Congestion Control and QoS

Network Congestion

拥塞即是网络上传输的包的数量达到或超过了网络能处理的包的数量。一般来说一个路由器或交换机利用率达到 80% 以上性能就会急剧下降，拥塞控制就是保持包的数量不至于超过使性能急剧下降的界限。

交换机会产生过载导致拥塞，原因就是包到达的速率超过了其转发包的速率，造成拥塞的原因是多种多样的

- 突然爆发的流量/很差的网络拓扑
- 包到达速率超过了离开链路的容量
- 包处理的速率小于包到达的速率
- 内存不足以缓存到达的包

当发生拥塞时，交换机便会采取对应的措施，或是丢弃队列中的包为新来的包让出空间，或是阻止更多的包到达拥塞的端口（链路层流控制）。拥塞会在网络中的交换机间传播，当发生拥塞后很快会遍及整个网络

Network Utilization

考虑延迟与网络荷载的关系，当网络中包的容量还在无拥塞的范围内时，延迟随着包的数量增长以指数级增长，当包的容量到达拥塞的等级时延迟趋于无穷大

考虑吞吐量与网络荷载的关系，当网络中包容量还在无拥塞的范围时，吞吐量随着包的数量的增长一同增长，而当拥塞发生时吞吐量开始下降

定义 Communication Power

$$Power = \frac{Throughput}{Delay}$$

易得 Power 在未发生拥塞的范围内先增后减，存在一个最大值

在实际的网络中，网络的性能往往受制于瓶颈效应，即一条链路上的性能受性能最差的一段链路限制

Mechanisms for Congestion Control

Choke Packet

Choke Packet 称为抑制分组，在拥塞的节点生成，并发送至源节点，具体实现时使用 ICMP 协议告知源节点，一旦发生拥塞导致的丢包就发送 Choke Packet

接收到 Choke Packet 源节点就会意识到发生了拥塞，之后如何处理就取决于具体实现，如源节点可以选择暂停一段时间的发送

Backpressure

Backpressure 选择逐跳向前发送 Choke Packet，一般用于传播时延大于传输时延的情况，接收到 backpressure 的所有结点都会减慢其发送的速率

Warning Bit

Warning Bit 是在包头中一个特殊的位，由交换机设置，当设置 warning bit 的时候即表示发生了拥塞，由接收到该包的端系统采取行动以降低供给载荷

warning bit 的设置可分为 backwards (BECN) 和 forwards (FECN)，其中 BECN 传递的方向与产生拥塞的包的方向相反，一般是用于告知发送方减少发送数据的速率，而 FECN 传递方向与产生拥塞的包的方向相同，用于告知接收方减缓数据要求

Congestion Window

Congestion Window 是端结点的拥塞控制，以包的超时作为网络拥塞的信号。如 TCP 中使用动态的发送窗口控制包的发送（详见 TCP 的拥塞控制）

Random Early Discard

RED 算法运行于路由器或交换机，与端结点的拥塞窗口机制相结合，是为了解决 TCP 的 global synchronization 问题，考虑多个 TCP 同时开始连接，他们会经历慢启动，达到一个很高的发送速率直至产生拥塞丢包，而丢包后 TCP 会再次进入慢启动，此时网络中的总流量会很低，直到同时开始慢启动的 TCP 连接再次导致流量爆发产生拥塞，形成循环

RED 算法即是为了解决这个问题，其在交换机或路由器的缓冲区全满之前就随机丢掉一些包，使得 TCP 连接错开进入慢启动，保证网络总的利用率高的同时不产生拥塞

RED 算法使用指数加权移动平均计算平均队列长度

$$\mathrm{avgLen} = (1-\omega) \times \mathrm{avgLen} + \omega \times \mathrm{sampleLen}$$

设定一个 MaxThreshold 与一个 MinThreshold

- 若 avgLen < MinThreshold，则将包放入队列
- 若 MinThreshold <= avgLen <= MaxThreshold，则以 p 的概率将包丢弃，以 $1-p$ 的概率将包放入队列
- 若 MaxThreshold < avgLen，直接丢弃

Traffic Shaping

Traffic Shaping 的思想就是在流量进入网络前将其整形，控制其发出的速率。Traffic Shaping 有两种算法，Leaky Bucket 和 Token Bucket

Leaky Bucket

Leaky Bucket 的思想就是将波动的流量以固定的速率转发出去，考虑一个桶底部有一个孔，水会以恒定的速度离开桶，不论其进入桶的流量如何。当桶满时（队列满）便会丢包

