节点缓存策略

1. 基本思想

节点为每个内容赋予一个缓存优先级，节点尽可能缓存所有内容，当空间不足时，缓存缓存优先级大的内容。缓存优先级由基本优先级、内容大小两个个参数形成，基本优先级由流行度、偏好度两个参数形成。节点每隔一段时间就更新所有内容的基本优先级，在内容到达后，节点通过基本优先级和内容大小计算内容的缓存优先级。通过查看缓存空间的大小和比较内容缓存优先级的大小决定要执行的操作。

1. 参数
2. 流行度(popu\_rate)
3. 含义：表示从某一节点观察，某一内容在网络中的流行程度。
4. 表示方法：一段时间内节点收到的对某一内容的interest packet的数量与此节点收到的interst packet的总数的比。
5. 偏好度(pref\_rate)
6. 含义：表示节点对于某一类内容的偏爱程度
7. 表示方法：一段时间内节点发出的对某一类内容的interest packet的数量与此节点发出的interest packet的总数的比。
8. 基本优先级(bprior)
9. 表示方法：popu\_weight\*popu\_rate+pref\_weight\*pref\_rate
10. 缓存优先级(cache\_prior)
11. 含义：表示单位内容的优先级，以防止优先级较高的超大型内容占用过多的缓存空间。
12. 表示方法：bprior / 内容大小，另外为了防止size太小或太大的内容出现不正常的缓存优先级，设置内容大小的上下限，只有在此限度内的内容才这样计算。
13. 流程

内容的初始基本优先级为0，节点每隔一段时间更新一次内容的基本优先级。更新公式为：

bprior=bprior\_old\*weight\_old + bprior\_new\*weight\_new

当节点收到某一data packet时，获取其大小content\_size，计算此内容缓存优先级为content\_prior

若空闲缓存空间大于content\_size，则立刻缓存此内容。

否则按缓存优先级从小到大的顺序检查节点中的缓存内容，并设

replace\_content\_size=0

replace\_content\_prior=0

每检查一个内容，将其缓存优先级加到replace\_content\_prior，将其大小加到replace\_content\_size。

若

replace\_content\_size >= content\_size &&

replace\_content\_prior <= content\_prior

则将遍历的内容删除，将此内容缓存；

若

replace\_content\_size < content\_size &&

replace\_content\_prior > content\_prior

则停止遍历，不缓存此内容。

若遍历结束，则不缓存此内容。