Scheduling in Wireless Networks With Packet-Level and Flow-Level Dynamics

Ramin Khalili

Reading Group: 01.02.2011
Outline

• network model
• packet-level scheduling
• flow-level scheduling
• multi-channel case
• conclusion
Network model

- wireless downlink network
- single base station with a single channel
- multi-user
- discrete-time
- goal: optimal allocation of available resource at BS (time) to users
Outline

• network model
• packet-level scheduling
• flow-level scheduling
• multi-channel case
• conclusion
Static user population

- Fix set of users \((N)\)
- a queue per each user
- input with average rates \((\lambda_1, \ldots, \lambda_N)\) in bits/sec
- **stability**: queue length process does not blow to infinity
Channel model

• time-varying channel:
  – $c(t)$: channel state at time $t$
  – $C$: set of channel states
  – $\pi_c$: average probability for state $c \in C$

• $R_{i,c(t)}$: rate of user $i$ if served at $t$ by BS
  – depend on $i$’s position and $c(t)$
Queue lengths

- $\Phi_i(t)$: queue length of user $i$ at time $t$

  $$\Phi_i(t+1) = \Phi_i(t) - D_i(t) + A_i(t)$$

- $A_i(t)$: number of arrived bits

- $D_i(t) = \min \{ \Phi_i(t), R_{i,c}(t) \}$ if $i$ is served, 0 o.w.
Stability analysis

• capacity region: set of all input rates $(\lambda_1, \ldots, \lambda_N)$ such that there exist $\varphi_{i,c} \geq 0$, 
$\sum_{i=1:N} \varphi_{i,c} = 1$ for all $c$, such that

$$\lambda_i < \sum_c \pi_c \varphi_{i,c} R_{i,c}$$
Stabilizing algorithm

• **MaxWeight scheduling:** at time $t$ schedule queue $i$ such that

$$i \in \arg \max_j R_{j,c}(t) \Phi_j(t)$$

• different versions of proof: Lyapunov drift, fluid limit technique
It is to note that ....

• MaxWeight scheduling:
  – needs to know current channel state $c(t)$
  – but no *a priori* information about $\pi_c$ and $(\lambda_1, \ldots, \lambda_N)$

• knowing $\pi_c$ and $(\lambda_1, \ldots, \lambda_N)$ cannot help to achieve better performance than MaxWeight
Outline

• network model
• packet-level scheduling
• flow-level scheduling
• multi-channel case
• conclusion
Dynamic user population

• number of users varies in time:
  – users arrive at different times
  – have finite numbers of bits to transmit
  – leave after their bits are transmitted
• flow refers to this finite size sessions
• stability condition: number of unserved users remain finite
A network with $K$ (finite and fix) distinct classes

- A user class defined by a pair of random variables $(R,F)$
- $R_{ki}(t)$: rate of $i$-th class-$k$ user if served at $t$
  - $R_{ki}(t), R_{ki}(t+1), \ldots$ are i.i.d copies of random variable $R_k$
  - $R_k^{\text{max}} = \sup\{x: \Pr(R_k = x) > 0\}$
- $F_{ki}$: traffic in bits generated by class-$k$ user $i$
  - $F_{k1}, F_{k2}, \ldots$ are i.i.d copies of random variable $F_k$ with mean $E\{F_k\}$
Stability analysis

• $\lambda_k$ (flows/sec): class-$k$ users arrival rate
  – arrivals are $i.i.d$ across time and classes

• capacity region: set of all user (flow) arrival rates $(\lambda_1, \ldots, \lambda_K)$ such that

$$\rho = \sum_{k=1}^{K} \lambda_k \mathbb{E}\left\{ \left[ F_k / R_k^{\text{max}} \right] \right\} \leq 1$$
Stabilizing algorithm

• workload-based scheduling
  – (i): serve class-k user $i$ at time $t$ if $R_{ki}(t) = R_k^{\text{max}}$ (ties broken arbitrary)
  – (ii): randomly schedule a user if no user satisfies (i)

• key idea: reduce workload by one at each time step
It is to note that ....

• **workload-based needs to know** $R_k^{\text{max}}$ and $\lambda_k$ for any $k$

• **workload-based scheduling with learning:**
  – a purely opportunistic algorithm
  – a user transmits if it sees its best channel state so far
  – BS needs only to know $c(t)$
Related work

• Borst 03 & 05, Borst-Bonald-Proutier 04 & 09
• Flow-level dynamics, but use time scale separation assumption
  – file sizes are large, so users see time average throughput region which is fixed or changes slowly
• Shneer 09, Srinkant 09 & 10 (presented in this talk) does not use such an assumption
Outline

• network model
• packet-level scheduling
• flow-level scheduling
• multi-channel case
• conclusion
Multiuser multichannel wireless downlink

- so far consider single channel scenario
- what if we have M (frequency) channels
- Ex: LTE

A network with 3 users and 2 channels
Packet-level throughput optimal algorithm

• MaxWeight scheduling is optimal
  – serve user $i$ over channel $j$ at time $t$ if
    
    $$i \in \arg \max_q R_{q,j,c(t)} \Phi_q(t)$$

    – $R_{q,j,c(t)}$ is rate of user $i$ if served at $t$ over channel $j$

• $\mu$-rule scheduler stabilizes network
  – MaxWeight is an specific case
Flow-level throughput optimal algorithm

- **channel-assignment**: determine number of times a user will transmit over a channel

- **workload-based scheduling**: schedule user $i$ over channel $j$ if it can send with its maximum rate
  
  - keeping in mind channel-assignment’s decision
Conclusion

• packet-level: static population of users
• flow-level: dynamic population of users (number of users could growth to infinity)
• which one is more realistic?
• packet-level and flow-level studies result different stability regions