关于分组交换网络和电路交换网络的接入能力分析

2018 级数据科学四班 陈宇铭 320180939611

1. 问题背景

1.1 题设情景

假设多个用户共享一条$1Mbps$链路，并且每个用户活跃周期是变化的。同时，设每个用户忙时的传输时要求$100kbs$的传输速率，但忙时仅以$p = 0.1$的概率产生数据。

1.2 问题概述

1.2.1 采用电路交换，能被同时支持的最大用户数$N_{c_{max}}$？
1.2.2 采用分组交换，设总用户数为N_p，求出在某一时刻超过k个用户同时使用网络的概率P，并给出三者的关系？

1.2.3 采用分组交换，是否存在一个用户承载数量的阈值$N_{p_{max}}$？

1.2.4 在该网络中，相较于电路交换，分组交换是否可以允许更多的用户使用该网络？
1.2.5 在该网络中，相较于电路交换，分组交换的效率是否更高？
1.2.6 分组交换和电路交换的区别？

2. 分析解答

2.1 电路交换

2.1.1 分析

在电路交换网络中，在端系统间通信会话期间，预留了端系统间沿路径通信所需要的资源。也就是说，在电路交换网络中，每个用户在传输数据前会建立一个端到端的电路，且在创建电路时，网络会在该链路上预留恒定的传输速率。同时，电路一旦被分配，在该用户未结束通信会话并释放电路之前，其他用户将无法使用该网络资源，即使某一时刻，线路上并没有数据传输。根据电路交换网络的特性及问题情景可知，题设网络需在所有时段内都必须为每个用户预留$100kbs$的传输速率，才能使得网络中每个用户的网络需求都能被满足。

2.1.2 题解

设题设电路交换网络同时支持的最多用户数为$N_{c_{max}}$，链路传输速率为Rb_{total}，每个用户所需要的传输速率为Rb_{user}

由上述分析得

$$N_{c_{max}} = \frac{Rb_{total}}{Rb_{user}} = \frac{1Mbps}{100kbs} = 10$$

因此，该电路交换链路始终仅能支持 10 个用户并发接入该网络。

2.2 分组交换

2.2.1 分析

在分组交换网络中，端系统彼此交换报文。与电路交换网络不同，用户在分组交换网络中进行网络会话通信的过程，不需要建立专属的链路并预留网络资源。不同的用户可以随时接入网络之中，在同一条链路上以分组为单位进行多路复用，大大提高网络利用效率。但是，如果当多个用户同时连入网络且超过网络载荷，可能会导致链路阻塞，进而导致时延，甚至引发丢包现象。

同时，根据问题情景，我们可知每个用户产生数据传输这个事件均相互独立且事件的结果只有“产生数据传输”和“未产生数据传输”两种状态。因此，我们可以将在某时刻内题设网络内每个用户接入网络状态类比为N_p(用户总人数)次独立重复的伯努利试验，其中k个用户恰好在该时刻内接入网络并产生数据传输的离散概率分布为二项分布。设p为用户连接网络并产生数据传输的概率，则k个用户恰好在该时刻内接入网络并产生数据传输的概率公式为

$$P(X=k)=C_{N_p}^{k}p^{k}(1-p)^{N_p-k}$$

2.1.2 题解

由上述分析得

$$P(X\geq k)=1\ -\sum_{i\,=\,0}^{k}C_{N_p}^{i}p^{i}(1-p)^{N_p-i}$$

因此，在某一时刻超过k个用户同时使用网络的概率为$1\ -\ \sum_{i\,=\,0}^{k}C_{N_p}^{i}p^{i}(1-p)^{N_p-i}$

2.3 分组交换阈值问题

2.3.1 分析

设题设分组交换网络在某时刻能承载的最多用户数为k_{max}。当某时刻同时接入题设分组交换网络的用户数超过k_{max}的概率值偏大时，则说明网络出现过载的概率也增大，易导致时延和网络的传输效率降低。

为了方便研究，我们将问题分为两种情况：第一种情况，假设当某时刻同时接入题设分组交换网络的用户数超过k_{max}的概率值小于 0.01 时，网络出现问题的概率小，几乎可以忽略不计，以此来寻找题设分组交换网络的用户承载数量阈值$N_{p_{max1}}$，在此情况中，网络几乎不发生问题，分组网络的性能与电路网络的性能基本无差异；第二种情况，假设当某时刻接入题设分组交换网络的用户数超过k_{max}的概率值小于 0.99 时，网络出现问题的概率大，但是还是存在满足用户网络需求的情况，为极端阈值$N_{p_{max2}}$.

