

无线 Mesh 网中基于网络编码感知的 分组公平调度机制

黄 辰,戴 彬,王芙蓉,张 波

(华中科技大学电子与信息工程系,湖北武汉 430074)

摘 要: 利用竞争无线媒介的用户的传输代价不同,本文提出了一种基于网络编码感知的分组公平调度机制 FSNC. FSNC 机制的基本调度单位是编码集,编码集中的用户将依据自身在编码集中的权重,公平地获取统计上的长期和短期系统服务时间. FSNC 机制是通过用户的 ARQ 反馈消息,感知用户的解码能力从而做出调度决策的.理论和仿真实验证明,FSNC 机制在改善多用户分集系统的用户的吞吐率的同时,能够为多用户提供公平的服务时间.

关键词: 网络编码;无线 Mesh 网络;公平调度

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2010) 01-0025-07

Network Coding Awareness Based Fairly Scheduling in Wireless Mesh Network

HUANG Chen, DAI Bin, WANG Fu-rong, ZHANG Bo

(Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China)

Abstract: In this paper, we exploit the fact that users contending for the wireless medium will have different transmission cost, and present a fairly scheduling scheme FSNC. FSNC adopts NC-aware scheduling while maintaining statistical short and long term fairness. FSNC makes scheduling policy based on the ARQ feedback message from users. Theoretic analysis and simulation results prove that FSNC can provide service time fairness while improving the expected good for each user in multiuser diversity system.

Key words: network coding; wireless Mesh network; fairly scheduling

1 引言

Mesh 网络可以为大范围无线用户提供高质量无线宽带接入服务.由于用户在业务实时性和链路可靠性上存在区别,使得多用户系统可通过发掘多用户分集增益,取得较大的系统吞吐率.当信道波动变化较大时,系统调度资源给信道增益高的用户,所带来的多用户分集增益也高.目前 3G 无线蜂窝系统的高速下行分组数据传输的实现中,辅以一定调度策略的多用户分集技术已经得到应用^[1].

最大化系统吞吐率的调度方式没有考虑到公平性,如果某个用户信道长期质量不佳,最大化吞吐率的调度方式将使系统无法为这样的用户提供服务保障.文献[2]提出的“哭泣的婴儿”问题指出,当用户的信道质量特别差时,如果仍强行满足该用户的通信需求,将导致其它用户的性能下降.因此在多用户分集系统中如何实

现公平性保障和吞吐率提高之间的均衡是当前一大研究难点和热点问题^[3].

无线网络编码技术^[4]允许中间节点对来自不同链路的数据帧进行线性编码操作,使得网络可以达到理论上的最大流最小割流量极限.网络编码被广泛用于 MAC 层差错控制,以恢复丢失的数据帧^[5].传统网络编码没有考虑到用户解码能力和信道状态的多样性,没有为用户提供差异性服务,可能导致部分用户的吞吐率下降、时延上升^[6].文献[7]指出,只有通过合理调度,才能充分发挥网络编码在提高吞吐率上的优势.同时结合已有的公平调度算法也可以改善网络编码在处理用户差异性上的不足.综上,网络编码关注的是网络整体吞吐率的提高,多用户调度关注的是用户的相对公平服务,网络编码与公平分组调度具有结合的优势与可能性,如何在两者之间进行有效的折中是本文的研究重点.

本文提出一种基于网络编码感知的分组公平调度

机制 FSNC(Fairly Scheduling based on Network Coding), 利用多用户分集和网络编码的特性, 在提高系统整体吞吐率的同时, 改善单个用户的吞吐率. FSNC 机制根据用户反馈的解码信息和物理信道信息, 感知用户当前的解码能力和信道状态, 向用户公平分配系统服务时间. 理论和仿真证明, FSNC 机制在获得高的多用户分集增益的同时, 还保证编码集中所有用户的吞吐率得到提高.

2 网络模型

2.1 无线信道模型

多用户通过时分复用的方式接入无线信道, 每个时隙中只允许一个用户传输数据. 下行调度通过 AP 控制实现, 上行调度通过用户的 ARQ 反馈实现, 用以向 AP 反馈用户的解码状态. 假定上行和下行信道是独立的. 本文采用 HDR^[1]的直接信道状态通知机制(explicit channel notification), 使调度机可获取翔实的底层信道状态信息.

