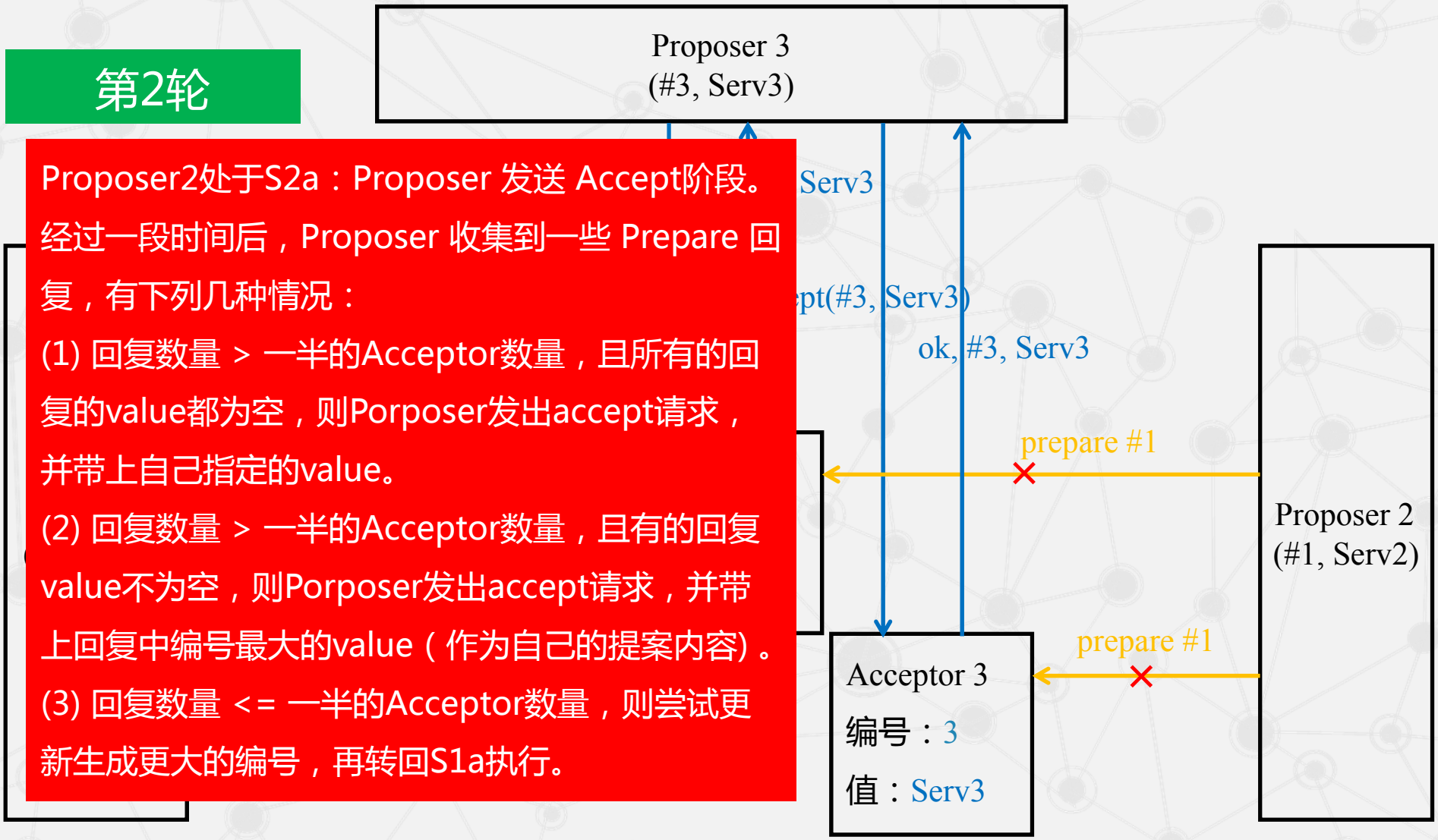
第2轮

Proposer2处于S2a：Proposer 发送 Accept阶段。经过一段时间后，Proposer 收集到一些 Prepare 回复，有下列几种情况：

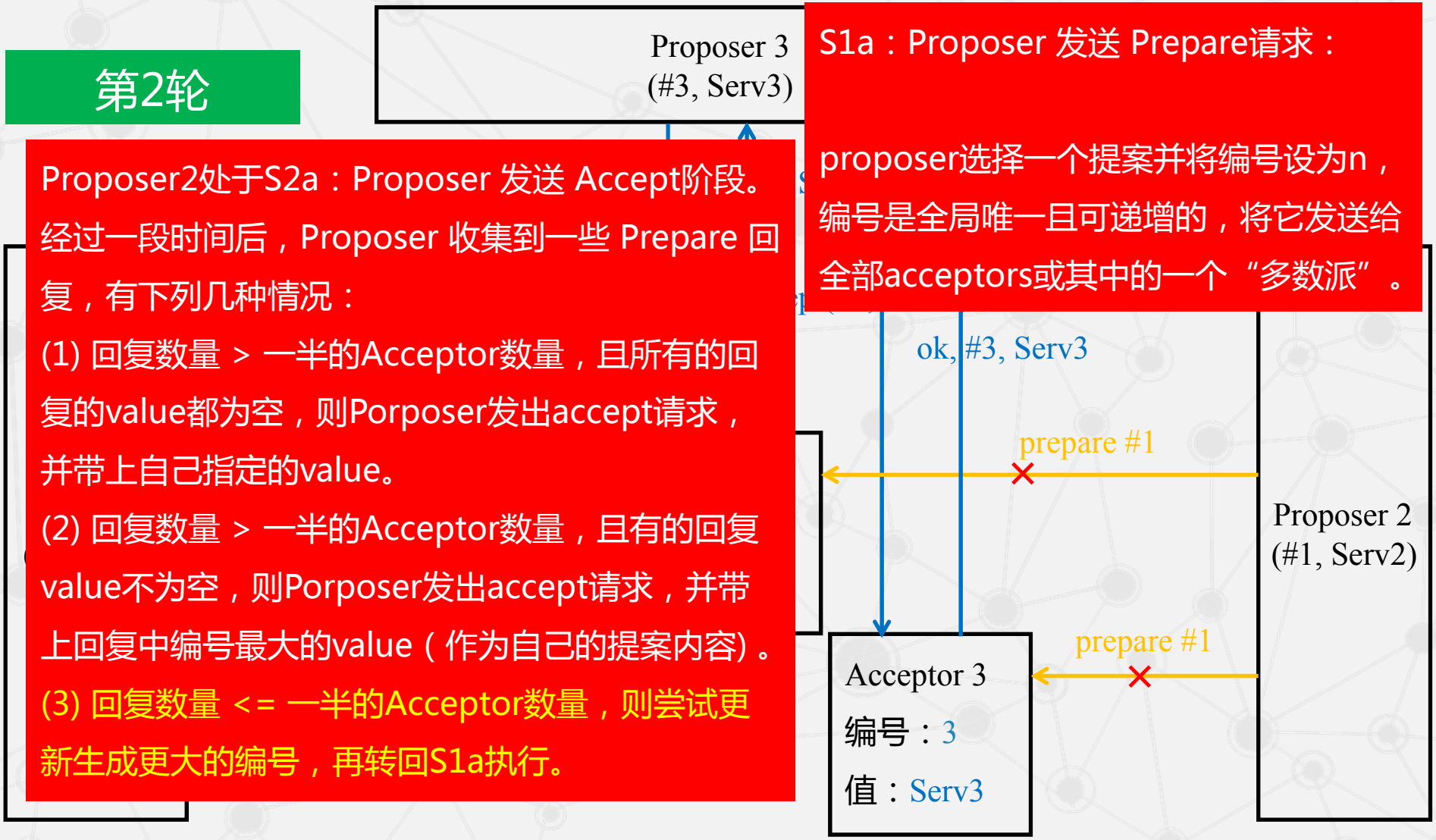
(1) 回复数量 > 一半的Acceptor数量，且所有的回复的value都为空，则Porposer发出accept请求，并带上自己指定的value。

(2) 回复数量 > 一半的Acceptor数量，且有的回复value不为空，则Porposer发出accept请求，并带上回复中编号最大的value（作为自己的提案内容）。

(3) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的编号，再转回S1a执行。



第2轮



S1a：Proposer 发送 Prepare请求：

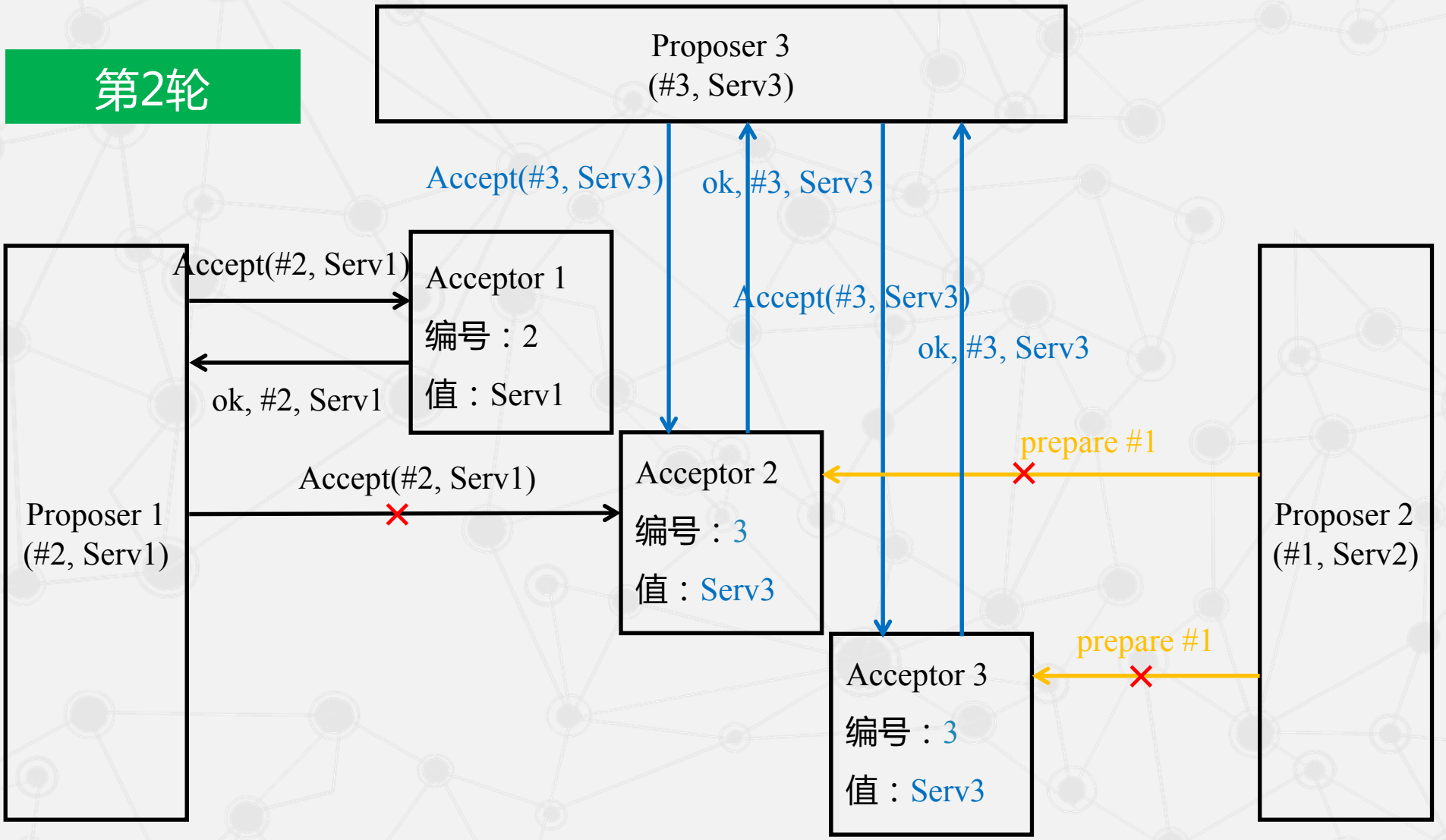
proposer选择一个提案并将编号设为n，编号是全局唯一且可递增的，将它发送给全部acceptors或其中的一个“多数派”。

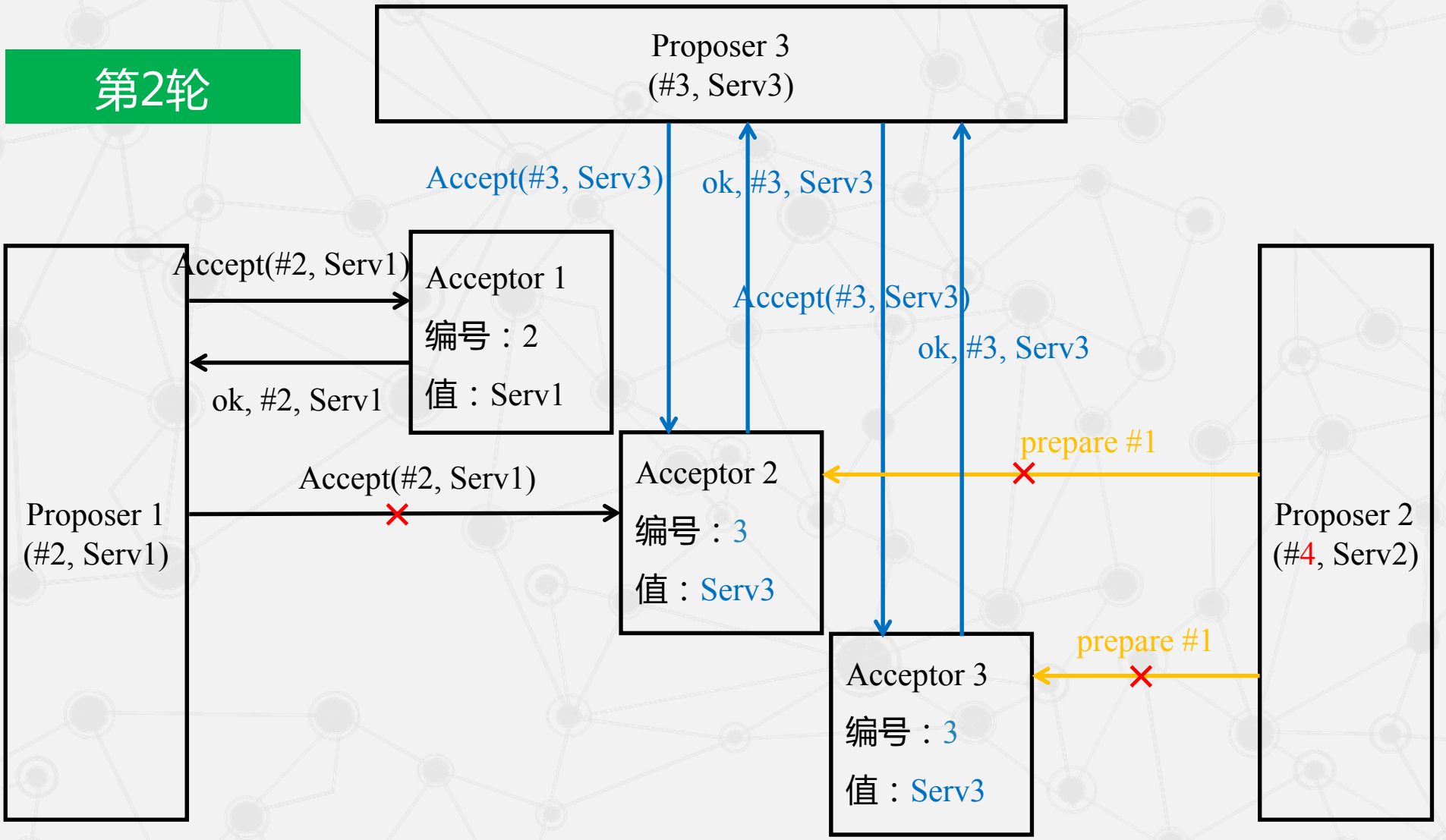
Proposer2处于S2a：Proposer 发送 Accept阶段。经过一段时间后，Proposer 收集到一些 Prepare 回复，有下列几种情况：

(1) 回复数量 > 一半的Acceptor数量，且所有的回复的value都为空，则Porposer发出accept请求，并带上自己指定的value。

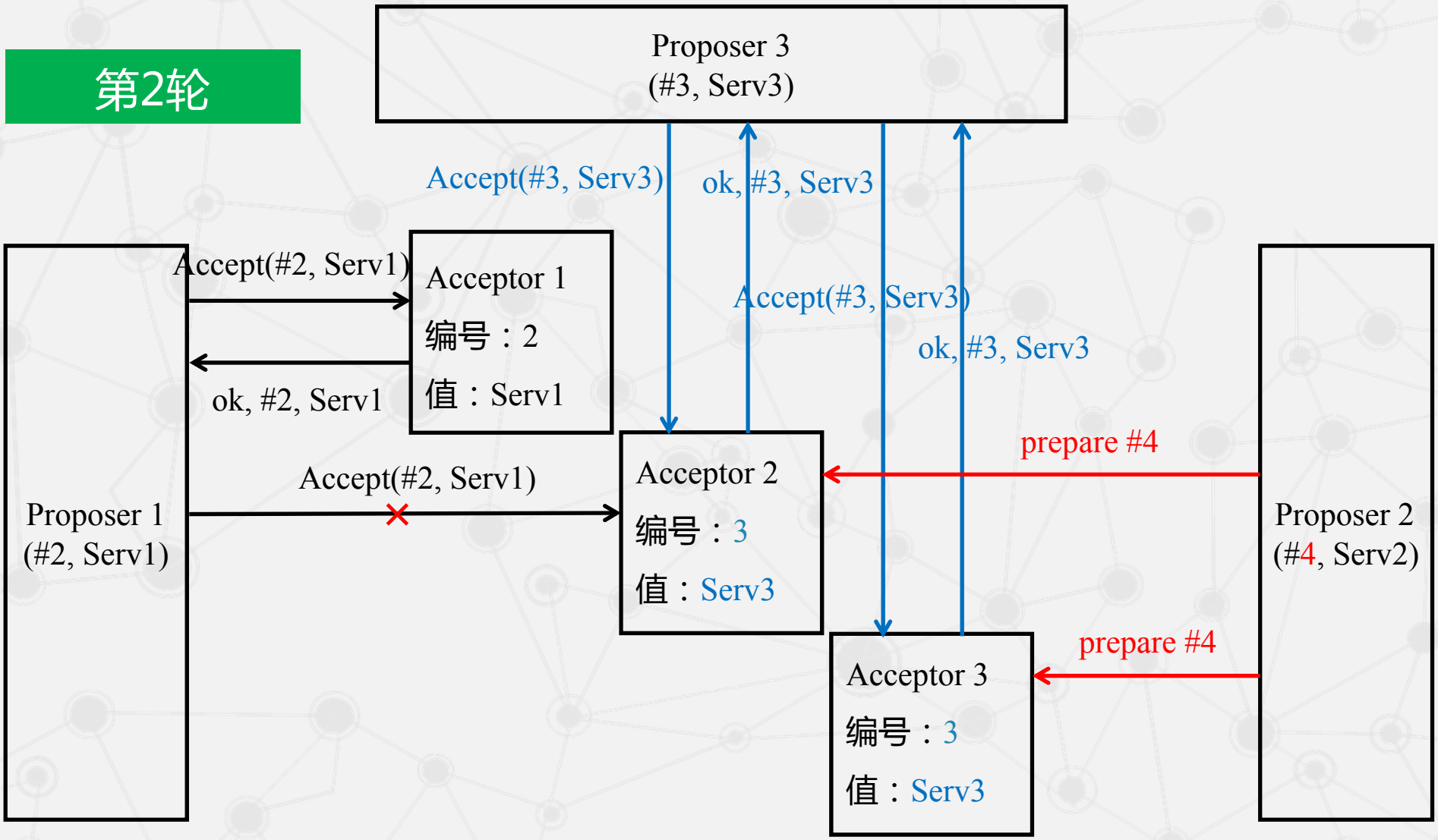
(2) 回复数量 > 一半的Acceptor数量，且有的回复value不为空，则Porposer发出accept请求，并带上回复中编号最大的value（作为自己的提案内容）。

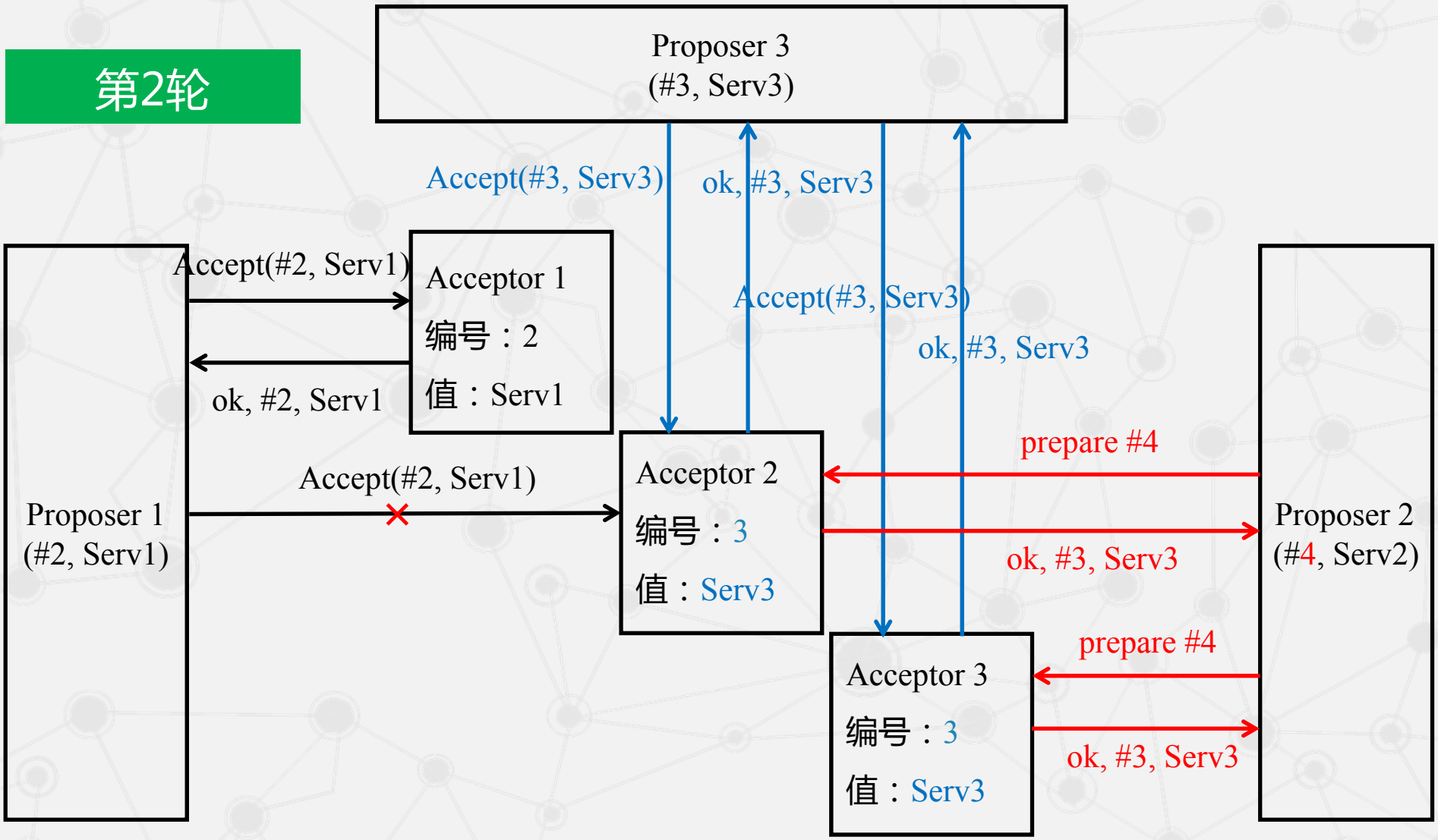
(3) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的编号，再转回S1a执行。


第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
ok, #3, Serv3
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#3, Serv3)
ok, #3, Serv3
prepare #1
Accept(#2, Serv1)
Proposer 1
(#2, Serv1)
Acceptor 2
编号：3
值：Serv3
Proposer 2
(#1, Serv2)
prepare #1
Acceptor 3
编号：3
值：Serv3

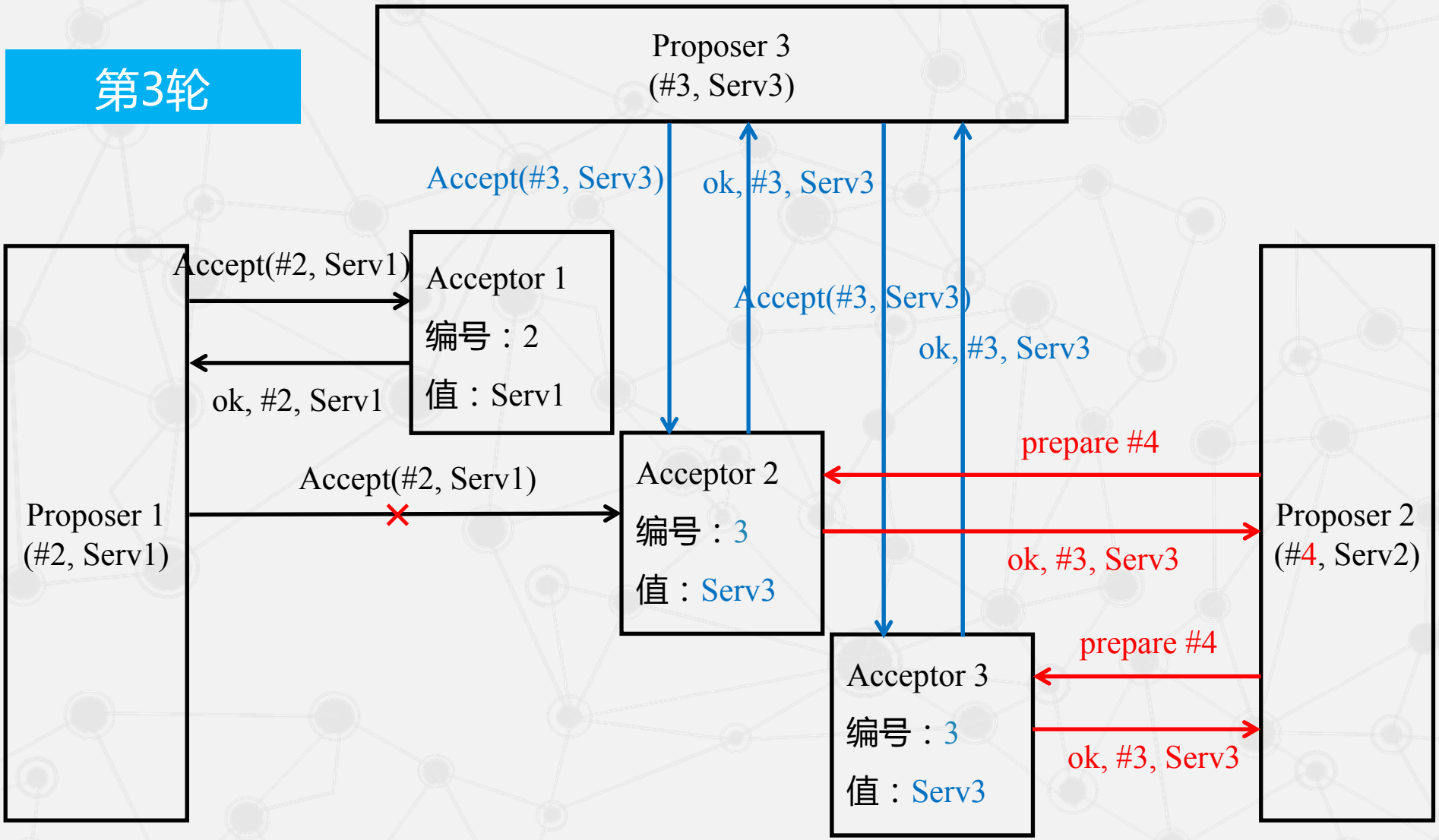

第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
ok, #3, Serv3
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#3, Serv3)
ok, #3, Serv3
prepare #1
Accept(#2, Serv1)
Proposer 1
(#2, Serv1)
Acceptor 2
编号：3
值：Serv3
Proposer 2
(#4, Serv2)
prepare #1
Acceptor 3
编号：3
值：Serv3

第2轮

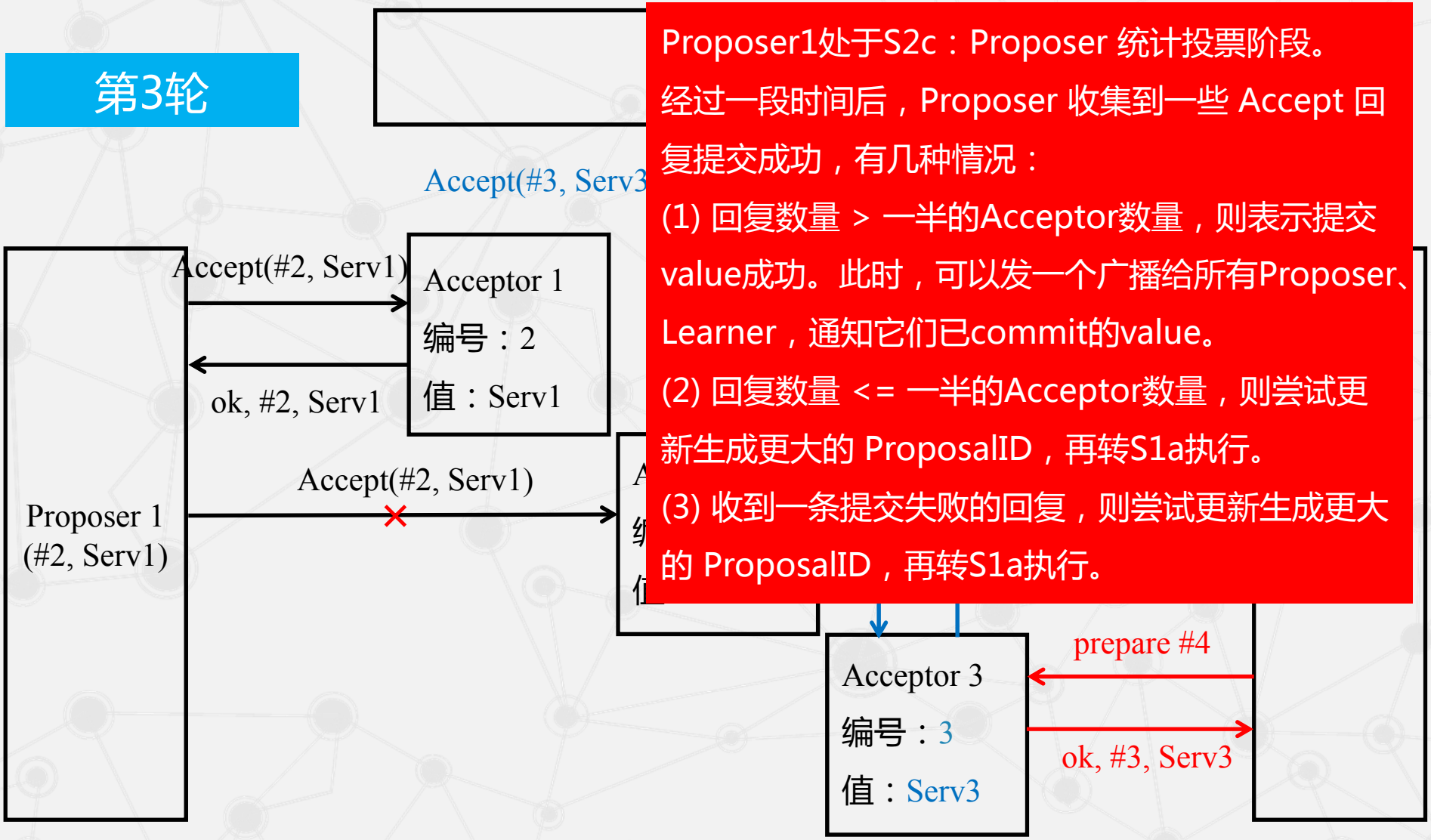

Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
ok, #3, Serv3
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#3, Serv3)
ok, #3, Serv3
prepare #4
Proposer 1
(#2, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：3
值：Serv3
Proposer 2
(#4, Serv2)
prepare #4
Acceptor 3
编号：3
值：Serv3


第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
ok, #3, Serv3
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#3, Serv3)
ok, #3, Serv3
prepare #4
Proposer 1
(#2, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：3
值：Serv3
ok, #3, Serv3
Proposer 2
(#4, Serv2)
prepare #4
Acceptor 3
编号：3
值：Serv3
ok, #3, Serv3

第3轮


第3轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
ok, #3, Serv3
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#3, Serv3)
ok, #3, Serv3
prepare #4
Proposer 1
(#2, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：3
值：Serv3
ok, #3, Serv3
Proposer 2
(#4, Serv2)
prepare #4
Acceptor 3
编号：3
值：Serv3
ok, #3, Serv3

第3轮



Proposer1处于S2c：Proposer 统计投票阶段。

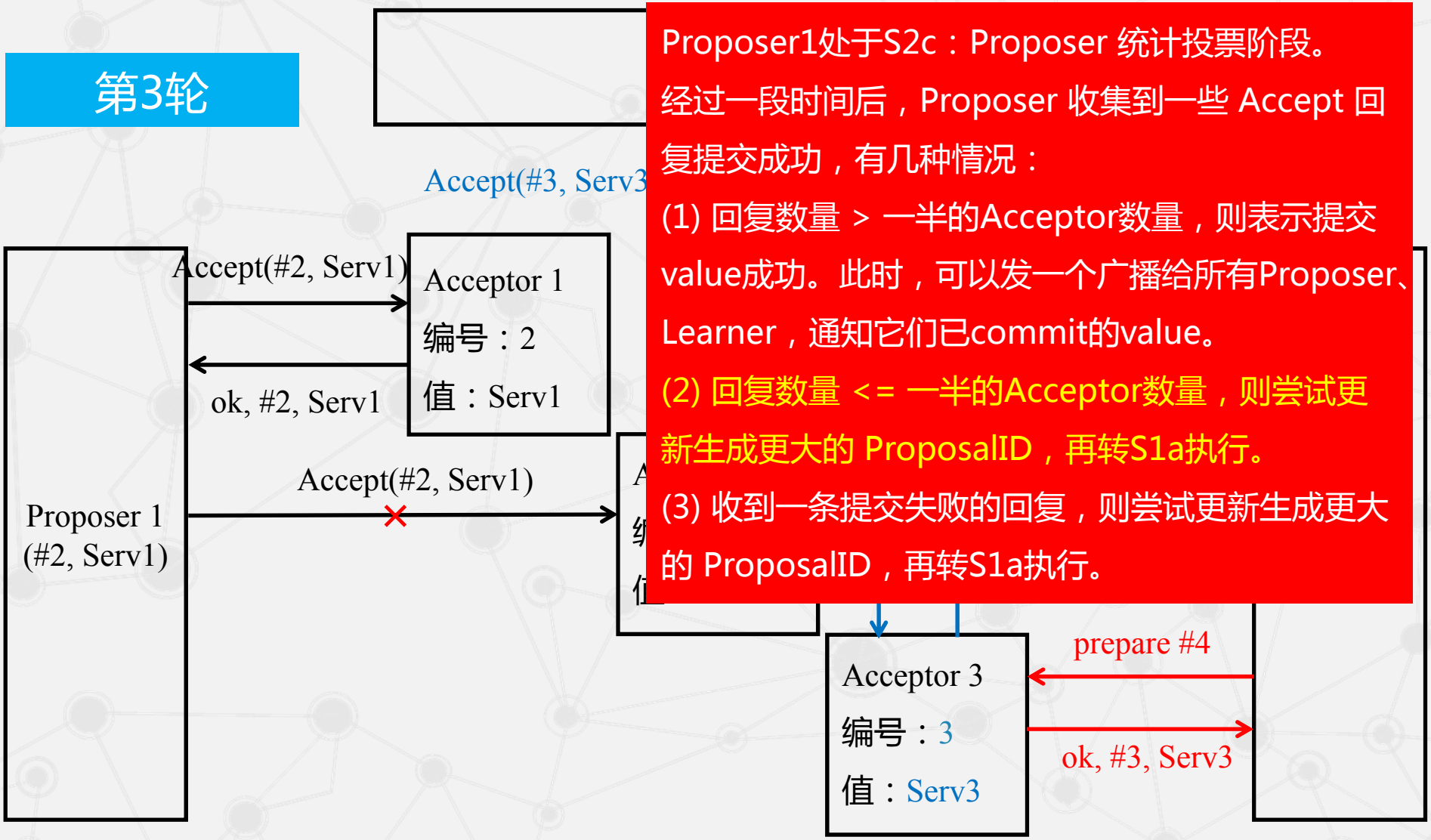
经过一段时间后，Proposer 收集到一些 Accept 回复提交成功，有几种情况：

(1) 回复数量 > 一半的Acceptor数量，则表示提交value成功。此时，可以发一个广播给所有Proposer、Learner，通知它们已commit的value。

