

网络层数据指标

用来从不同的角度或者维度来衡量网络层模块的性能、稳定性、正确性等表现。

1. 收发包

网络层最基础最核心的功能就是收发包。

主要分为三种方式：

- 单播（点对点收发包）
- 跨网广播（从 A 网络广播到 B 网络）
- 全网广播（广播至该网络所有节点）

收发包最重要、最核心的指标如下：

1.1. TPS

收包的 TPS 值与多个因素有关，通常用来衡量网络层或者整个节点的性能

1.2. 丢包率

由于网络不可达、网络拥堵、节点程序内部负载较高等原因都有可能造成丢包率升高。通过该值能够衡量整个系统或者节点当前时刻的状态。

1.3. 包类型

收发包的包类型情况做统计，能够定位那个模块或者哪种消息类型是否过多或者网络是否存在风暴等问题

1.4. 包大小

可以统计收发包的平均包大小，用来衡量发包的内容是否设置合理。

上面几点是收发包指标中比较基本的几个，下面重点介绍下广播。

2. 广播

以下是广播模块的数据指标。

2.1. 广播播覆盖率

定义：通常指一次广播完成后能够有效覆盖的节点占总结点数的百分比

由于广播算法采用了洪泛的一些思想，以及真实网络可能存在波动或不可达的一些现象，故有可能造成某些节点无法收到此次广播包的情况，那么我们统计这个传播覆盖率基本能衡量此次广播的有效性。

2.2. 丢包率

定义：丢包率也即一次广播完成后没有被覆盖到的节点占总结点数的百分比

丢包率 = 100% - 广播覆盖率

2.3. 广播最大跳数

定义：完成一次广播后，所需要的最大跳数

区块链底层网络是基于分布式的 p2p 网络，广播本质上就是一条消息在这个网络中各个节点转发的现象，每经过一个节点，我们称为一跳。

广播最大跳数通常用来衡量广播算法的正确性，实际情况下不会太关注这个指标（实际情况会对最大跳数做限制，比如 Hop_Max = 20）

2.4. 广播平均跳数（广播跳数范围）

定义：完成一次广播后，对各个节点收到这个广播包的跳数情况做统计，计算跳数情况的分布图或者平均跳数

通常用来衡量一次广播的效率。

比如 [0,5) 范围有多少；[5,10) 范围有多少；[10,15) 范围有多少；[15, n) 范围有多少。理想情况应该是按照正态分布图的概率进行分布，即大多数可能集中在中间的范围，比如 [5,10) 范围。

2.5. 广播平均传播时间

定义：完成一次广播后，对各个节点收到这个广播包的传播时间情况做统计，计算传播时间分布图或者平均传播时间

该指标基本等同于上面一个指标：广播平均跳数

广播平均传播时间能够衡量广播的效率以及整个系统的负载情况。如果偏高，有可能是广播算法不够良好；也有可能是此时网络较差或者网络负载较高（网络拥堵、节点 CPU 过高）等。

3. 选举组网

以下是选举组网相关的数据指标。

3.1. 选举组网的正确性

定义： 网络层根据选举模块的选举数据进行组网，形成多层网络，需要正确的把同一个网络角色的节点根据 P2P 的方式进行组网

该指标是选举组网的最基本也是最重要的指标，P2P 组网是基于 Root 网络进行组网，模糊查找节点，这些过程不是百分之百确定的，比如网络不可达，防火墙限制等原因，最终会造成组出来的 P2P 网络与选举数据不符。

3.2. 选举组网的时间

定义： 一次选举完成之后，从网络层开始组网到最终正确的组网花费的时间

该指标通常用来衡量 P2P 组网的效率、网络的质量等。实际情况由于选举是提前半轮选举的，那么这个时间间隔是足够用来组网的了。

3.3. 选举组网销毁

定义： 选举节点过期后，是否正确的剔除节点或者退出网络

该指标多用于定位问题，跟踪丢包问题等。

4. 节点负载

节点负载包含多个指标，用来衡量节点程序的整体负载以及压力情况。

节点负载可以在不同的场景下进行观察，举个例子：

- 没有外部交易压力，观察负载情况
- 持续压测 1h、10h、1d、3d、1w，观察负载情况
- 停止压力后，观察负载恢复情况
- ...（根据实际场景思考和变换）

4.1. CPU

CPU 可以从多个角度来衡量，比如节点程序整体的 CPU，比如网络层模块的 CPU 等。

并且 CPU 情况可以扩展一下：

- 根据外部输入压力的不同（交易的频率、类型等），做不同场景的统计
- 统计各个分片网络的整体平均 CPU（比如高级节点群、共识节点群）

4.2. Mem

内存消耗指标，基本场景同 CPU 指标。

可以在不同的场景或者模块中持续观察

4.3. Bandwidth

网络流量指标，基本场景同 CPU 指标。

可以在不同场景或者模块中持续观察