Throughout-Optimal Scheduling

What is the right way to select the actions $\vec{p}(t)$ to be taken across the links in our wireless network model? The answer to this question depends on the objective of the system. We will here present a scheduling scheme that obtains *throughput optimality*. In doing so, we will present some of the key engineering ideas in network scheduling. We will also discuss implications of our scheduling scheme to a few real-world network architectures.

Scheduling Policy

In any scheduling scheme, we need to introduce the concept of **queuing**. In our context, a queue is a "waiting area" where received transmissions are stored before they are forwarded over a link.

Formally, let q_l denote the queue length of link l. We are considering a slotted system here, where time is indexed by t, and without loss of generality, we assume the length of a timeslot is 1:

- The channel state $K(t)$ is fixed within a timeslot.
- Our scheme uses a fixed action $\vec{p}(t)$ within a timeslot.
- The capacity of each link is also fixed within a timeslot:

$$\vec{r}(t) = g(\vec{p}(t), K(t)) \tag{1}$$

The queue length then evolves as

$$q_l(t+1) = \left[q_l(t) + \sum_s H_s^l A_s(t) - r_l(t)\right]^+, \tag{2}$$

where $A_s(t)$ is the random arrivals to link l at time t, with $\mathbb{E}[A_s(t)] = X_s$, and $[x]^+ = \max\{0, x\}$. $r_l(t)$ can be thought of as the "drain rate" of the queue. We let $\vec{q}(t) = (q_1(t), ..., q_L(t))$ denote the vector of queue lengths.

The key intuition is that we need to make sure the queues do not explode to infinity. To drain queues faster, we would like to choose $\vec{p}(t)$ such that $\vec{r}(t)$ is large. If the current rate of link l cannot support the offered load, q_l will increase, so we should increase r_l, possibly at the cost of other links.

Consider the following policy: Pick $\vec{p}(t)$ at each time t such that

$$\vec{p}(t) = \underset{\substack{\vec{p} \in \mathcal{H} \\ \vec{r} = g(\vec{p}, K(t))}}{\arg\max} \sum_l q_l(t) \cdot r_l. \tag{3}$$

This gives larger weight to r_l if $q_l(t)$ is large, i.e., to prioritize this link more at time t. We will show that this scheduling policy will be able to stabilize all queues for any offered load $\vec{X} \in \Lambda$.

It is important to note that this policy does not require knowledge of the flow rates $\vec{X}$ or the channel state distributions λ_k. It only needs the current channel state $K(t)$:

- The policy is queue-length based, which provides current state information about the system.
- The policy is adaptive based on the current network conditions. Further, this is an online solution, as decisions are made in each timeslot.

Before proving optimality, we can briefly consider why some other policies would definitely *not* work:

1. Since our goal is throughput maximization, suppose we choose the schedule that maximizes $\sum_l r_l$, or $\sum_l w_l r_l$ for fixed w_l. This would end up prioritizing the links with the highest achievable rates, with queues building up at other links.
2. On the other extreme, suppose we focus entirely on the queue length, and choose the schedule that maximizes $\sum_l q_l(t) \mathbb{1}_{\{p_l(t)>0\}}$. In other word, the objective function improves when we schedule links that have larger queues. But merely scheduling transmissions on a link says nothing about the rate achieved on that link.

Therefore, it is important to consider both the rates $\vec{r}(t)$ and the queues $\vec{q}(t)$ in making scheduling decisions.

Lyapunov Stability

Our main theoretical approach will draw from **Lyapunov stability**. Roughly speaking, a system is said to be Lyapunov stable if a solution starting near an equilibrium point will stay near the equilibrium forever.

Formally, we will seek a Lyapunov function $V(\vec{q})$ of the queue $\vec{q}$ with the following properties:

