我们考虑从主机 A 到主机 C 的连接，该连接经过路由器 R1 和 R2。A－C 连接与 D－B 连接共享路由器 R1，并与 B－D 连接共享路由器 R2。对极小的 λ_{in} 值，路由器缓存的溢出是很少见的（与拥塞情况 1、拥塞情况 2 中的一样），吞吐量大致接近供给载荷。对稍大的 λ_{in} 值，对应的吞吐量也更大，因为有更多的初始数据被发送到网络中并交付到目的地，溢出仍然很少。因此，对于较小的 λ_{in}，λ_{in} 的增大会导致 λ_{out} 的增大。

在考虑了流量很小的情况后，下面分析当 λ_{in}（因此 λ'_{in}）很大时的情况。考虑路由器 R2。不管 λ_{in} 的值是多大，到达路由器 R2 的 A－C 流量（在经过路由器 R1 转发后到达路由器 R2）的到达速率至多是 R，也就是从 R1 到 R2 的链路容量。如果 λ'_{in} 对于所有连接（包括 B－D 连接）来说是极大的值，那么在 R2 上，B－D 流量的到达速率可能会比 A－C 流量的到达速率大得多。因为 A－C 流量与 B－D 流量在路由器 R2 上必须为有限缓存空间而竞争，所以当来自 B－D 连接的供给载荷越来越大时，A－C 连接上成功通过 R2（即由于缓存溢出而未被丢失）的流量会越来越小。在极限情况下，当供给载荷趋近于无穷大时，R2 的空闲缓存会立即被 B－D 连接的分组占满，因而 A－C连接在 R2 上的吞吐量趋近于 0。这又一次说明在重载的极限情况下，A－C 端到端吞吐量将趋近于 0。这些考虑引发了供给载荷与吞吐量之间的权衡，如图 3-48 所示。

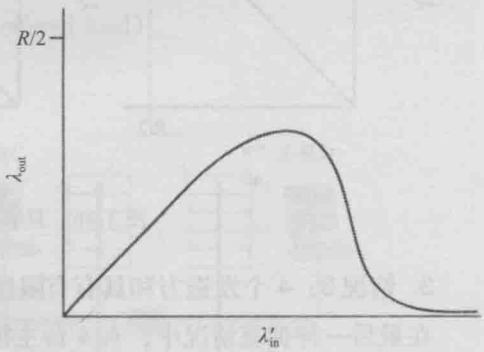

图 3-48 具有有限缓存和多跳路径时的情况 3 性能

当考虑由网络所做的浪费掉的工作量时，随着供给载荷的增加而使吞吐量最终减少的原因是明显的。在上面提到的大流量的情况中，每当有一个分组在第二跳路由器上被丢弃时，第一跳路由器所做的将分组转发到第二跳路由器的工作就是“劳而无功”的。如果第一跳路由器只是丢弃该分组并保持空闲，则网络中的情况是幸运的（更准确地说是糟糕的）。需要指出的是，第一跳路由器所使用的将分组转发到第二跳路由器的传输容量用来传送不同的分组可能更有效益。（例如，当选择一个分组发送时，路由器最好优先考虑那些已经历过一定数量的上游路由器的分组。）所以，我们在此又看到了由于拥塞而丢弃分组的另一种代价，即当一个分组沿一条路径被丢弃时，每个上游路由器用于转发该分组到丢弃该分组而使用的传输容量最终被浪费掉了。

3.6.2 拥塞控制方法

在 3.7 节中，我们将详细研究 TCP 用于拥塞控制的特定方法。这里，我们指出在实践中所采用的两种主要拥塞控制方法，讨论特定的网络体系结构和具体使用这些方法的拥塞控制协议。

在最为宽泛的级别上，我们可根据网络层是否为运输层拥塞控制提供了显式帮助，来区分拥塞控制方法。

- 端到端拥塞控制。在端到端拥塞控制方法中，网络层没有为运输层拥塞控制提供显式支持。即使网络中存在拥塞，端系统也必须通过对网络行为的观察（如分组丢失与时延）来推断之。我们将在 3.7 节中将看到，TCP 必须通过端到端的方法解决拥塞控制，因为 IP 层不会向端系统提供有关网络拥塞的反馈信息。TCP 报文