2.2 公平性度量模型

与某个权值成比例的公平性指数在有线网络中常被用来刻画在时间区间 (t_1, t_2) 内被持续积压的两个数据流 i 和 j 所获得的服务的差异. 一般有线网络的调度机都会保证比例公平性指数有一个常数值上界. FSNC 机制的调度机为适应无线异构网络环境, 不能采用定值上界, 而需要一个统计上的公平性指数, 反映无线信道的短期波动. 为区分时间公平性与吞吐率公平性, 定义 $\alpha_i(t_1, t_2)$ 为用户在时间区间 (t_1, t_2) 中分配到的服务时间, 并进一步定义统计上的时间接入公平性指数为

$$\Pr\left(\left|\frac{\alpha_i(t_1, t_2)}{\phi_i} - \frac{\alpha_j(t_1, t_2)}{\phi_j}\right| \geq x\right) \leq f(x)$$

3 FSNC 机制的调度算法

MAC 层网络编码的基本传输单位是编码帧. Mesh 网络的 AP 挑选用户成员构成多个备选网络编码集, 并基于调度目标函数调度为一个网络编码集服务, 向网络编码集成员发送编码帧. 设 g_p 为编码帧 p 所在的网络编码集. 每向编码集传输编码帧 p 之后, FSNC 机制会再次选定一个新的网络编码集服务, 因此 p 可作为所有动态网络参数的时间索引.

3.1 长期公平性控制

FSNC 机制借鉴了与银行借贷类似的思想, 在用户中分配称为 credit 的货币单位^[8], 通过 credit 的相互补偿控制长期公平性, 并改造这一思想使其支持网络编码. 设 $K_i(p)$ 为时刻 p 用户 u_i 拥有的 credit, 编码集 g 的 credit 为所有成员的 credit 之和, 即 $K_g(p) = \sum_{i \in g} K_i(p)$. 编码集获取系统服务时间的同时其 credit 会减少; 如果其 credit 不足, 产生了“亏欠”, 其他未

被调度的编码集会“借贷” credit 给它. credit 积累越多, 代表该用户获得调度的可能性越大.

基于 credit 的编码集调度流程如下: 当 AP 在 $(p-1)$ 时刻完成了一个编码帧的发送之后, 会根据图 3 的编码成员选择算法 MemSelect(), 从积压的用户集合 $B(p)$ 中选择若干用户作为下一个时隙 p 的服务对象. 如果传输代价高, 调度机可能推迟调度该编码集, 但随着时间推进, 调度机一再被延迟的用户最终会积累足够的 credit 以克服传输代价的影响, 获得被调度的机会. 图 1 是 credit 的更新算法:

$$\begin{aligned} g_{p+1} &= \text{MemberSelect}(B(p+1)); \\ K_{g_{p+1}}(p+1) &= \sum_{i \in g_{p+1}} K_i(p); \\ K_{g_{p+1}}(p+1) &= \max(0, K_{g_{p+1}}(p) - T_{g_{p+1}}(p)); \\ \text{for } (i=1; i \leq |g_{p+1}|; i++) \\ &\quad \text{if } (i \in B(p+1) \quad \&\& \quad i \in g_{p+1}) \\ &\quad \quad K_i(p+1) = K_i(p) - \max\left(\frac{T_{g_{p+1}}(p) - K_{g_{p+1}}(p)}{\phi_{g_{p+1}}}, 0\right) \phi_i \\ \text{for } (i=1; i \leq |B(p+1)|; i++) \\ &\quad \text{if } (i \in B(p+1) \quad \&\& \quad i \notin g_{p+1}) \\ &\quad \quad K_i(p+1) = K_i(p) + \max\left(\frac{T_{g_{p+1}}(p) - K_{g_{p+1}}(p)}{\phi_{g_{p+1}}}, 0\right) \phi_i \end{aligned}$$

图 1 编码集 credit 更新算法

算法设计思路如下: 上一个编码帧 p 的传输结束后, AP 将运行 Memselect() 算法构造新的编码集 g_{p+1} . 如果 g_{p+1} 的 credit $(K_{g_{p+1}}(p))$ 高于系统分配的服务时间 $(T_{g_{p+1}}(p))$, 则减去该服务时间. 否则其 credit 会被置 0. 产生的亏欠使其它被积压的用户的 credit 得到积累, 积累数量与各用户的权值成比例.

3.2 调度目标函数的设置

FSNC 机制的调度目标函数以提高编码集的吞吐率为目标, 包括关于吞吐率的编码集效用函数、编码集的 credit、传输代价函数, 以及编码集的系统服务时间需求. 函数定义如下:

$$\hat{g}(p) = \arg \max_g \frac{U_g(p) + K_g(p) - C_g(p) - T_g(p)}{\phi_g}$$

其中 U_g 是关于吞吐率的效用函数, C_g 是调度代价函数. $T_g(p)$ 代表当第 p 个编码帧传输完了之后, 编码集 g 传输下一个编码帧所需的系统服务时间需求. ϕ_g 代表编码集的业务优先级权值.