(2) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的 ProposalID，再转S1a执行。

(3) 收到一条提交失败的回复，则尝试更新生成更大的 ProposalID，再转S1a执行。

第3轮



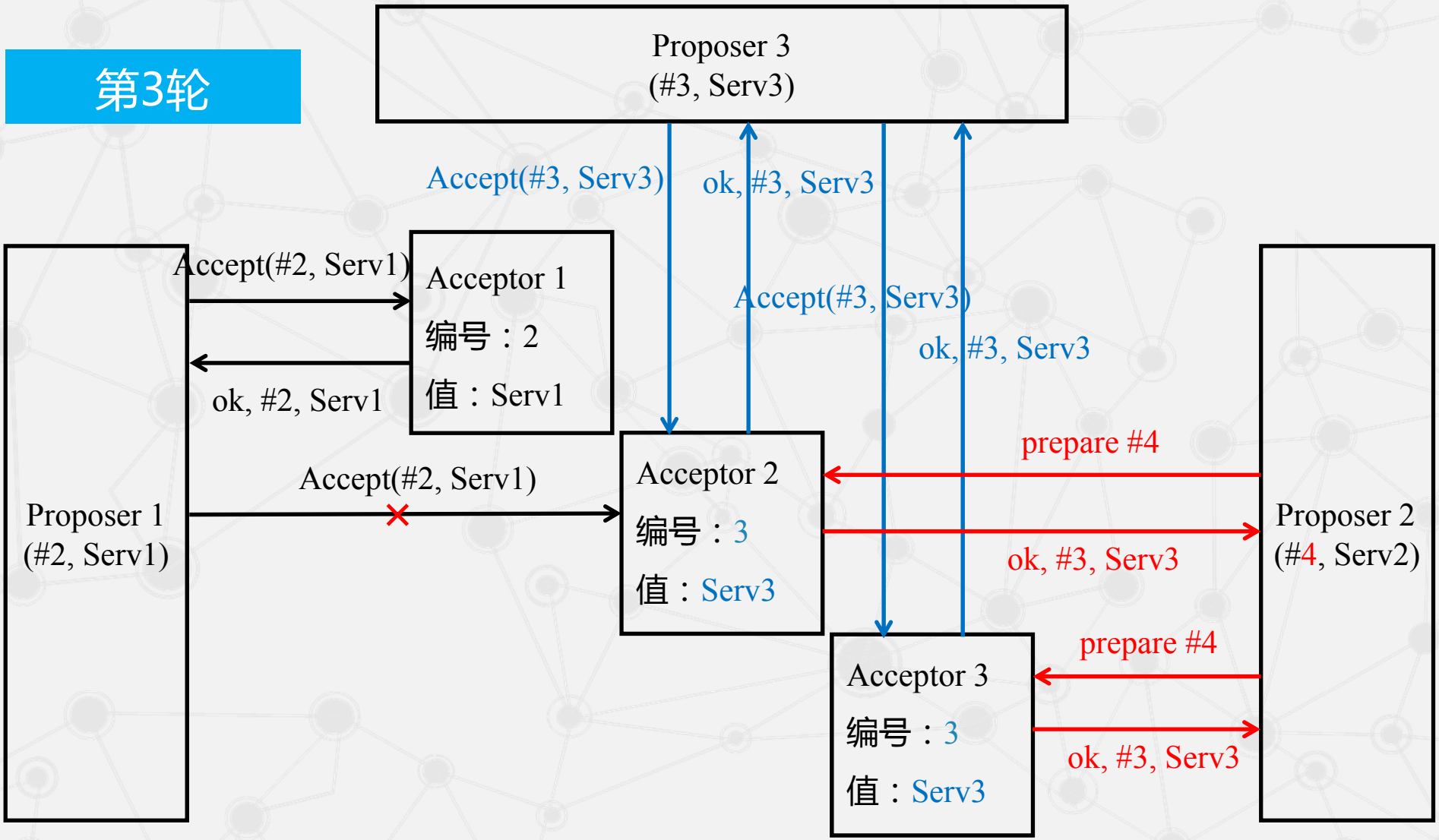
Proposer1处于S2c：Proposer 统计投票阶段。

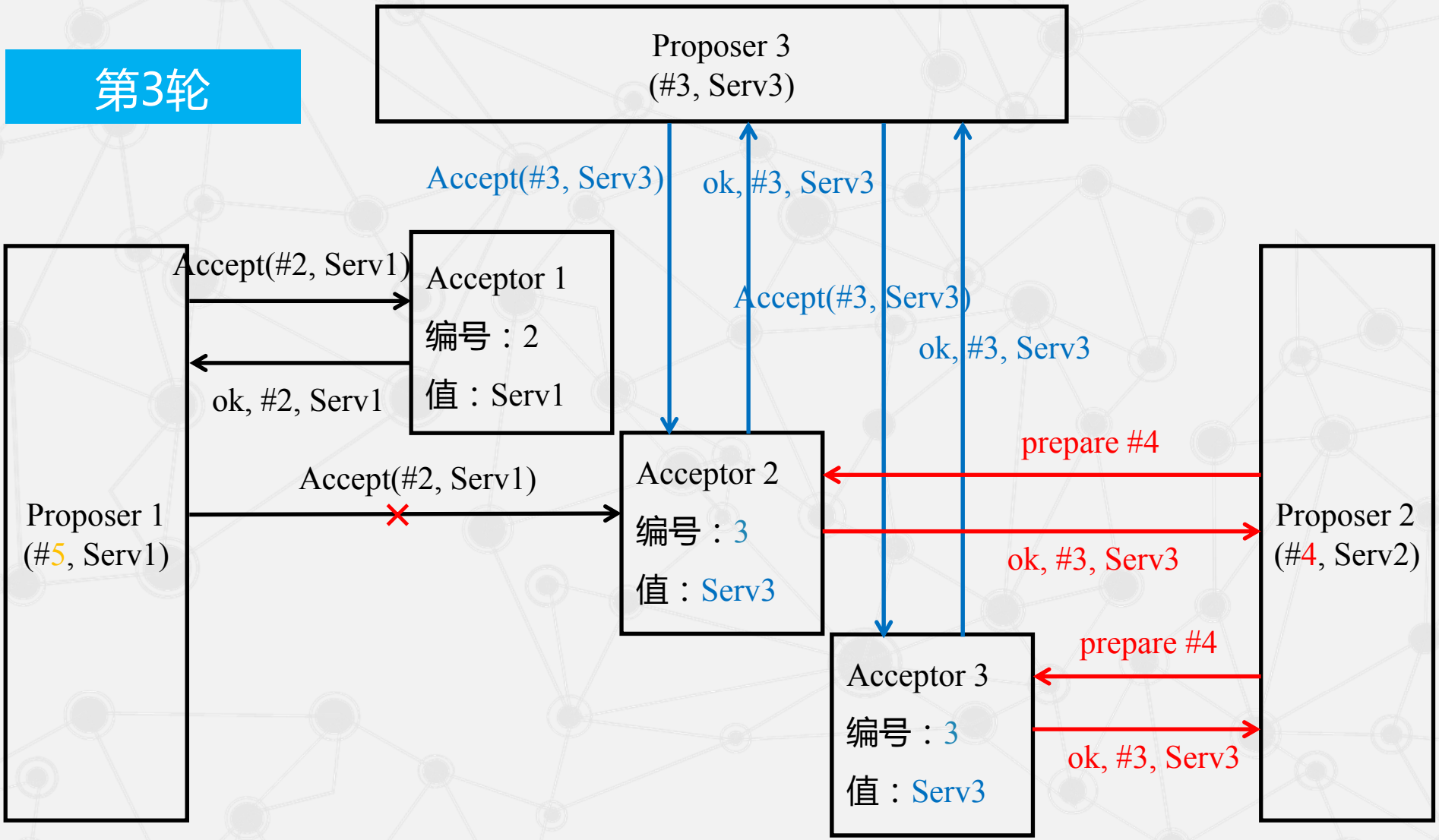
经过一段时间后，Proposer 收集到一些 Accept 回复提交成功，有几种情况：

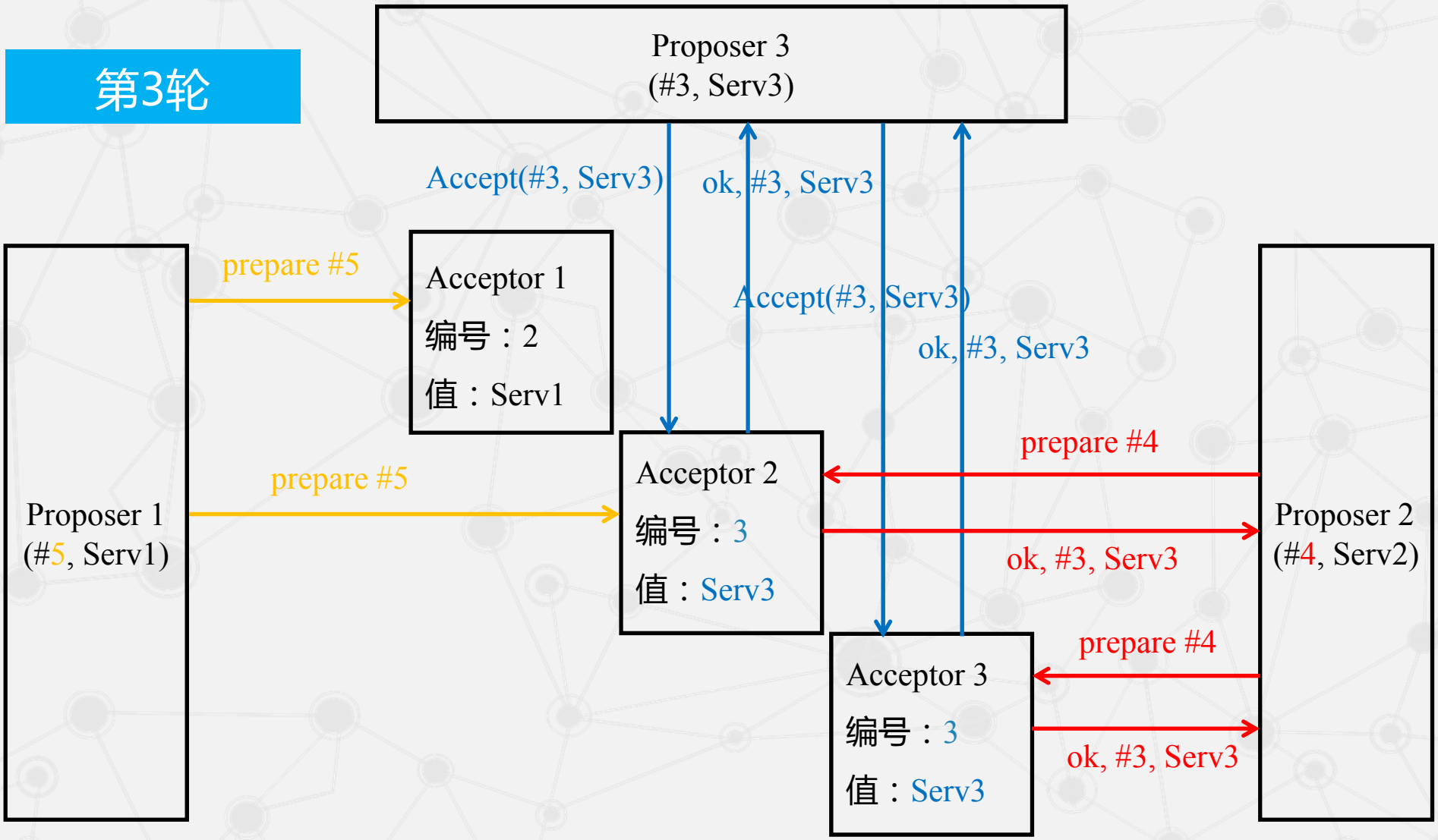
(1) 回复数量 > 一半的Acceptor数量，则表示提交value成功。此时，可以发一个广播给所有Proposer、Learner，通知它们已commit的value。

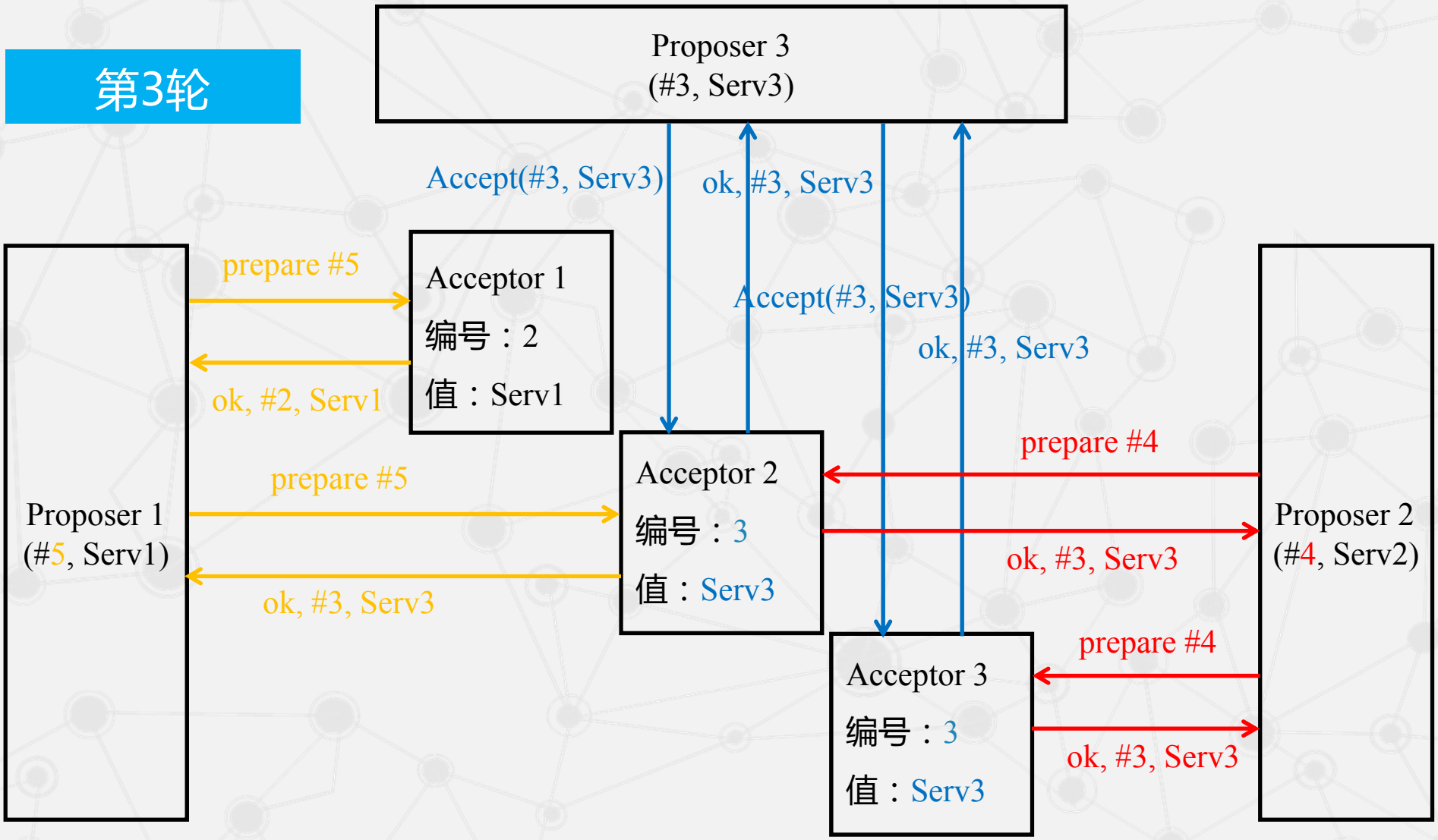
(2) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的 ProposalID，再转S1a执行。

(3) 收到一条提交失败的回复，则尝试更新生成更大的 ProposalID，再转S1a执行。

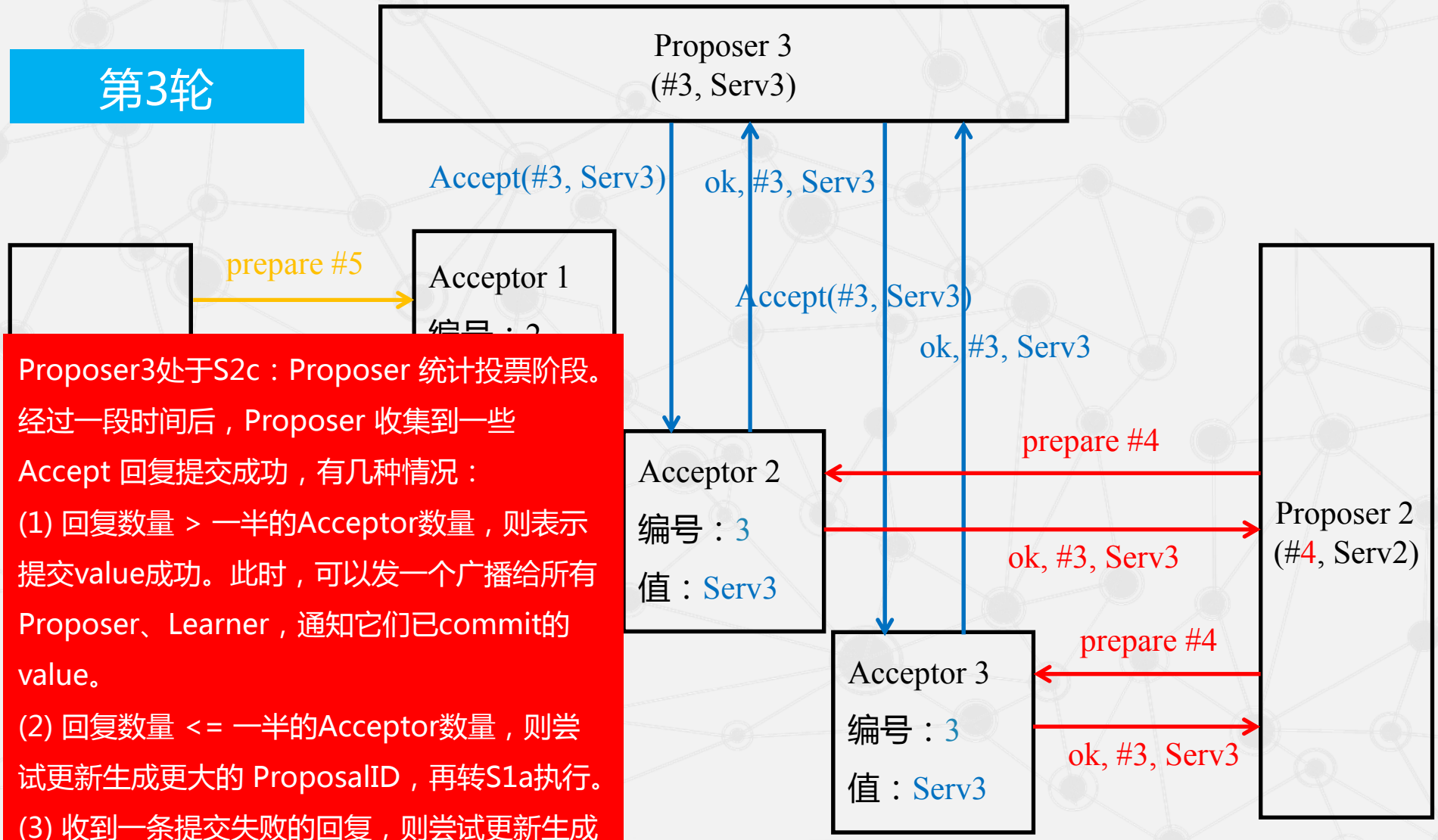

第3轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
ok, #3, Serv3
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#3, Serv3)
ok, #3, Serv3
prepare #4
Proposer 1
(#2, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：3
值：Serv3
ok, #3, Serv3
Proposer 2
(#4, Serv2)
prepare #4
Acceptor 3
编号：3
值：Serv3
ok, #3, Serv3


第3轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
ok, #3, Serv3
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#3, Serv3)
ok, #3, Serv3
prepare #4
Proposer 1
(#5, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：3
值：Serv3
ok, #3, Serv3
Proposer 2
(#4, Serv2)
prepare #4
Acceptor 3
编号：3
值：Serv3
ok, #3, Serv3


第3轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
ok, #3, Serv3
prepare #5
Acceptor 1
编号：2
值：Serv1
Accept(#3, Serv3)
ok, #3, Serv3
prepare #4
Proposer 1
(#5, Serv1)
prepare #5
Acceptor 2
编号：3
值：Serv3
Proposer 2
(#4, Serv2)
ok, #3, Serv3
prepare #4
Acceptor 3
编号：3
值：Serv3
ok, #3, Serv3


第3轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
ok, #3, Serv3
prepare #5
Acceptor 1
编号：2
值：Serv1
Accept(#3, Serv3)
ok, #3, Serv3
ok, #2, Serv1
prepare #4
prepare #5
Acceptor 2
编号：3
值：Serv3
Proposer 1
(#5, Serv1)
Proposer 2
(#4, Serv2)
ok, #3, Serv3
ok, #3, Serv3
prepare #4
Acceptor 3
编号：3
值：Serv3
ok, #3, Serv3

第3轮



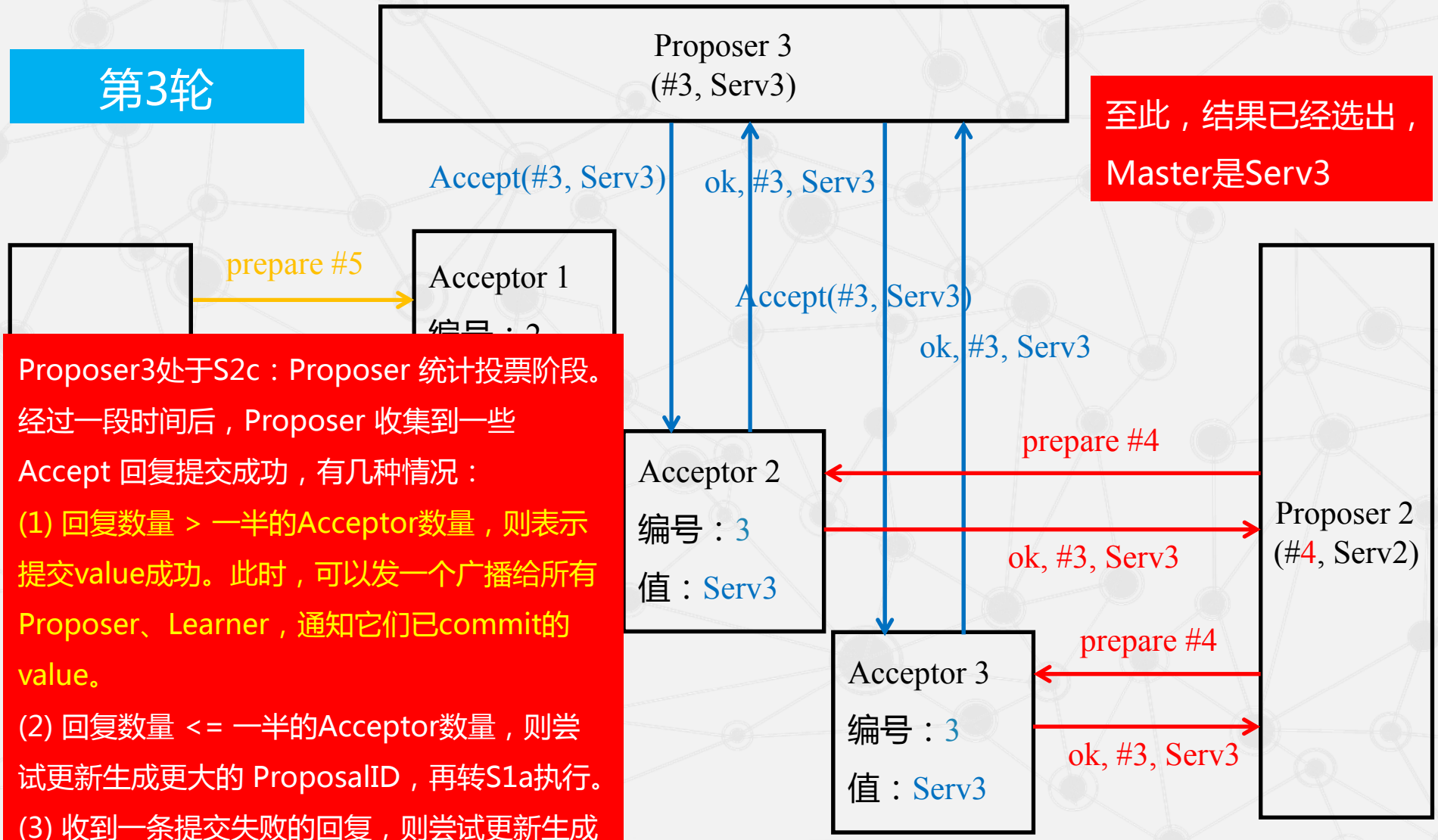
Proposer3处于S2c：Proposer 统计投票阶段。经过一段时间后，Proposer 收集到一些 Accept 回复提交成功，有几种情况：

(1) 回复数量 > 一半的Acceptor数量，则表示提交value成功。此时，可以发一个广播给所有Proposer、Learner，通知它们已commit的value。

(2) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的 ProposalID，再转S1a执行。

(3) 收到一条提交失败的回复，则尝试更新生成更大的 ProposalID，再转S1a执行。

第3轮



Proposer3处于S2c：Proposer 统计投票阶段。经过一段时间后，Proposer 收集到一些 Accept 回复提交成功，有几种情况：

(1) 回复数量 > 一半的Acceptor数量，则表示提交value成功。此时，可以发一个广播给所有Proposer、Learner，通知它们已commit的value。

(2) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的 ProposalID，再转S1a执行。

(3) 收到一条提交失败的回复，则尝试更新生成更大的 ProposalID，再转S1a执行。

【例子2】

在server1、server2和server3中选一个Master

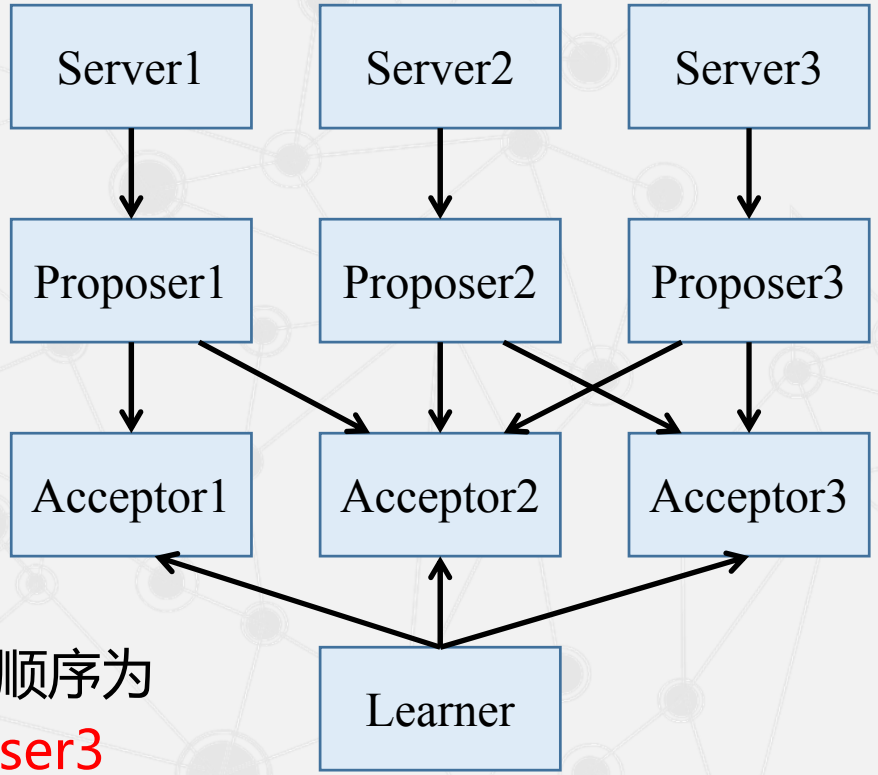
已知：

Proposer1的编号为2；

Proposer2的编号为1；

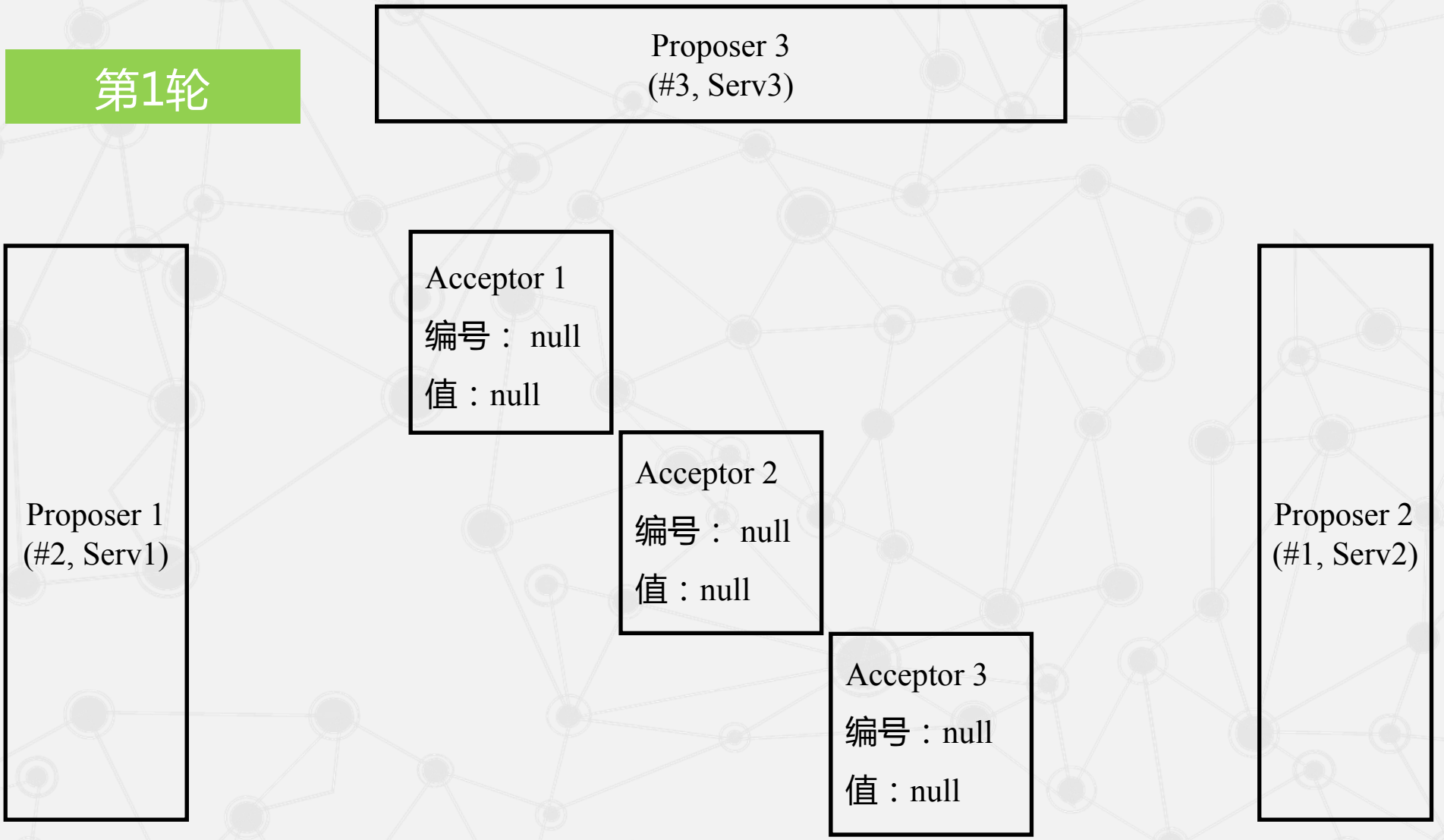
Proposer3的编号为3；

Proposer向Acceptor提交决议的顺序为Proposer1、Proposer2、Proposer3

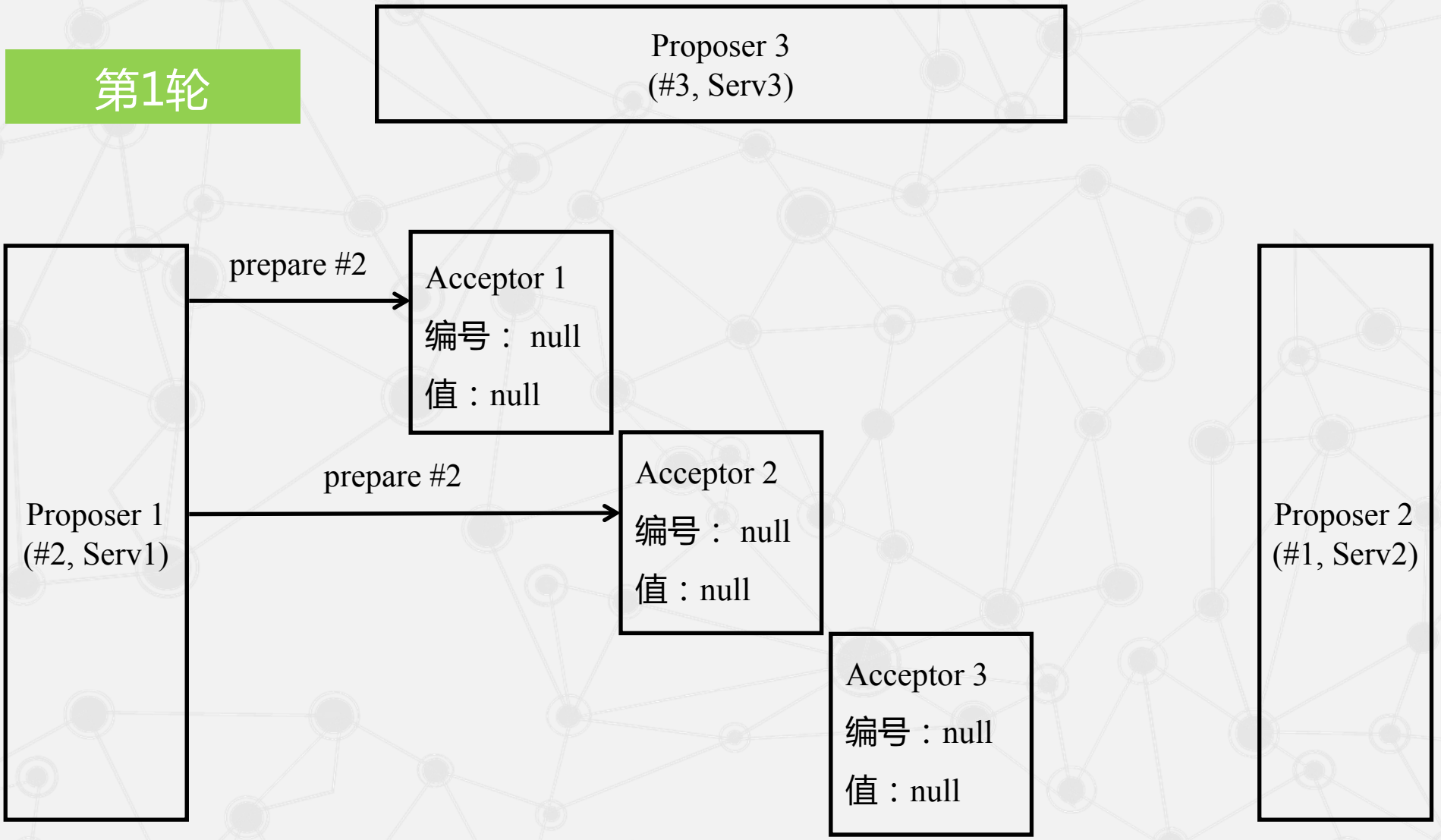


试分析：

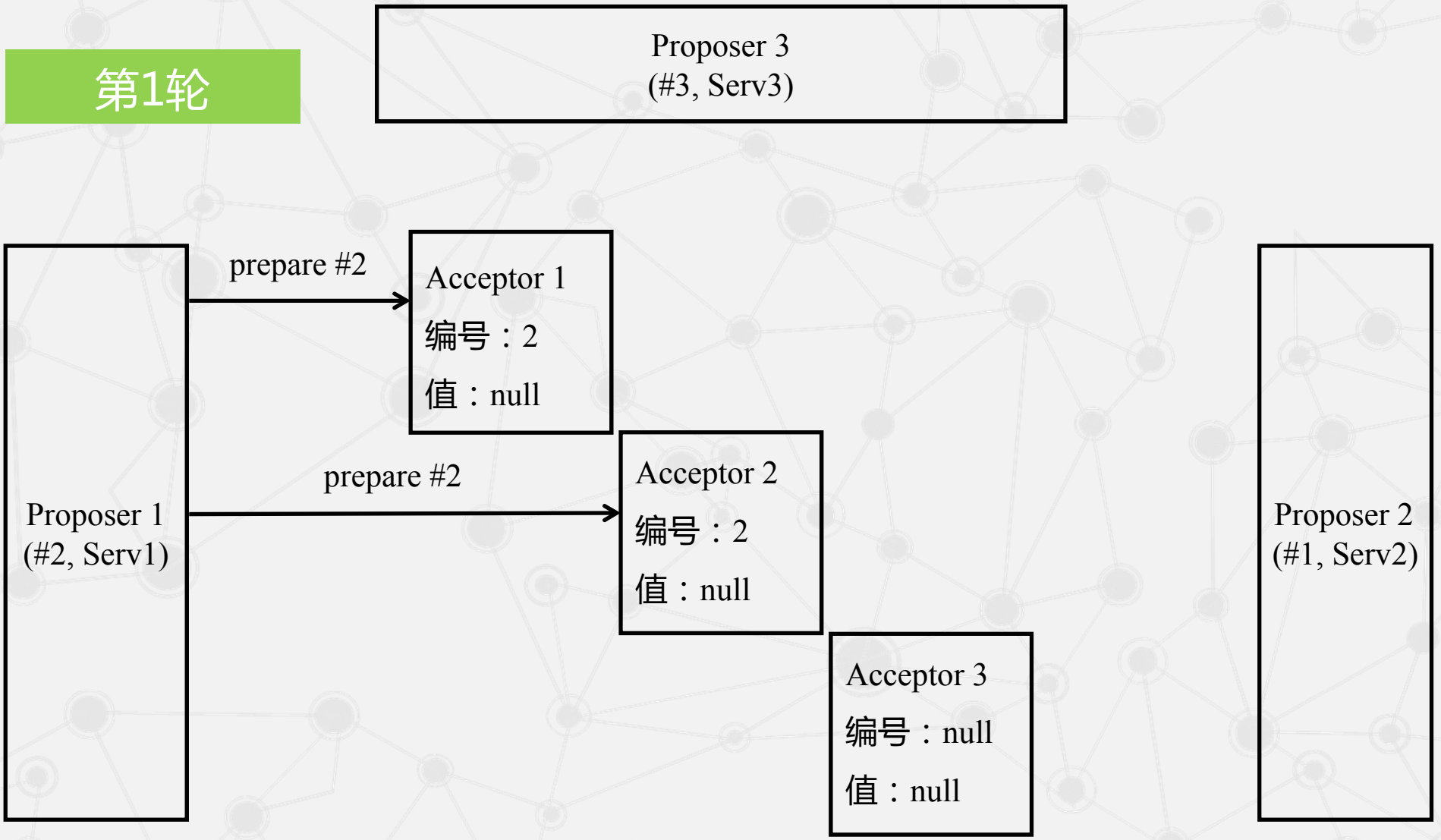
最终选取出的Master是哪台服务器？写出分析过程。


第1轮
Proposer 3
(#3, Serv3)
Acceptor 1
编号： null
值：null
Acceptor 2
编号： null
值：null
Acceptor 3
编号：null
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#1, Serv2)

