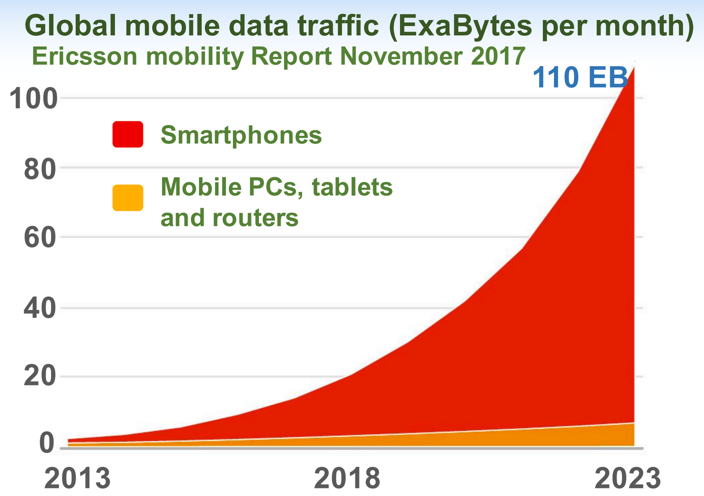


Switching Constrained Max-Weight Scheduling for Wireless Networks

Soumya Basu, and Sanjay Shakkottai

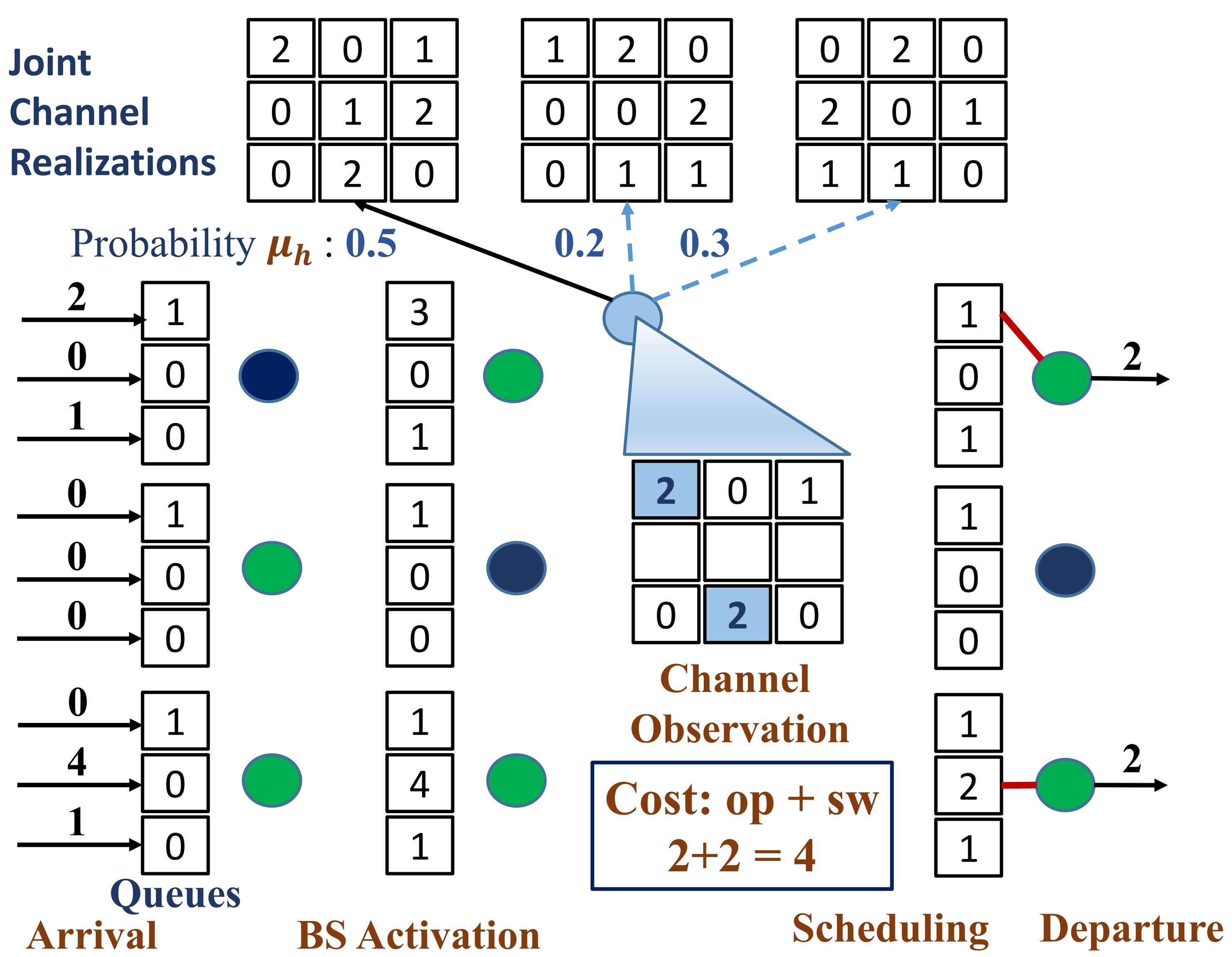
The University of Texas at Austin

The University of Texas at Austin
Electrical and Computer Engineering
Cockrell School of Engineering