Leaky Bucket 的输出速率是恒定不变的

Token Bucket

Token Bucket 的思想和 Leaky Bucket 相似，使用 Token 控制发送的速率，以一个恒定的速率向一个桶中放入 Token，每当发出一个包时，取出一个 Token。Token Bucket 提供了一个可变的速率，但是保证速率不会超过某个限度，比如在流量突发增多时可以通过快速消耗 Bucket 中的 Token 将其转发，但消耗完后转发的速率就受限与产生 Token 的速率，Token Bucket 在提供更灵活的速率时保证了总的速率不会超过某个限度

Internet QoS

随着互联网的飞速发展，各种各样的应用应运而生，比如大量数据交互的客户端服务器应用，有大量图形要素的网页，实时的通讯与视频等，此时网络中的尽力而为（best effort）已经无法满足需求，需要支持 QoS （服务质量）。

QoS 是一种控制机制，能够针对不同用户或者不同数据流采用相应不同的优先级，或者是根据应用程序的要求，保证数据流的性能达到一定的水准。QoS 的保证对于容量有限的网络来说十分重要，特别对于流媒体应用，这些应用对传输速率和延时都很敏感

不同的流对网络性能的需求都不同，常见的网络性能有

- reliability，即保证信息传递可靠
- delay，传播的延时
- jitter，网络的波动情况

- bandwidth，即传输时的吞吐量

根据不同应用对网络性能的需求不同，可将其分为弹性流和非弹性流。非弹性流的应用在面对网络拥塞时也不会减少或退让对性能的要求，有时候在资源不足时需要拒绝服务，为弹性流留出一些空间

在一般的 IP 网络中无法满足对于性能的需求，因为排队时延和拥塞状况往往是不可预测的，此时有两种解决方案

- 提前声明需求，使用 resource reservation functions，对应于 ISA (Integrated Services Architecture)
- 在传输时声明需求，使用 IP 报文头中的 QoS 字段，对应于 DS (Differentiated Services)

Integrated Services Architecture

ISA 中将可分辨的 IP 报文流组织为一个 flow ，根据以下原则组织

- 使用同样的 QoS 参数
- 基于同样的源/目的 IP 地址，端口号以及协议类型（TCP/UDP）
- 单向或组播

ISA 要求路由器实现对应的标准，即 ISA functions

ISA functions

路由算法：ISA 中的路由算法中链路的代价根据 QoS 的参数的不同而不同，不仅仅是简单的根据延时作为 cost，且基于具有类似 QoS 的流的类别进行路由/转发

排队原则：使用优先级队列，且使用多个队列，用以满足不同流的需求（QoS and best effort）

丢包策略：选择性地丢包而不是丢弃新来的包

保留协议：RSVP，对一个给定的 QoS 为新流量预留资源

准入控制：确定请求的 QoS 中是否有足够的资源给新的流

流量控制数据库：包含用于流量控制的参数

管理代理：修改流量控制数据库，引导准入控制模块设置策略

ISA services

ISA 中的服务可分为两个等级

- guaranteed service
  - 排队时延不会超过某个上界
  - 没有排队时的丢包
  - 一般用于租用的专线
- controlled load service
  - 没有保证排队时延的上界，但是在路由时会优先调度
  - 很高的送达率
  - 一般用于网络上的音视频应用
- best effort (default)

对于每个种类中的每个特定的流，根据其 QoS 的参数的值为其设定流量规范 （TSpec, Traffic specification），对于超过 TSpec 的流量仅提供 best effort 服务

resource allocation and RSVP

resource reservation

对于 IP 网络中的每个应用间的 session，ISA 需要提供 QoS 的保证。路由器需要维护为每个 session 分配的资源的状态信息，故需要一个类似于虚电路一样的机制，并且根据资源情况接受或拒绝建立一个新 session 的请求

session 是单向的，指向接收者，包括一个有特定目的和传输层协议的数据流，一般可以用一个三元组标记一个特定的 session：<目的 IP，运输层协议，目的端口>

一个 RSVP 的请求包含一个流描述符 <R-Spec, T-Spec>，其中

- R-Spec：指定所需的 QoS，用于在数据包调度程序中设置参数
- T-Spec：定义 QoS 下的流量特性，用于设置数据包分类器

RSVP: resource reservation protocol

RSVP 用于在端到端路径上预留资源以支持 QoS

RSVP 有如下特性

- 可以经由非 RSVP 的路由器预留资源
- 支持 IPv4 与 IPv6，与路由协议独立
- 能处理动态路由和组播成员改变的情况

在 RSVP 运行过程中，接收方将 RSVP 请求向发送者方向发送，中继的路由器将其信息传入准入模块判断能否预留资源，并且为连接维护一个定时更新的软状态，然后继续将 RSVP 请求向上传播

