IEEE Communication Theory Workshop

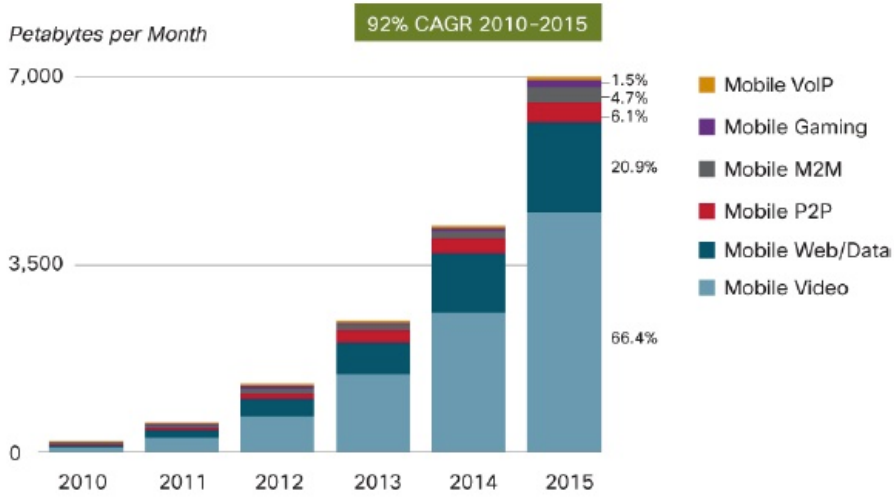
Caching at the Edge: Throughput Scaling Laws of Wireless Video Streaming

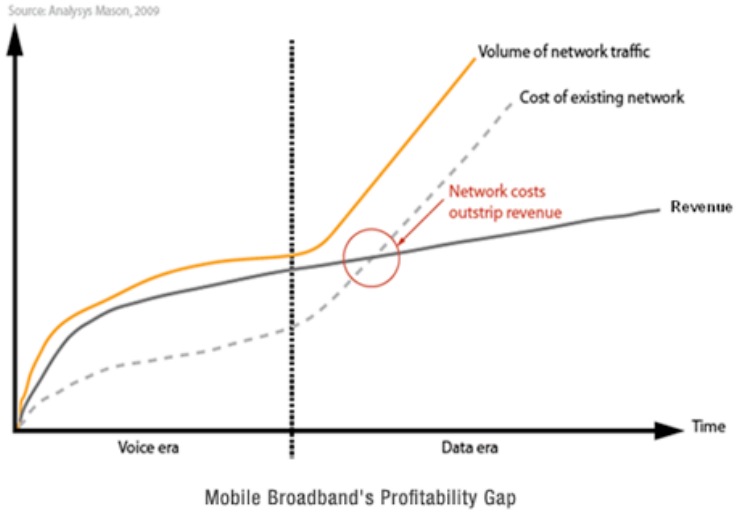
Giuseppe Caire
University of Southern California/Technical University of Berlin

(Joint work with: D. Benabothla, K. Shanmugam, N. Golarezai, M. J. Neely, A. Dimakis, A. F. Molisch, M. Ji, A. Tulino, J. Llorca)