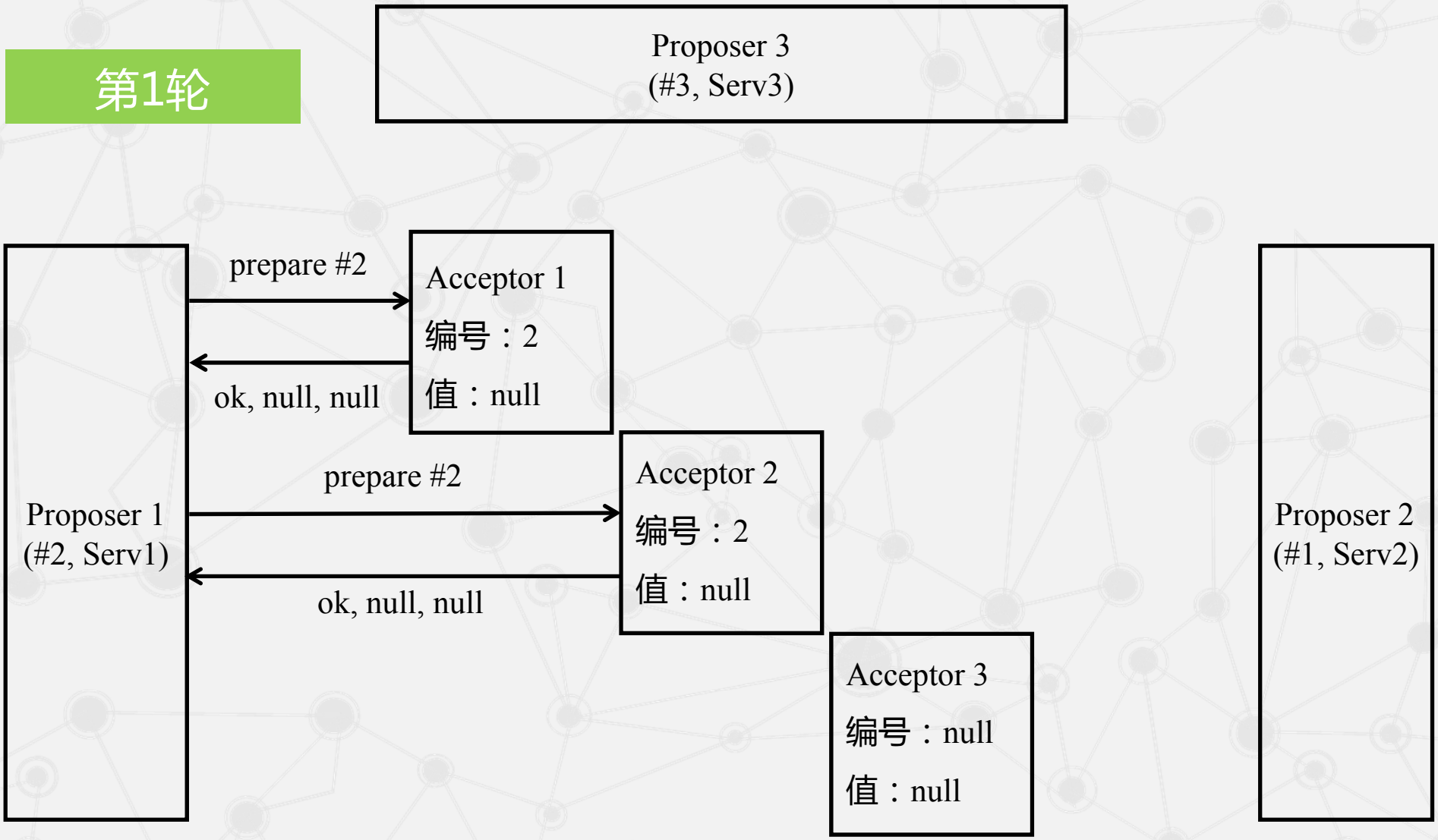
第1轮


Proposer 3
(#3, Serv3)
prepare #2
Acceptor 1
编号： null
值：null
Proposer 1
(#2, Serv1)
prepare #2
Acceptor 2
编号： null
值：null
Acceptor 3
编号：null
值：null
Proposer 2
(#1, Serv2)

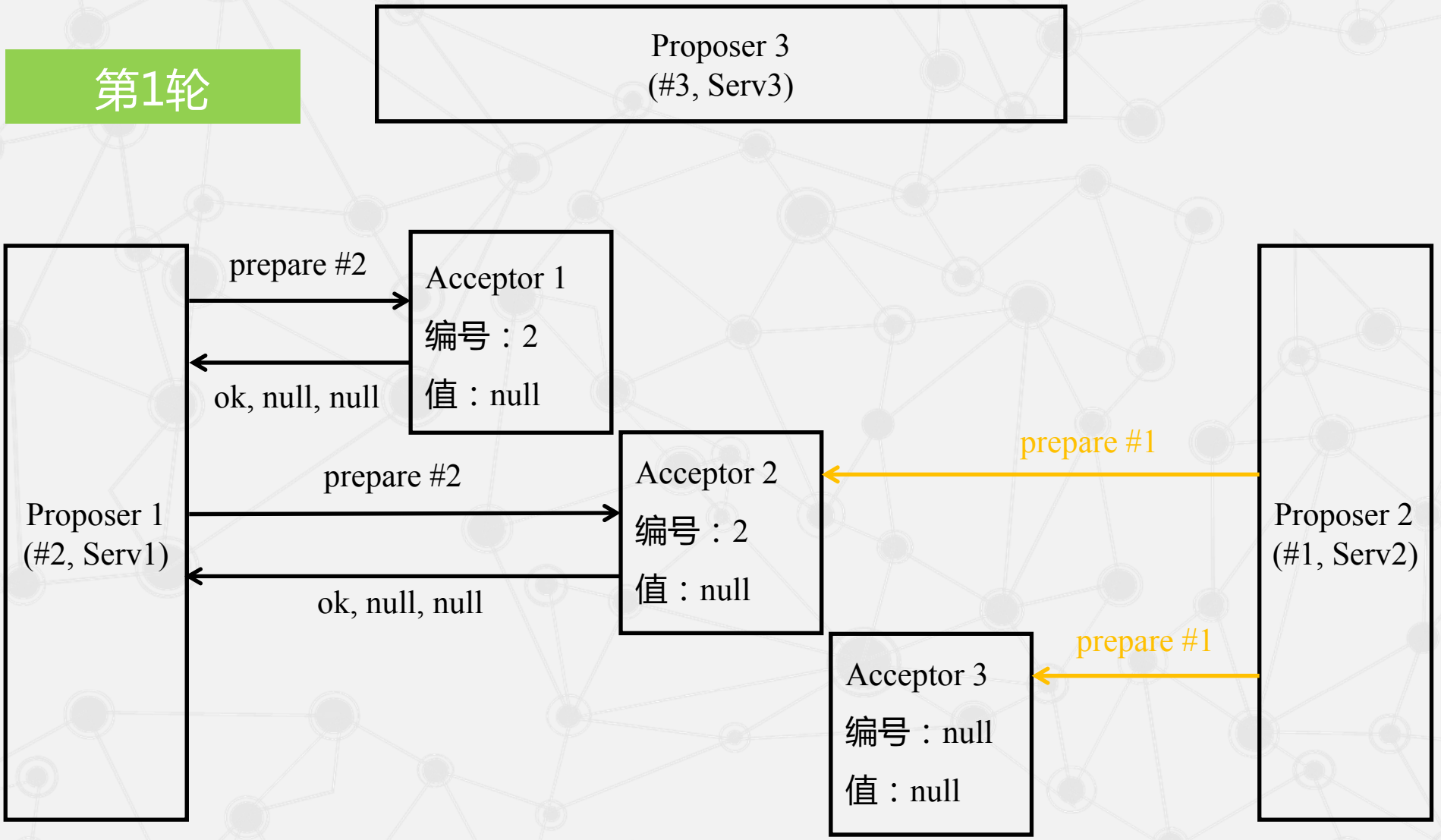
第1轮

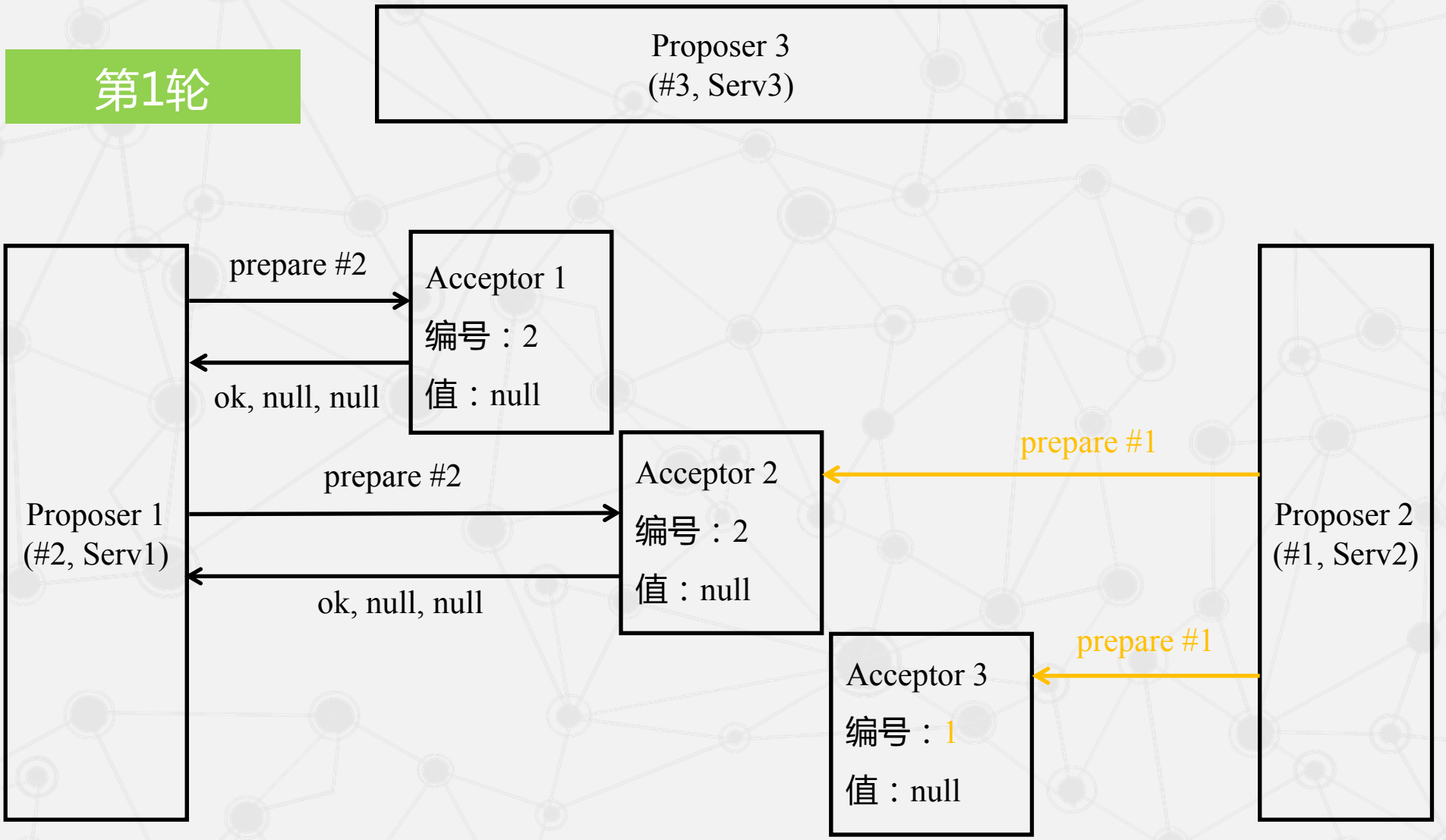

Proposer 3
(#3, Serv3)
prepare #2
Acceptor 1
编号：2
值：null
Proposer 1
(#2, Serv1)
prepare #2
Acceptor 2
编号：2
值：null
Acceptor 3
编号：null
值：null
Proposer 2
(#1, Serv2)

第1轮

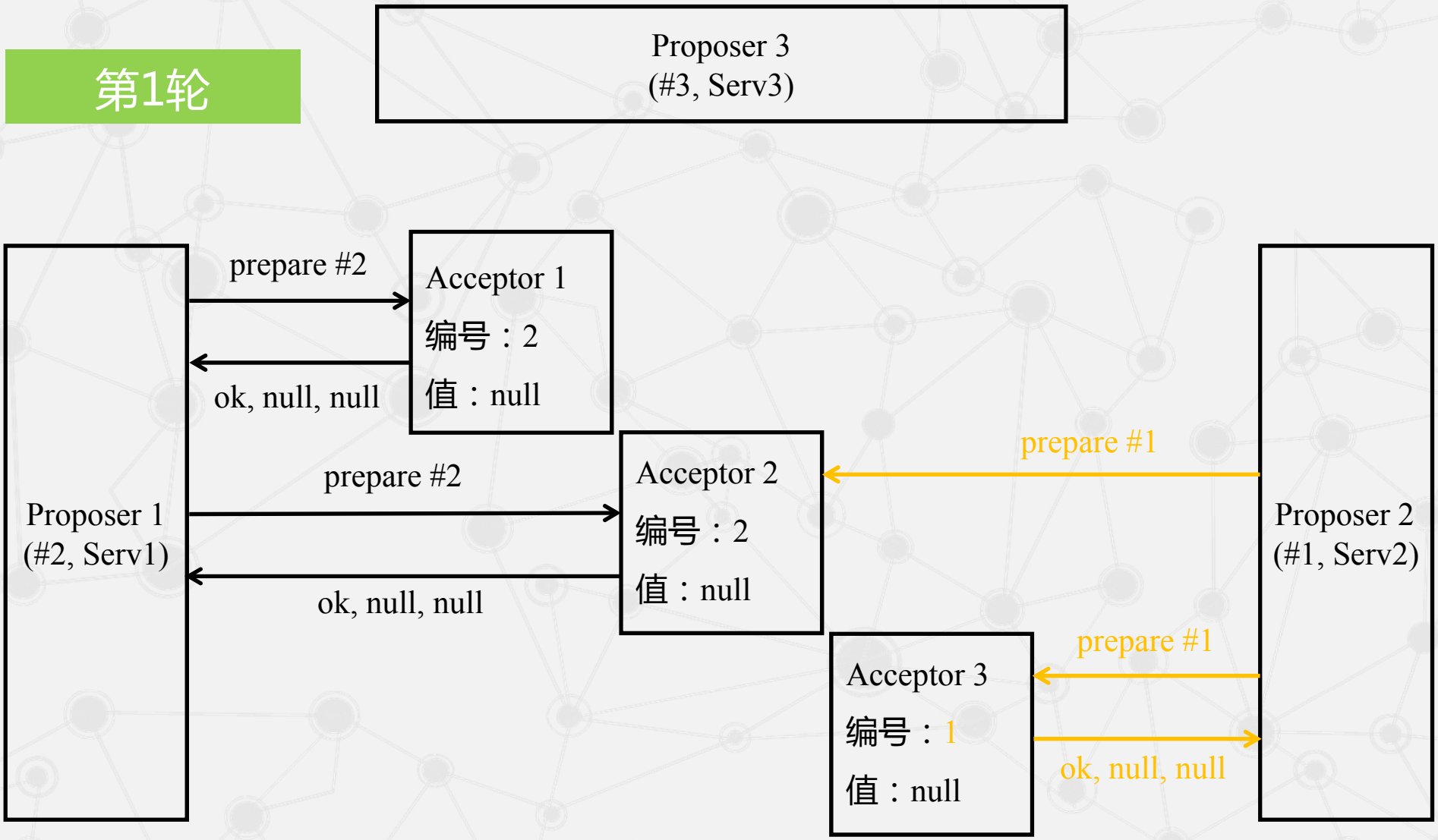

Proposer 3
(#3, Serv3)
prepare #2
Acceptor 1
编号：2
值：null
ok, null, null
Proposer 1
(#2, Serv1)
prepare #2
Acceptor 2
编号：2
值：null
ok, null, null
Acceptor 3
编号：null
值：null
Proposer 2
(#1, Serv2)

第1轮

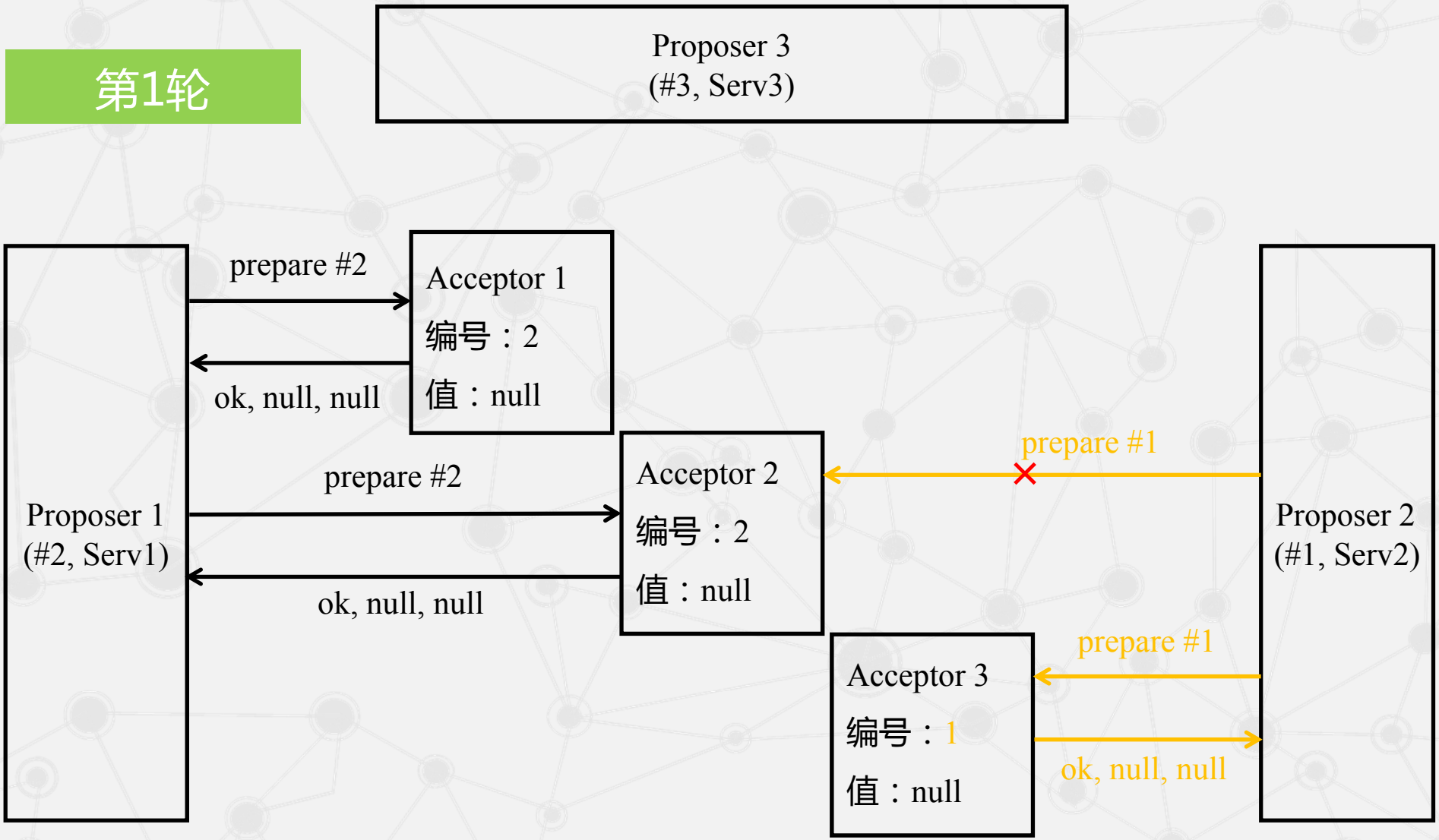

Proposer 3
(#3, Serv3)
Proposer 1
(#2, Serv1)
prepare #2
ok, null, null
Acceptor 1
编号：2
值：null
prepare #2
ok, null, null
Acceptor 2
编号：2
值：null
prepare #1
prepare #1
Acceptor 3
编号：null
值：null
Proposer 2
(#1, Serv2)

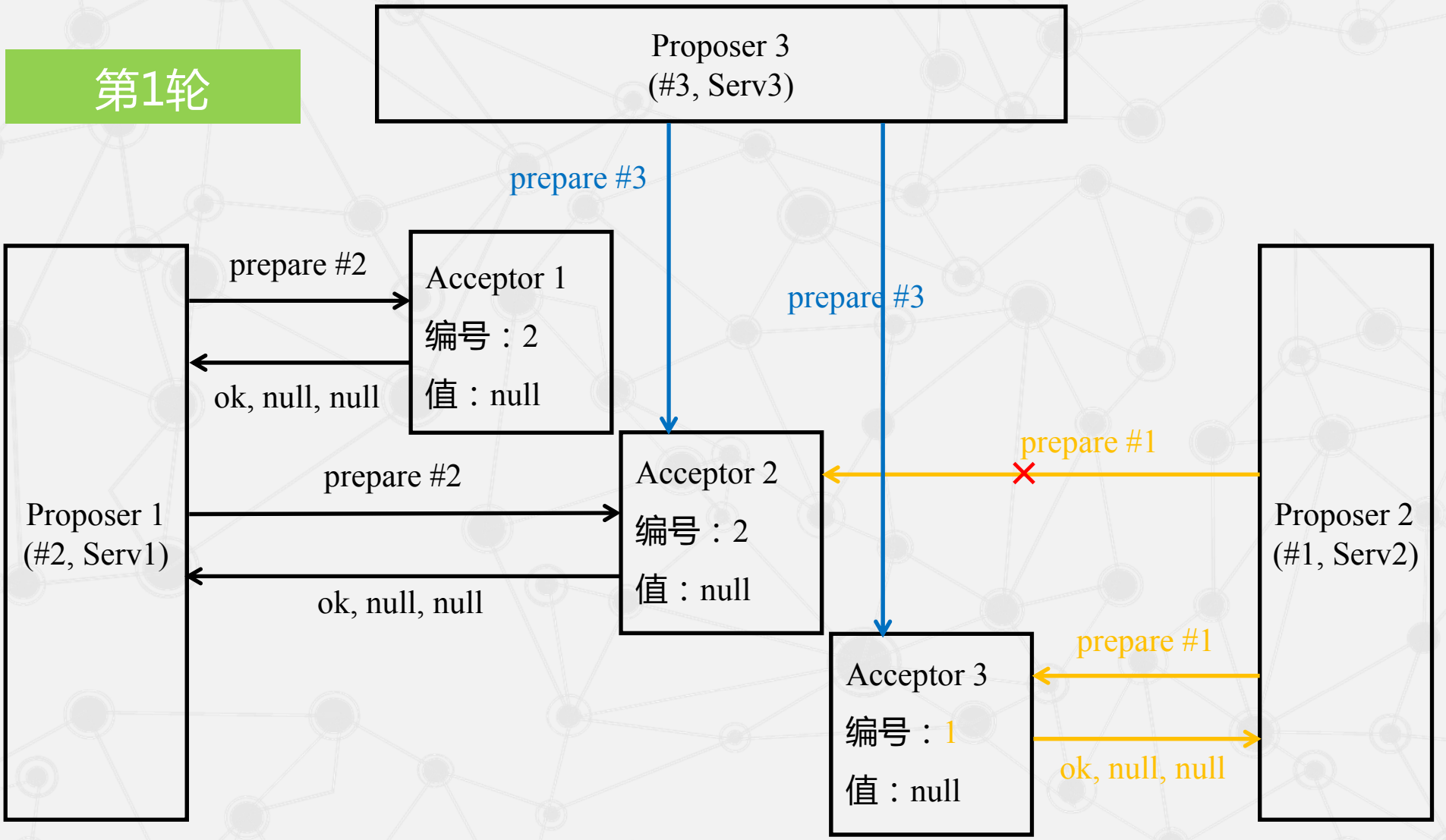

第1轮
Proposer 3
(#3, Serv3)
Proposer 1
(#2, Serv1)
prepare #2
ok, null, null
Acceptor 1
编号：2
值：null
prepare #2
ok, null, null
Acceptor 2
编号：2
值：null
prepare #1
Acceptor 3
编号：1
值：null
prepare #1
Proposer 2
(#1, Serv2)

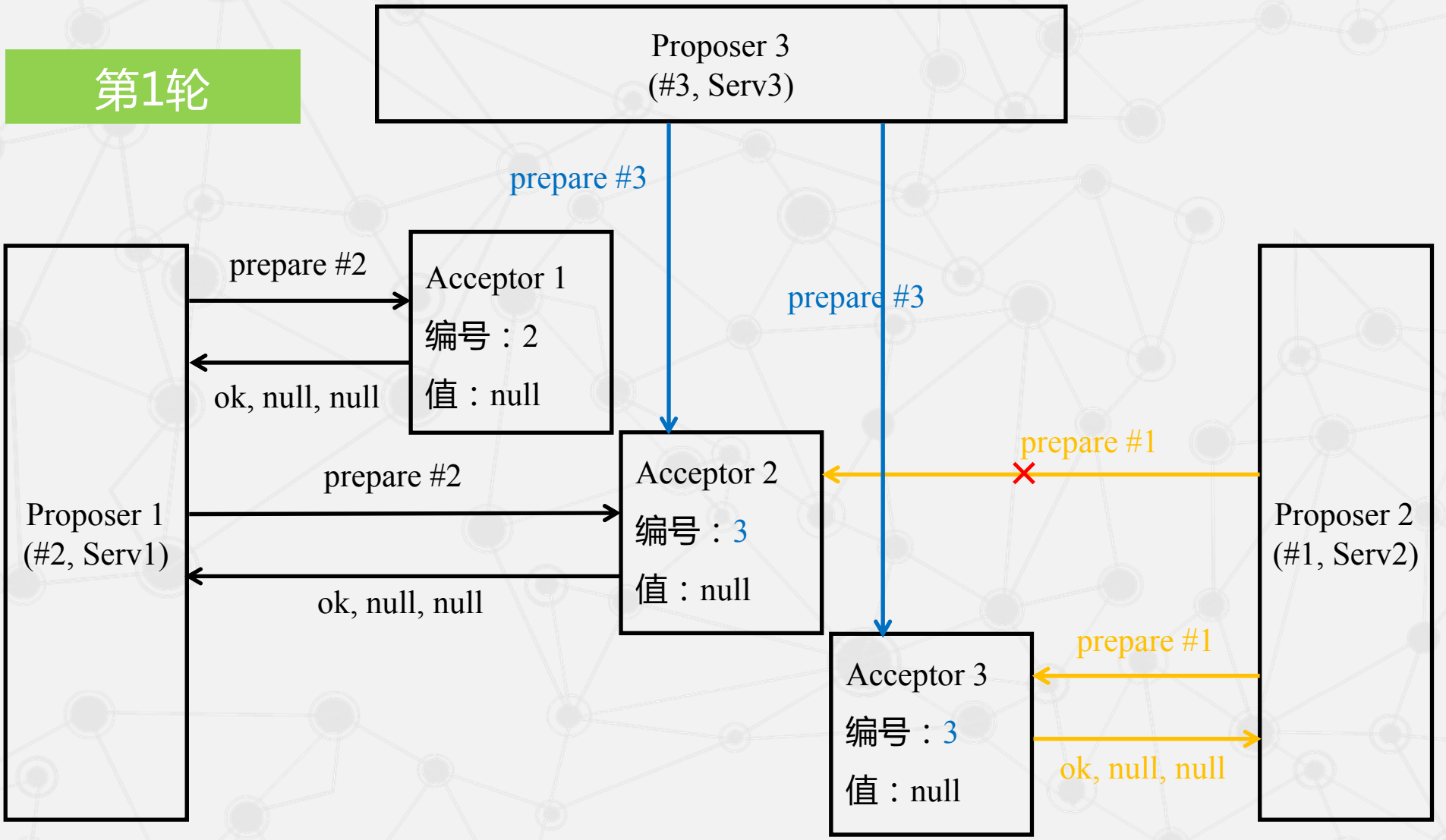
第1轮

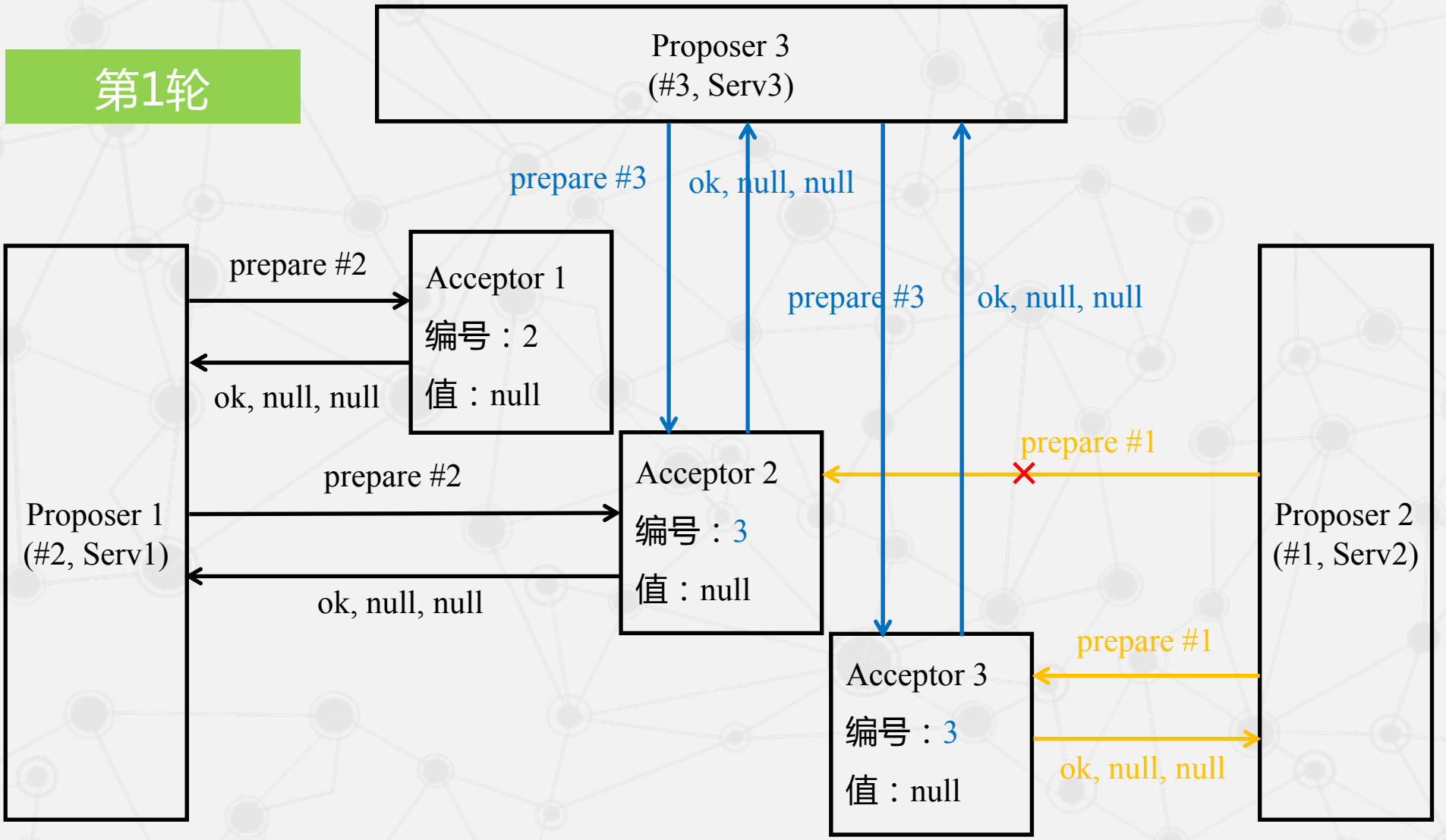

Proposer 3
(#3, Serv3)
Proposer 1
(#2, Serv1)
prepare #2
ok, null, null
Acceptor 1
编号：2
值：null
prepare #2
ok, null, null
Acceptor 2
编号：2
值：null
prepare #1
Proposer 2
(#1, Serv2)
prepare #1
ok, null, null
Acceptor 3
编号：1
值：null

第1轮


Proposer 3
(#3, Serv3)
Proposer 1
(#2, Serv1)
prepare #2
ok, null, null
Acceptor 1
编号：2
值：null
prepare #2
ok, null, null
Acceptor 2
编号：2
值：null
prepare #1
Proposer 2
(#1, Serv2)
prepare #1
ok, null, null
Acceptor 3
编号：1
值：null

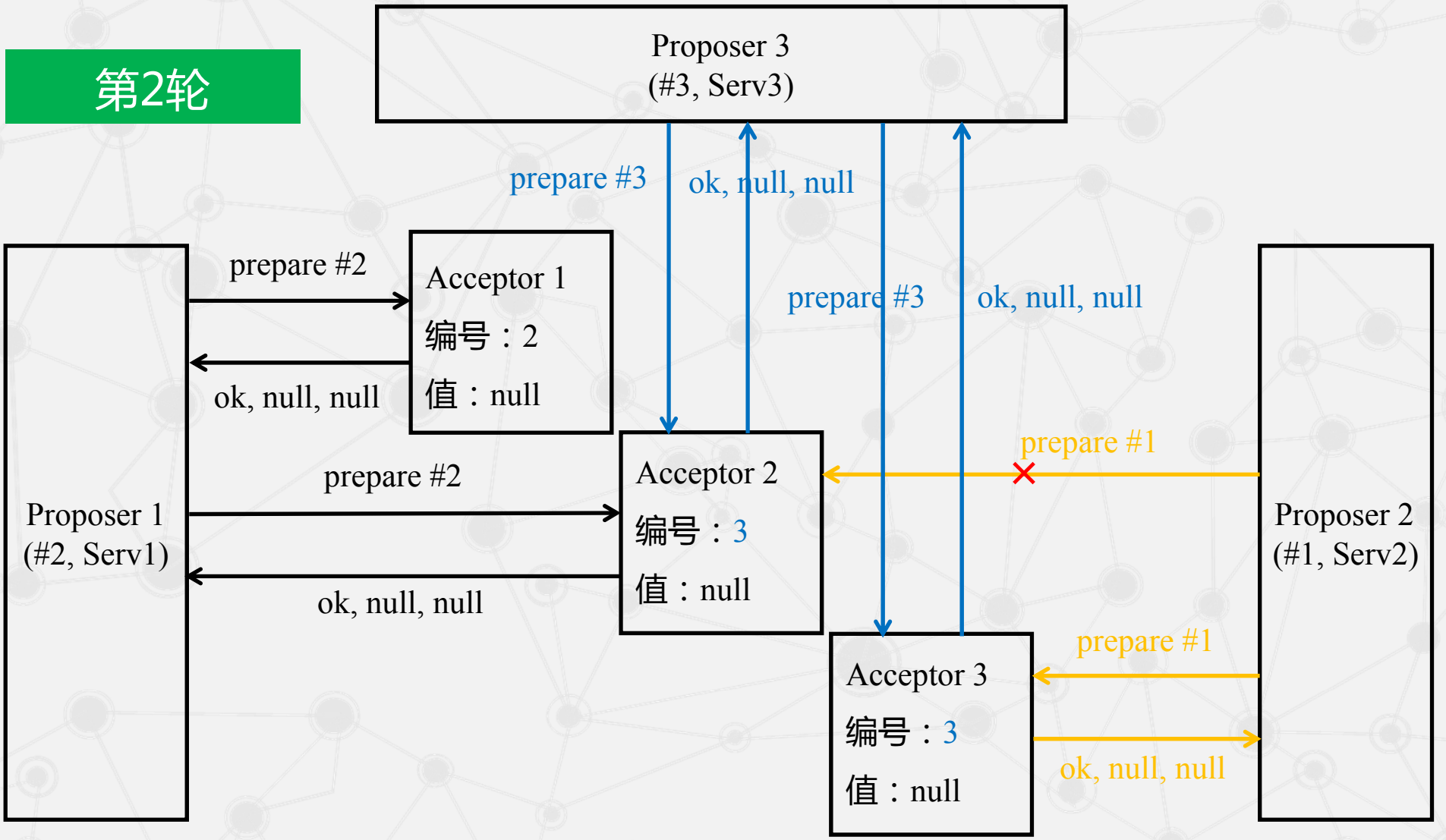

第1轮
Proposer 3
(#3, Serv3)
prepare #3
prepare #3
prepare #2
Acceptor 1
编号：2
值：null
ok, null, null
Proposer 1
(#2, Serv1)
prepare #2
Acceptor 2
编号：2
值：null
ok, null, null
prepare #1
Proposer 2
(#1, Serv2)
Acceptor 3
编号：1
值：null
prepare #1
ok, null, null