4 FSNC 机制的编码算法

编码算法包括三个子算法: 编码队列控制子算法、编码成员选择子算法和网络编码子算法.

4.1 编码队列控制子算法

若直接将网络编码用于分组调度, 信源将在缓存

原始数据帧队列上付出更高的代价.为此,FSNC 机制改变了以往在用户解码结束后才丢弃数据帧的队列管理思想,而是在用户感知到数据帧之后,即丢弃该数据帧的思想,以缩短信源缓存的数据帧队列长度.

定义 1(数据帧的可感知性) 数据帧 p 可感知代表用户拥有足够的信息,可以计算出一个包含 p 的多数据帧的线性组合.

定义 2(知识空间) 用户的解码状态可以用向量空间来表示,称为用户的知识空间.在有限域中,全局编码系数向量构成了用户的知识空间.

定义 3(自由度) 自由度指用户的知识空间维数.

定义 4(虚拟队列) 信源和用户在知识空间维数上的差异称为虚拟队列的大小.

在定义 1 的基础上,FSNC 机制规定:只要数据帧 p 对所用用户可感知,缓存队列就可删除该帧.可感知即丢弃的思想降低了缓存队列的长度,用户感知到新的数据帧就相对于增加其知识空间的自由度.这和在网络编码中通过控制有限域的大小,以高概率保证信宿每次收到的编码帧能增加信宿的系数解码矩阵的秩是一样的^[9].图 2 显示了一个时隙中与编码集相关的各个事件的相对时序.

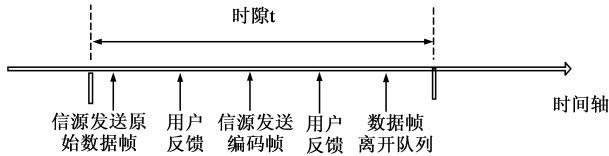


图2 编码集构造相关事件的时序

用户的知识空间用编码系数矩阵的简化行阶梯形式^[10]表示.设 V 是用户的知识空间, V 的基矩阵唯一地表示知识空间,可以通过高斯消元法得到.在知识空间的基矩阵中,每行的第一个非零元素称为主元(pivot),任何含有主元的列称为主元列(pivot column).主元列的数目等于非零行的数目,即 $\dim |V|$.

第 k 个到达信源的数据帧具有索引 k ,用 P_k 表示.基矩阵的第 k 列就代表数据帧 P_k .如果具有索引 k 的数据帧可感知,则用户知识空间的基矩阵的第 k 列是个主元列.否则主元在第 k 列之前的行向量的线性组合会在第 k 列的左侧生成非 0 元素,导致基矩阵的第一个非零元不可能在第 k 列,即数据帧 P_k 不可感知.数据帧的可感知与否同基矩阵的上述关系保证了信源可从反馈信息计算出用户知识空间的基矩阵,从而获知用户还有哪些数据帧没有感知到.进一步,用户可感知的数据帧的数目等于其知识空间基矩阵的维数.

4.2 编码集成员选择子算法

影响成员选择算法的因素包括链路可靠性、数据帧发送速率、数据帧大小,以及用户的解码能力.

(1) 用户的解码能力

p_g 代表发送给编码集 g 所有用户的编码帧, p_i 是发送给用户 $i(i \in g)$ 的原始数据帧,解码能力指用户 i 收到了编码帧 p_g 之后解码得到 p_i 的可能性.设信源的知识空间的自由度为 m ,用户 u_i 的知识空间的自由度为 y_i ,则用户 u_i 的解码能力 $A_i^g = \frac{y_i}{m}$.

(2) 编码集的吞吐率

吞吐率指单位时间内传输的数据量(bit).编码集 g 的吞吐率为其所有用户的吞吐率之和.经典的 COPE 算法已经证明,在单跳网络中启发式的网络编码算法是可以提高编码集成员的吞吐率之和的.