2.3.2 题解

设链路传输速率为Rb_{total}，每个用户所需要的传输速率为Rb_{user}

则

$$k_{max} = \frac{Rb_{total}}{Rb_{user}} = \frac{1Mbps}{100kbs} = 10$$

由上述分析得

$$P(X \geq k_{max}) = 1 - \sum_{i=0}^{k_{max}} C_{N_{p_{max}}}^{i} p^{i}(1-p)^{N_{p_{max}}-i} \leq 0.01$$

编写相应的 Python 程序计算$N_{p_{max}}$

```python
import numpy as np
from scipy.stats import binom
min = 0.01
k = 10
p = 0.1
Ns = np.array(range(1, 201))
Ps = np.array([round(1 - binom.cdf(k=k, p=p, n=n), 6) for n in Ns])
P_critical = Ps[Ps >= min][0]
P_critical_x = np.where(Ps == P_critical)[0][0]
```

解之得

$$\begin{cases} N_{p_{max1}} = 50 \\ N_{p_{max2}} = 197 \end{cases}$$

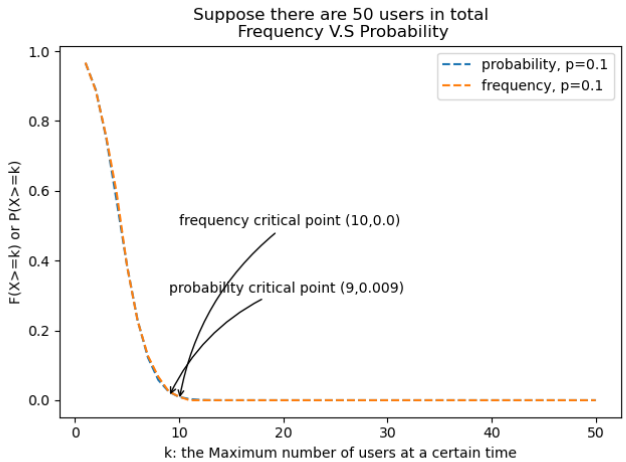
因此，分组交换网络的用户数在 50 以下时，出故障的概率仅有 0.01，基本不会出问题，远超电路交换网络。同时，在最极端差的情况下，分组交换网络也能接入 197 人。

3. 模拟实验

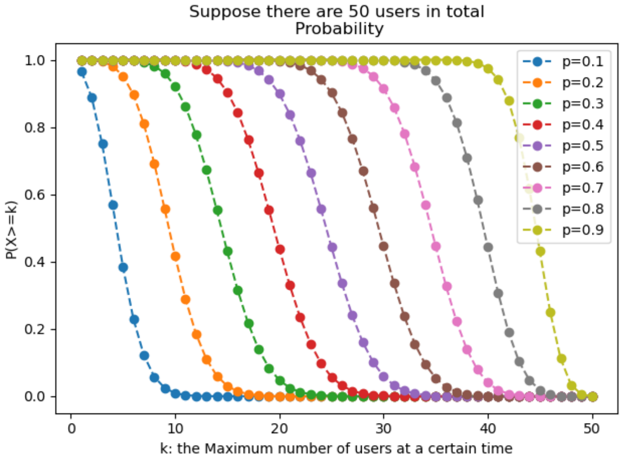
为了更好地研究分组交换网络的性能以及特性，我通过 python 程序进行模拟实验。设在分组网络中，承载用户总人数为N_p，在某时刻能承载的最多用户数为k，用户连接网络并产生数据传输的概率为p。