RSVP 包含两个基础的信息

- path 信息由发送方发给接收方，包含了路径上经过的每个路由器，用于反向定向，在传递 path 信息时也可收集信息用于预测 QoS
- recv 信息由接收方发向发送方，跟随 path 信息中的路径，沿途创建并维护预留资源的信息

在路由器和端中需要处理资源预留状态和路径状态，并且定期刷新（path 信息和 recv 信息），且必须紧跟动态路由的变化，当路由改变时预留的资源也需要改变。故应用程序需要定期续订预留的资源，否则途中的状态信息会过期。

使用 teardown 信息可以立刻删除预留资源的状态和路径信息。

在组播中预留资源与普通端对端相似，只是路径变为了组播树。在发送 recv 信息的过程中会经历 merge

Disadvantage of ISA

开销过大：需要为每个流维护一个状态信息，在流量很大时不实用

复杂：结构十分复杂，且需要专门的支持 ISA 的路由器，在路径上若有一个不支持 ISA 的路由器，就会变回 best effort 服务

不够灵活：仅定义了少许几个服务等级

Differentiated Services

characteristic

DS 提供简单易实现且开销低的机制以保证 QoS，并且支持基于性能区分的网络服务范围。DS 不同于 ISA，在网络核心（路由）上的实现较为简单，而在边界路由上较复杂

DS 使用 IPv4 中的 ToS 字段或是 IPv6 中的 Traffic Class 字段，且定义了 Service Level Agreement (SLA)，SLA 是一个两方协议，通常是使用 DS 的用户和网络供应商签订，用户只需要选择适当的 DS，在对待流量时相同 DS 的流量不做区分

基于 DS 确定每跳时的行为 (PHB, Per-Hop-Behavior)，如排队和转发，边界路由器需要分类和整形不合规的流量

SLA

SLA 约定了各种服务的参数，如期待的吞吐量，延时，丢包可能性，网络波动等

SLA 也约定了流量配置，如使用令牌桶时的参数（产生令牌的速率，桶大小等），并且约定了如何处理超出范围的流量

DS domain

DS domain 由网络供应商提供，比如移动，电信和联通有各自的 DS domain，而在同一 DS domain 中对于 DS 的处理策略是相同的

DS 间的边界路由由网络服务提供者维护，因为顾客可能在另一个 domain 中（e.g.，作为中国移动的用户，你的流量可能要通过联通的 domain），在流量穿过不同的 DS domain 时，需要衡量不同 DS 所能提供的服务表现，并为来自其他 domain 的流量匹配最符合的 DS

DS architecture

对于边界路由器，需要为每个流处理流量，并且将流量标记为符合流量配置文件的和超出范围的

对于网络核心路由器，按照不同 DS 的类别处理流量，并且根据在边界路由器的标记来缓存调度流量，仅仅为符合配置的流量提供 QoS 保证

边界路由器需要对每个流进行流量分类和调节，包括

- 分类：将流依据 DS 分为不同的类
- 测量：测量流量是否符合配置文件
- 标记器：如果需要则根据测量结果标记流量
- 整形器：使用令牌桶整形流量
- 丢包：超出配置文件过多的包直接丢弃

核心路由器只需要根据 DS 类别和源/目的地址、上层协议等将流量分类，并在排队和丢包时按照处理原则与优先级处理即可（PHB）

PHB: Per-Hop Behavior

PHB 定义了 DS domain 中每一跳时对特定 DS 的处理优先级和原则，不同的 PHB 会导致转发性能的不同。一般分为两种

- Expedited Forwarding：加急转发，提供低丢包率，低延时，低抖动，保证带宽，一般是点对点的连接或是租用的专线，但是在现有网络环境下难以实现，一般定义最小离开速率，并且保证到达速率始终小于路由器的最小离开速率（这样不会在队列中堆积以产生延时和丢包）
- Assured Forwarding：确保转发，提供优于 best effort 的服务，一般实现时显式分配路由器，而不用预留资源。转发时的性能取决于当前 DS 所属类还有多少资源剩余以及当前 DS

所属类的载荷，若发生了拥塞，则还取决于丢包时该类的优先级。流量的分类工作在边界路由完成，故核心路由只需根类别转发即可，开销很小。核心路由使用 RED 算法进行流量控制