(3) 编码集成员选择算法

为了保证解码成功率和提高编码集吞吐率之间取得均衡,FSNC 机制制定了一个编码集成员的数量门限 G_T .编码集的成员选择与两个因素有关,一个是用户的丢包数量,一个是用户的 credit 值.每轮算法将所有用户分别按丢包数量和 credit 值从高到低排成两个列表,从两个列表的中段长度为 G_T 的区间中,搜索同时位于两个列表中的用户.如果搜索到的用户数量不足 G_T ,则区间长度会增加一倍,直至搜索到的成员数量达到 G_T 为止.图 3 给出了用户搜索算法 MemberSelect().AP 维护的 Lost_Frame 列表记录了传输过程由于丢失原始数据帧,需要传输编码帧的用户.

```

g = ∅;
i = 1;
loop
PacketLossList = Lost_Frame(PacketLoss);
CreditList = Lost_Frame(Credit);
Length(PacketLossList) = i * G_T;
Length(CreditList) = i * G_T;
for(j = 1; j ≤ |Length(PacketLossList)|; j++)
if(m ∈ PacketLossList && m ∈ CreditList)
ĝ = g ∪ m;
if(|ĝ| = G_T)
break;
if(|ĝ| < G_T)
i = i + 1;
return ĝ

```

图3 编码集成员选择算法

4.3 网络编码子算法

FSNC 机制采用的编码算法,是在 COPE 算法的基础上加入对用户知识空间的分析.信源根据 ARQ 消息携带的用户反馈信息,计算出用户的知识空间,并针对各个用户最早的未感知的原始数据帧进行网络编码.当某个数据帧对所有用户可感知,该帧就可以从信源缓存队列中删除.该编码算法对用户知识空间的基矩

阵的操作如下:

(1)建立用户的原始知识空间:空矩阵 $\mathbf{B}_1, \dots, \mathbf{B}_n$, 代表 n 个用户的 n 个虚拟队列;将 a 个新的数据帧加入到信源物理队列的末尾,并在矩阵 $\mathbf{B}_j(j=1, \dots, n)$ 的每一行都添加 a 个 0;

(2)数据帧传输:设 $\{u_1, u_2, \dots, u_m\}$ 为用户最早的未看见的数据帧的索引集合,按照升序排列($m \leq n$, 这是因为可能存在不同的用户具有相同的最早未感知到的数据帧,也可以按照降序排列,相应地基矩阵也会发生变化),如果信源物理队列为空,不作任何操作;否则计算 COPE 算法提出的启发式网络编码方法计算编码

帧 $p_g = \sum_{i=1}^m \alpha_i p_{ui}$, 其中 $\alpha_i \in F_q$;

(3)用户知识空间信息反馈:对于编码集中的每个用户 $j(j=1, \dots, n)$,如正确接收到了步骤(2)传输的编码帧,则将该编码帧的编码系数向量作为矩阵 \mathbf{B}_j 的一行加入,并执行高斯消元得到知识空间的 RREF 基;之后用户 j 将信息反馈给信源 AP;

(4)更新虚拟队列:收到用户反馈之后,信源首先更新知识空间矩阵 \mathbf{B}_j 的主元的索引集合以及被所有用户感知到的数据帧的集合,并将被所有用户感知到的数据帧的集合从新的知识空间矩阵中减去;

(5)更新物理队列:从信源传输队列中删除 S_Δ 中的数据帧;一个传输时隙结束,进入下一个时隙,返回到步骤(2).

4.4 编码集的信源队列分析

多用户分集系统中存在一个物理队列,多个虚拟队列,由 AP 统一管理.物理队列长度反映了信源用于缓存待编码的数据帧的开销.下面比较的是两种队列管理思想下的物理缓存队列的长度.

假定编码有限域的大小可以保证编码帧是新颖的.首先研究的是稳态下的虚拟队列,定义 $Q(t)$ 为时隙 t 结束时物理队列的长度; $Q_j(t)$ 为时隙 t 结束时第 j 个虚拟队列的长度.在描述 $Q_j(t)$ 状态变化的马尔科夫链中, λ 为新数据帧进入信源的概率, γ 是用户正确接收数据帧的概率. $\bar{\lambda} = 1 - \lambda$, $\bar{\gamma} = 1 - \gamma$.显而易见,该马尔科夫链是正递归的.

再定义 $S(t)$ 为时隙 t 结束时,AP 的数据帧集合; $V(t)$ 为时隙 t 时候 AP 的知识空间; $V_j(t)$ 为时隙 t 结束时用户 j 的知识空间; $S_j(t)$ 为到时刻 t 结束为止,用户 j 收到编码帧集.

引理 1 可感知即丢弃思想中,AP 的物理队列长度的上界为虚拟队列长度之和,即 AP 和用户之间积压的自由度之和.