Curacao, May 25-28, 2014

Wireless operators' nightmare

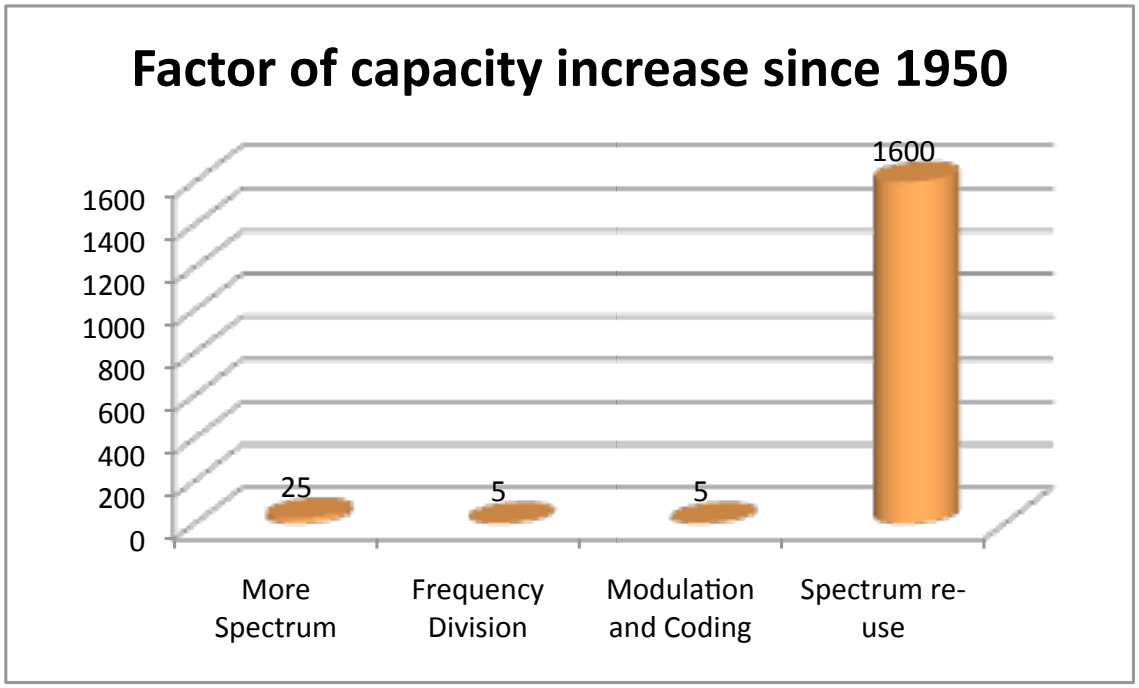




- 100x Data traffic increase, due to the introduction of powerful multimedia capable user devices.
- Operating costs not matched by revenues.

A Clear Case for Denser Spatial Reuse

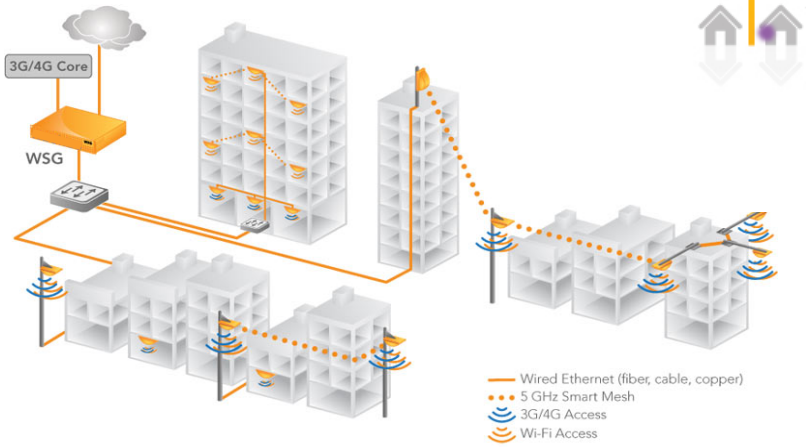
- If user-destination distance is $O(1/\sqrt{n})$, with transport capacity $O(\sqrt{n})$, we trivially achieve $O(1)$ throughput per user.



Dense infrastructure is happening!