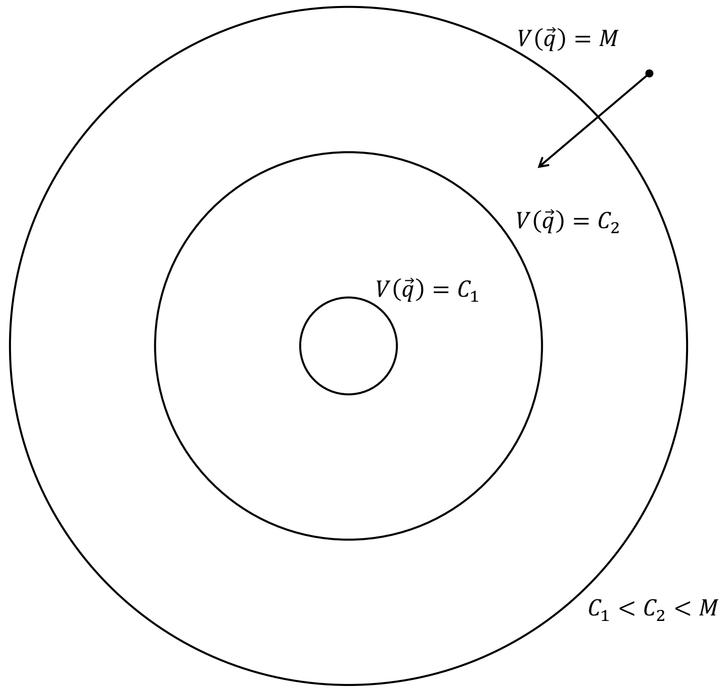
1. $V(\vec{q}) \geq 0$ for all $\vec{q}$, and $V(\vec{q}) \to +\infty$ as $||\vec{q}|| \to +\infty$.
2. Whenever $||\vec{q}(t)|| \geq M$ for some $M > 0$,
$$\mathbb{E}\big[V(\vec{q}(t+1)) - V(\vec{q}(t)) \mid \vec{q}(t)\big] \leq -\epsilon||\vec{q}(t)|| \tag{4}$$
for some $\epsilon > 0$.

The first property is pretty straightforward to satisfy, and establishes that the Lyapunov function $V(\vec{q})$ will grow with the size of $\vec{q}$. The second property then states that whenever $V(\vec{q})$ grows large, it will experience a **negative drift** proportional to the the size of $\vec{q}$. This implies that $\vec{q}(t)$ cannot explode to infinity. Visually, if we consider contour plots of $V(\vec{q})$, once we arrive at a point $V(\vec{q}) > M$, the system will experience a negative drift towards back inside the region:

Note also that the negative drift condition can be relaxed to
$$\mathbb{E}\big[V(\vec{q}(t+1)) - V(\vec{q}(t)) \mid \vec{q}(t)\big] \leq -\epsilon||\vec{q}(t)|| + M, \tag{5}$$
where M and ϵ can be taken as any positive constants. Their exact values are not important for our purpose here, as we only need to ensure that the drift becomes negative once $||\vec{q}||$ becomes large, and that it is proportional to $||\vec{q}||$. Thus, when $M \leq \frac{\epsilon}{2}||\vec{q}(t)||$, we can express the condition as
$$\mathbb{E}\big[V(\vec{q}(t+1)) - V(\vec{q}(t)) \mid \vec{q}(t)\big] \leq -\frac{\epsilon}{2}||\vec{q}(t)||. \tag{6}$$
When we derive the drift, we can ignore all terms that are bounded, and focus on those terms that grow with $\vec{q}$.

Proof of Throughput-Optimality

We will show that the function
$$V(\vec{q}) = \frac{1}{2}\sum_{l} q_l^2 \tag{7}$$



can serve as the Lyapunov function, and that the policy in (3) corresponds to maximizing the negative drift of this function.

Proof: Based on the queue evolution in (2), we have

$$\begin{aligned}
(q_l(t+1))^2 &\leq \Big(q_l(t) + \sum_s H_s^l A_s(t) - r_l(t)\Big)^2 \qquad (8)\\
&= (q_l(t))^2 + 2q_l(t)\Big[\sum_s H_s^l A_s(t) - r_l(t)\Big] \\
&\qquad + \Big[\sum_s H_s^l A_s(t) - r_l(t)\Big]^2 \qquad (9)
\end{aligned}$$

Since $A_s(t)$ is finite and $r_l(t)$ is bounded in practice, the rightmost term will always be finite, and there exists a constant M_l such that

$$(q_l(t+1))^2 \leq (q_l(t))^2 + 2q_l(t)\Big[\sum_s H_s^l A_s(t) - r_l(t)\Big] + M_l^2. \qquad (10)$$

Based on the definition of $V(\vec{q})$ in (7), then,

$$V(\vec{q}(t+1)) \leq V(\vec{q}(t)) + \sum_l q_l(t)\Big[\sum_s H_s^l A_s(t) - r_l(t)\Big] + \frac{1}{2}\sum_l M_l^2. \qquad (11)$$

Applying the expectation from the left side of (6),

$$\mathbb{E}\Big[V(\vec{q}(t+1)) - V(\vec{q}(t)) \mid \vec{q}(t)\Big] \;\; \leq \;\; \sum_l q_l(t)\Big[\sum_s H_s^l X_s\Big] - \underbrace{\sum_l q_l(t)\mathbb{E}\Big[r_l(t) \mid q_l(t)\Big]}_{(a)} + \frac{1}{2}\sum_l M_l^2. \qquad (12)$$

We see that term (a) matches the objective function in our policy (3). In other words, the max-weight policy is chosen to minimize the negative drift in the Lyapunov function $V(\vec{q})$.