证明 在数论的基本原理中,任何集合 S 的子集都满足

$$|S| - \left| \bigcap_{i=1}^k S_i \right| \leq \sum_{i=1}^k (|S| - |S_i|) \quad (1)$$

当有数据帧没有被所有用户感知,则物理队列的长度为 $Q(t) = |S| - \left| \bigcap_{i=1}^k S_i \right|$,再根据用户感知的数据帧的数目等于其知识空间的维数,有 $|S_j(t)| = \dim[V_j(t)]$.于是式的右边变为 $\sum_{j=1}^n [\dim[V(t)] - \dim[V_j(t)]]$,即虚拟队列的长度之和. 得证.

由上述分析和引理 1 易得以下命题:

命题 1 采用了可感知即丢弃思想的信源物理缓存队列的长度要短于采用解码即丢弃思想的队列长度.相应地 FSNC 机制就能够有效降低了为分组调度服务的原始数据帧的缓存开销.

5 FSNC 机制的协议分析

5.1 吞吐率分析

编码集的构造结束之后,被调度的编码集将整体获得系统服务时间.编码集再根据内部各无线用户的业务权重,进一步向各个用户分配服务时间,提高各用户的吞吐率.FSNC 机制分配的用于向被调度的编码集传输编码帧 p_g 的系统服务时间为 T_g . $A(g, i) = \delta_i^g$ 代表了分配给编码集中的用户 u_i 的服务时间,服务时间分配应该满足 $\sum_{i \in g} \delta_i^g = T_g$.

定义 5(有效编码集) 有效编码集为采用了网络编码之后,编码集中的用户的吞吐率之和,不低于没有采用网络编码的任何用户的吞吐率.

FSNC 机制的用户服务时间分配方式为:编码集向系统申请到的总服务时间,按照单个用户的业务权值占有所有用户的业务权值之和的比例,向各个用户分配系统服务时间.为了反映用户获得的系统服务时间与其吞吐率之间的关系,给出了如下定义:

定义 6 给定编码集 g ,定义 $\frac{L_g \cdot \gamma_i \cdot A_i^g}{\delta_i^g}$ 为编码集 g 中的用户 u_j ,在分配到了服务时间 δ_i^g 后获得的吞吐率.

在定义 6 的基础上,将服务时间按照上述方式进行分配,在用户 u_i 的解码能力保持为 $A_i^g = \frac{\gamma_i}{m}$ 的时候,根据定义 5 关于有效编码集的定义可以看出,是可以保证编码集用户的吞吐率之和,都不会低于在没有采用 FSNC 机制时,这些用户的吞吐率之和.

5.2 公平性分析

FSNC 机制采用统计上的公平性指数,即没有设置固定门限值,从而作使信道质量差的用户可以在连续被积压几个调度时间区间后,积累相当多的 credit.命题 2 证明基于网络编码的调度具有统计上的公平性保证.

命题 2 对于网络编码集 i 和 j 在时间区间上 $(t_1,$

t_2)被持续积压,在任意两个时隙上,均有概率上的公平性保证(注意,FSNC 机制并不是提供严格的公平性保证,这是因为在公平性度量模型中采用的是统计上的时间接入公平性指数)。

证明 在公平性度量模型中已经指出,统计上的时间接入公平性指数可以表示为

$$\Pr\left(\left|\frac{\alpha_i(t_1, t_2)}{\phi_i} - \frac{\alpha_j(t_1, t_2)}{\phi_j}\right| \geq x\right) \leq f(x)$$

根据调度目标函数的定义,对于任意两个时隙,均有以下的不等式成立

$$\begin{aligned} \alpha_i(t_1, t_2) + K_i(p_i) &\leq T_i^{\max} + C_i(p_i) \\ \Rightarrow \alpha_i(t_1, t_2) &\leq T_i^{\max} + C_i(p_i) \\ \Rightarrow \frac{\alpha_i(t_1, t_2)}{\phi_i} &\leq \frac{T_i^{\max} + C_i(p_i)}{\phi_i} \end{aligned}$$