Small cells centrally managed

Enterprise WiFi networks
WiFi offloading networks



Next generation cellular networks

More bandwidth re-use

Problems:

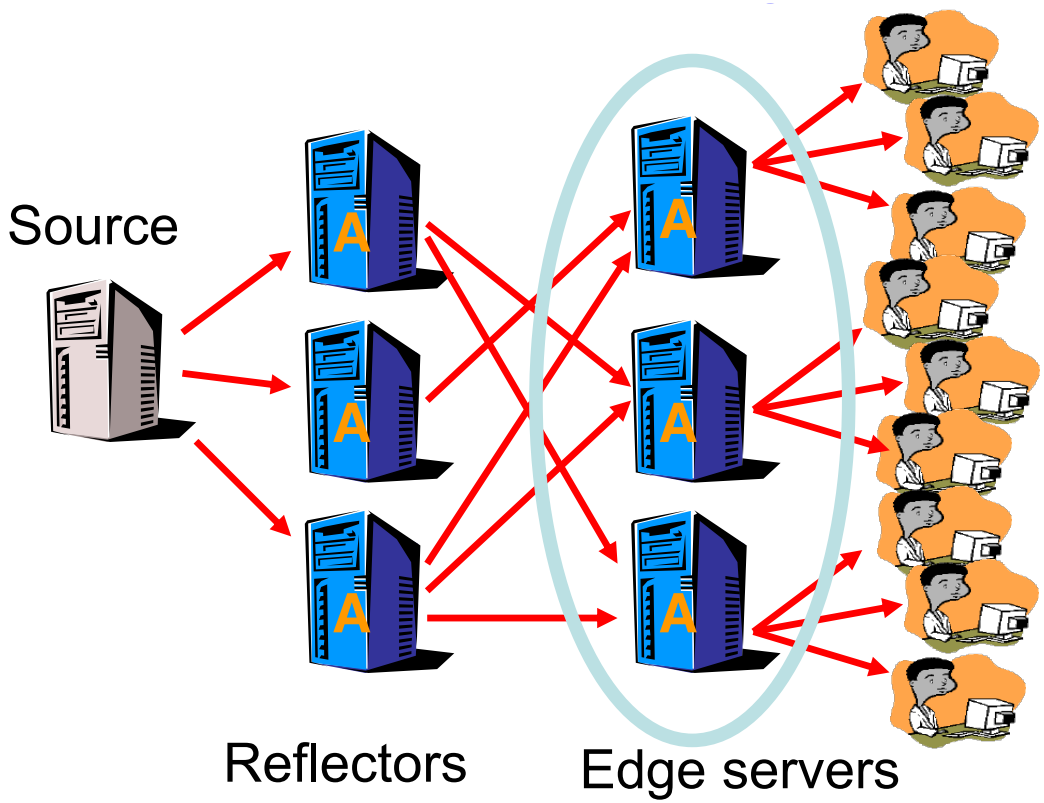
- Interference management, SoN, user plane and control plane separation, all what we have talked about in this workshop ...
- **Backhaul bottleneck.**

Video-Aware Wireless Networks

- Video is responsible for 66% of the traffic demand increase.
- Internet browsing for another 21%.
- On-demand video streaming and Internet browsing have important common features:
  1. **Asynchronous content reuse** (traffic generated by a few popular files, which are accessed in a totally asynchronous way).
  2. **Highly predictable demand distribution** (we can predict what, when and where will be requested).
  3. **Delay tolerant, variable quality,** ideally suited for best-effort (goodbye QoS, welcome QoE).