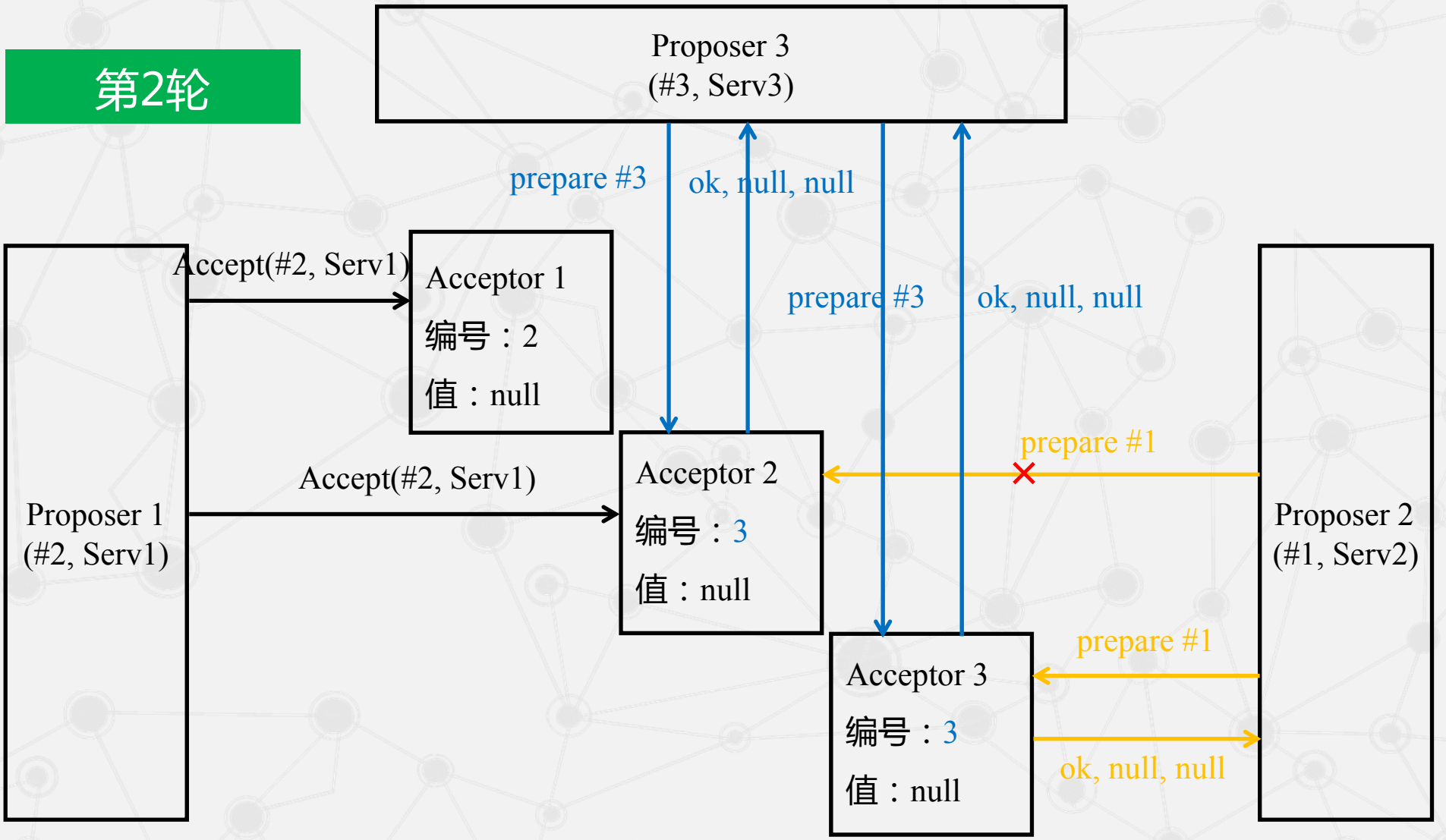

第1轮
Proposer 3
(#3, Serv3)
prepare #3
prepare #3
prepare #2
Acceptor 1
编号：2
值：null
ok, null, null
Proposer 1
(#2, Serv1)
prepare #2
Acceptor 2
编号：3
值：null
ok, null, null
prepare #1
Proposer 2
(#1, Serv2)
Acceptor 3
编号：3
值：null
prepare #1
ok, null, null

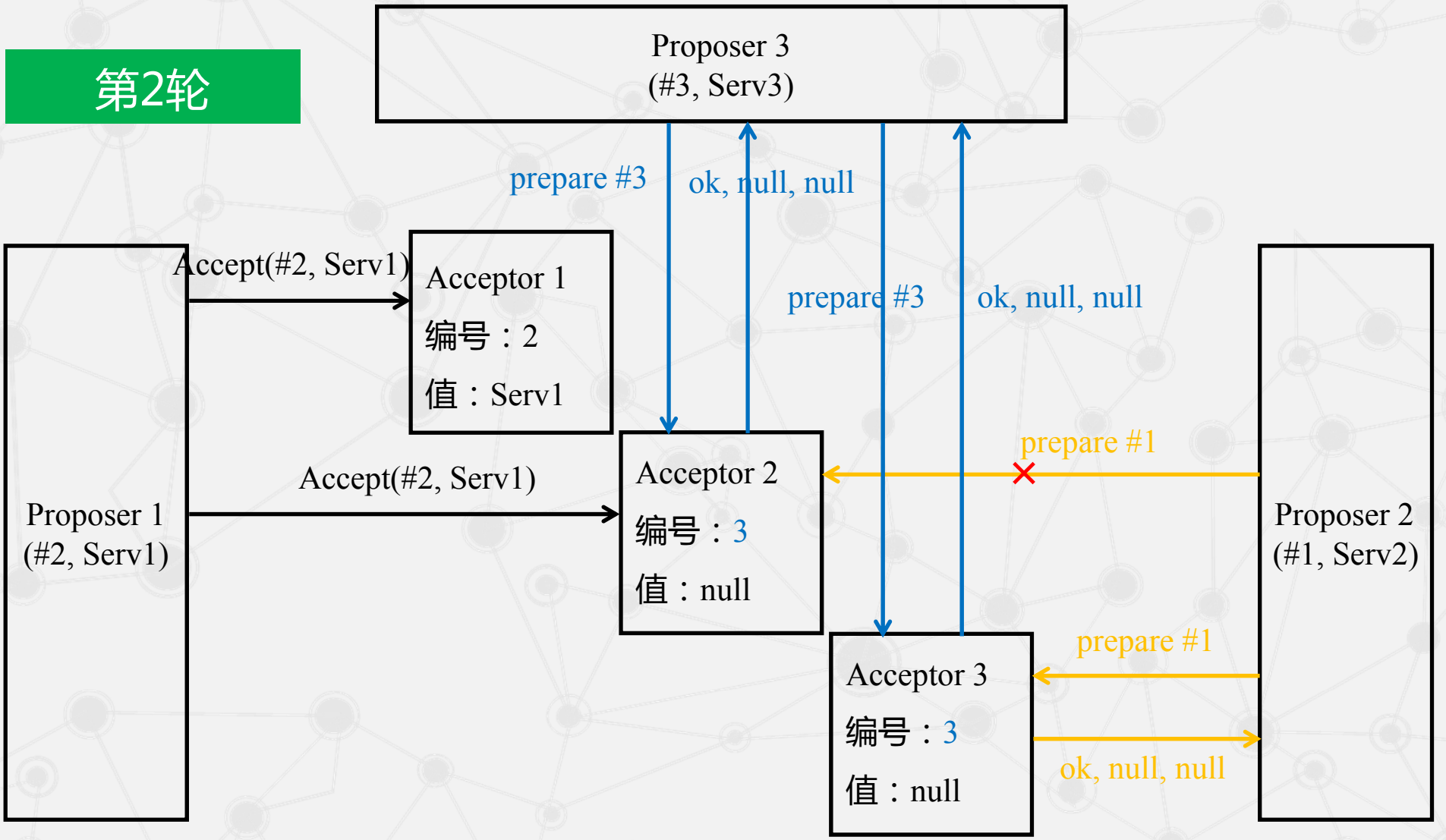

第1轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
prepare #2
ok, null, null
Acceptor 1
编号：2
值：null
prepare #1
prepare #2
ok, null, null
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#1, Serv2)
prepare #1
ok, null, null
Acceptor 3
编号：3
值：null

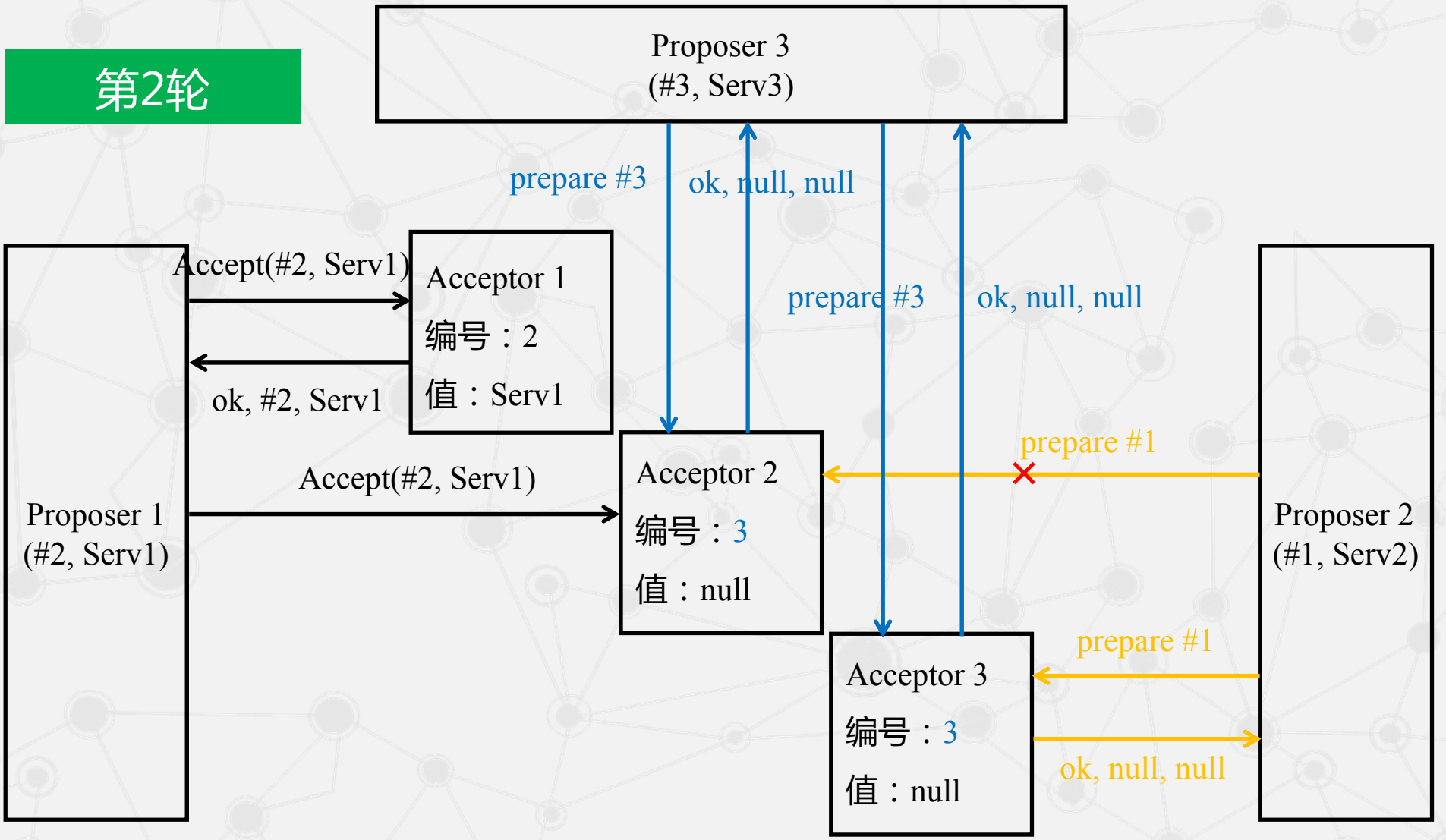
第2轮

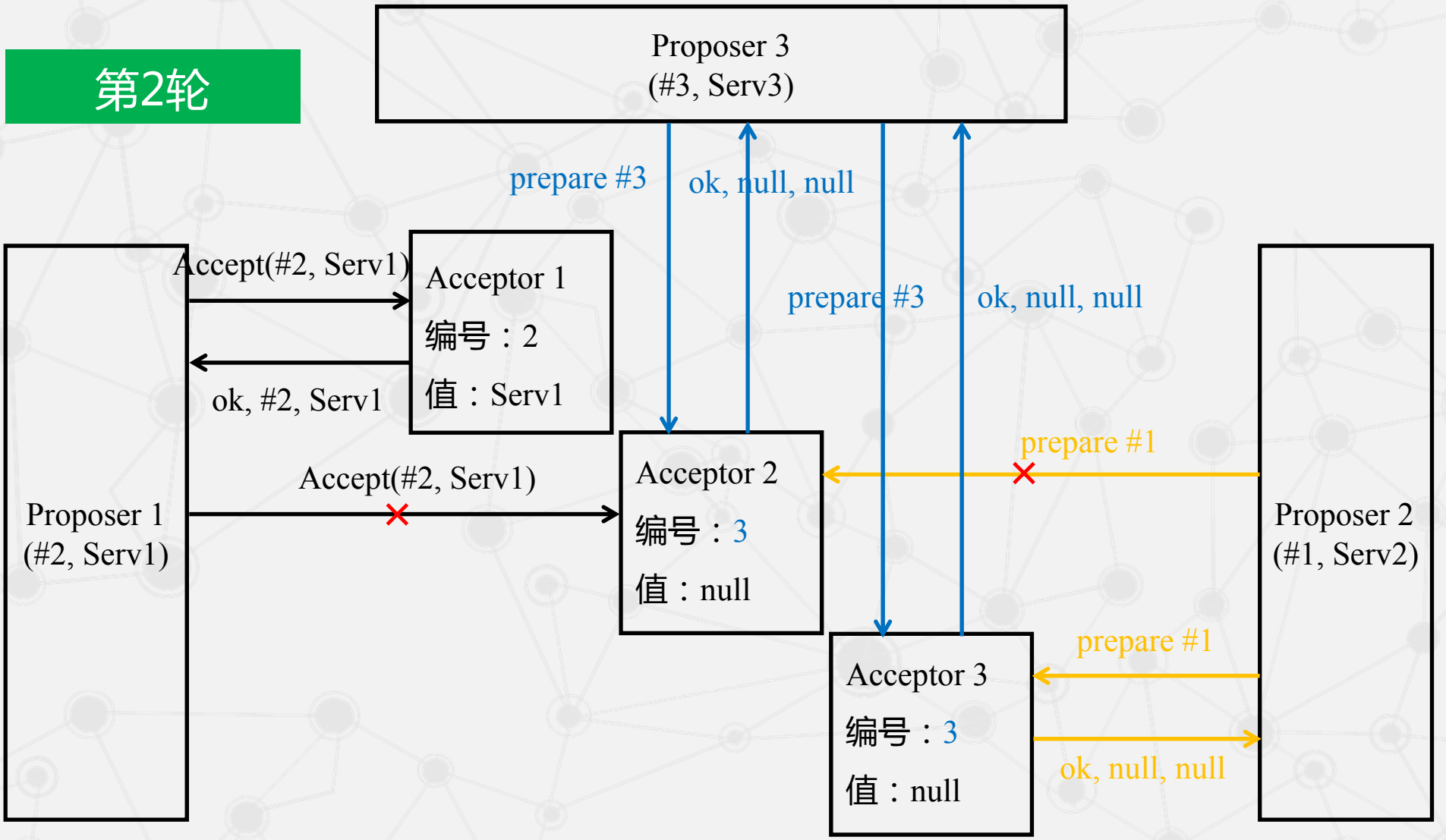
一些Proposer开始进入accept阶段


第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
prepare #2
Acceptor 1
编号：2
值：null
ok, null, null
Proposer 1
(#2, Serv1)
prepare #2
Acceptor 2
编号：3
值：null
ok, null, null
prepare #1
Proposer 2
(#1, Serv2)
Acceptor 3
编号：3
值：null
prepare #1
ok, null, null


第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
Accept(#2, Serv1)
Acceptor 1
编号：2
值：null
prepare #1
Accept(#2, Serv1)
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#1, Serv2)
prepare #1
Acceptor 3
编号：3
值：null
ok, null, null


第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
prepare #1
Accept(#2, Serv1)
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#1, Serv2)
prepare #1
Acceptor 3
编号：3
值：null
ok, null, null


第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
prepare #1
Accept(#2, Serv1)
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#1, Serv2)
prepare #1
Acceptor 3
编号：3
值：null
ok, null, null


第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
prepare #1
Accept(#2, Serv1)
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#1, Serv2)
prepare #1
Acceptor 3
编号：3
值：null
ok, null, null

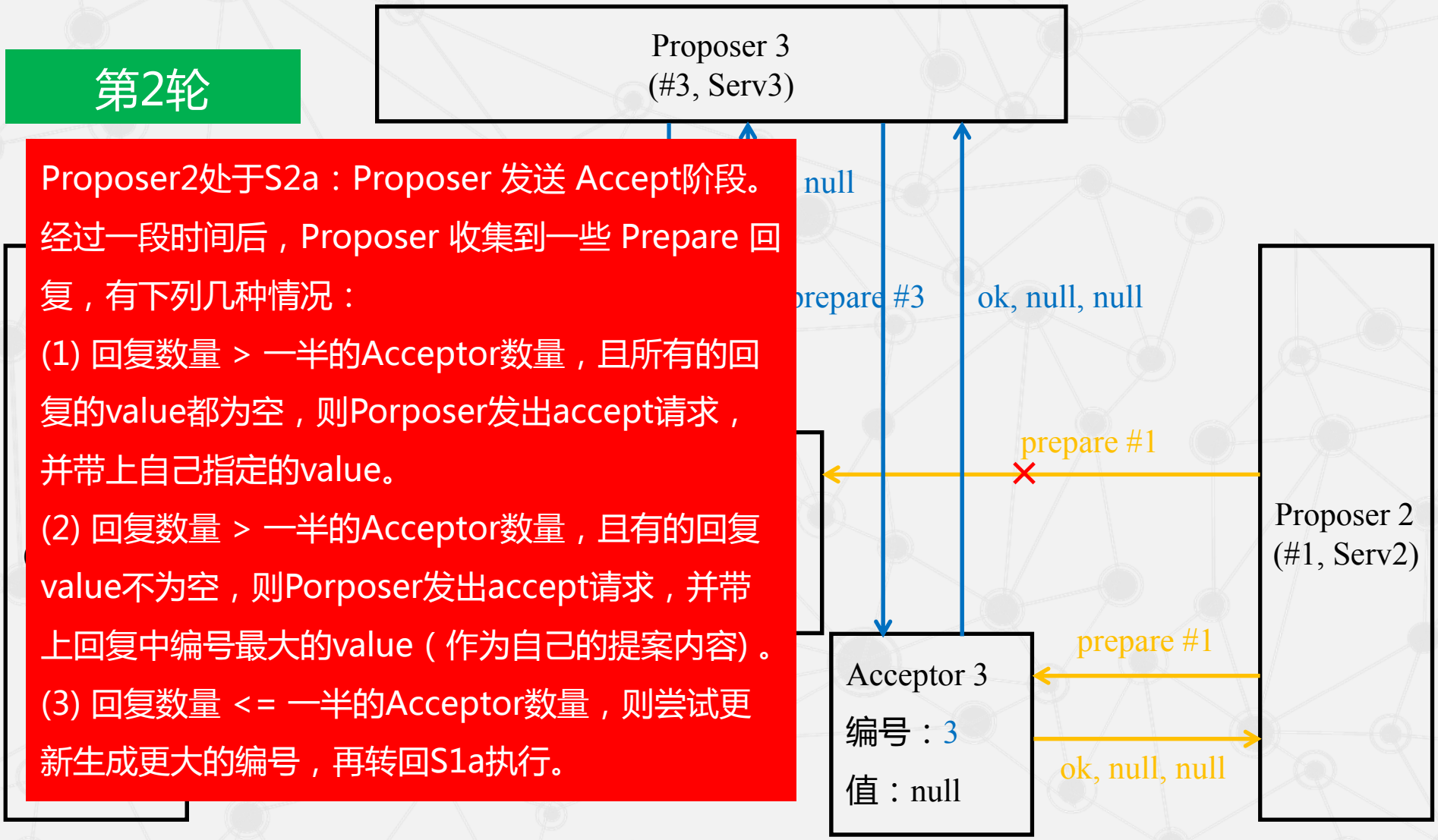
第2轮

Proposer2处于S2a：Proposer 发送 Accept阶段。经过一段时间后，Proposer 收集到一些 Prepare 回复，有下列几种情况：

(1) 回复数量 > 一半的Acceptor数量，且所有的回复的value都为空，则Porposer发出accept请求，并带上自己指定的value。

(2) 回复数量 > 一半的Acceptor数量，且有的回复value不为空，则Porposer发出accept请求，并带上回复中编号最大的value（作为自己的提案内容）。

(3) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的编号，再转回S1a执行。



第2轮

S1a：Proposer 发送 Prepare请求：

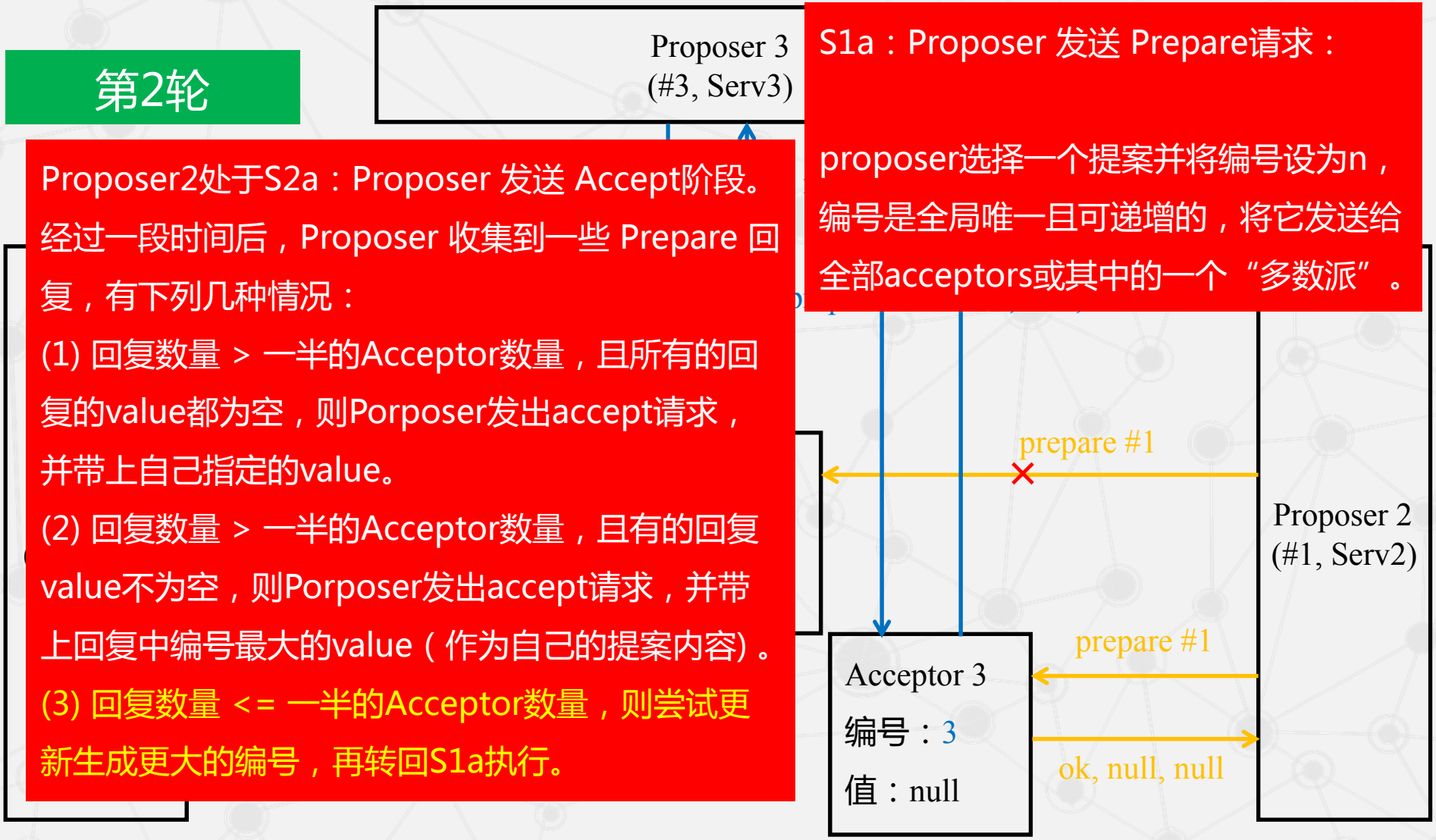
proposer选择一个提案并将编号设为n，编号是全局唯一且可递增的，将它发送给全部acceptors或其中的一个“多数派”。

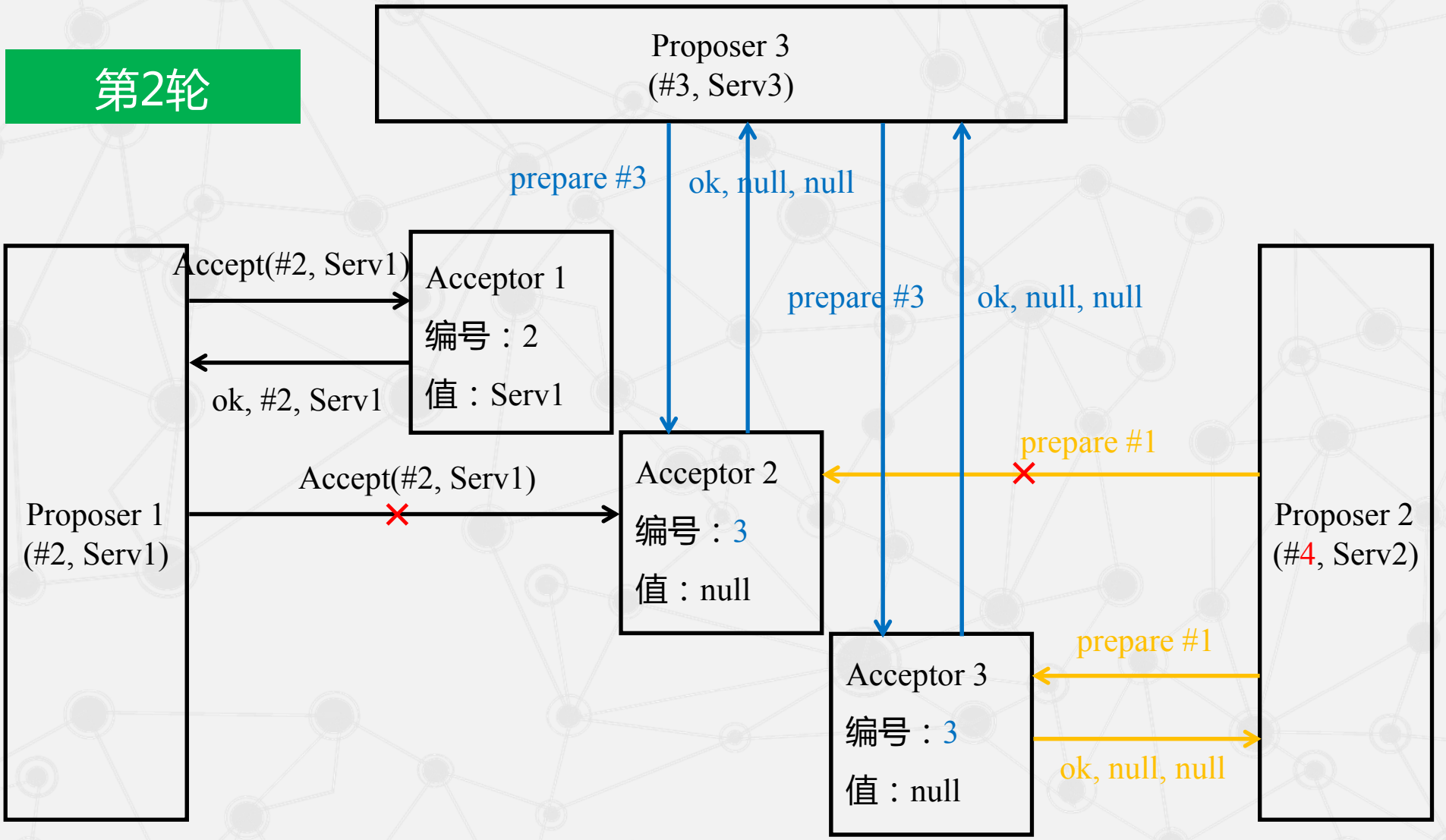
Proposer2处于S2a：Proposer 发送 Accept阶段。经过一段时间后，Proposer 收集到一些 Prepare 回复，有下列几种情况：

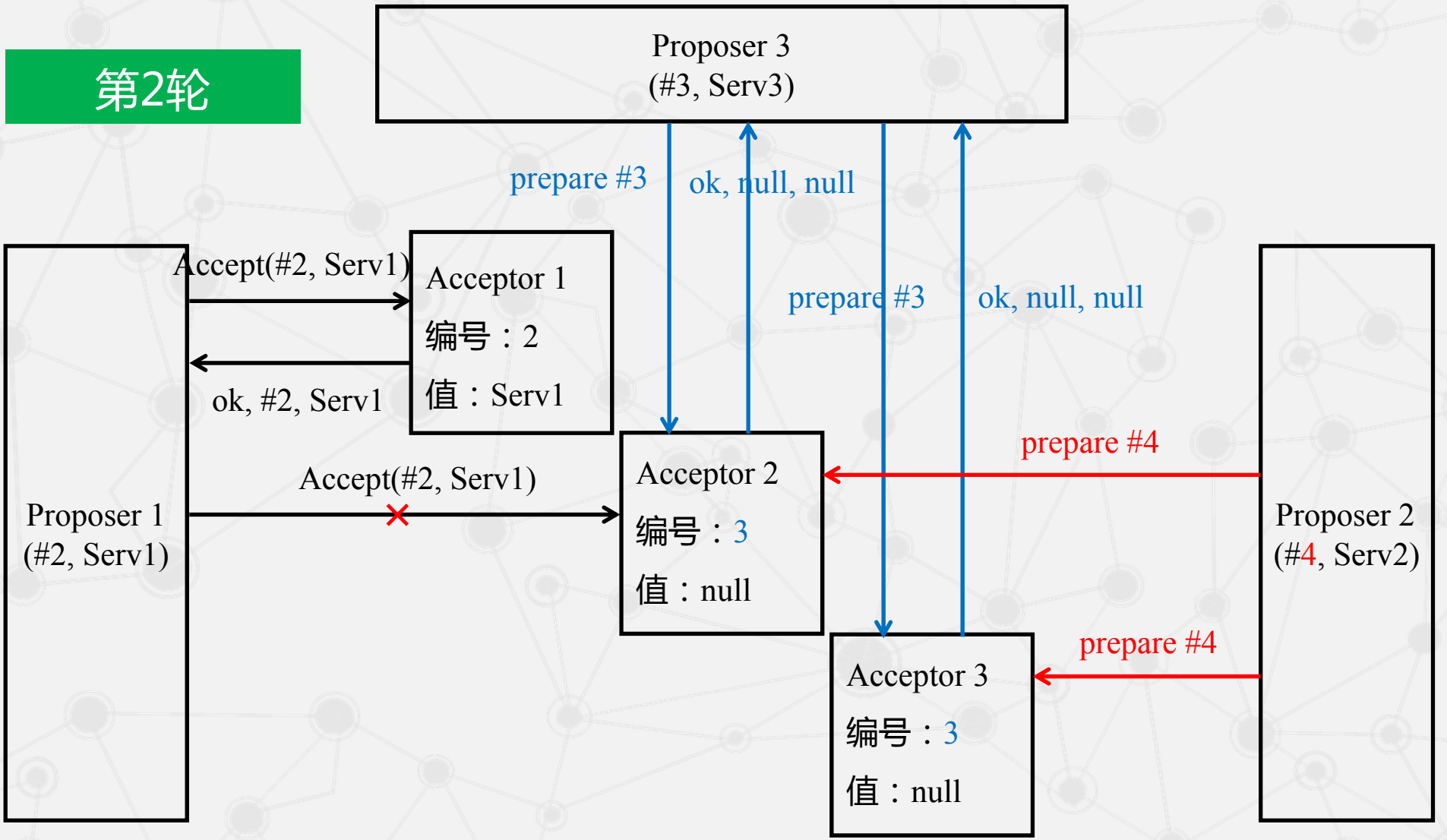
(1) 回复数量 > 一半的Acceptor数量，且所有的回复的value都为空，则Porposer发出accept请求，并带上自己指定的value。

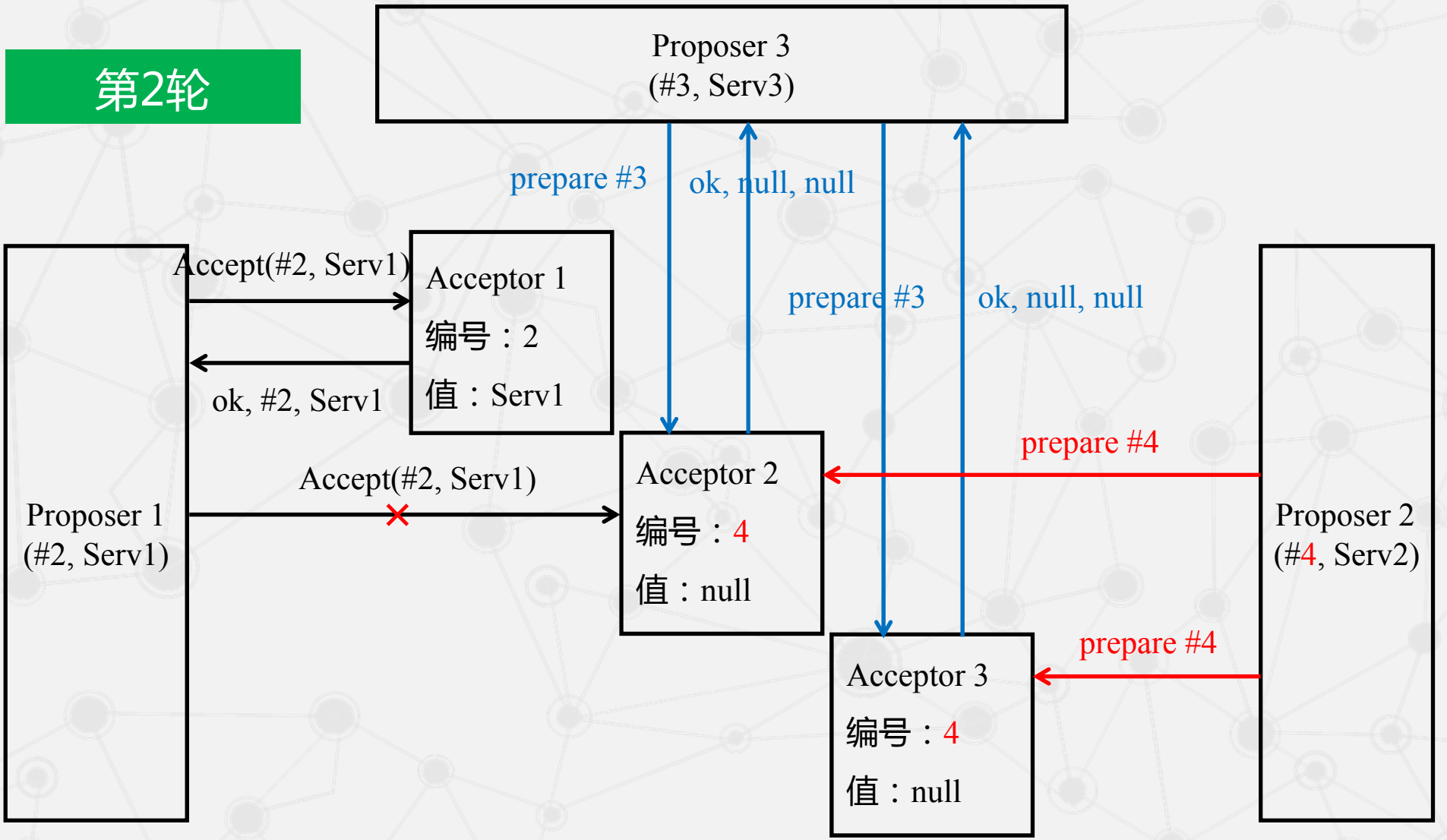
(2) 回复数量 > 一半的Acceptor数量，且有的回复value不为空，则Porposer发出accept请求，并带上回复中编号最大的value（作为自己的提案内容）。