3.1 实验一、给定承载用户总人数为N_p

3.1.1 探究频率与概率的关系以及$P(X \geq k)$与某时刻能承载的最多用户数k的关系

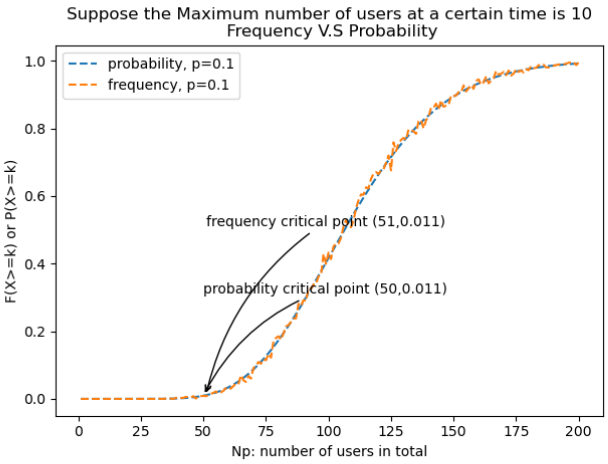


3.1.2 探究$P(X \geq k)$与用户连接网络并产生数据传输的概率p的关系

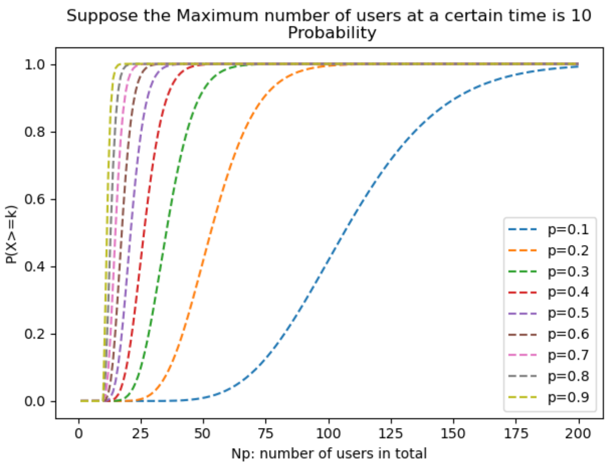


3.2 实验二、给定在某时刻能承载的最多用户数为 k

3.2.1 探究频率与概率的关系以及$P(X \geq k)$与承载用户总人数N_p的关系



3.2.2 探究$P(X \geq k)$与用户连接网络并产生数据传输的概率p的关系



3.3 实验结论

3.3.1 实验一

由实验数据可得，当用户连接网络并产生数据传输的概率p恒定时，承载用户总人数N_p固定为 50 时，随着某时刻能承载的最多用户数k的增加，$P(X \geq k)$减少，也就是说某时刻同时接入超过k个用户的概率减少；当改变用户连接网络并产生数据传输的概率p时，整体图像向右移动，即$P(X \geq k)$的值增大，也就是说某时刻同时接入超过k个用户的概率增大。当$k \geq k_{max}$时，则$P(X \geq k)$可以表示网络产生堵塞和故障等事件的概率，也就是说，当用户连接网络并产生数据传输的概率的值p增大时，该网络产生堵塞和故障的概率也随之增大。

3.3.2 实验二

由实验数据可得，当用户连接网络并产生数据传输的概率p恒定时，在某时刻能承载的最多用户数k固定为 10，随着承载用户总人数N_p的增加，$P(X \geq k)$也将随之增加，也就是说某时刻同时接入超过k个用户的概率将增加。当$k \geq k_{max}$时，则承载用户总人数N_p增大时，该网络产生堵塞和故障的概率也随之增大，故网络存在一定的阈值，超过该阈值网络的性能将下降；当改变用户连接网络并产生数据传输的概率p时，整体图像向左移动，也就是代表承载用户总人数N_p将随着p的增大而减少，故在相同的网络环境下（即某时刻能承载的最多用户数k相同），用户连接网络并产生数据传输的概率p越大，网络能承载的用户总量就越小。

3.3.3 总结

根据上述的分析，我们可以清楚地分析出承载用户总人数N_p，在某时刻能承载的最多用户数k，用户连接网络并产生数据传输的概率p这三个值之间的关系。同时，也能确认，当$N_p > k$时，分组交换网络明显比电路交换网络具有更高的接入能力，进而效率也更加地高。