Dense Cellular Networks

- **Dense deployment** of base stations (BS) to support peak data traffic
- **Dynamic** activation and de-activation of BS to **optimize energy** usage
- **Fast activation** dynamics is **necessary** to maintain **data rate**
- **Fast activation** dynamics leads to **large switching overhead**,

  e.g. **hand-offs**, **state exchange** among BSs, and BS **start-up** costs



System Model

- **Downlink** time-slotted system consists of **N BSs** and **M users**
- **Queue** $Q_{nm}(t)$ correspond to the queue for **(n,m)** BS-user pair
- An i.i.d. **joint arrival A(t)** (N×M) is realized: $A_{nm}(t)$ packets for Q_{nm}
- An i.i.d. **joint channel H(t)** (N×M) is realized: state h with prob. μ_h
- Two stages of decisions **Activation**, and **Scheduling** from active BSs
- **Step 1: Activate** a subset of **BSs**, **J(t)**, in timeslot **t**
- **Step 2: Observe channel** from **active BSs**: row **n** of **H(t)** iff BS **n** is ON
- **Step 3: Schedule** an 'active BS'-user matching **S(t)** from **S(J(t), H(t))**
- **Cost of operation + switching**: $C(t) = |J(t)| + |J(t-1)\Delta J(t)|$
- **Departure D(t):** $D_{nm}(t) = H_{nm}(t)$ if (BS **n** is ON & serves User **m**), o/w **0**

  Queue Update: $Q_{nm}(t+1) = \left(Q_{nm}(t) + A_{nm}(t) - D_{nm}(t)\right)^+$

BS Activation and Scheduling

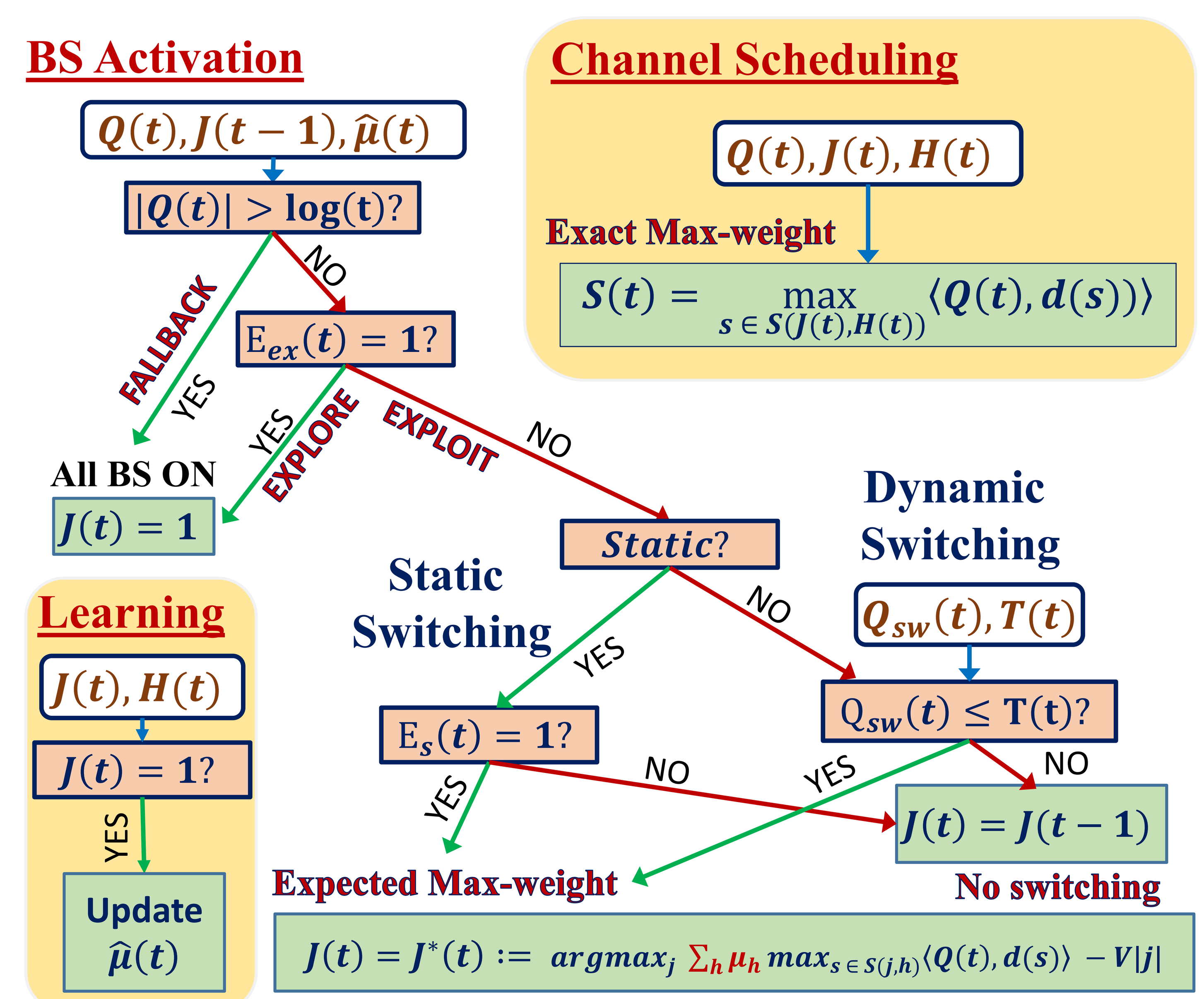
- Two key decisions: **BS activation** and **Channel Scheduling**
- **BS activation** is further split into two decisions
  - **When to switch?** Switch at a very slow rate(Constrained Switching)
  - **What to switch to?** Expected Max-weight with activation penalty
- **Channel state** is unknown before BS activation
  - **Exploration-exploitation** tradeoff in learning channel states
- **Channel Scheduling:** Exact max-weight with known channel state