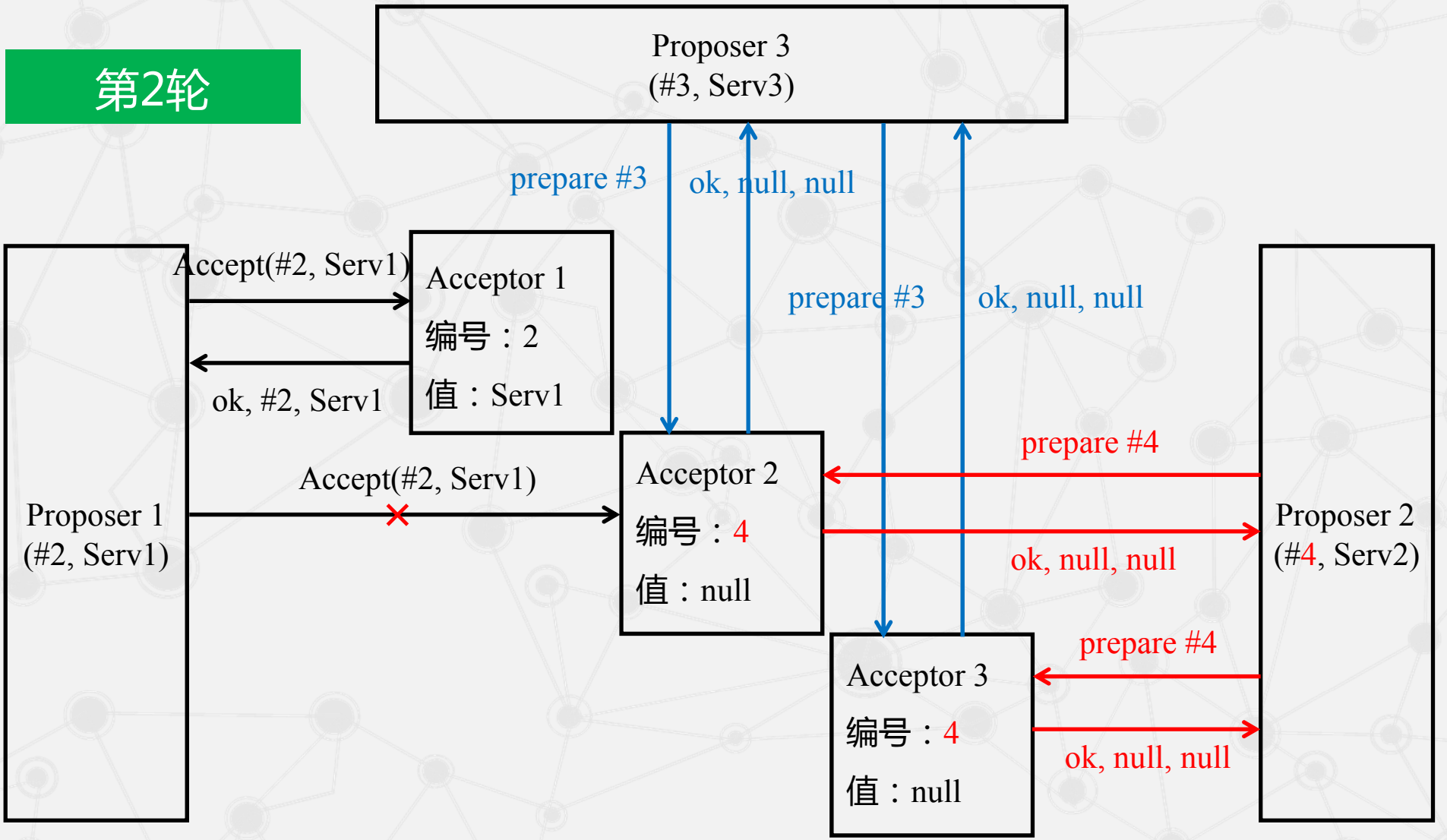
(3) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的编号，再转回S1a执行。

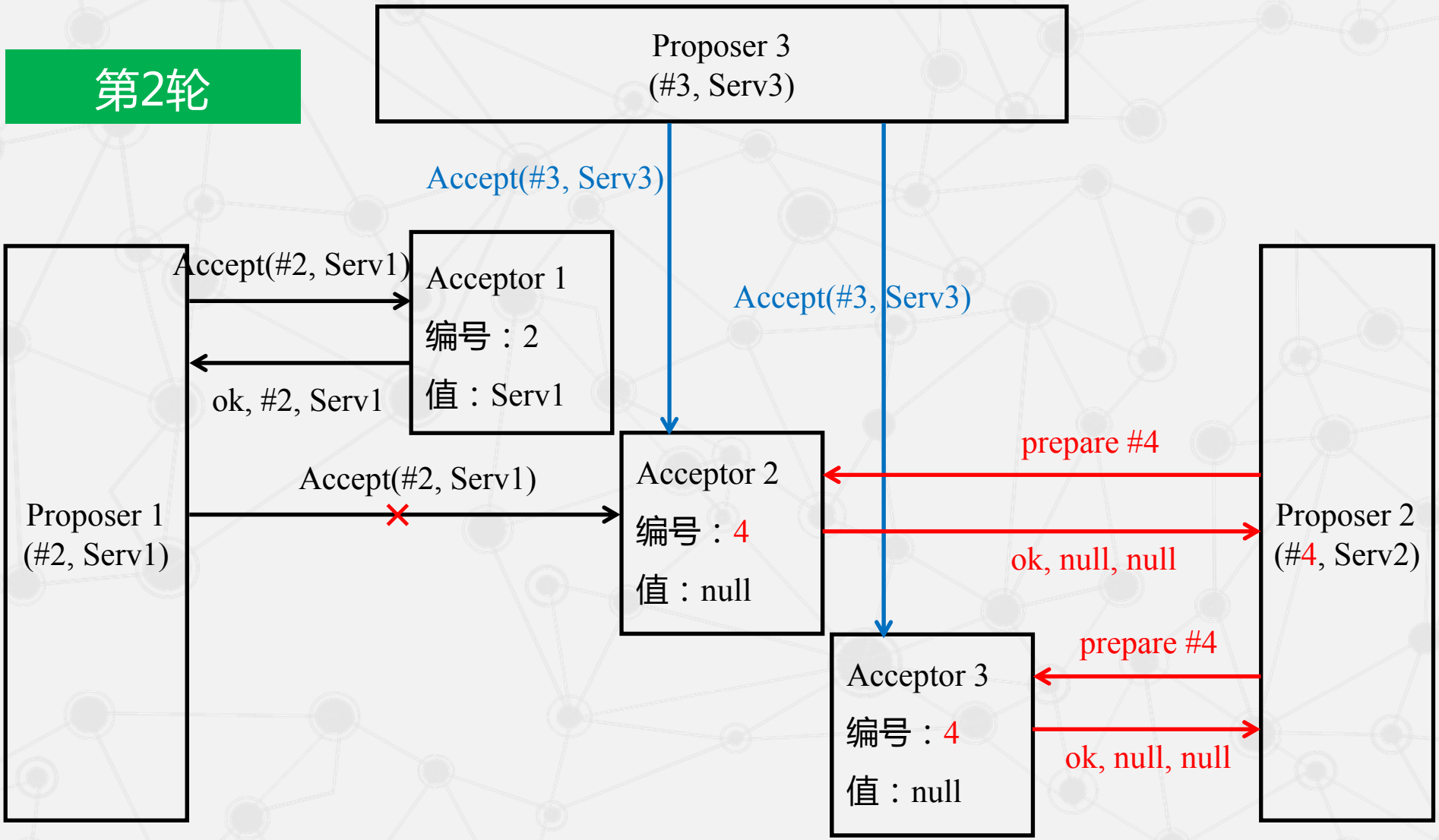


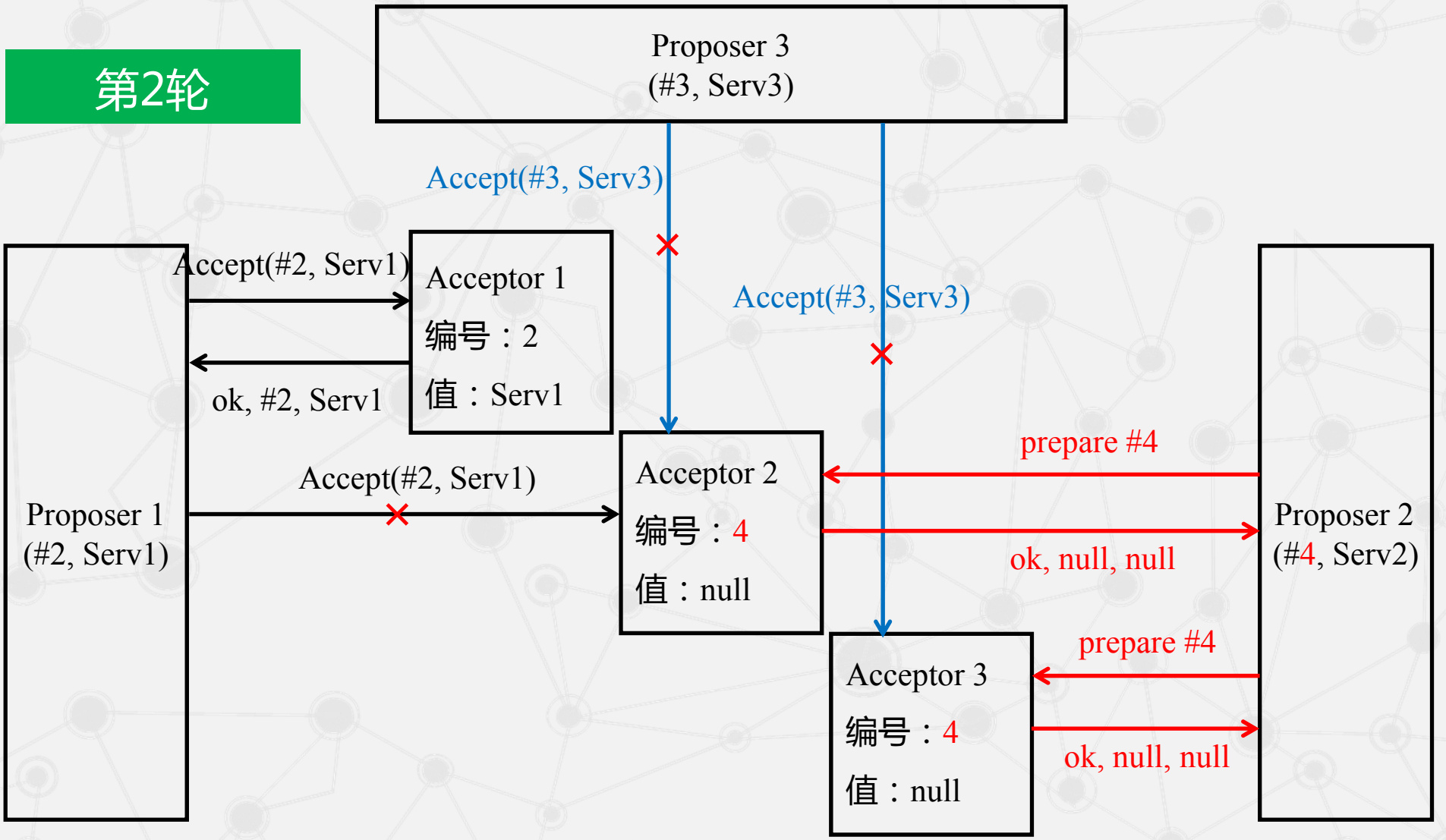

第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
prepare #1
Accept(#2, Serv1)
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#4, Serv2)
prepare #1
Acceptor 3
编号：3
值：null
ok, null, null


第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
prepare #4
Accept(#2, Serv1)
Acceptor 2
编号：3
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#4, Serv2)
prepare #4
Acceptor 3
编号：3
值：null

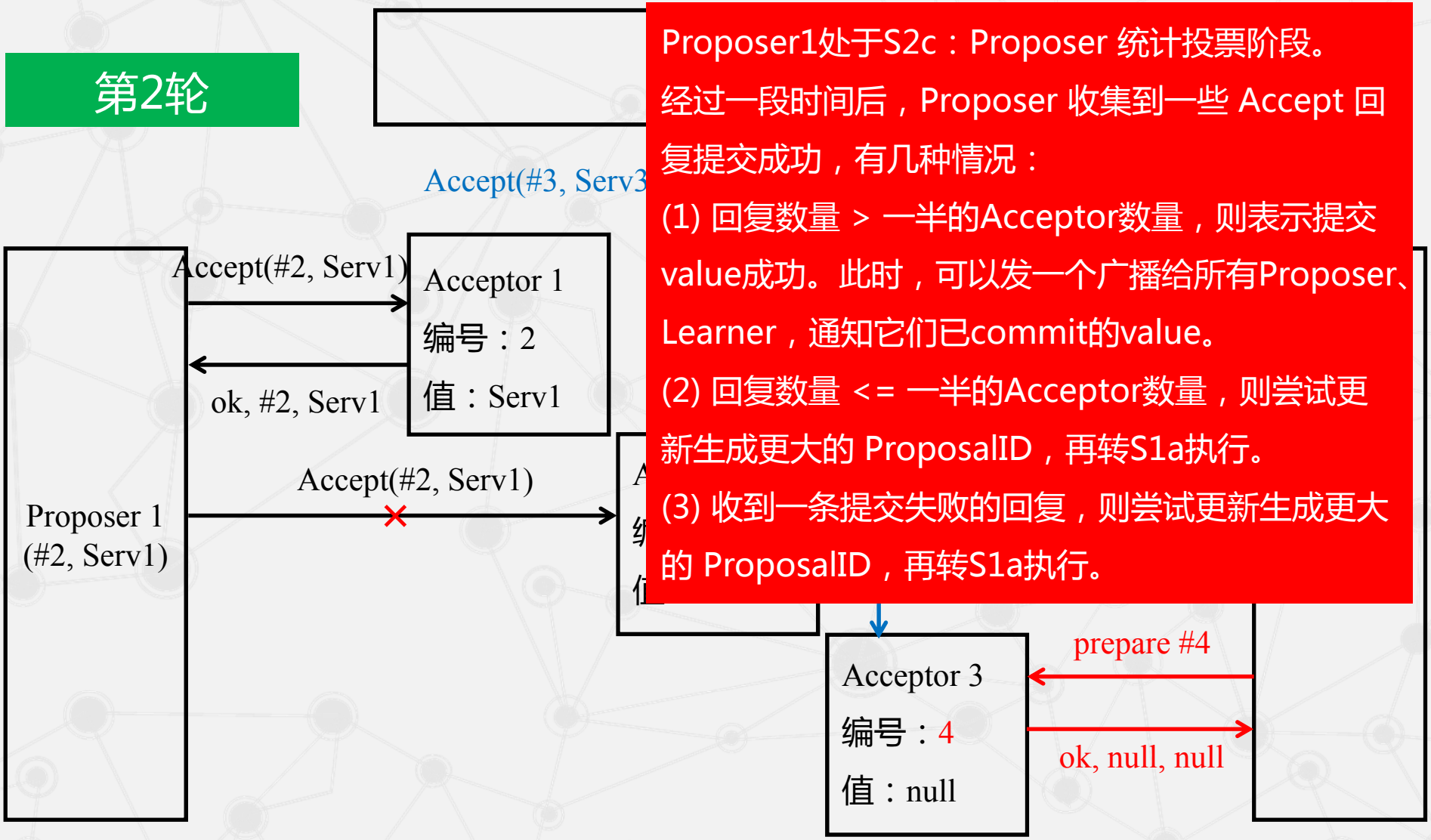

第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
prepare #4
Accept(#2, Serv1)
Acceptor 2
编号：4
值：null
Proposer 1
(#2, Serv1)
Proposer 2
(#4, Serv2)
prepare #4
Acceptor 3
编号：4
值：null


第2轮
Proposer 3
(#3, Serv3)
prepare #3
ok, null, null
prepare #3
ok, null, null
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#2, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：4
值：null
prepare #4
ok, null, null
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：null
prepare #4
ok, null, null


第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#2, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：4
值：null
prepare #4
ok, null, null
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：null
prepare #4
ok, null, null


第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#2, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：4
值：null
prepare #4
ok, null, null
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：null
prepare #4
ok, null, null

第2轮



Proposer1处于S2c：Proposer 统计投票阶段。

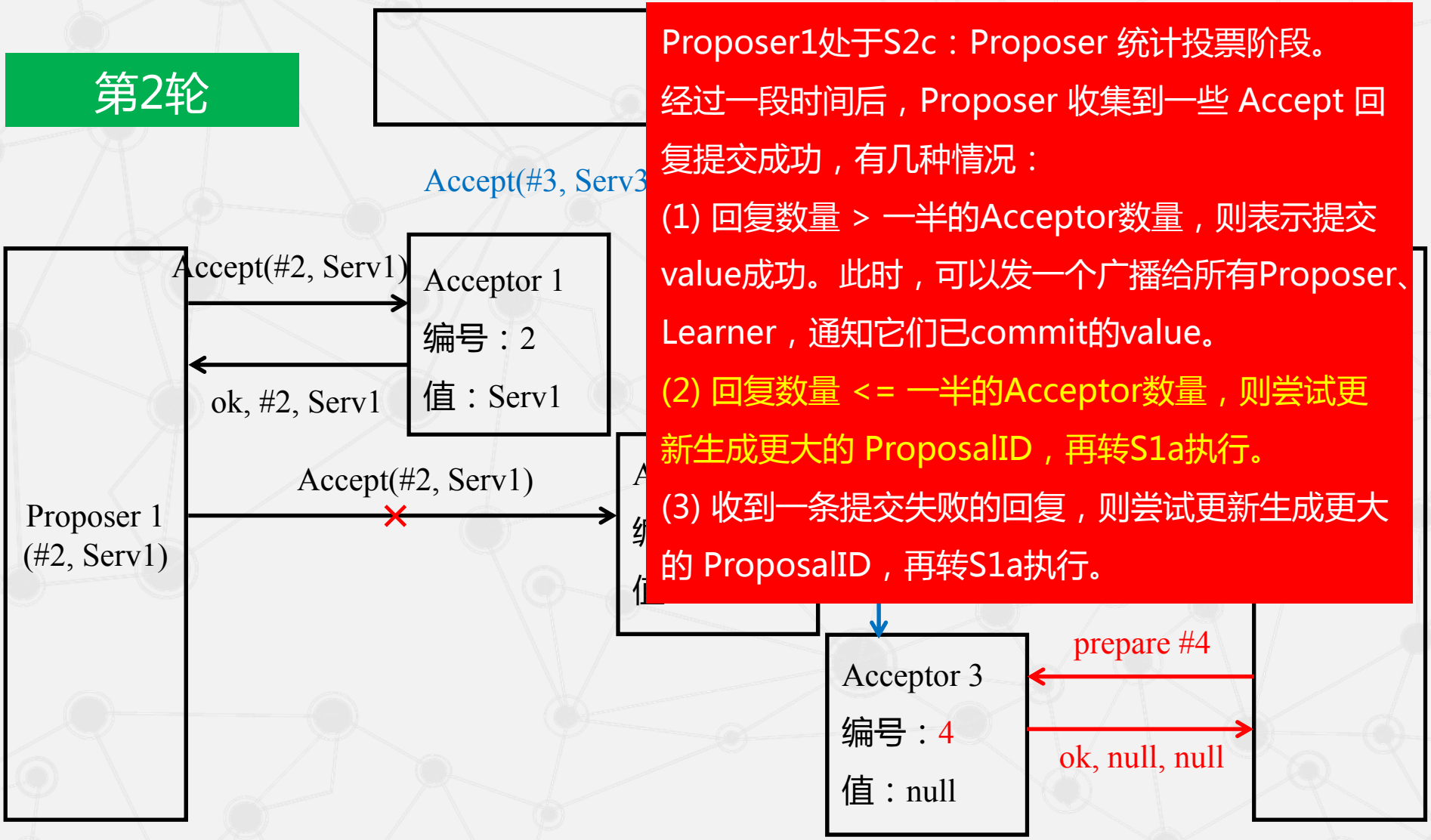
经过一段时间后，Proposer 收集到一些 Accept 回复提交成功，有几种情况：

(1) 回复数量 > 一半的Acceptor数量，则表示提交value成功。此时，可以发一个广播给所有Proposer、Learner，通知它们已commit的value。

(2) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的 ProposalID，再转S1a执行。

(3) 收到一条提交失败的回复，则尝试更新生成更大的 ProposalID，再转S1a执行。

第2轮



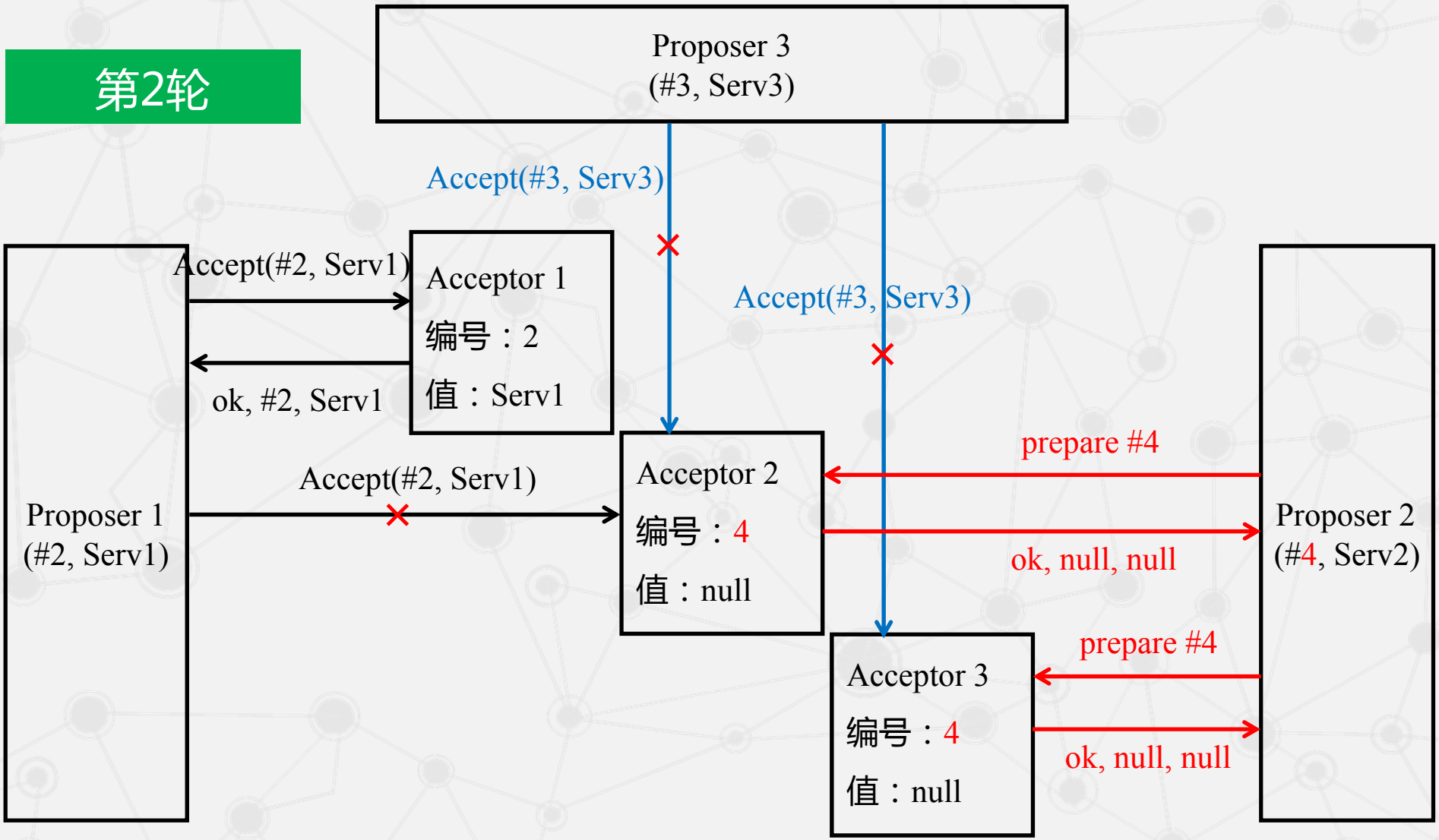
Proposer1处于S2c：Proposer 统计投票阶段。

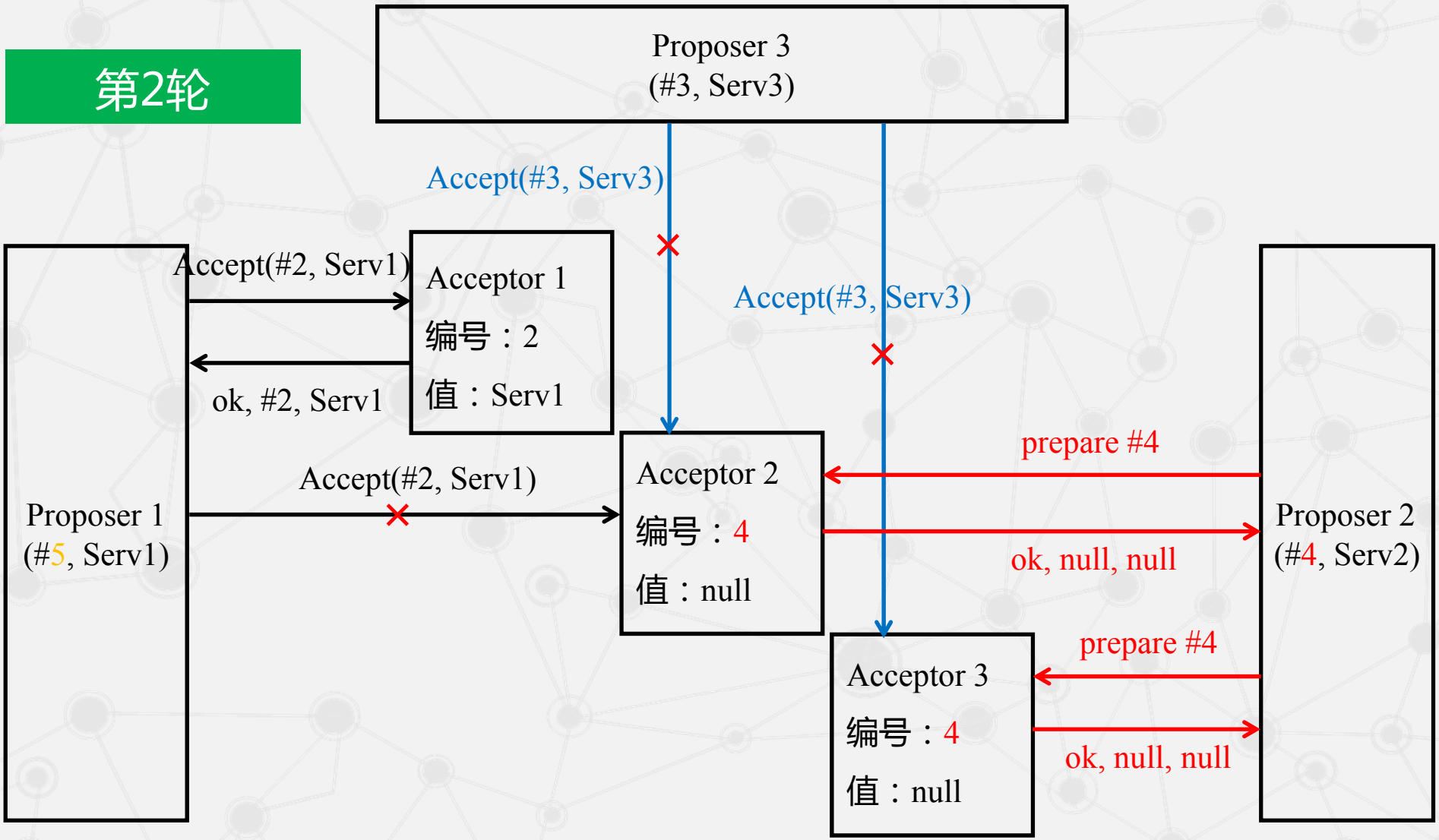
经过一段时间后，Proposer 收集到一些 Accept 回复提交成功，有几种情况：

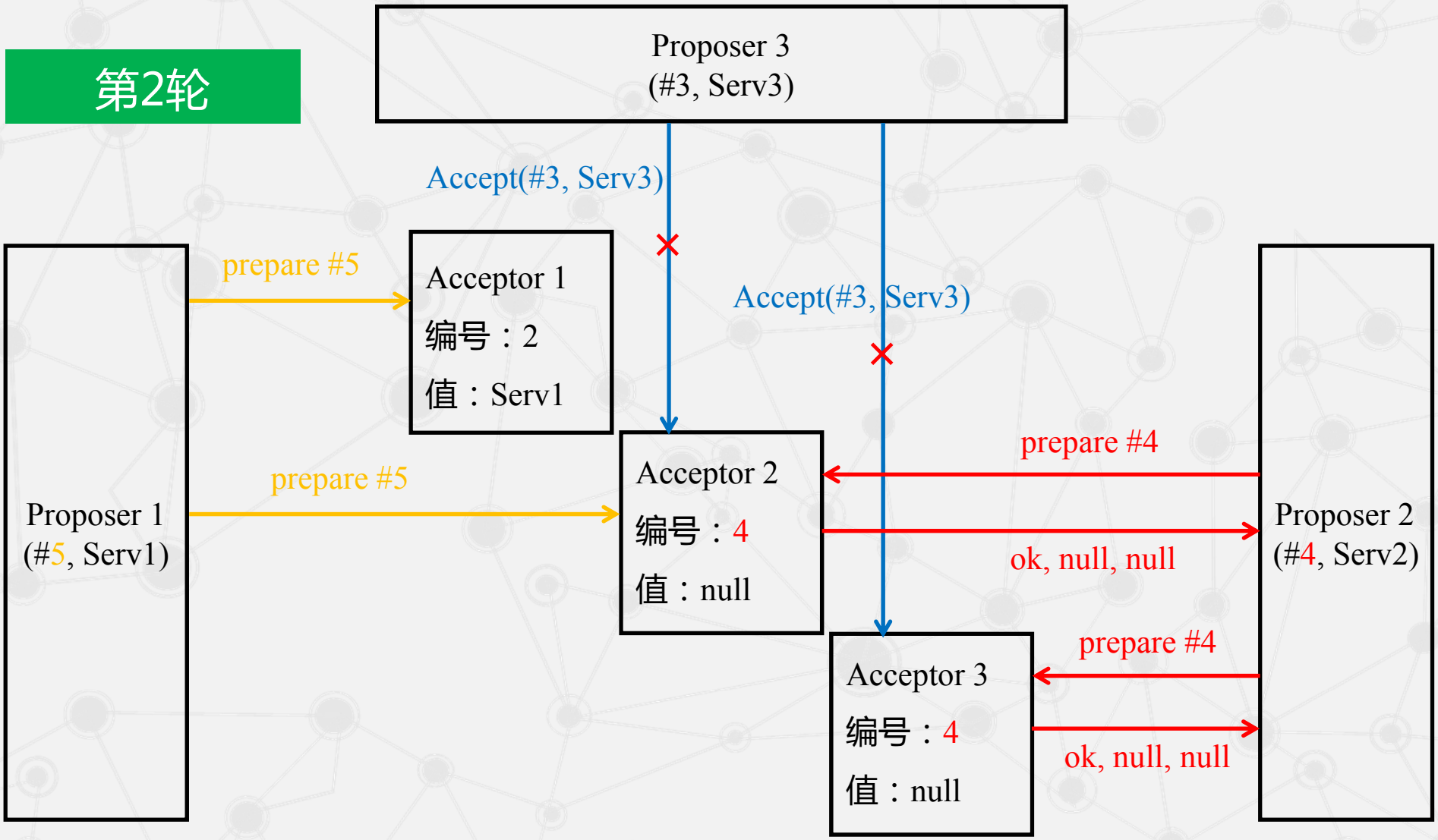
(1) 回复数量 > 一半的Acceptor数量，则表示提交value成功。此时，可以发一个广播给所有Proposer、Learner，通知它们已commit的value。

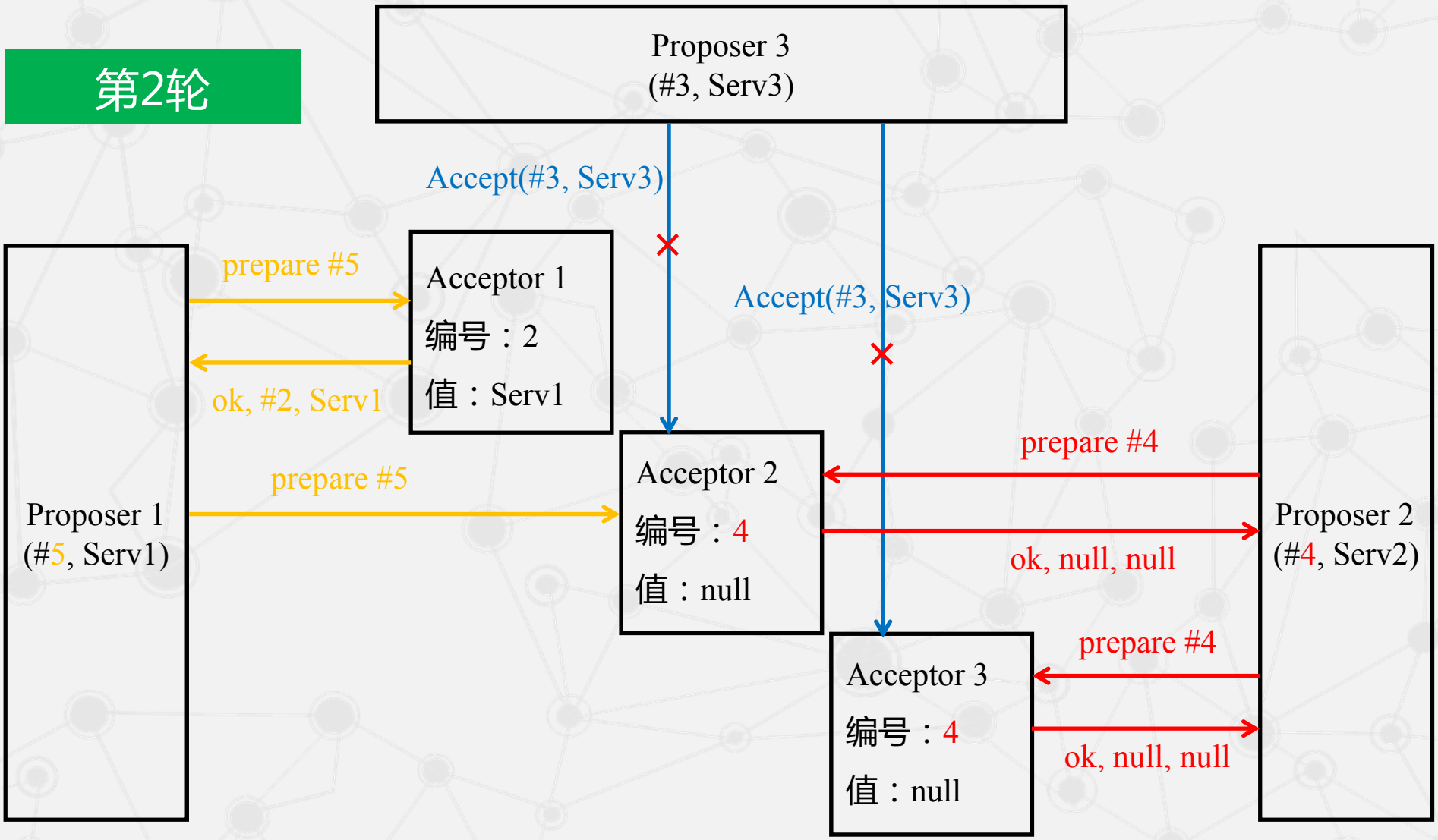
(2) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的 ProposalID，再转S1a执行。

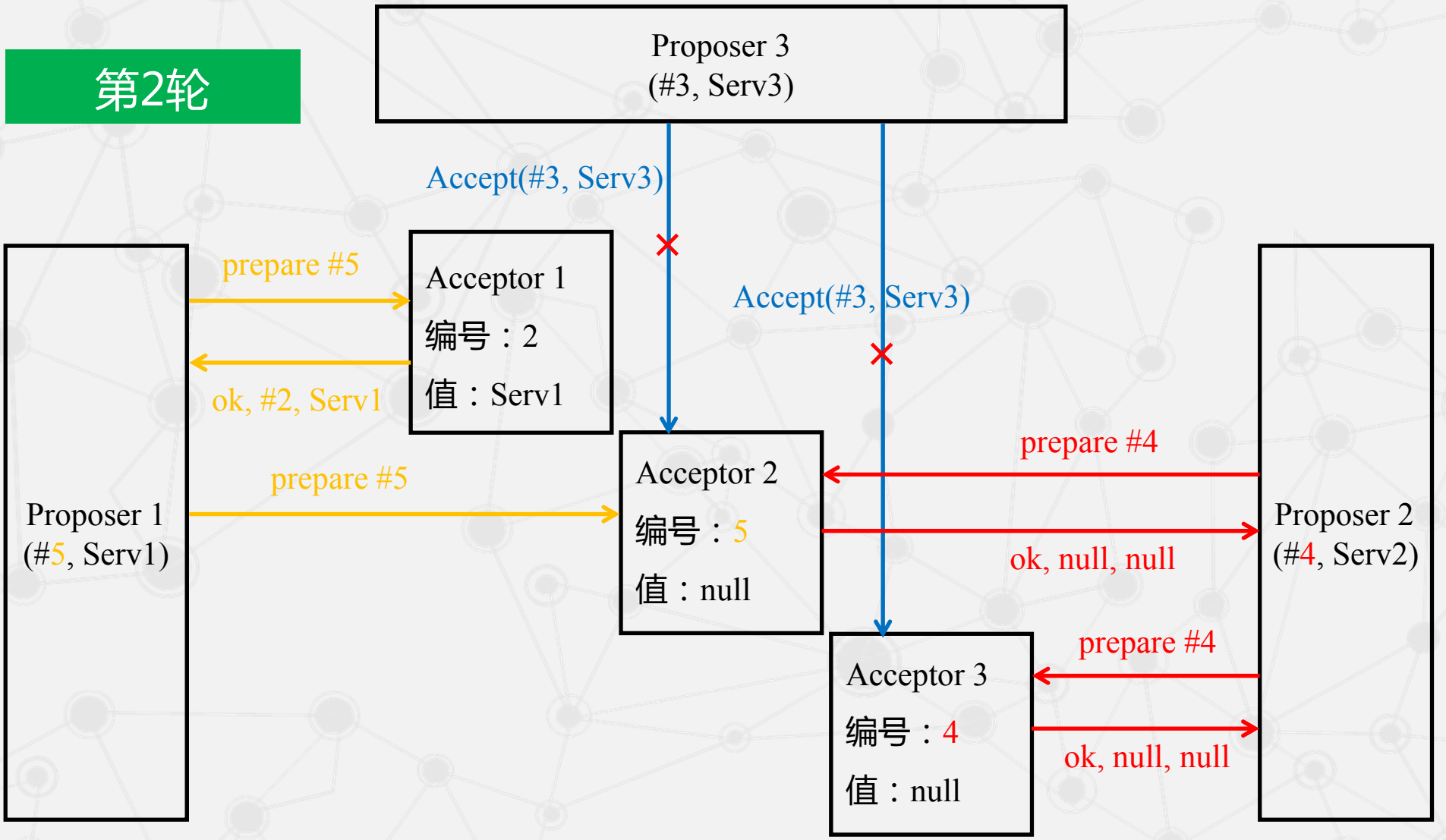
(3) 收到一条提交失败的回复，则尝试更新生成更大的 ProposalID，再转S1a执行。