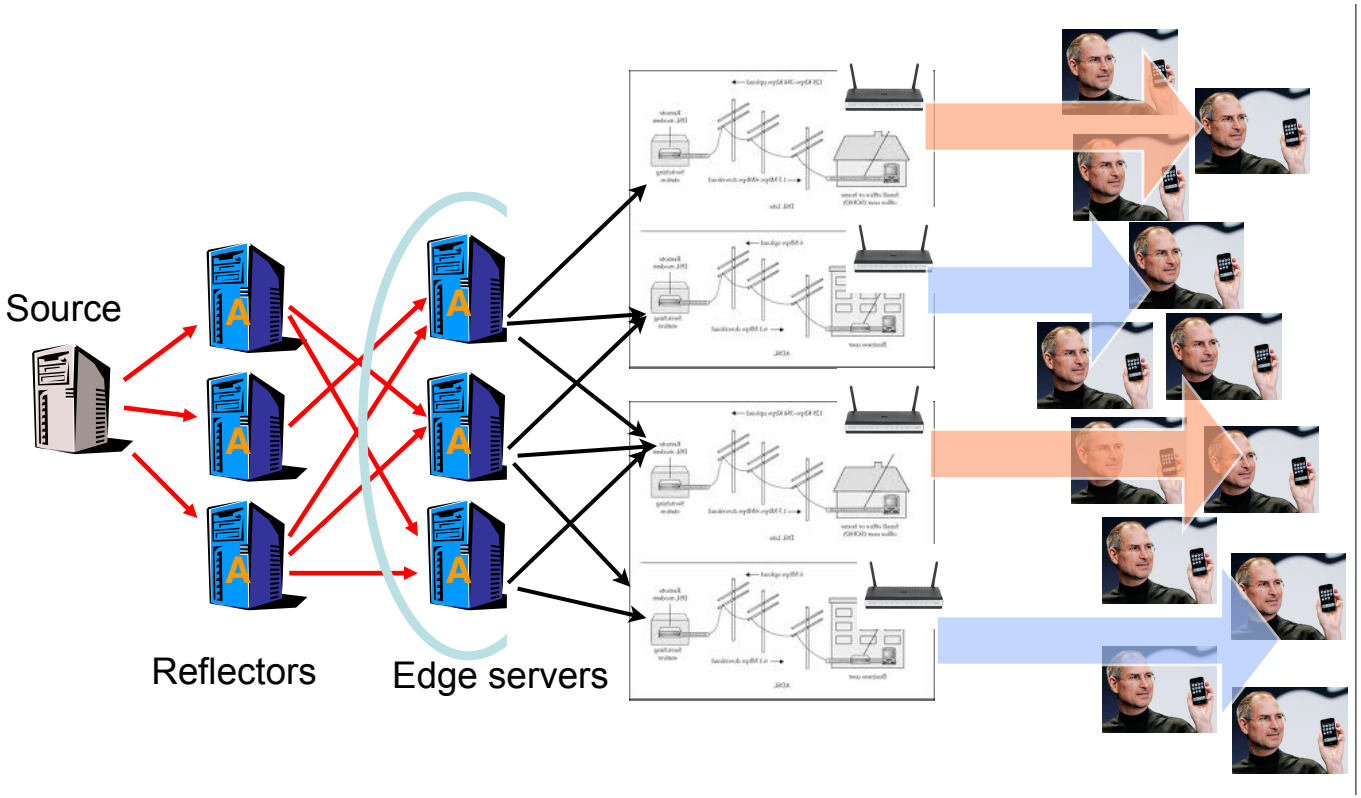
Well-Known Solution in Wired Networks: CDNs

- Caching is implemented in the core network (e.g., Akamai).
- Transparent and agnostic to the wireless segment.



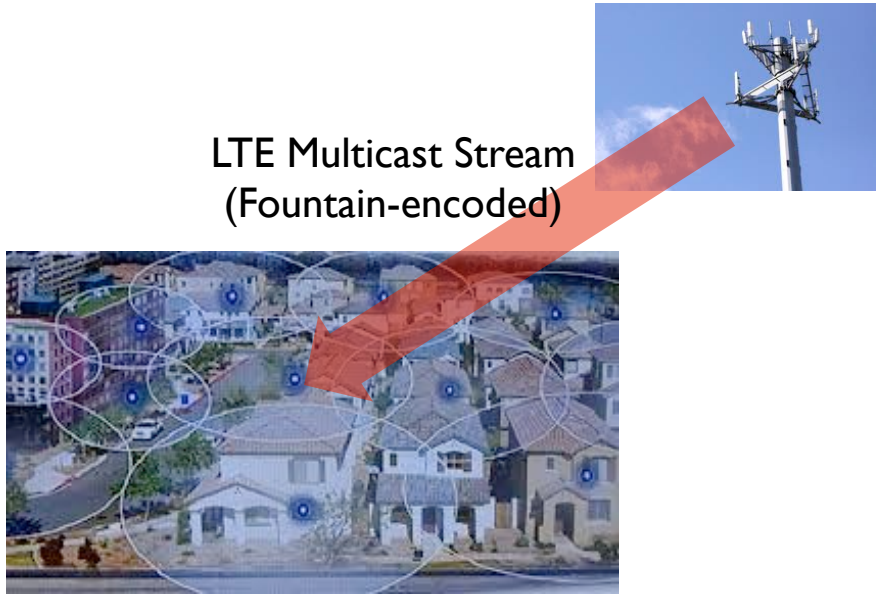
Why the Problem is Not (Yet) Solved?

