



链滴

理解 Eureka 的 P2P 通讯

作者: [88250](#)

原文链接: <https://ld246.com/article/1498726165535>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```
<p>Eureka 客户端会和在一个可用区 (Zone) 的服务端进行通讯，如果通讯失败或者服务端没有客户端在一个可用区，则客户端将进行失效转移：对其他可用区的服务端发起通讯。</p>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<p>服务端接收到客户端信息后，会进行一些列<a href="https://link.ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2FNetflix%2Feureka%2Fwiki%2FUnderstanding-eureka-client-server-communication" target="_blank" rel="nofollow ugc">操作</a>将信息同步给其他服务端节点。如果某步操作失败，信息将在下一次心跳时同步给其他服务端节点。</p>
<p>当一个 Eureka 服务端节点启动后，它将向其他服务端节点获取所有实例的注册信息。服务端获到实例列表后将根据信息创建续约 (renew) 相关数据并准备接收来自客户端的续约请求。如果在某时刻客户端续约失败 (15 分钟内低于 85%)，服务端将停止实例过期防止该实例注册信息丢失。</p>
<p>在 Netflix 内部，上述过程称作<strong>自我保护</strong>模式，是 Eureka 客户端和服务端通讯时发生断网的一个保护机制。在这个场景下，服务端将尝试保存住已有的注册信息。此时客户端获的实例列表中有可能会有的实例已经不能正常服务了，客户端需要自己保证 Eureka 服务端返回的实例不存在或不响应情况下是弹性的，最好的处理方式就是对该实例的调用设置较短超时并尝试其他服务。</p>
<p>当服务端没法从其他节点获取注册信息时，它将等待 5 分钟让客户端注册自己。服务端会尽可能提供完整信息给客户端，所以可能会导致调用流量集中在某些实例上。</p>
<p>Eureka 服务端-服务端之间的通讯也使用客户端-服务端通讯同样的机制，参数调整可参考 <a href="https://link.ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2FNetflix%2Feureka%2Fblob%2Fmaster%2Feureka-core%2Fsrc%2Fmain%2Fjava%2Fcom%2Fnetflix%2Feureka%2FEurekaServerConfig.java" target="_blank" rel="nofollow ugc">configurations</a>。</p>
<p><strong>服务端节点之间断网会发生什么？</strong></p>
<ul>
<li>节点之间心跳会失败，检查到后进入自我保护模式防止注册列表丢失</li>
<li>在孤立服务节点上可能会有注册事件，部分客户端可能获取到新的注册实例，部分客户端获取不</li>
<li>网络连通后将恢复到正常状态，新注册信息将被同步到所有服务节点</li>
</ul>
<p>最后需要强调的是，在断网期间，服务端会尽可能保持可用，但不同的客户端可能会看到不同的数据。</p>
<p>译自：<a href="https://link.ld246.com/forward?goto=https%3A%2F%2Fgithub.com%2FNetflix%2Feureka%2Fwiki%2FUnderstanding-Eureka-Peer-to-Peer-Communication" target="_blank" rel="nofollow ugc">Understanding Eureka Peer to Peer Communication</a></p>
```