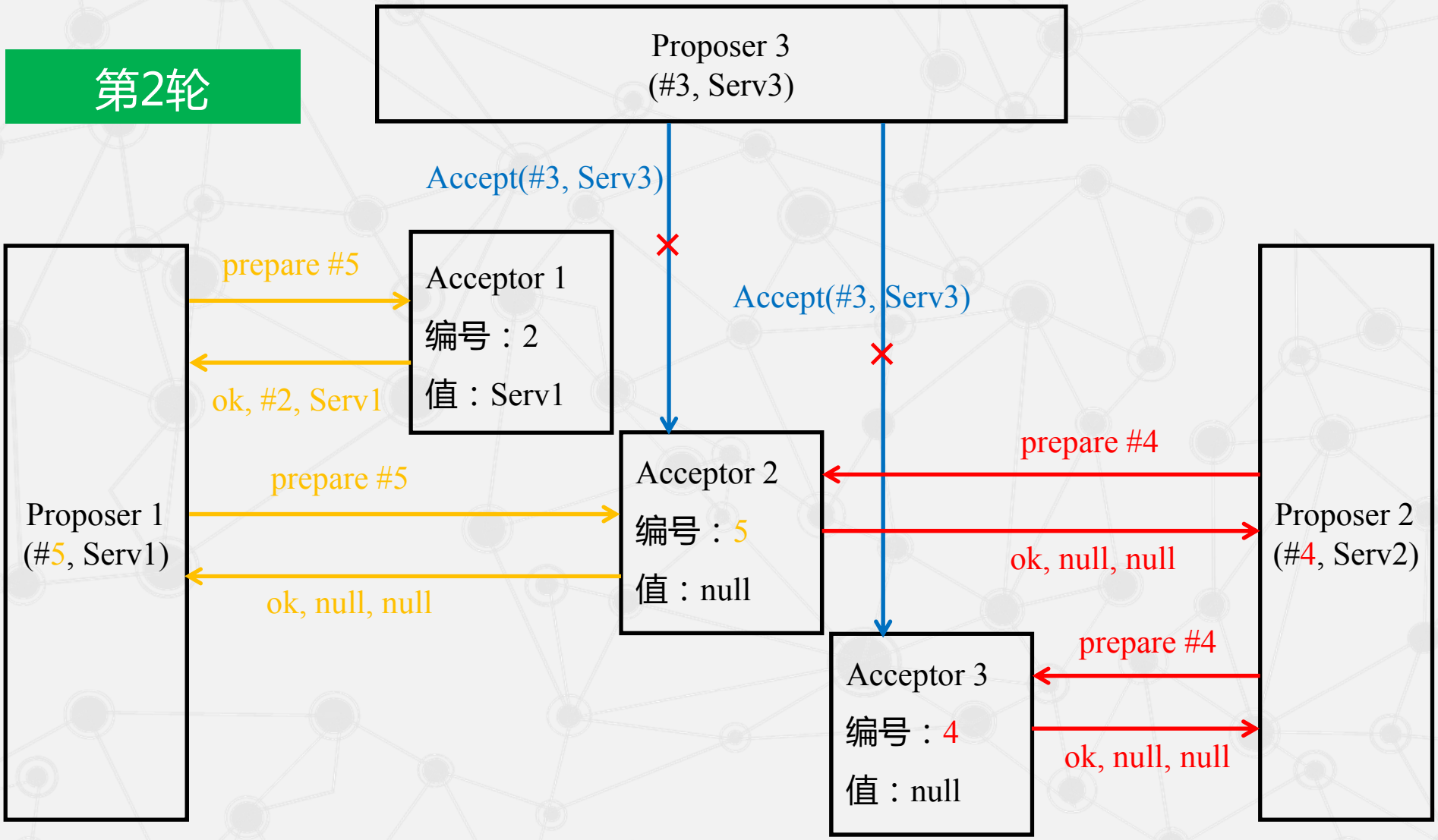

第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#2, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：4
值：null
prepare #4
ok, null, null
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：null
prepare #4
ok, null, null

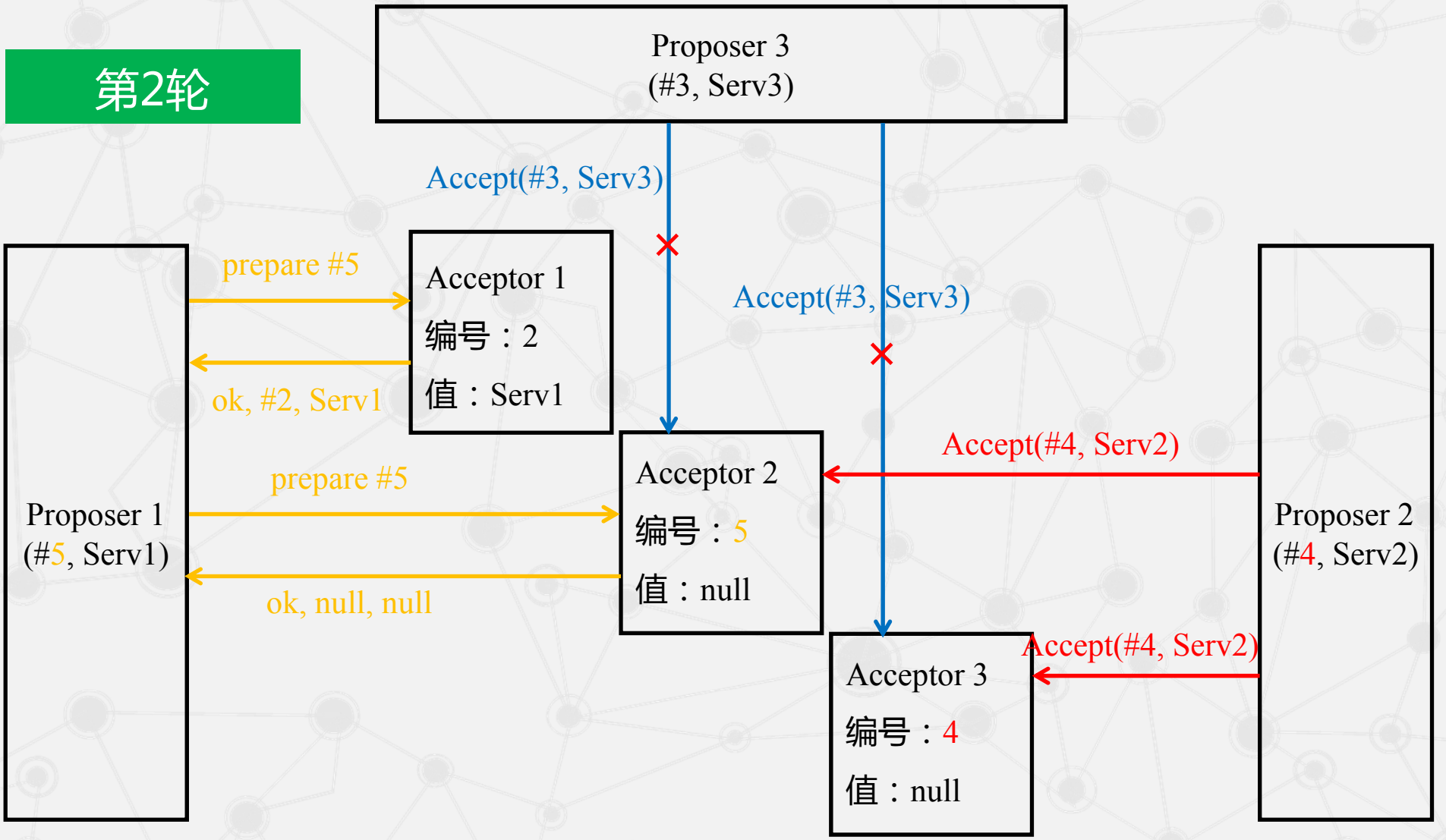

第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
Accept(#2, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#5, Serv1)
Accept(#2, Serv1)
Acceptor 2
编号：4
值：null
prepare #4
ok, null, null
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：null
prepare #4
ok, null, null

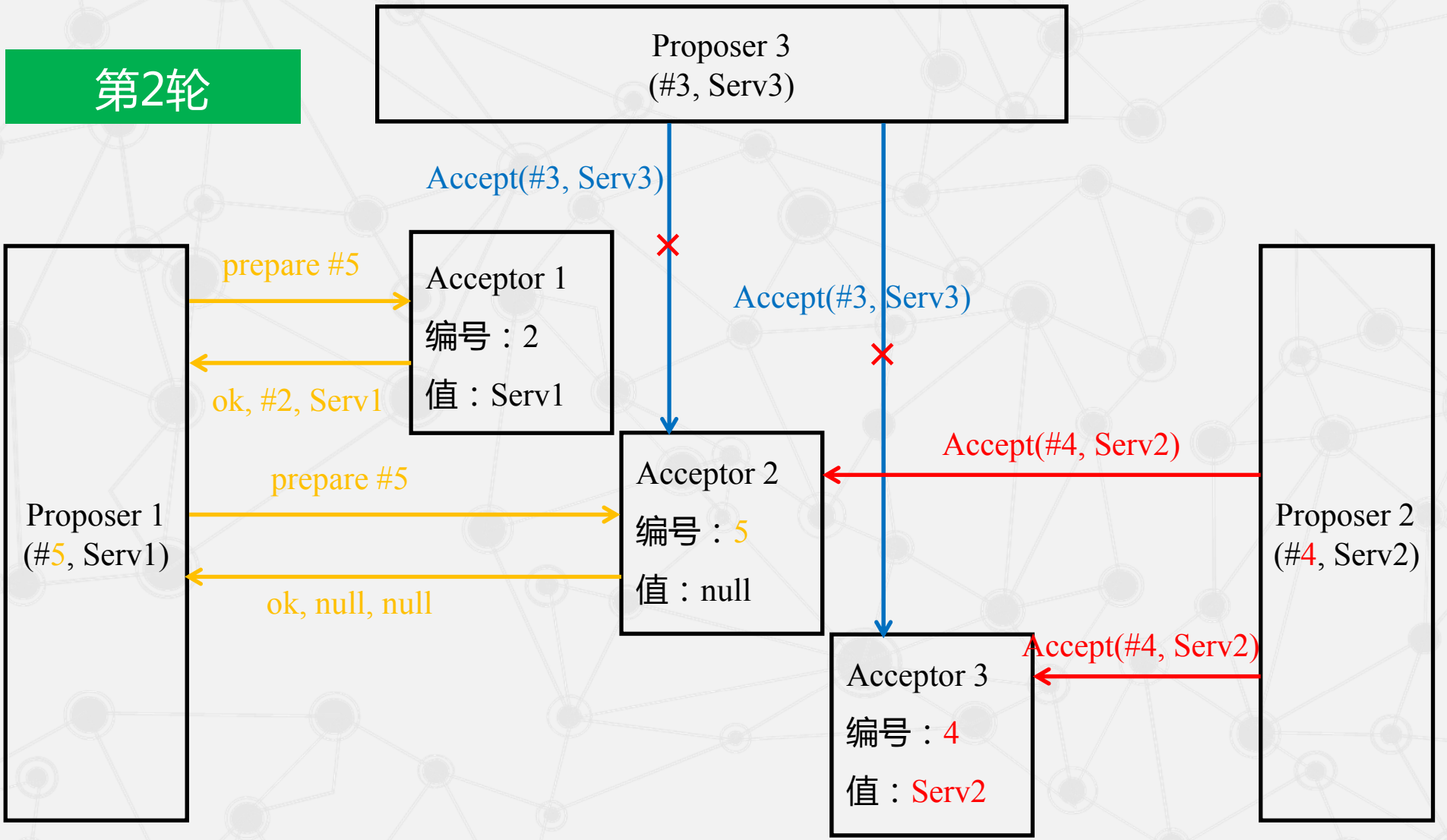

第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
prepare #5
Acceptor 1
编号：2
值：Serv1
Proposer 1
(#5, Serv1)
prepare #5
Acceptor 2
编号：4
值：null
prepare #4
ok, null, null
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：null
prepare #4
ok, null, null

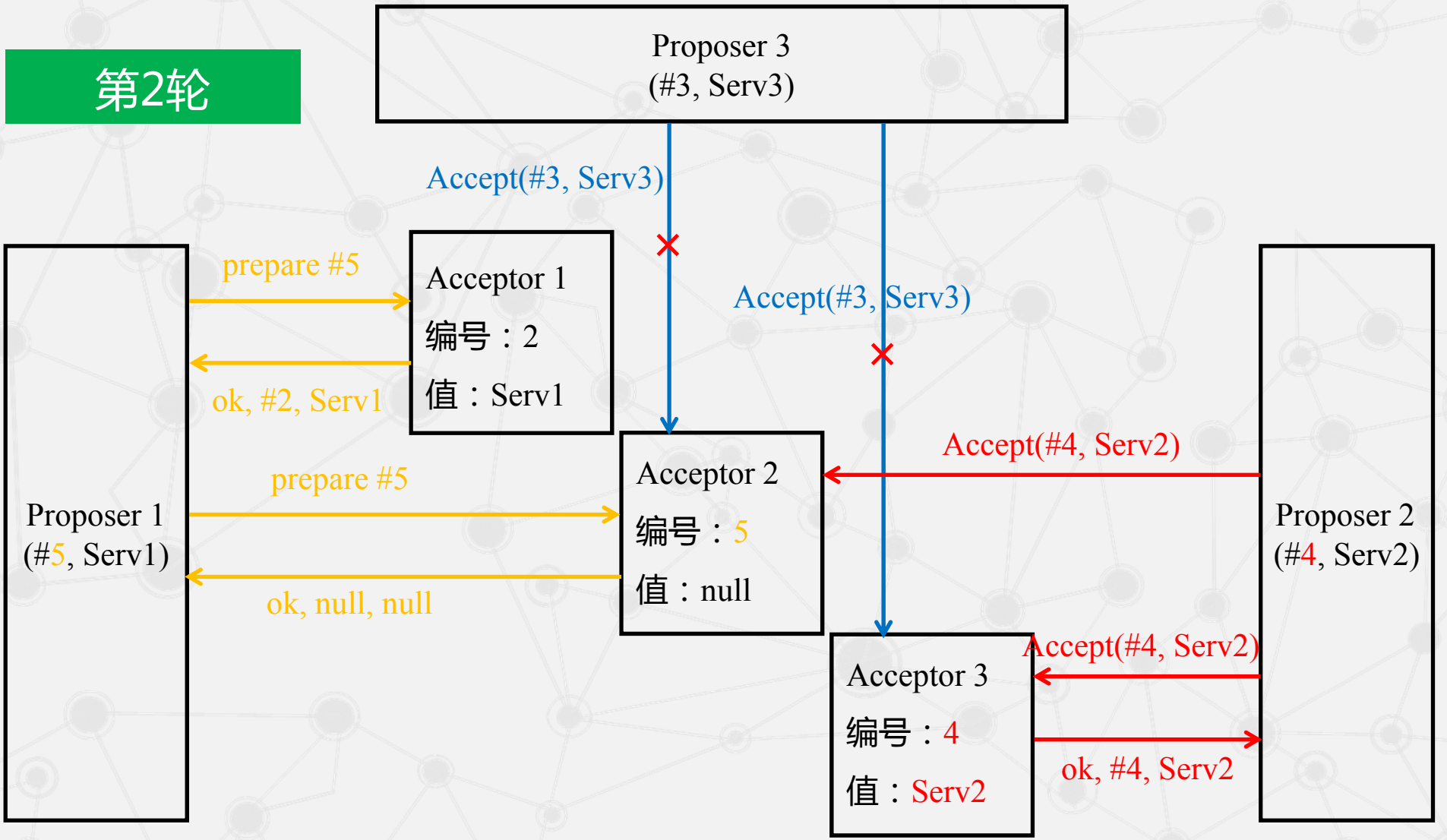

第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
prepare #5
ok, #2, Serv1
Acceptor 1
编号：2
值：Serv1
Proposer 1
(#5, Serv1)
prepare #5
Acceptor 2
编号：4
值：null
prepare #4
ok, null, null
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：null
prepare #4
ok, null, null

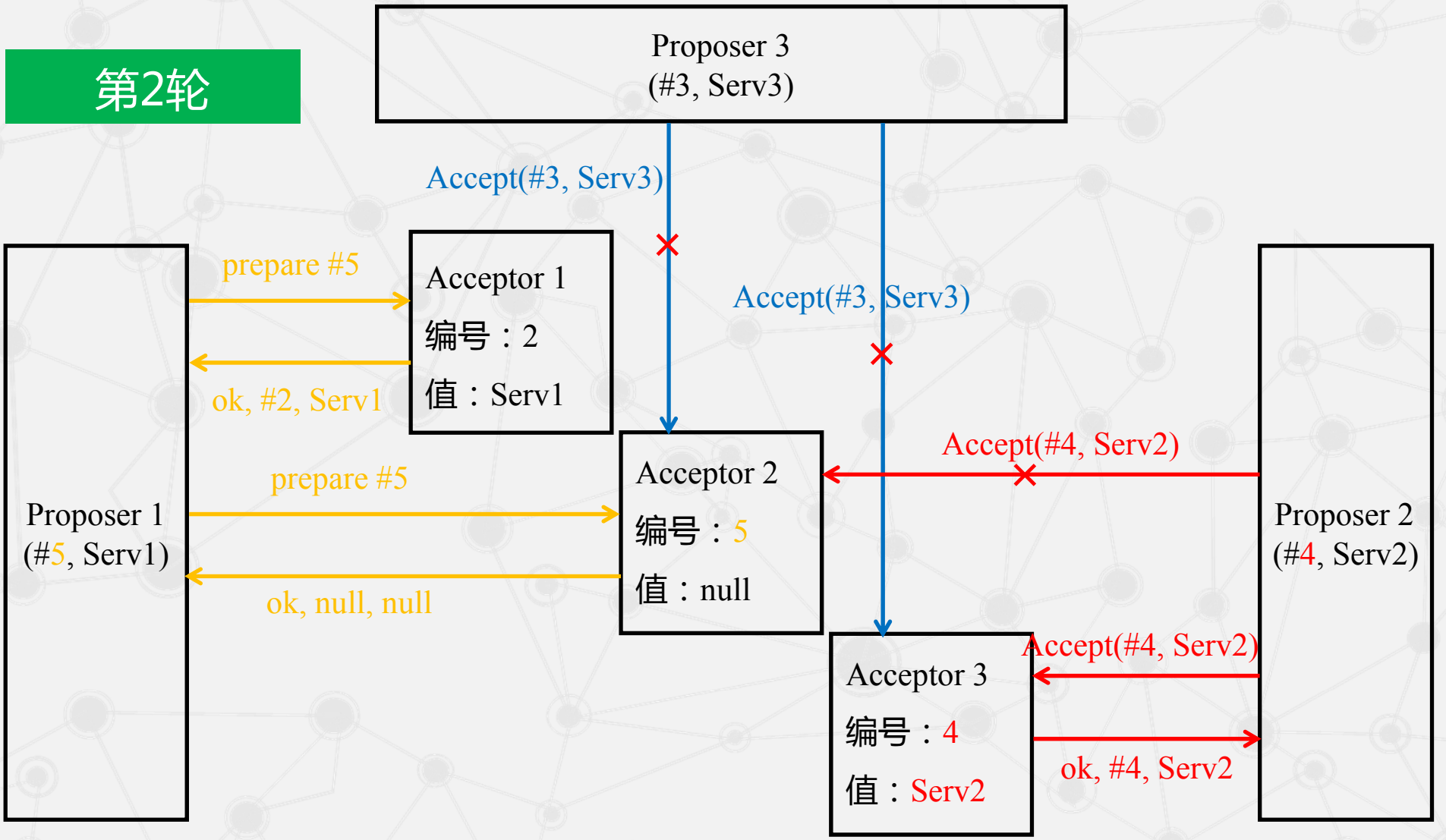

第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
prepare #5
ok, #2, Serv1
Acceptor 1
编号：2
值：Serv1
Proposer 1
(#5, Serv1)
prepare #5
Acceptor 2
编号：5
值：null
prepare #4
ok, null, null
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：null
prepare #4
ok, null, null

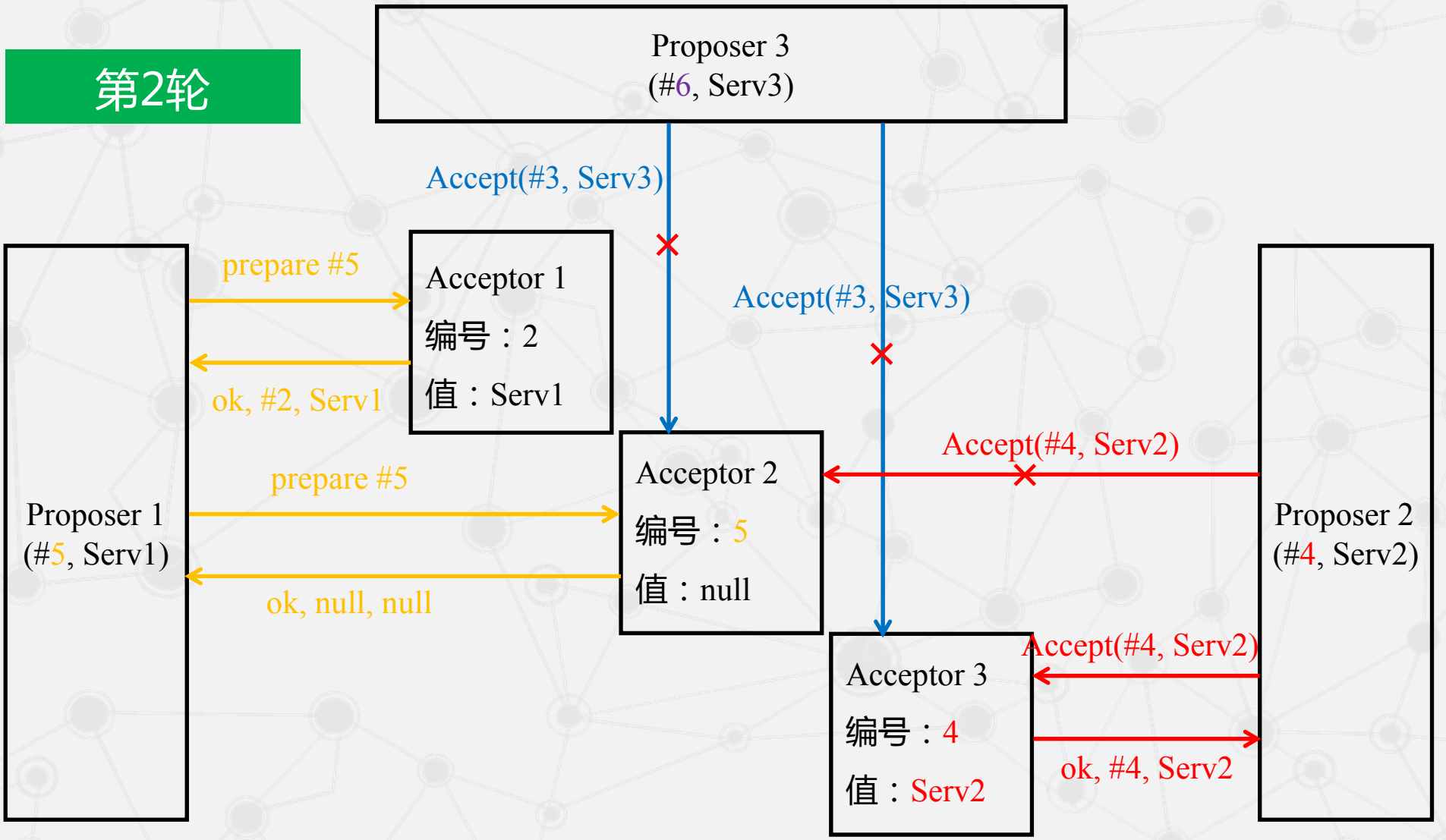

第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
prepare #5
ok, #2, Serv1
Acceptor 1
编号：2
值：Serv1
Proposer 1
(#5, Serv1)
prepare #5
ok, null, null
Acceptor 2
编号：5
值：null
prepare #4
ok, null, null
Proposer 2
(#4, Serv2)
prepare #4
ok, null, null
Acceptor 3
编号：4
值：null

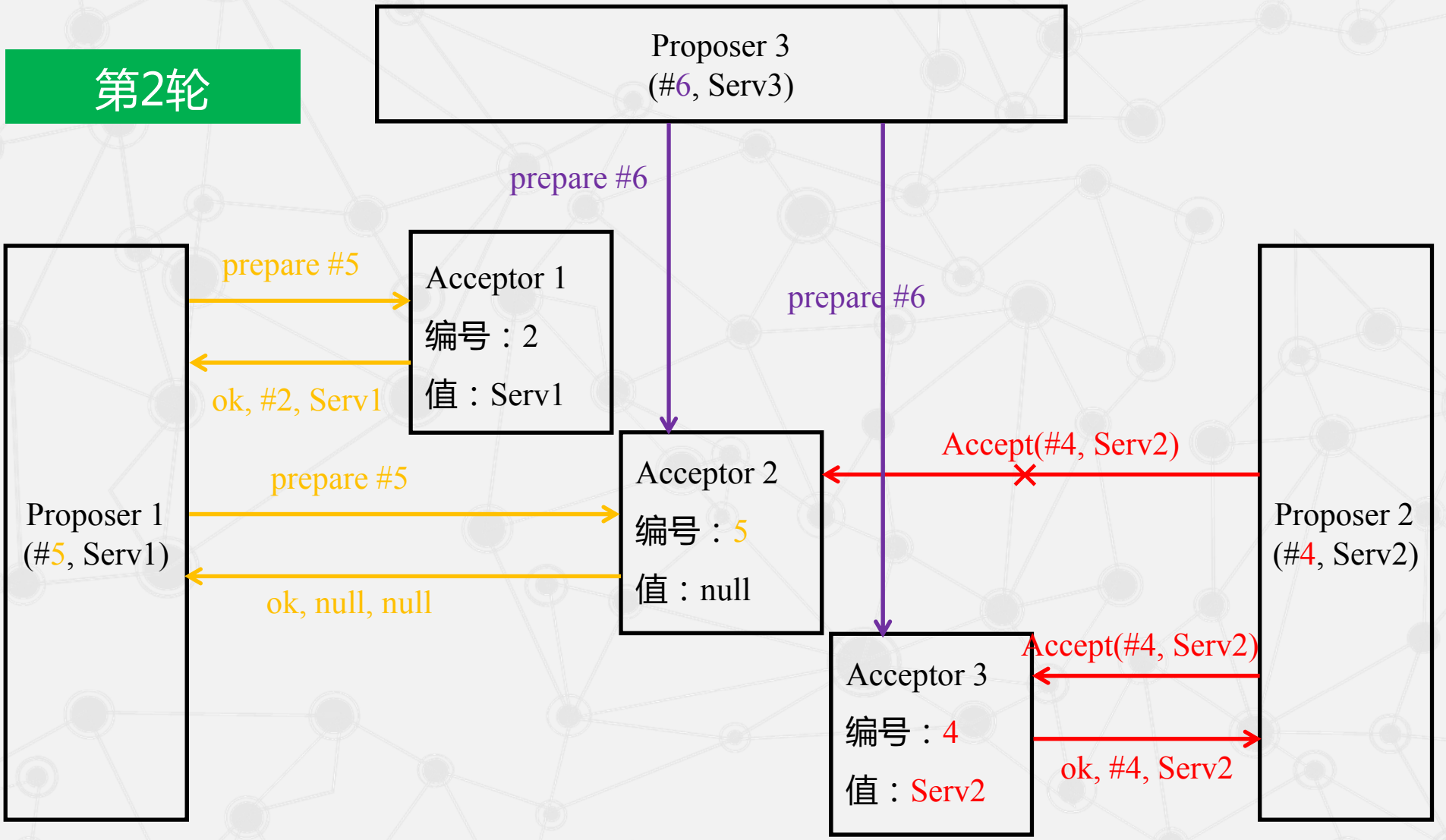

第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
prepare #5
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#5, Serv1)
prepare #5
Acceptor 2
编号：5
值：null
ok, null, null
Accept(#4, Serv2)
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：null
Accept(#4, Serv2)

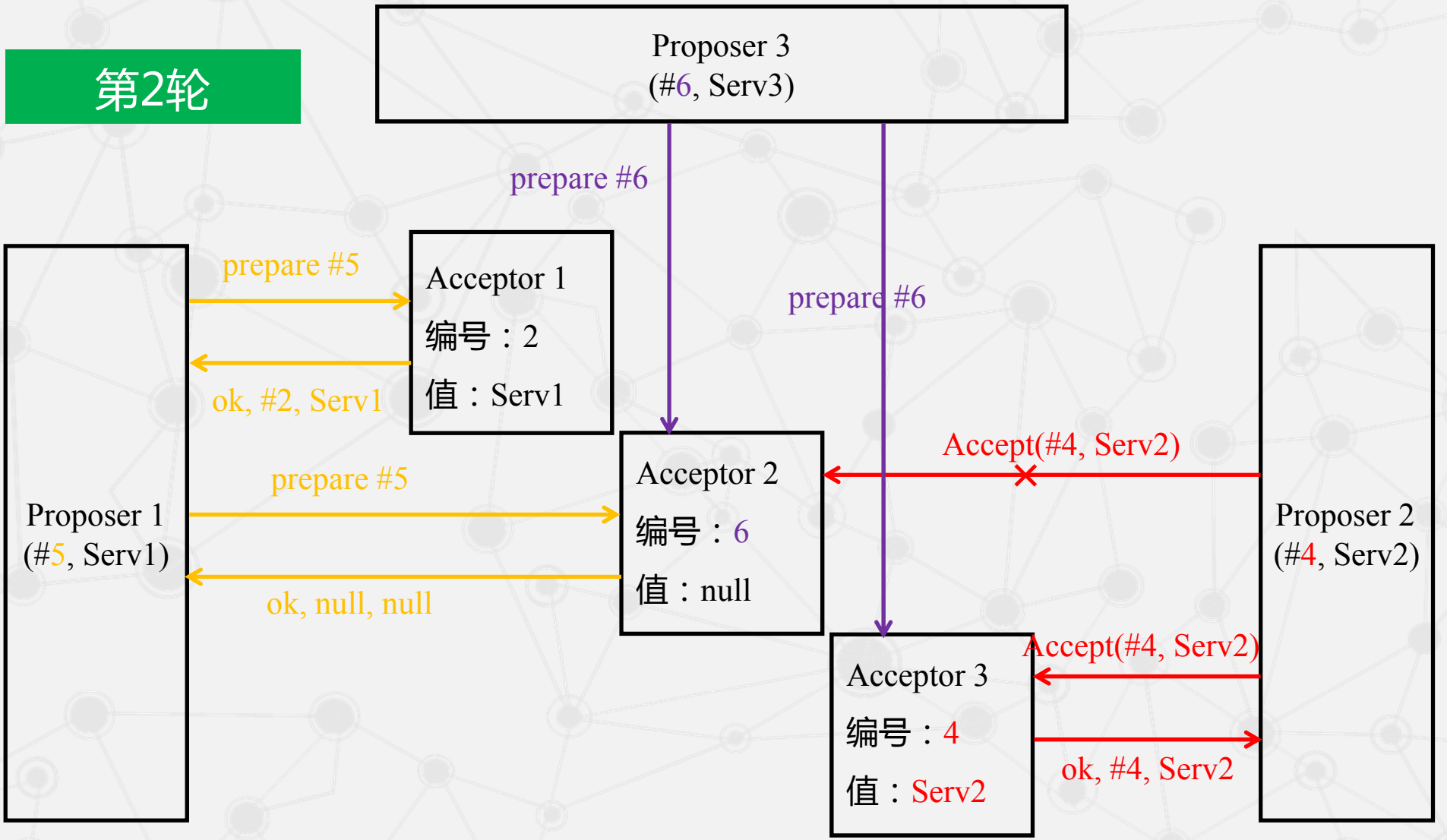

第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
prepare #5
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#5, Serv1)
prepare #5
Acceptor 2
编号：5
值：null
ok, null, null
Accept(#4, Serv2)
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：Serv2
Accept(#4, Serv2)

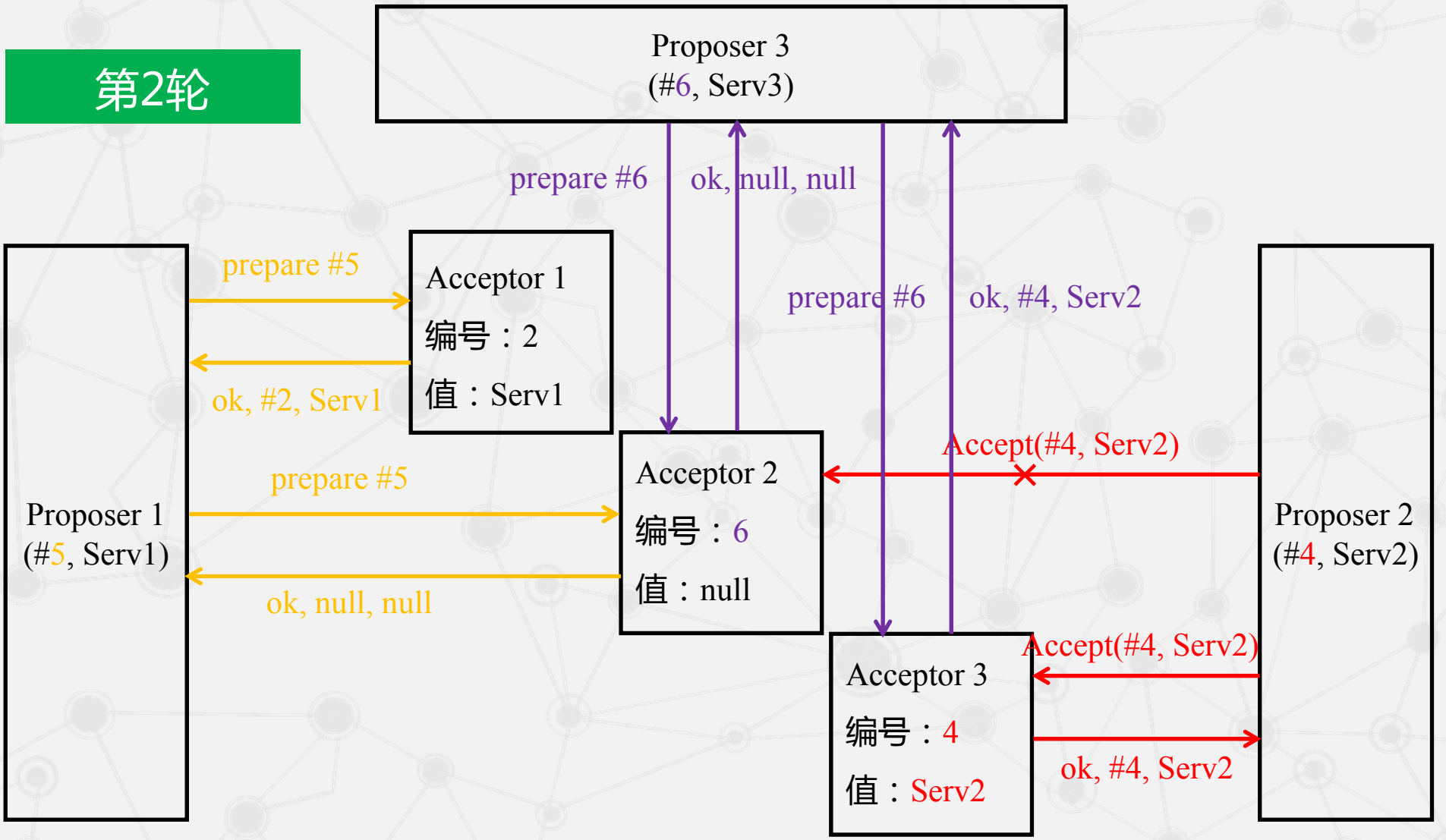

第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
prepare #5
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#4, Serv2)
Proposer 1
(#5, Serv1)
prepare #5
Acceptor 2
编号：5
值：null
ok, null, null
Proposer 2
(#4, Serv2)
Accept(#4, Serv2)
Acceptor 3
编号：4
值：Serv2
ok, #4, Serv2

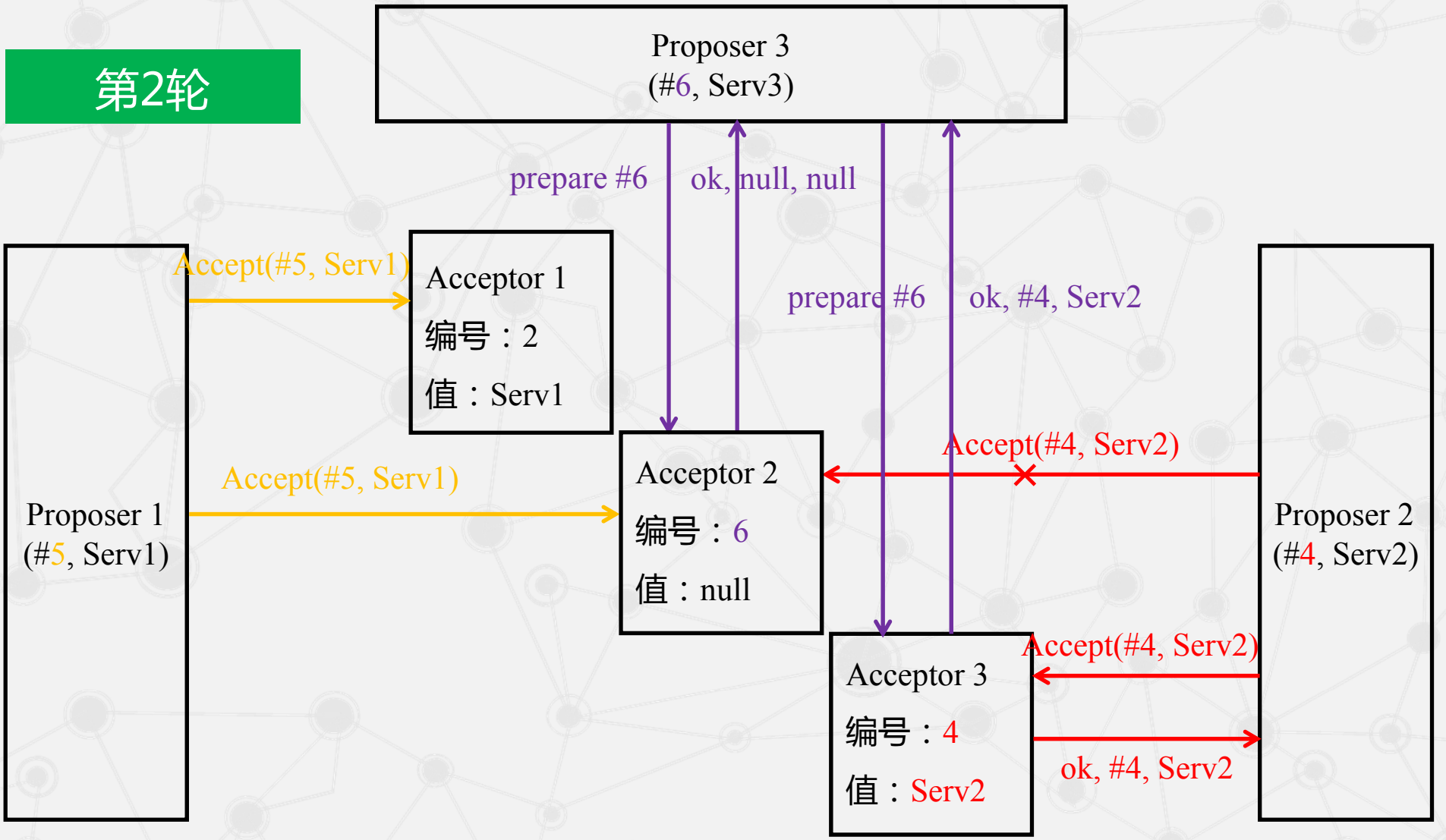

第2轮
Proposer 3
(#3, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
prepare #5
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#4, Serv2)
Proposer 1
(#5, Serv1)
prepare #5
Acceptor 2
编号：5
值：null
ok, null, null
Proposer 2
(#4, Serv2)
Accept(#4, Serv2)
Acceptor 3
编号：4
值：Serv2
ok, #4, Serv2