- The wired backhaul to small cells is weak or expensive.
- The wireless capacity of macro-cells is not sufficient.

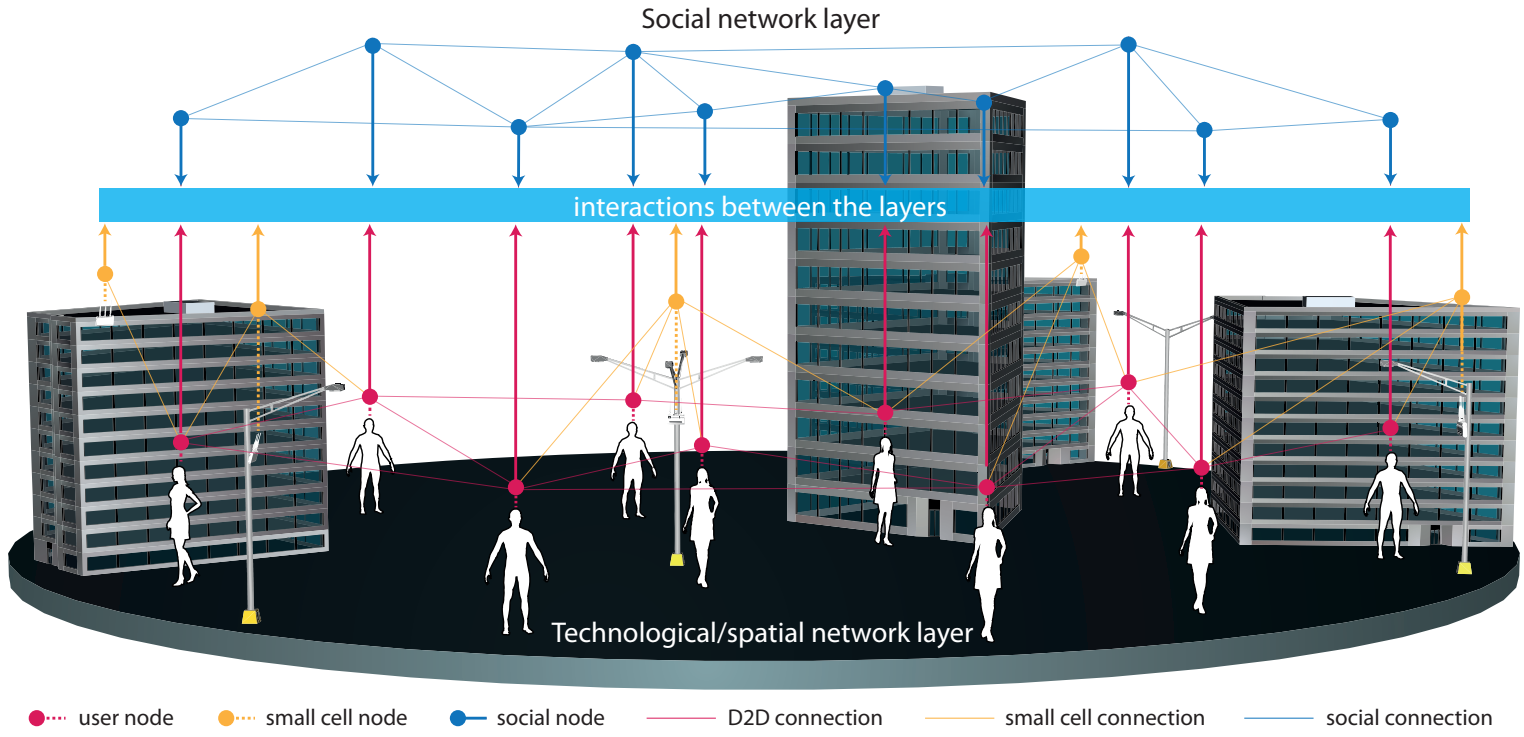


Caching at the Wireless Edge

- Femto-Caching: deploy "helper" nodes everywhere.
- Replace expensive fast backhaul with inexpensive storage capacity.
- Re-use the LTE macro-cellular network to refresh caches at off-peak times.
- Example: 4TB nodes $\times$ 100 nodes/km^2 $=$ 400 TB/km^2 of distributed storage capacity, with plain today's technology.