进一步有

$$\begin{aligned} \frac{\alpha_i(t_1, t_2)}{\phi_i} &\leq \frac{T_i^{\max} + C_i(p_i)}{\phi_i} \\ \Rightarrow \frac{\alpha_i(t_1, t_2)}{\phi_i} - \frac{T_i^{\max} + x}{\phi_i} &\leq \frac{C_i(p_i) - x}{\phi_i} \\ \Rightarrow \Pr\left(\frac{\alpha_i(t_1, t_2)}{\phi_i} \geq \frac{T_i^{\max} + x}{\phi_i}\right) &\leq \Pr\left(\frac{C_i(p_i)}{\phi_i} \geq \frac{x}{\phi_i}\right) \\ \Rightarrow \Pr\left(\left|\frac{\alpha_i(t_1, t_2)}{\phi_i} - \frac{\alpha_j(t_1, t_2)}{\phi_j}\right| \geq \sum \frac{T_i^{\max} + x}{\phi}\right) & \\ \leq \Pr\left(\frac{C_i(p_i)}{\phi_i} + \frac{C_j(p_j)}{\phi_j} \geq \sum \frac{x}{\phi}\right) & \end{aligned}$$

本命题得证。

5.3 调度目标函数分析

(1) 传输代价函数的设计

命题 2 证明了公平性约束是传输代价的函数,因此给定统计上的公平性需求以及信道状态分布时,就可以设计反映用户链路状态的传输代价函数.定义 M_i 为代表用户 u_i 的链路状态的随机变量.根据命题 2,首先给出代表公平性约束要求的函数 $b(x)$.

$$\Pr\left(\left|\frac{\alpha_i(t_1, t_2)}{\phi_i} - \frac{\alpha_j(t_1, t_2)}{\phi_j}\right| \geq \sum \frac{T_i^{\max} + x}{\phi}\right) \leq b(x) \quad (2)$$

不妨设 $b(x) = \frac{x^2 - x + 1}{2}$,当式(3)满足的时候,式(2)成立,对于多用户有

$$\Pr\left(\sum \frac{C}{\phi} \geq \sum \frac{x}{\phi}\right) \leq \frac{x^2 - x + 1}{2} \quad (3)$$

由于每个用户的传输代价函数满足独立同分布,因此式(3)进一步简化为

$$\Pr(C_i \leq x) \geq \sqrt{\frac{1+x-x^2}{2}} \quad (4)$$

假设公平性约束函数 $g(x)$ 要求随着 x 的增长而快速下降为 0,设 δ 是可以调节公平性与系统性能的参数.

假定 M_i 是一个满足均匀分布的随机变量,则相应传输代价函数 C_i 与具体应用有关,还和信道状态有关,由于传输代价越高,系统的吞吐率就会减少得越多,因此可以设为传输代价函数为 $C_i = -\delta \log(1 - M_i)$.通过定义不同的传输代价函数,系统可调节对于公平性的要求.

(2) 效用函数的设计

定义编码集 g 的效用函数 U_g 为该编码集的吞吐率 χ_g 的函数,对该效用函数的要求是, U_g 是关于 χ_g 的增函数且有界,定义如下

$$U_g = \delta T_{\max} \cdot (1 - e^{-\frac{\chi_g}{r_{\max}}}) \cdot \sum A_i^g$$

其中 T_{\max} 为未采用网络编码时,编码集中的用户的系统服务时间需求的最大值,而 r_{\max} 为每个用户的链路传输速率的最大值. U_g 的定义有两个特性:① U_g 的上限值为 $\delta T_{\max} \cdot \sum A_i^g$,其中 δ 和传输代价函数一样,可以调节公平性和系统性能之间的均衡;② 当编码集的成员选择使得 χ_g 取得最大值的时候, U_g 也可以同时取得最大值.

(3) 调度目标函数

完成了传输代价函数和效用函数的设计,按 3.2 节的设计,FSNC 机制的调度目标函数的表达式为

$$\hat{g}(p) = \arg \max_g \frac{\delta T_{\max} (1 - e^{-\frac{\chi_g(p)}{r_{\max}}}) \cdot \sum A_i^g + K_g(p) + \delta \log(1 - M_i) - T_g(p)}{\phi_g}$$

6 仿真及试验分析

本节将 FSNC 机制同经典的 COPE 协议^[11](基于网络编码的单播通信协议)以及 WFQ 机制^[12](基于数据流权重的分组公平调度机制)进行比较.仿真环境是具有单个无线 AP,多个用户的无线 Mesh 网络,用户数量、链路丢包率可调.