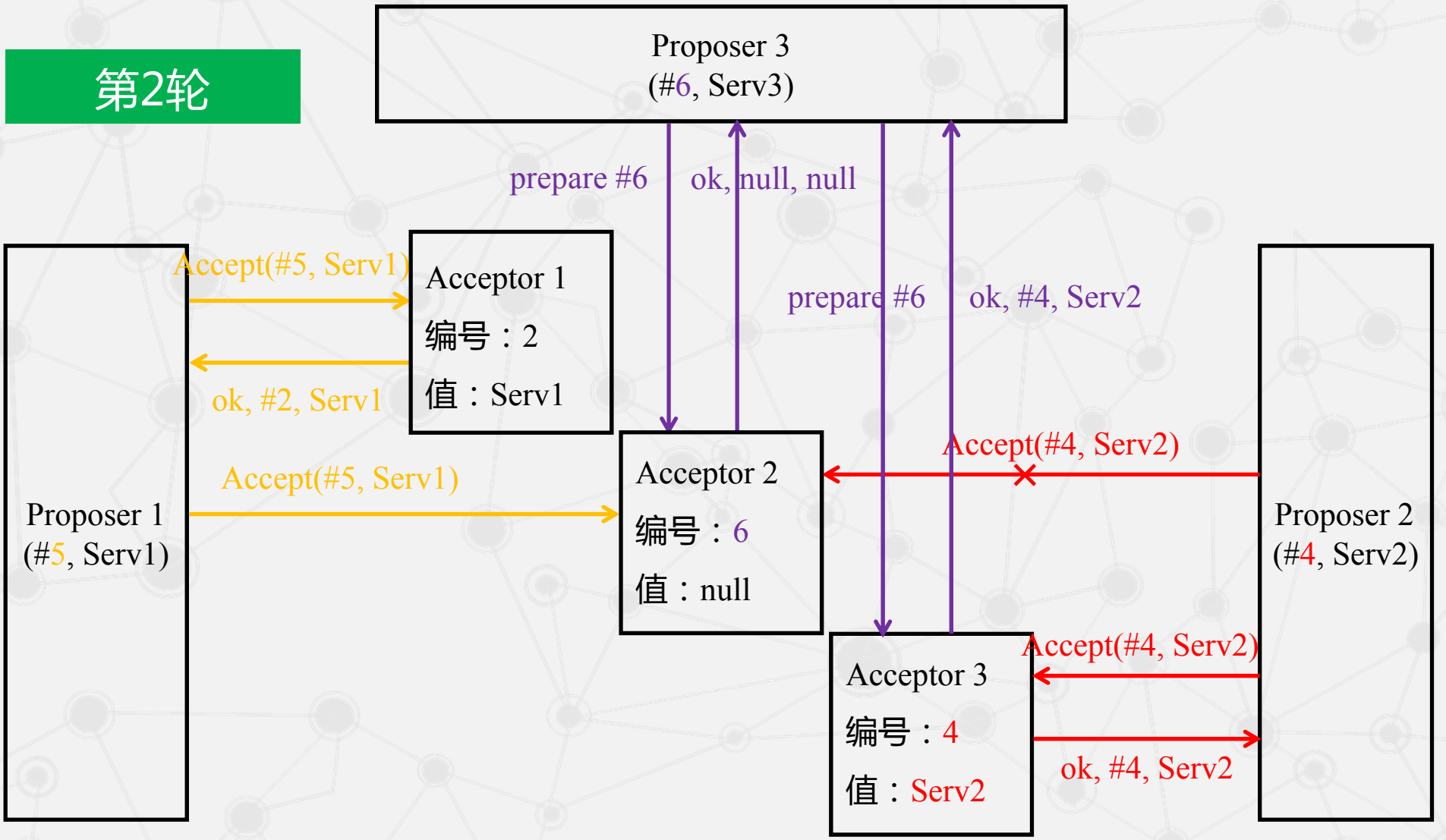

第2轮
Proposer 3
(#6, Serv3)
Accept(#3, Serv3)
Accept(#3, Serv3)
prepare #5
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#5, Serv1)
prepare #5
Acceptor 2
编号：5
值：null
ok, null, null
Accept(#4, Serv2)
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：Serv2
Accept(#4, Serv2)
ok, #4, Serv2

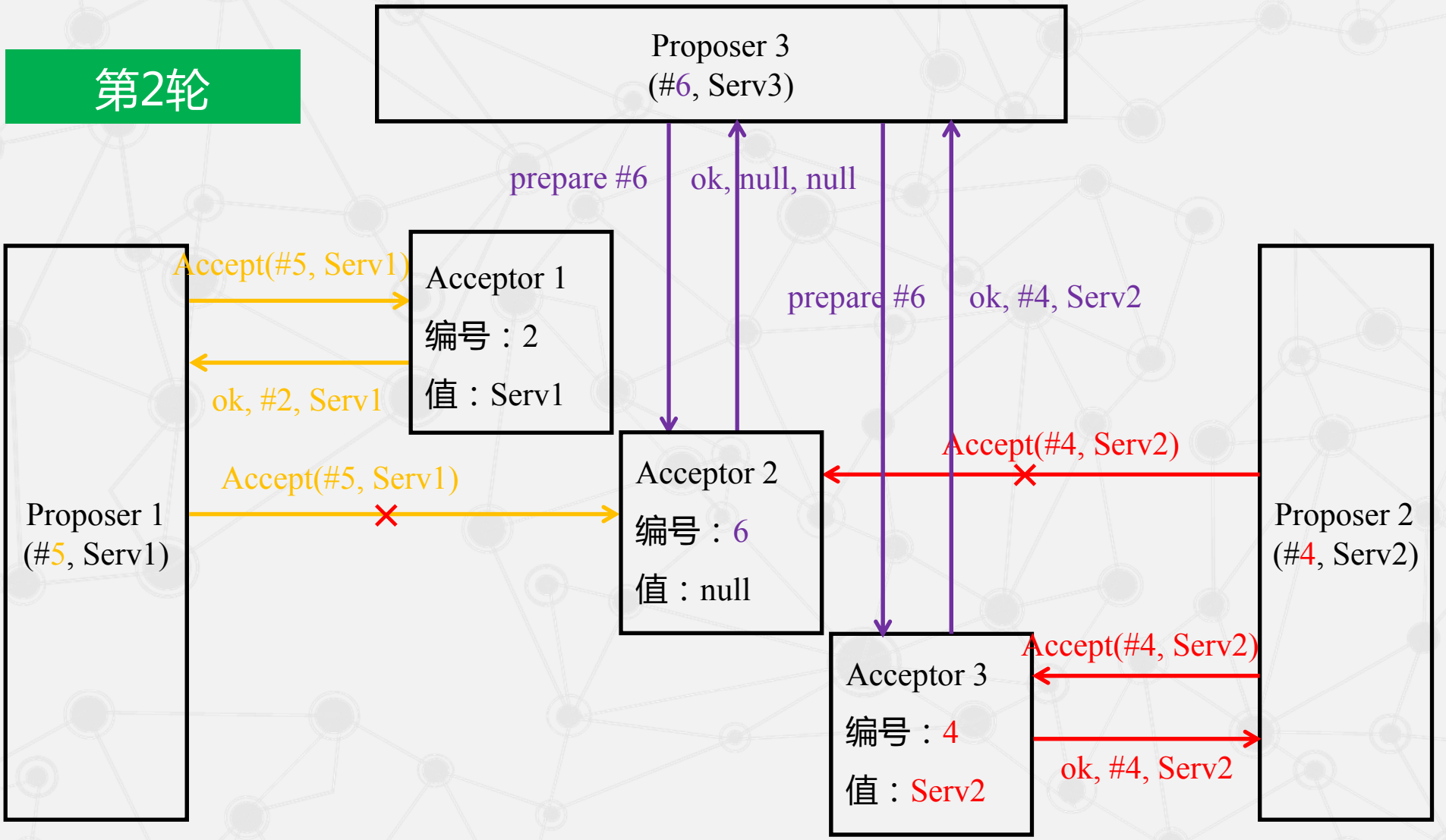

第2轮
Proposer 3
(#6, Serv3)
prepare #6
prepare #6
prepare #5
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#5, Serv1)
prepare #5
Acceptor 2
编号：5
值：null
ok, null, null
Accept(#4, Serv2)
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：Serv2
Accept(#4, Serv2)
ok, #4, Serv2

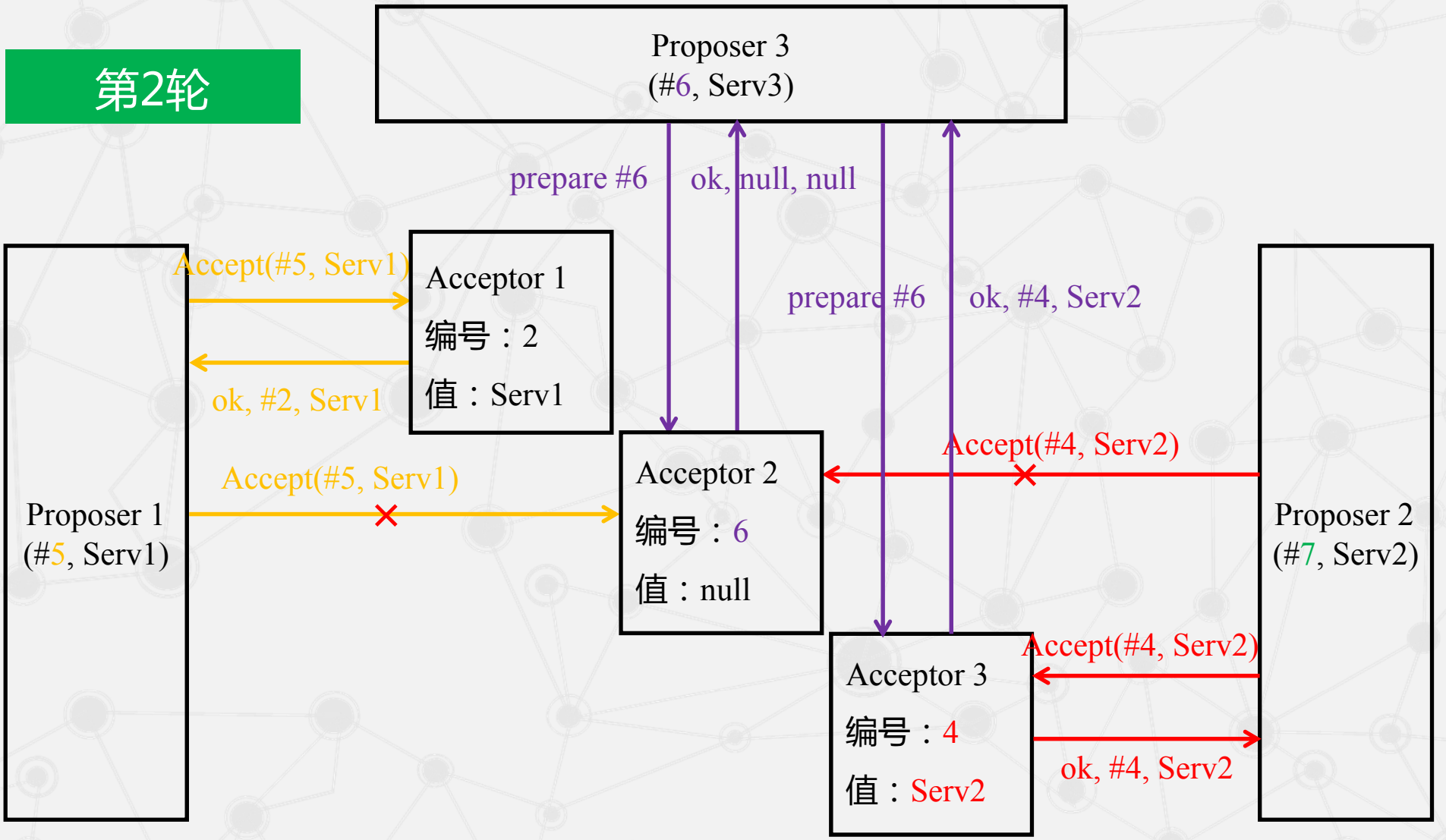

第2轮
Proposer 3
(#6, Serv3)
prepare #6
prepare #6
prepare #5
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#5, Serv1)
prepare #5
Acceptor 2
编号：6
值：null
ok, null, null
Accept(#4, Serv2)
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：Serv2
Accept(#4, Serv2)
ok, #4, Serv2

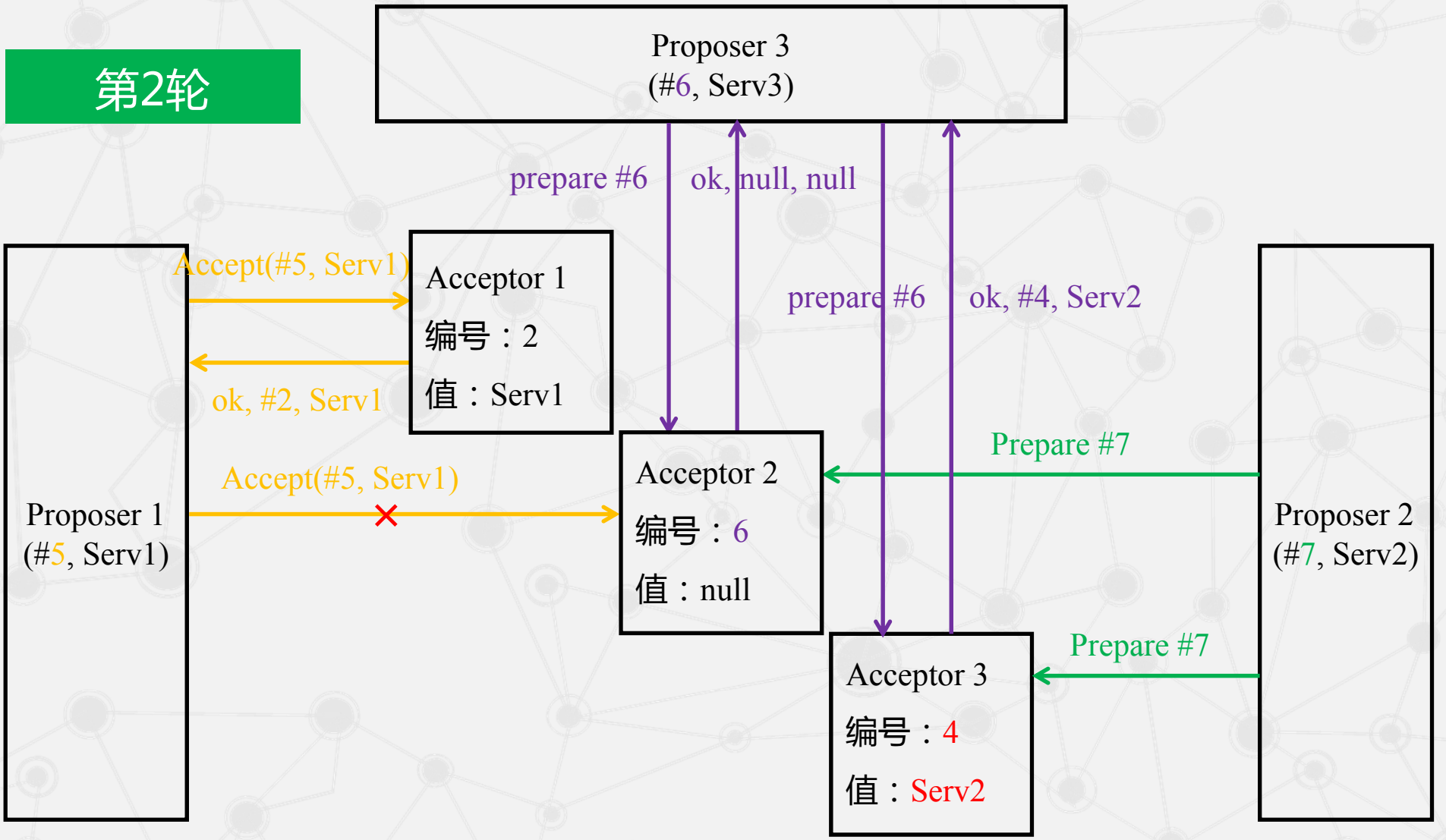

第2轮
Proposer 3
(#6, Serv3)
prepare #6
ok, null, null
prepare #6
ok, #4, Serv2
prepare #5
ok, #2, Serv1
Acceptor 1
编号：2
值：Serv1
Accept(#4, Serv2)
prepare #5
ok, null, null
Proposer 1
(#5, Serv1)
Acceptor 2
编号：6
值：null
Proposer 2
(#4, Serv2)
Accept(#4, Serv2)
ok, #4, Serv2
Acceptor 3
编号：4
值：Serv2

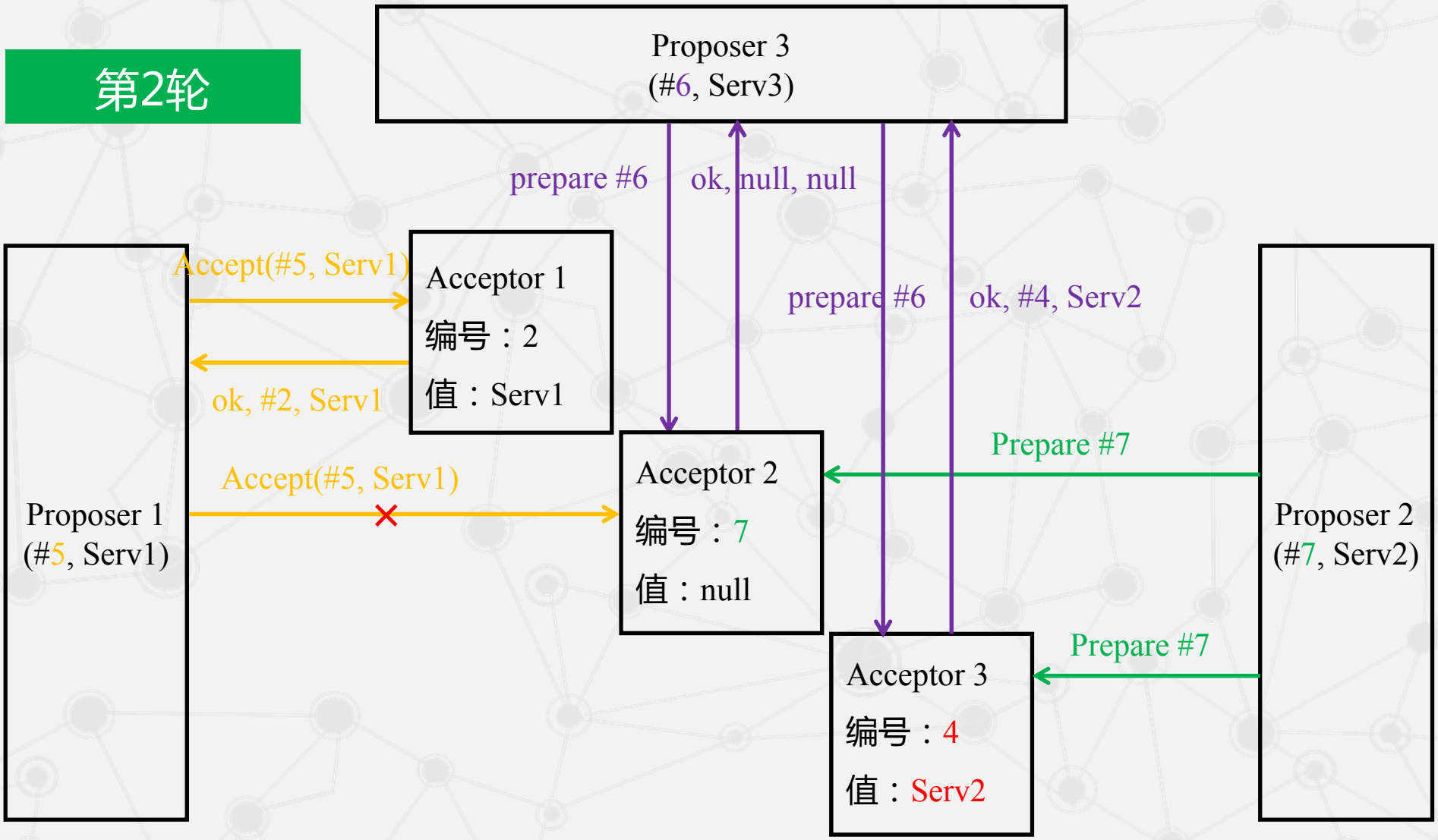

第2轮
Proposer 3
(#6, Serv3)
prepare #6
ok, null, null
Accept(#5, Serv1)
Acceptor 1
编号：2
值：Serv1
prepare #6
ok, #4, Serv2
Accept(#4, Serv2)
Proposer 1
(#5, Serv1)
Accept(#5, Serv1)
Acceptor 2
编号：6
值：null
Proposer 2
(#4, Serv2)
Accept(#4, Serv2)
Acceptor 3
编号：4
值：Serv2
ok, #4, Serv2

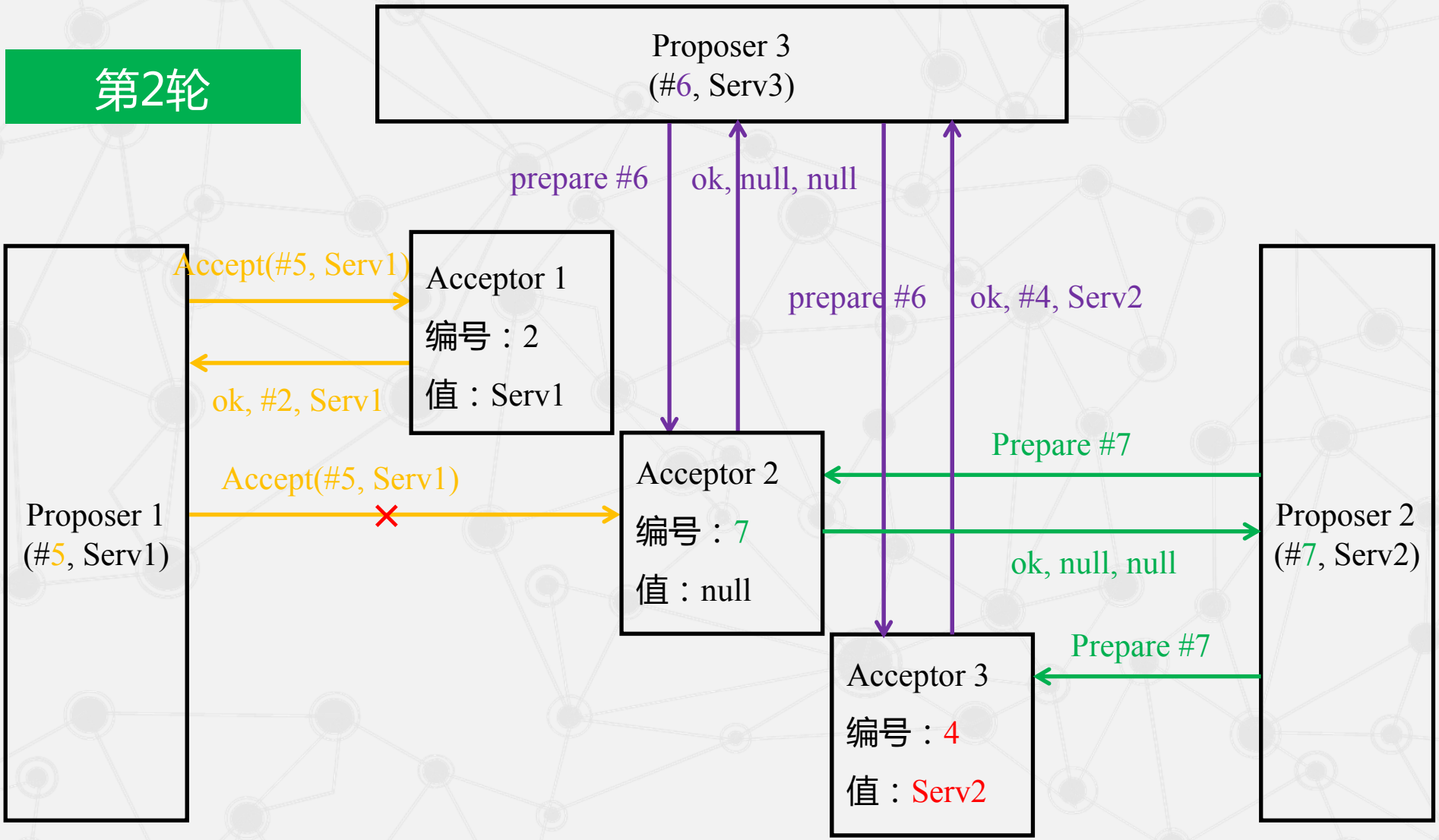

第2轮
Proposer 3
(#6, Serv3)
prepare #6
ok, null, null
Accept(#5, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
prepare #6
ok, #4, Serv2
Accept(#4, Serv2)
Proposer 1
(#5, Serv1)
Accept(#5, Serv1)
Acceptor 2
编号：6
值：null
Proposer 2
(#4, Serv2)
Acceptor 3
编号：4
值：Serv2
Accept(#4, Serv2)
ok, #4, Serv2

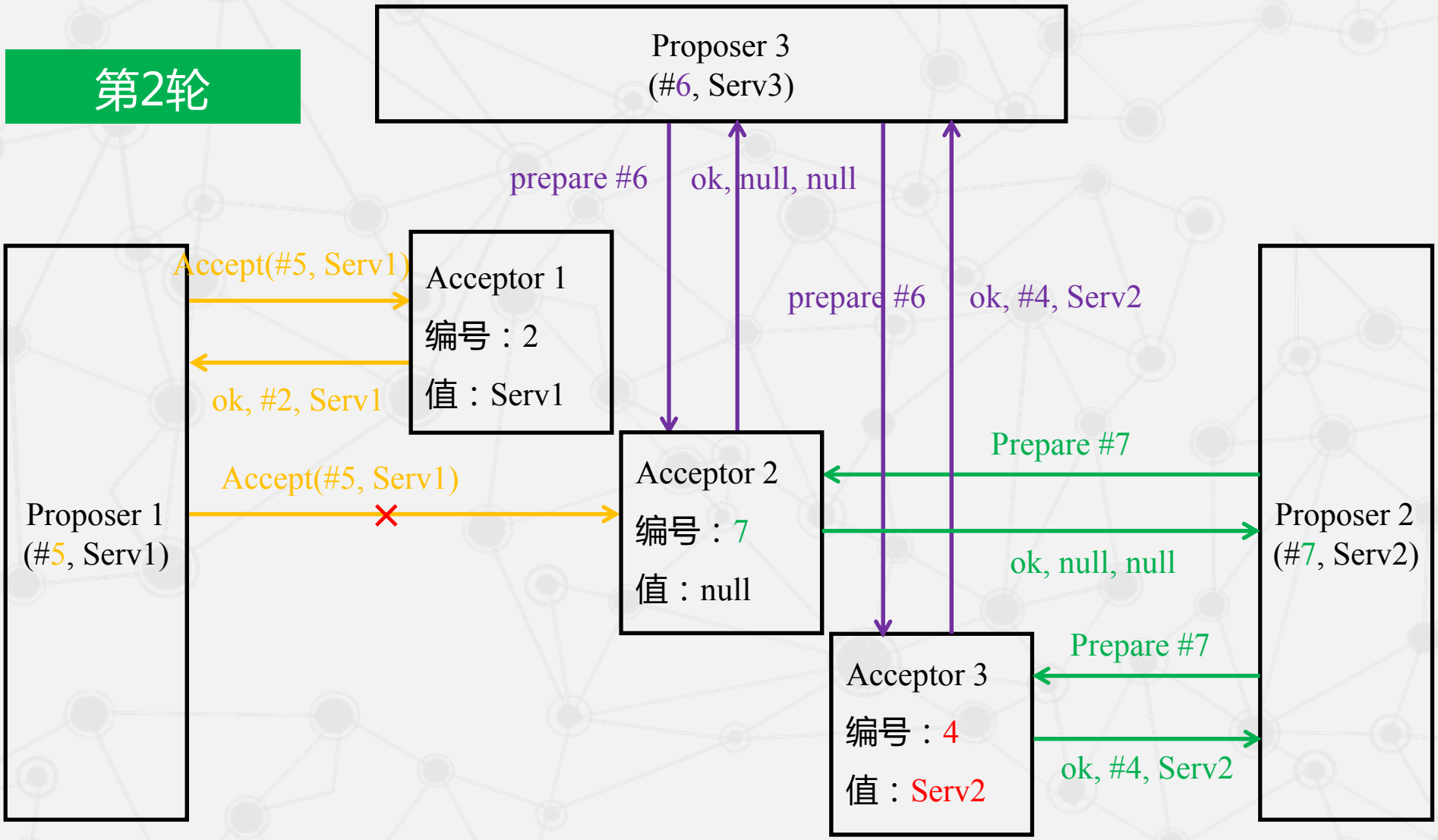

第2轮
Proposer 3
(#6, Serv3)
prepare #6
ok, null, null
prepare #6
ok, #4, Serv2
Accept(#5, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#4, Serv2)
Accept(#5, Serv1)
Proposer 1
(#5, Serv1)
Acceptor 2
编号：6
值：null
Proposer 2
(#4, Serv2)
Accept(#4, Serv2)
Acceptor 3
编号：4
值：Serv2
ok, #4, Serv2


第2轮
Proposer 3
(#6, Serv3)
prepare #6
ok, null, null
Accept(#5, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
prepare #6
ok, #4, Serv2
Accept(#4, Serv2)
Accept(#5, Serv1)
Proposer 1
(#5, Serv1)
Acceptor 2
编号：6
值：null
Proposer 2
(#7, Serv2)
Accept(#4, Serv2)
Acceptor 3
编号：4
值：Serv2
ok, #4, Serv2

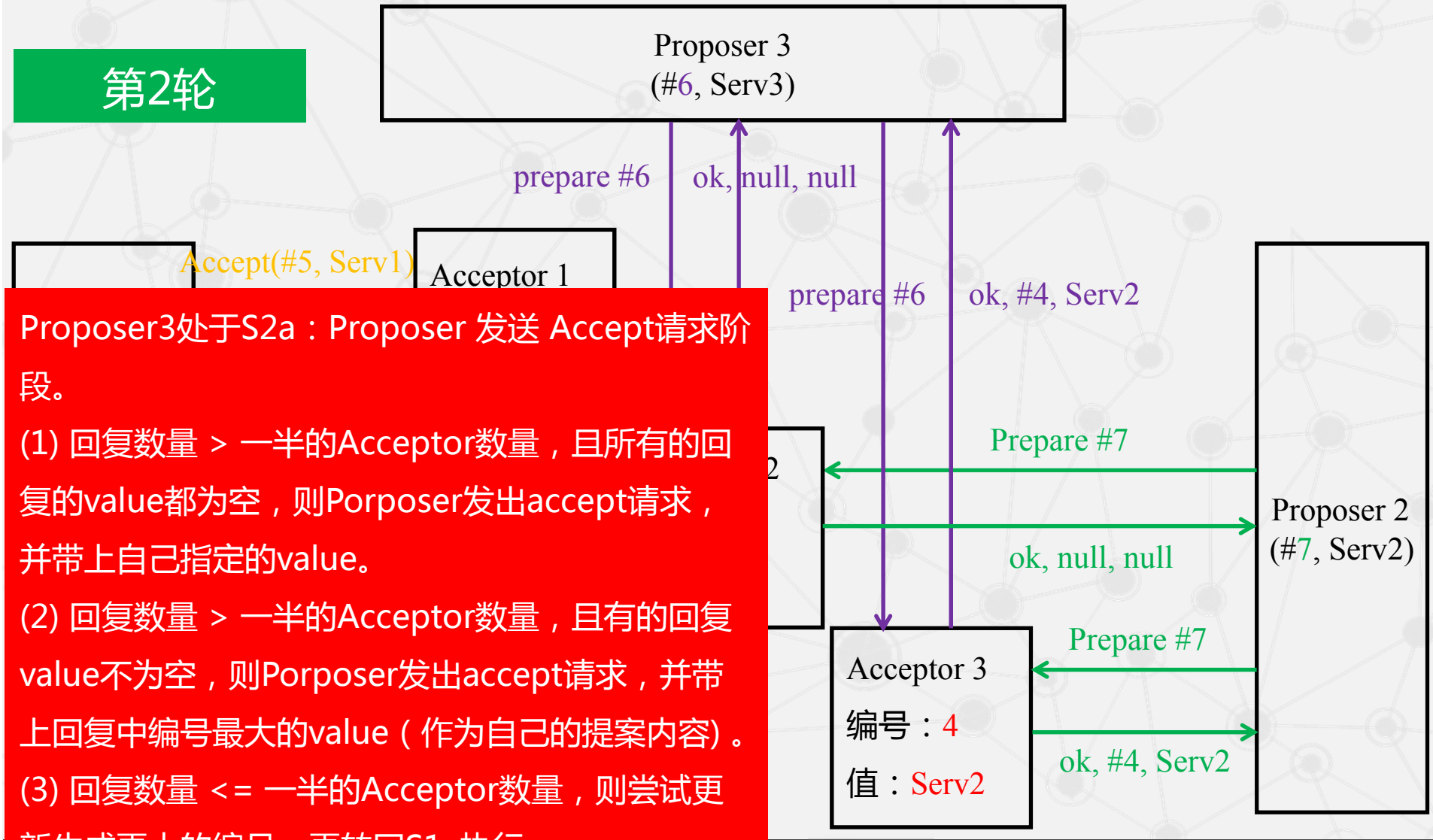

第2轮
Proposer 3
(#6, Serv3)
prepare #6
ok, null, null
prepare #6
ok, #4, Serv2
Accept(#5, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Prepare #7
Accept(#5, Serv1)
Proposer 1
(#5, Serv1)
Acceptor 2
编号：6
值：null
Proposer 2
(#7, Serv2)
Prepare #7
Acceptor 3
编号：4
值：Serv2


第2轮
Proposer 3
(#6, Serv3)
prepare #6
ok, null, null
prepare #6
ok, #4, Serv2
Accept(#5, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Prepare #7
Accept(#5, Serv1)
Proposer 1
(#5, Serv1)
Acceptor 2
编号：7
值：null
Proposer 2
(#7, Serv2)
Prepare #7
Acceptor 3
编号：4
值：Serv2


第2轮
Proposer 3
(#6, Serv3)
prepare #6
ok, null, null
prepare #6
ok, #4, Serv2
Accept(#5, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#5, Serv1)
Accept(#5, Serv1)
Acceptor 2
编号：7
值：null
Prepare #7
ok, null, null
Proposer 2
(#7, Serv2)
Acceptor 3
编号：4
值：Serv2
Prepare #7


第2轮
Proposer 3
(#6, Serv3)
prepare #6
ok, null, null
prepare #6
ok, #4, Serv2
Accept(#5, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Proposer 1
(#5, Serv1)
Accept(#5, Serv1)
Acceptor 2
编号：7
值：null
Prepare #7
ok, null, null
Proposer 2
(#7, Serv2)
Acceptor 3
编号：4
值：Serv2
Prepare #7
ok, #4, Serv2