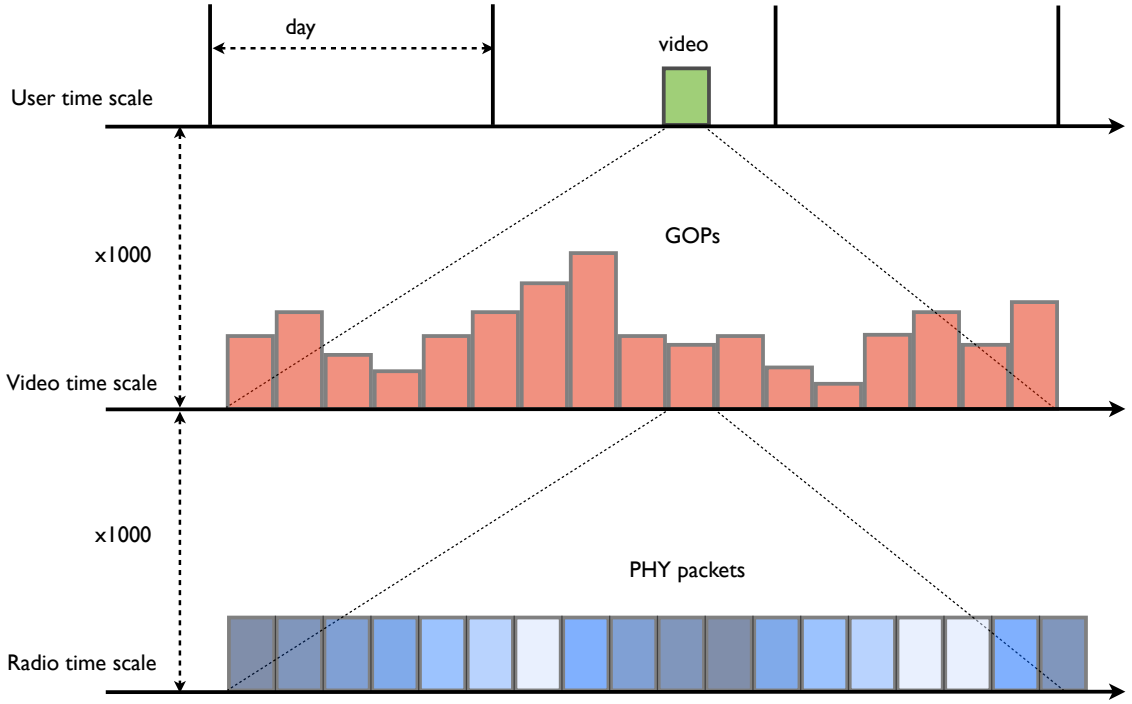
The Big Picture



- Proactive Caching: what to cache, where and when: predicting the user behavior in space and time.

Time-Scale Decomposition

- Cache placement, predictive caching at the time-scale of content popularity evolution.
- Scheduling at the time scale of the streaming sessions (video chunks).
- Underlying PHY resource allocation at the time scale of PHY slots.