4. 拓展分析

假设用户对网络的需求并不稳定时，某些用户需求的突增会对网络造成影响。在电路交换网络中，若某一时刻某用户对网络的需求突增，但由于其在一开始就分配了固定的资源，电路交换网络无法为该用户提供更多的网络需求，因此即使网络中无其他用户，该用户也无法使用其他的网络资源，只能按照之前的网络资源进行使用，会大幅度增大时间的成本。而在分组交换网络中，若某一时刻某用户对网络的需求突增，但网络中并无其他用户，该用户可以使用全部的网络资源，从而大幅度降低时间成本，达到效率最大化。由此可知，分组交换网络能灵活地适应用户及环境的变化，其效率是高于电路交换网络的。

5. 附录

5.1 实现代码

5.1.1 计算函数

使用 python 的 scipy 工具包实现对二项分布累计概率的求解

```
def count(n, k, p):
    return round(1-binom.cdf(k=k, p=p, n=n), 6)
```

5.1.2 模拟函数

使用 python 的 random 工具包进行独立重复的伯努利实验，以模拟分组交换的网络情况

```
def packet_switching_simulation(n, k, p, times):
    total_busy_users = 0
    for times in range(times):
        users_status = np.random.binomial(1, p, n)
        busy_users = users_status[users_status == 1]
        if len(busy_users) >= k+1:
            total_busy_users += 1
    return total_busy_users / times
```

5.2.1 绘图程序

使用 python 的 matplotlib 工具包对实验结果进行绘制

```
min = 0.01
n = 50
p = 0.1
Ks = np.array(range(1,n+1))
times = 1000
label_f = 'frequency, p={}'.format(p)
label_p = 'probability, p={}'.format(p)

Fs = np.array([packet_switching_simulation(n,k,p,times) for k in Ks])
Ps = np.array([count(n, k, p) for k in Ks])

P_critical = Ps[Ps <= min][0]
P_critical_x = np.where(Ps == P_critical)[0][0]

F_critical = Fs[Fs <= min][0]
F_critical_x = np.where(Fs == F_critical)[0][0]

plt.title('Suppose there are {} users in total\n Frequency V.S Probability'.format(n))
plt.plot(Ks, Ps, '--',label = label_p)
plt.plot(Ks, Fs, '--',label = label_f)

plt.annotate('frequency critical point ({},{})'.format(F_critical_x, round(F_critical, 3)),
             xy=(F_critical_x, F_critical), xytext=(F_critical_x, F_critical + 0.5),
             arrowprops=dict(arrowstyle='->',
                             connectionstyle='arc3,rad=.2'))
plt.annotate('probability critical point ({},{})'.format(P_critical_x, round(P_critical, 3)),
             xy=(P_critical_x, P_critical), xytext=(P_critical_x, P_critical+0.3),
             arrowprops=dict(arrowstyle='->',
                             connectionstyle='arc3,rad=.2'))
plt.legend()
plt.xlabel('k: the Maximum number of users at a certain time')
```

5.2 实验数据（部分）

Np	k	p	F(X>k)	P(X>k)
50	1	0.1	0.968969	0.966214
50	2	0.1	0.895896	0.888271
50	3	0.1	0.764765	0.749706
50	4	0.1	0.576577	0.568802
50	5	0.1	0.366366	0.383877
50	6	0.1	0.21021	0.229773
50	7	0.1	0.123123	0.122145
50	8	0.1	0.05005	0.057867
50	9	0.1	0.014014	0.024538
50	10	0.1	0.008008	0.009355
50	11	0.1	0.007007	0.00322
50	12	0.1	0.001001	0.001005
50	13	0.1	0	0.000285
50	14	0.1	0	7.40E-05
50	15	0.1	0	1.70E-05
50	16	0.1	0	4.00E-06
50	17	0.1	0	1.00E-06
50	18	0.1	0	0
50	19	0.1	0	0
50	20	0.1	0	0
50	21	0.1	0	0
50	22	0.1	0	0
50	23	0.1	0	0
50	24	0.1	0	0
50	25	0.1	0	0
50	26	0.1	0	0
50	27	0.1	0	0
50	28	0.1	0	0
50	29	0.1	0	0
50	30	0.1	0	0
50	31	0.1	0	0
50	32	0.1	0	0
50	33	0.1	0	0
50	34	0.1	0	0
50	35	0.1	0	0
50	36	0.1	0	0
50	37	0.1	0	0
50	38	0.1	0	0