第2轮



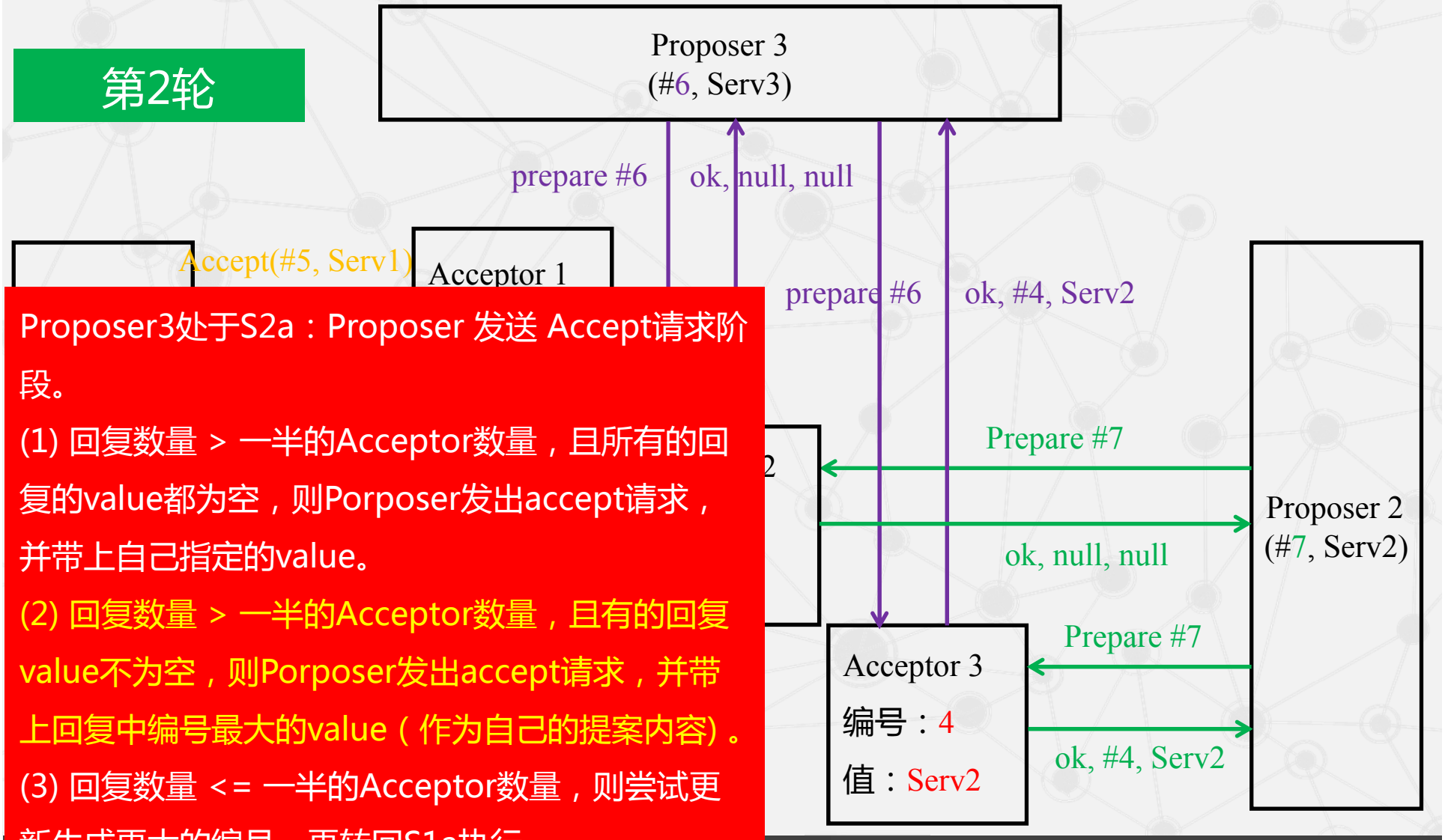
Proposer3处于S2a：Proposer 发送 Accept请求阶段。

(1) 回复数量 > 一半的Acceptor数量，且所有的回复的value都为空，则Porposer发出accept请求，并带上自己指定的value。

(2) 回复数量 > 一半的Acceptor数量，且有的回复value不为空，则Porposer发出accept请求，并带上回复中编号最大的value（作为自己的提案内容）。

(3) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的编号，再转回S1a执行。

第2轮

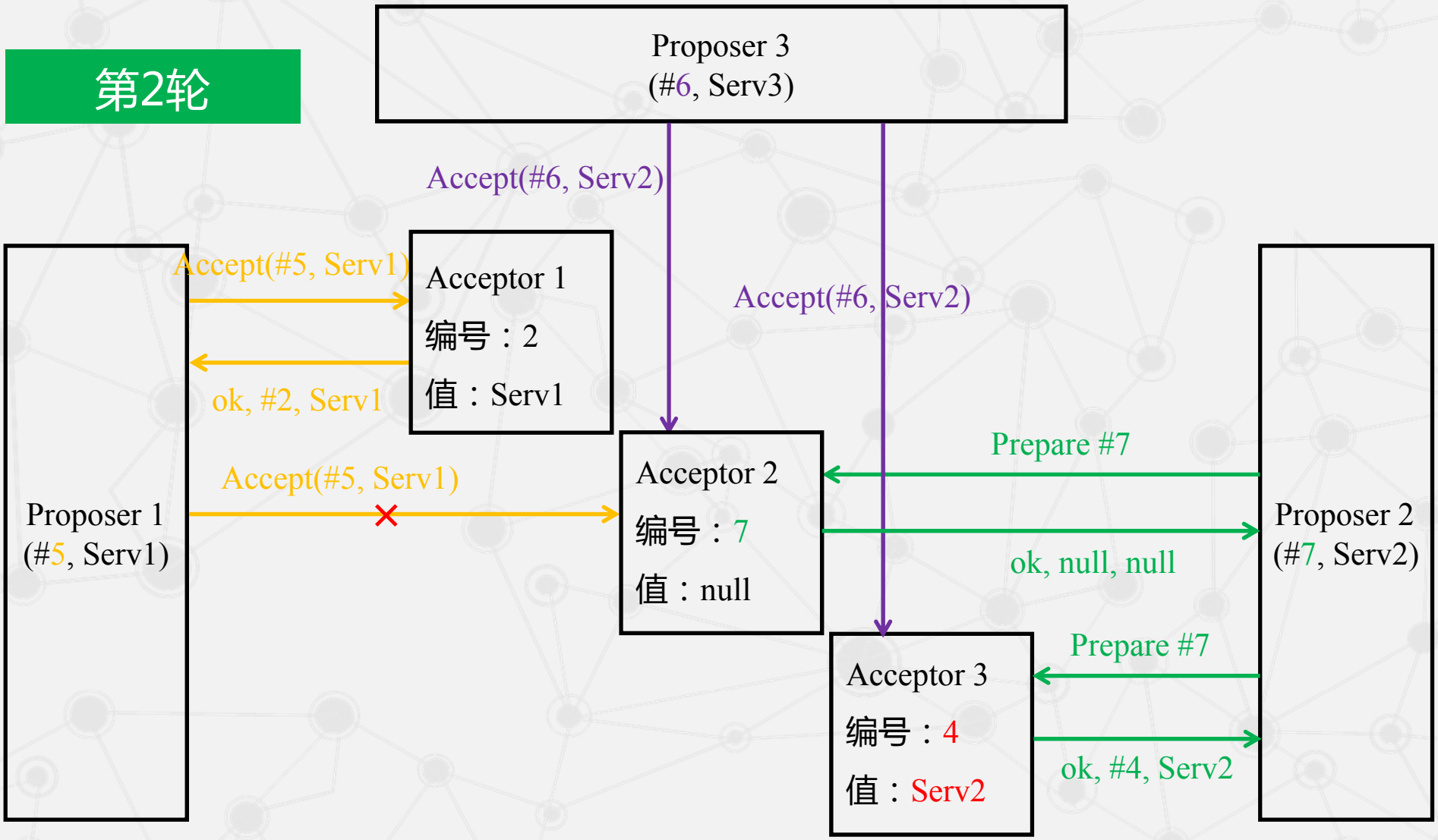


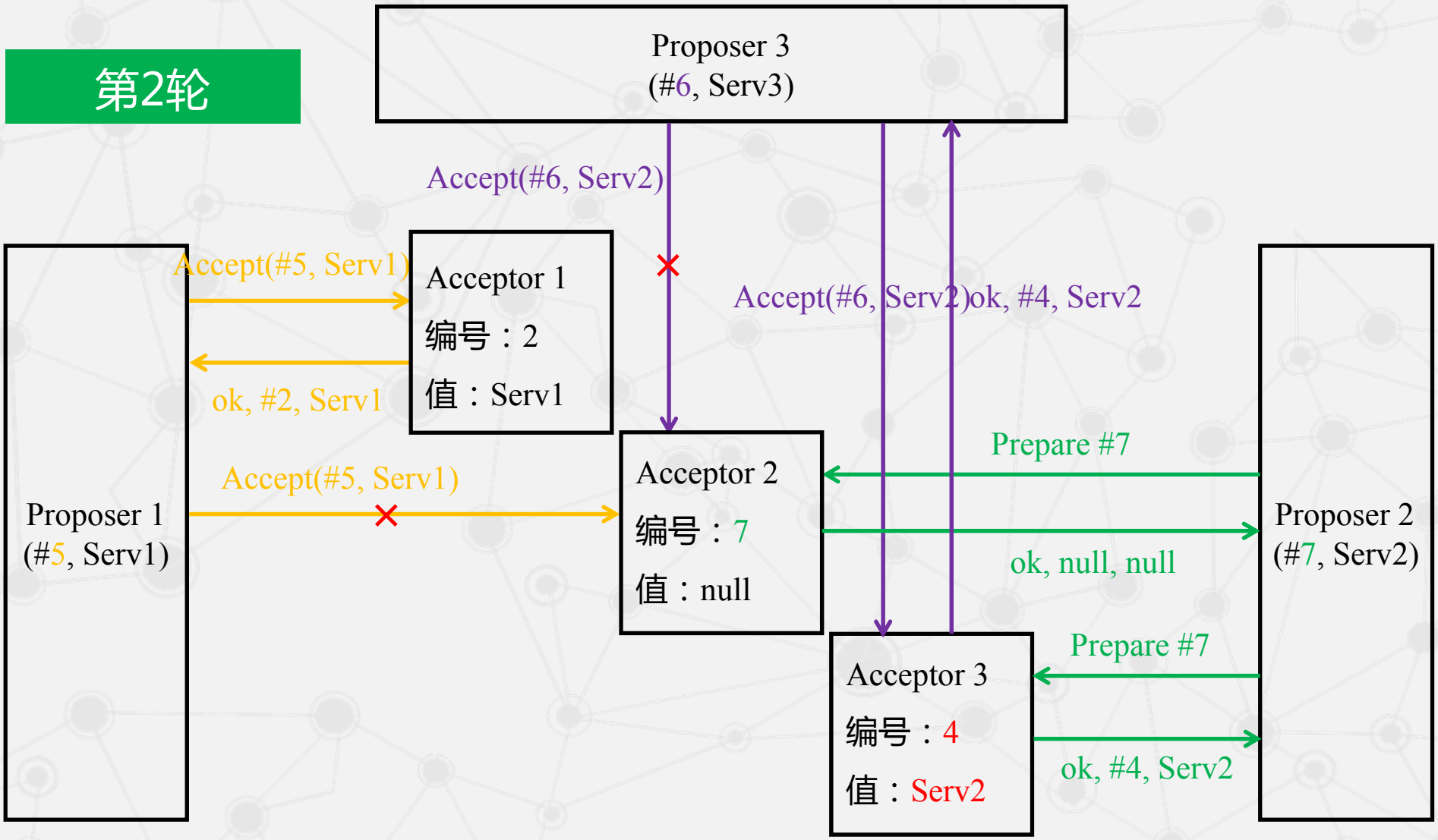
Proposer3处于S2a：Proposer 发送 Accept请求阶段。

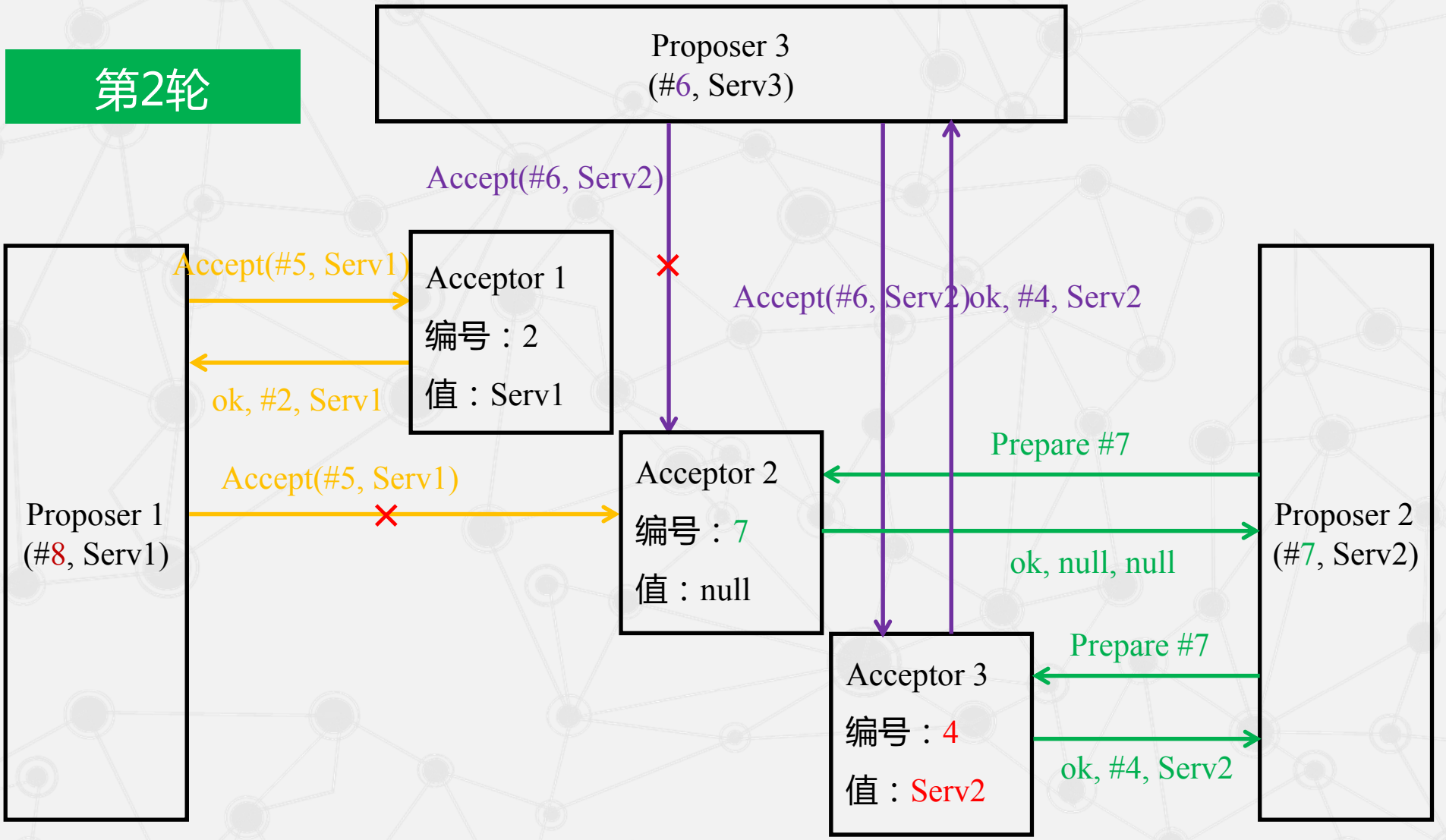
(1) 回复数量 > 一半的Acceptor数量，且所有的回复的value都为空，则Porposer发出accept请求，并带上自己指定的value。

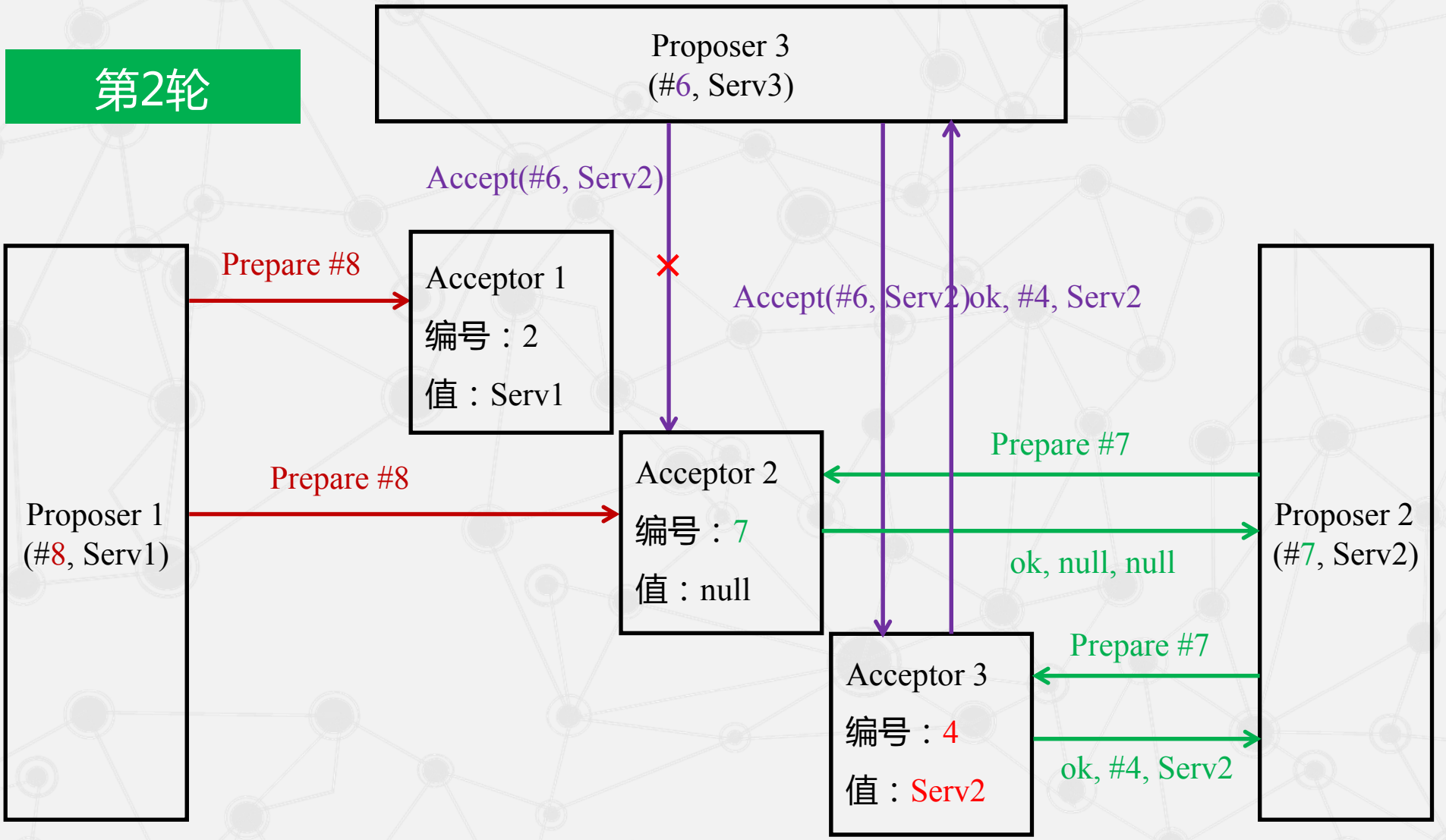
(2) 回复数量 > 一半的Acceptor数量，且有的回复value不为空，则Porposer发出accept请求，并带上回复中编号最大的value（作为自己的提案内容）。

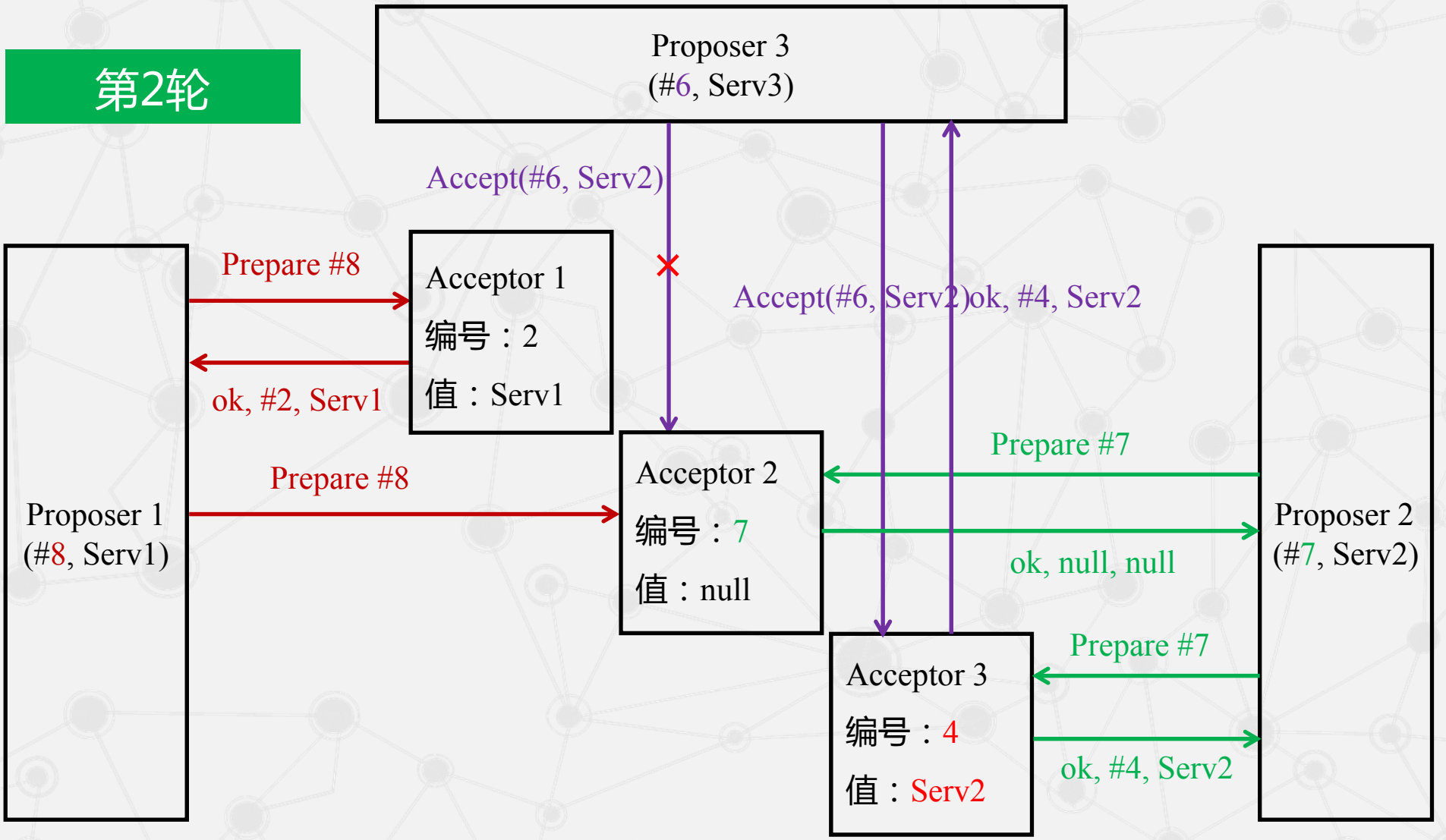
(3) 回复数量 <= 一半的Acceptor数量，则尝试更新生成更大的编号，再转回S1a执行。

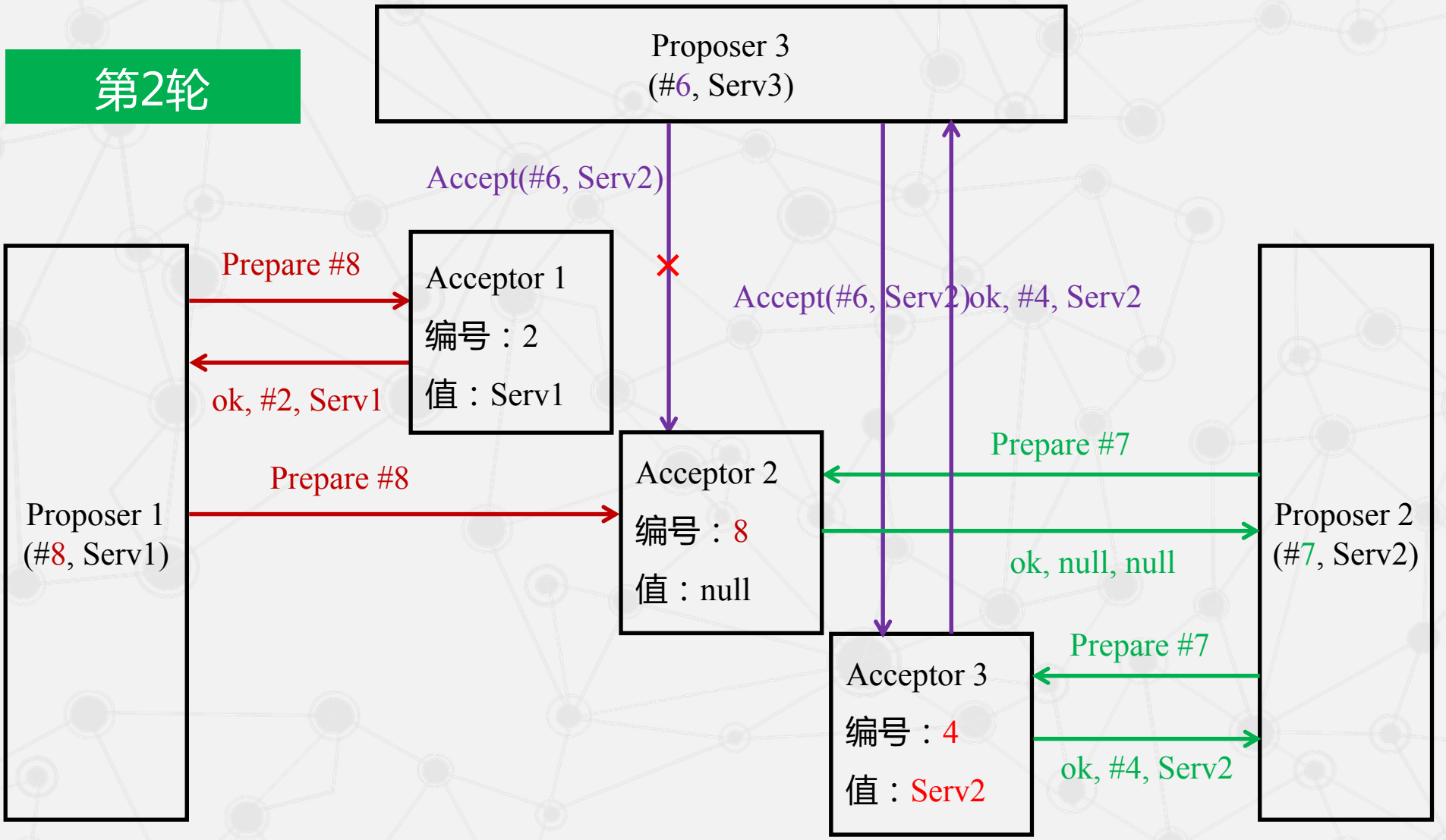

第2轮
Proposer 3
(#6, Serv3)
Accept(#6, Serv2)
Accept(#5, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#6, Serv2)
Proposer 1
(#5, Serv1)
Accept(#5, Serv1)
Acceptor 2
编号：7
值：null
Prepare #7
ok, null, null
Proposer 2
(#7, Serv2)
Acceptor 3
编号：4
值：Serv2
Prepare #7
ok, #4, Serv2

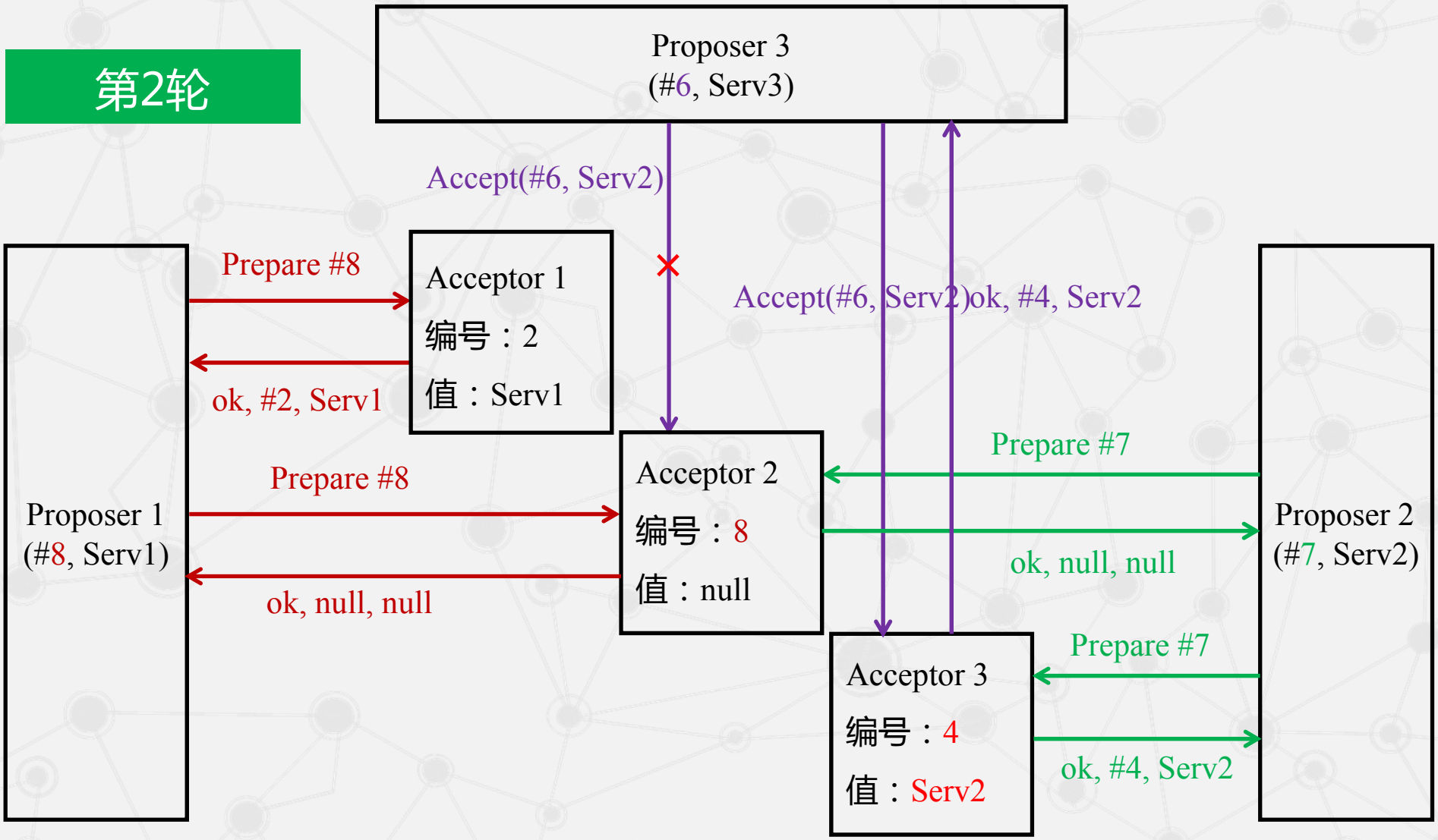

第2轮
Proposer 3
(#6, Serv3)
Accept(#6, Serv2)
Accept(#5, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#6, Serv2)ok, #4, Serv2
Proposer 1
(#5, Serv1)
Accept(#5, Serv1)
Acceptor 2
编号：7
值：null
Prepare #7
ok, null, null
Proposer 2
(#7, Serv2)
Acceptor 3
编号：4
值：Serv2
Prepare #7
ok, #4, Serv2

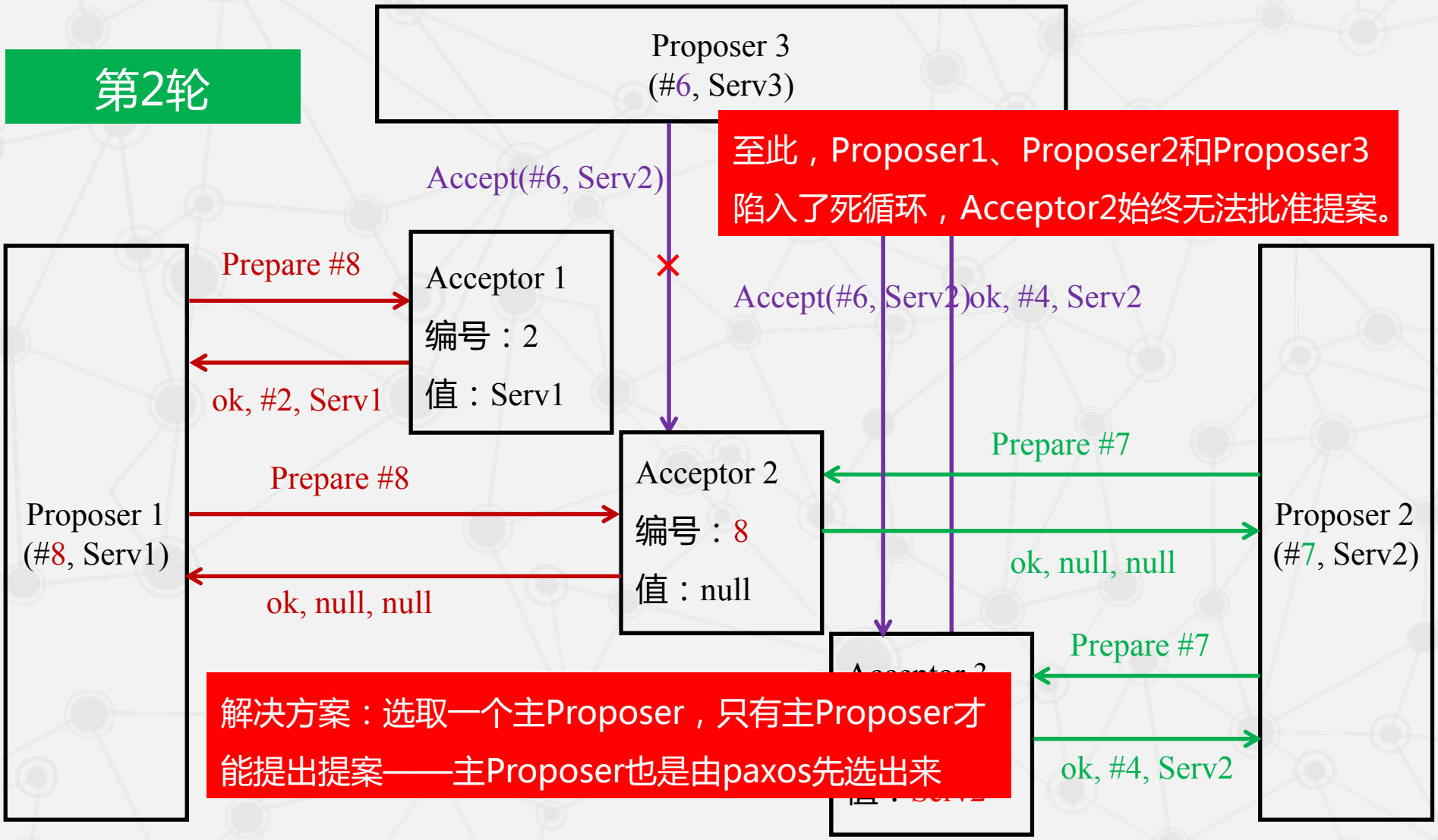

第2轮
Proposer 3
(#6, Serv3)
Accept(#6, Serv2)
Accept(#5, Serv1)
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#6, Serv2)ok, #4, Serv2
Proposer 1
(#8, Serv1)
Accept(#5, Serv1)
Acceptor 2
编号：7
值：null
Prepare #7
ok, null, null
Proposer 2
(#7, Serv2)
Acceptor 3
编号：4
值：Serv2
Prepare #7
ok, #4, Serv2


第2轮
Proposer 3
(#6, Serv3)
Accept(#6, Serv2)
Prepare #8
Acceptor 1
编号：2
值：Serv1
Accept(#6, Serv2)ok, #4, Serv2
Proposer 1
(#8, Serv1)
Prepare #8
Acceptor 2
编号：7
值：null
Prepare #7
ok, null, null
Proposer 2
(#7, Serv2)
Acceptor 3
编号：4
值：Serv2
Prepare #7
ok, #4, Serv2


第2轮
Proposer 3
(#6, Serv3)
Accept(#6, Serv2)
Prepare #8
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#6, Serv2)ok, #4, Serv2
Proposer 1
(#8, Serv1)
Prepare #8
Acceptor 2
编号：7
值：null
Prepare #7
ok, null, null
Proposer 2
(#7, Serv2)
Acceptor 3
编号：4
值：Serv2
Prepare #7
ok, #4, Serv2


第2轮
Proposer 3
(#6, Serv3)
Accept(#6, Serv2)
Prepare #8
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#6, Serv2)ok, #4, Serv2
Prepare #7
Proposer 1
(#8, Serv1)
Prepare #8
Acceptor 2
编号：8
值：null
ok, null, null
Proposer 2
(#7, Serv2)
Prepare #7
Acceptor 3
编号：4
值：Serv2
ok, #4, Serv2


第2轮
Proposer 3
(#6, Serv3)
Accept(#6, Serv2)
Prepare #8
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#6, Serv2)ok, #4, Serv2
Prepare #7
Proposer 1
(#8, Serv1)
Prepare #8
Acceptor 2
编号：8
值：null
ok, null, null
ok, null, null
Proposer 2
(#7, Serv2)
Prepare #7
Acceptor 3
编号：4
值：Serv2
ok, #4, Serv2


第2轮
Proposer 3
(#6, Serv3)
Accept(#6, Serv2)
至此，Proposer1、Proposer2和Proposer3
陷入了死循环，Acceptor2始终无法批准提案。
Prepare #8
Acceptor 1
编号：2
值：Serv1
ok, #2, Serv1
Accept(#6, Serv2)ok, #4, Serv2
Proposer 1
(#8, Serv1)
Prepare #8
Acceptor 2
编号：8
值：null
ok, null, null
Prepare #7
Proposer 2
(#7, Serv2)
ok, null, null
Prepare #7
ok, #4, Serv2
解决方案：选取一个主Proposer，只有主Proposer才
能提出提案——主Proposer也是由paxos先选出来

Learners学习（获取）被选定的value有如下三种方案：

方案一

Acceptor接收了一个提案，就将该提案发送给所有Learner

优点：Learner能快速获取被选定的value
缺点：通信次数较多

方案二

Acceptor接收了一个提案，就将该提案发送给主Learner，主Learner再通知其他Learner

优点：通信次数减少

缺点：主Learner可能出现故障

方案一

Acceptor接收了一个提案，就将该提案发送给一个Learner集合，Learner集合再通知其他Learner

优点：集合中Learner个数越多，可靠性越好
缺点：网络通信复杂度高


本章未完待续