图 4 比较的是在用户具有相同的信道状态的情况下,三种传输机制的吞吐率增益与链路丢包率之间的关系.吞吐率增益是指相对于普通的 Mesh 网络传输协议 802.11s,在同样的信道状态下采用特定的传输辅助机制后,吞吐率的提升程度.对于 WFQ 机制,由于所有用户具有相同的传输速率和链路可靠性,WFQ 机制无法利用多用户分集获得吞吐率增益.对于 FSNC 机制,在丢包率达到 60%之前,FSNC 机制可以持续改善系统的吞吐率,最高可以获得高达 70%的吞吐率增益.但是当丢包率进一步上升,FSNC 机制对于吞吐率的改善程度开始下降.这是因为用户反馈的 ARQ 消息大量丢失,FSNC 机制无法获得节点当前的解码能力,也就无法做出合适的编码集成员选择,从而限制了其在吞吐率的提高上发挥的作用.对于 COPE 机制,依赖网络编码固

有的鲁棒性,随着链路丢包率的上升,能够持续取得一定的吞吐率增益.但由于其采用的随机线性网络编码方式并不针对节点的解码能力调整编码策略,从网络全局来看,COPE 机制对于吞吐率的提升能力是有限的(不超过 40%).从仿真结果可见,在链路可靠性较好的时候(丢包率低于 50%),FSNC 机制在吞吐率提升效率上的优势还是非常明显的.

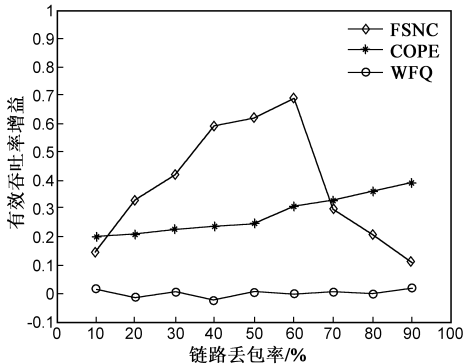


图4 吞吐率增益同链路丢包率的关系

FSNC 机制能够提升吞吐率的本质原因在于,每次发送完一次编码数据包之后,用户会将将自己的知识空间信息反馈给 AP. AP 调整编码数据包的组成,这样相对于 COPE 算法,是可能减少编码数据包的发送次数的.因为 COPE 是根据用户的丢包情况,一次性发送指定数量的、指定构成的编码数据包,在发送期间,即便信道状态发生了变化,不会改变原定发送的编码数据包的数量和构成.在这期间出现了新的丢包,也只能在下一轮编码数据包的发送中才会进行调整.当然 FSNC 机制为此需要付出的代价就是需要多占用用户到 AP 的上行信道,用于发送反馈信息.

在用户异构的情况下,用户间的链路可靠性和权重是存在差异的.链路丢包率差异 $\Delta\gamma_L = \gamma_{L_{max}} - \gamma_{L_{min}}$ 的变化范围从 0 到 0.7,图 5 比较了在此条件下三种机制的系统整体吞吐率与链路丢包率波动程度的关系. WFQ 机制和 FSNC 机制都可以依据用户的权重合理调度用户接入无线信道,因此受链路状态波动的影响较

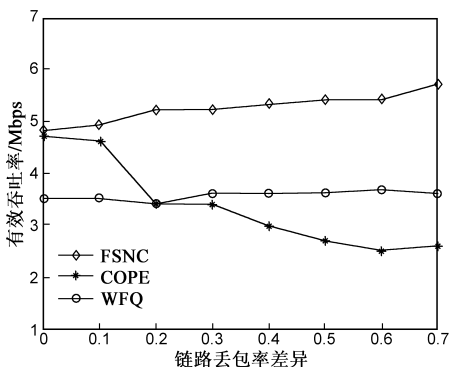


图5 系统整体吞吐量与链路丢包率波动程度的关系

小.但不同的是 FSNC 机制由于网络编码的引入,使得调度单位由 WFQ 机制的数据流转变为编码集,在编码集的构成过程中以最大化编码集的整体吞吐率为目标,相比只是单纯保证各个用户公平性的 WFQ 机制,FSNC 机制在吞吐率上的提升程度最高可达 30%.而 COPE 机制虽然在链路波动小的时候可以取得较高吞吐率,但是随着链路波动的进一步剧烈,由于没有采用任何合理的调度机制,个别信道质量差的用户由于数据帧的丢失持续请求系统服务,从而占据大量的信道资源,使得整体吞吐率大幅度下降,在链路丢包率差异在 20% 的时候,COPE 机制的整体吞吐率甚至低于 WFQ 机制,从而失去了网络编码在提升吞吐率上的优势.