Now, we must show that the resulting drift satisfies (6). Recall the expression for the capacity region from the last lecture:

$$\Lambda = \sum_k \lambda_k \cdot \mathsf{Conv\_hull}\{g(\vec{p}, k) \mid \vec{p} \in \mathcal{H}\}. \qquad (13)$$

Consider any $\vec{X}$ that lies strictly inside Λ. Since the region is convex, then a small perturbation $(1+\epsilon)\vec{X}$ for some $\epsilon > 0$ will lie in the region as well. Since this point is in Λ, there must exist a collection of actions $\vec{p}_k^m \in \mathcal{H}$ with fractions α_k^m and $\sum_m \alpha_k^m = 1$ such that:

$$(1+\epsilon)\Big[\sum_s H_s^l X_s\Big] \leq \sum_k \lambda_k\Big(\sum_m \alpha_k^m g(\vec{p}_k^m, k)\Big) \qquad \text{for all } l. \qquad (14)$$

Ultimately, we need to bound $(1+\epsilon)\Big[\sum_s H_s^l X_s\Big]$ in terms of the negative drift term (a) in (12). To do this, we will look for an upper bound on the right hand side of (14). With our policy (3) in place, at any time t,

$$\sum_l q_l(t) r_l(t) = \max_{\vec{r}=g(\vec{p},K(t))} \sum_l q_l(t) \cdot r_l. \qquad (15)$$

Thus, for any action m and any channel state k,

$$\sum_l q_l(t) g_l(\vec{p}_k^m, k)\mathbb{1}_{\{K(t)=k\}} \leq \sum_l q_l(t) r_l(t)\mathbb{1}_{\{K(t)=k\}}, \qquad (16)$$

where $g_l(\vec{p}_k^m, k)$ is the rate obtained on link l from $\vec{p}_k^m$, and $r_l(t)$ is the rate from the optimal action. Taking the expected value of both sides of this expression conditioned on $\vec{q}(t)$, we have

$$\sum_l q_l(t)\Big[\sum_k \lambda_k\big(\sum_m \alpha_k^m g_l(\vec{p}_k^m, k)\big)\Big] \;\; \leq \;\; \sum_l q_l(t)\mathbb{E}[r_l(t)\mathbb{1}_{\{K(t)=k\}} \mid \vec{q}(t)]$$

$$= \sum_l q_l(t)\mathbb{E}[r_l(t) \mid \vec{q}(t)], \tag{17}$$

where the indicator functions become absorbed by the expectation over channel states. Now, we can combine this with (14) by summing both sides of (14) over links to fit the right hand side of (14) with the left hand side of (17):

$$(1+\epsilon)\sum_l q_l(t)\Big[\sum_s H_s^l X_s\Big] \leq \sum_l q_l(t)\mathbb{E}[r_l(t) \mid \vec{q}(t)] \tag{18}$$

Finally, using (18) back in (12), we have

$$\begin{aligned}
\mathbb{E}\big[V(\vec{q}(t+1)) - V(\vec{q}(t)) \mid \vec{q}(t)\big] &\leq -\epsilon\sum_l q_l(t)\Big[\sum_s H_s^l X_s\Big] + \frac{1}{2}\sum_l M_l^2 \\
&\leq -\epsilon\Big[\min_l \sum_s H_s^l X_s\Big]\Big(\sum_l q_l(t)\Big) + \frac{1}{2}\sum_l M_l^2 \\
&\leq -\frac{\epsilon}{2}\Big[\min_l \sum_s H_s^l X_s\Big]\Big(\sum_l q_l(t)\Big), \qquad (19)
\end{aligned}$$

where this last step comes from assuming $\sum_l M_l^2 \leq \epsilon(\min_l \sum_s H_s^l X_s)\sum_l q_l(t)$, or in other words,

$$\sum_l q_l(t) \geq \frac{\sum_l M_l^2}{\epsilon\Big[\min_l \sum_s H_s^l X_s\Big]}. \tag{20}$$

Based on (19), the Lyapunov condition (6) holds as soon as the size of $\vec{q}(t)$ rises above a certain threshold. Thus, our policy in (3) stabilizes queues for any offered load in Λ, giving throughput optimality.

Implications