Need for a New Approach

- **Greedy Optimization Techniques (Primal-Dual, Drift+Penalty)**
  - Frequent BS state change as switching cost is not optimized
- **Reinforcement learning with bounded queue length**
  - Prohibitive computation for large queue lengths
  - Complex packet drop vs optimality tradeoffs
- **Sticky BS selection using static-split rule + MW scheduling**
  - Large delay as BS selection is non-adaptive to queue lengths

Learning Aided Switching and Scheduling (LASS)

- **Parameters:** Switching rate, ϵ_s and Penalty scale, V
- **Independent R.V.s:** Switch: E_s(t)~Ber(ϵ_s), Explore: E_{ex}(t)~Ber(log(t)/t)



Dynamic Switching

- Uses two variables **Switch Counter: T(t)** and **Switch Queue :** $Q_{sw}(t)$
- **Switch counter** keeps count of the time since $J^*(t)$ is scheduled
- **Switch queue** counts the number of switching events that exceeds rate ϵ_s

  Switch queue: $Q_{sw}(t+1) = \left(Q_{sw}(t) - E_s(t) + 1\{Q_{sw}(t) \le T(t)\}\right)^+$

  Switch counter: $T(t+1) = 1\{J(t) = J^*(t)\}(1 + T(t))$

Performance Guarantees

- Assumptions on **system parameters**
  - **Capacity gap** $\epsilon_g > 0$, and **bounded** arrivals and departures
  - **Optimal cost** of the system C^*_{avg} with **no switching constraints**
- **Algorithm parameters**: Switching rate, ϵ_s and Penalty scale, V
- **Performance metrics** of interest:
  - Time average of queue lengths: Q_{avg} and costs: C_{avg}
  - Tail bounds for queue lengths: $\mathbb{P}(|Q(t)| \ge x)$
- For **LASS** with **Static** switching and LASS with **Dynamic** switching
  - $Q_{avg} \le O\left(\frac{C^*_{avg}}{\epsilon_g} + V + \frac{NM}{\epsilon_g \epsilon_s}\right)$ and $C_{avg} \le C^*_{avg} + O\left(\epsilon_s + \frac{NM}{V\epsilon_s}\right)$
- For **LASS Static**: $\mathbb{P}(|Q(t)| \ge x) \le \exp(-\Theta(\epsilon_s \epsilon_g)\, x) + O\left(\frac{\log(t)}{t}\right)$
- For **LASS Dynamic**: $\mathbb{P}(|Q(t)| \ge x) \le \exp(-\Theta(\epsilon_g)\, x) + O\left(\frac{\log(t)}{t}\right)$

Simulation Results

- Four algorithms are simulated for **8 Users and 3 BSs** until convergence:
  - **DP**: Greedy Drift + Penalty
  - **LSG**: LASS Static with geometric inter arrival between $E_s(t)$
  - **LSF:** LASS Static with fixed inter arrival between $E_s(t)$
  - **LD:** LASS with Dynamic switching
- **First plot:** Q_{avg} of **DP < LD < LSF < LSG** (V = 100, load = 0.9)
- **Second plot:** C_{avg} of **DP > LD ≈ LSF ≈ LSG** (V = 100, ϵ_s = 0.1)
- **Third and Fourth plot:** Separation of queue length tail distribution
  - **DP < LD < LSF << LSG** (V = 100, load = 0.9)
  - Differences are more pronounced for smaller ϵ_s