图 6 比较的是 FSNC 机制和 COPE 机制的用户吞吐率与系统分配给用户的服务时间的关系,将用户按照其链路丢包率从低到高排序.由于 COPE 机制没有考虑到用户在解码能力和信道状态上的区别,所有编码帧对应的用户都得到了相同的系统服务时间.从图中可见,COPE 机制在系统的整体吞吐量上虽然有所提升,但是部分信道质量高的用户却因为信道质量差的用户过多占据了带宽反而出现了吞吐率的下降,低于没有采用网络编码的普通路由传输方式.而 FSNC 机制在编码过程中考虑到了信道状态和用户解码能力,采用统计上的时间分配算法,使得信道质量差的用户对于其它用户的影响是可控的,因此 FSNC 机制不仅仅提高了系统吞吐率,同时也改善了每个用户的吞吐率,这和 5.1 节关于吞吐率的分析结论是一致的.

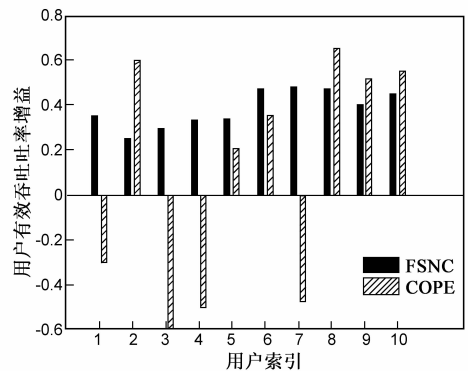


图6 用户吞吐率增益与系统服务时间分配的关系

7 结束语

本文提出了一种基于网络编码感知的分组公平调度机制 FSNC,将网络编码和公平调度相结合,在提高了编码集的整体吞吐率的同时,保证多用户可以公平地得到服务.仿真结果显示,与经典的网络编码单播通信协议 COPE,以及分组公平调度机制 WFQ 相比,FSNC 可以取得较好的吞吐率和公平性的均衡.

参考文献:

[1] P Bender, P Black, M Grob, et al. CDMA/HDR: a bandwidth

- efficient high speed wireless data service for nomadic users[J]. IEEE Communications Magazine, 2000, 38(7): 70 – 77.
- [2] D Koutsonikolas, Y C Hu, C C Wang. Pacifier: High-throughput, reliable multicast without “crying babies” in wireless mesh networks[A]. Proceedings of the 28th Conference on Computer Communications (INFOCOM09) [C]. Rio de Janeiro, Brazil, 2009.
- [3] W Rhee, J Cioffi. On the capacity of multiuser wireless channels with multiple antennas[J]. IEEE Transactions on Information Theory, 2003, 49(10): 2580 – 2595.
- [4] R Ahlswede, N Cai, S Y R Li, R W Yeung. Network information flow[J]. IEEE Transactions on Information Theory, 2000, 46(4): 1204 – 1216.
- [5] D Nguyen, T Nguyen, B Bose. Wireless broadcasting using network coding [A]. Proceedings of NetCod [C]. IEEE Press, 2007.
- [6] C Chu, D Yang, M Chen. Using network coding for dependent data broadcasting in a mobile environment[A]. Proceedings of Global Telecommunications Conference (GLOBECOM' 07) [C]. IEEE Press, 2007. 5346 – 5350.
- [7] P Chaporkar, A Proutiere. Adaptive network coding and scheduling for maximizing throughput in wireless networks [A]. Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking [C]. ACM Press, 2007. 135 – 146.
- [8] B Bensaou, K Chan, D Tsang. Credit-based fair queueing (CBFQ): a simple and feasible scheduling algorithm for packet networks[A]. Proceedings of IEEE ATM Workshop [C]. IEEE Press, 1997. 589 – 594.
- [9] D Lun, M Medard, R Koetter, M Effros. On coding for reliable communication over packet networks[J]. Physical Communication, 2008, 1(1): 3 – 20.
- [10] M Artin, P Algebra, Prentice Hall. Englewood Cliffs, New Jersey [DB]. MR 1129886 (92g:00001). 1991.
- [11] S Katti, H Rahul, W Hu, D Katabi, M Medard, J Crowcroft. XORs in the air: Practical wireless network coding[J]. IEEE/ACM Transactions on Networking, 2008, 16(3): 497 – 510.
- [12] A Parekh, R Gallager, I Center, Y Heights. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case[J]. IEEE/ACM Transactions on Networking, 1994, 2(2): 137 – 150.

作者简介:



黄 辰 男, 1983 年出生于湖北武汉. 华中科技大学博士生, 主要研究方向为无线网络编码. E-mail: szo094@gmail.com

戴 彬 男, 1977 年出生于湖北武汉. 华中科技大学讲师, 主要研究方向为无线网络编码.