Let's cut the BS

- At this point ..... the classical objections are:

  1. How do you convince the users to share their on-board memory?
  2. How do you convince the users to share their battery power?
  3. How do you convince the content providers to put their content on the user devices?
  4. When critics run out of arguments .... what about privacy?

- All the above argument are non-technical and easily countered (e.g., Google Android on-board Firewall to keep cached content inaccessible to the users).

- Users are already sharing their content spontaneously ... imagine if they have a service subscription incentive.

- Most importantly ... this is not my business (let's Bizdev people figure this out).

Throughput Scaling Laws of One-Hop Caching Networks

- [M. Ji, GC, A. F. Molisch, arXiv:1302.2168]: D2D network, random demands (known distribution), random (decentralized) caching:

$$T = \Theta\left(\max\left\{\frac{M}{m}, \frac{1}{n}\right\}\right), \quad p_o \in (0,1)$$

- [M. Maddah-Ali, U. Niesen, arXiv:1209.5807]: one sender (BS) many receivers (multicast only), arbitrary demands:

$$T = \Theta\left(\max\left\{\frac{M}{m}, \frac{1}{n}\right\}\right), \quad p_o = 0$$

- [M. Ji, GC, A. F. Molisch arXiv:1405.5336]: D2D network, arbitrary demands:

$$T = \Theta\left(\max\left\{\frac{M}{m}, \frac{1}{n}\right\}\right), \quad p_o = 0$$

Good and Bad News

- Moore's Law for bandwidth (!!): in the regime of $nM \gg m$, if you double the on-board device memory (M) you double the per-user minimum throughput.

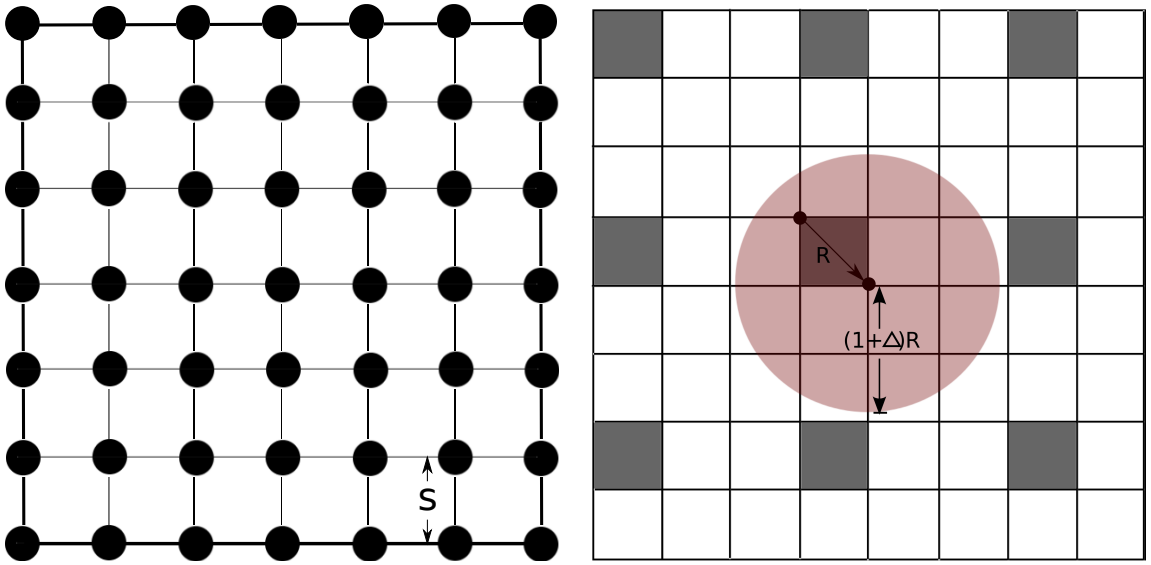
- This remarkable behavior is achieved in two ways:

  1. caching entire files and exploiting the spatial frequency reuse (dense D2D network);
  2. caching sub-packets of files and exploiting network coded multicasting (both BS and D2D).

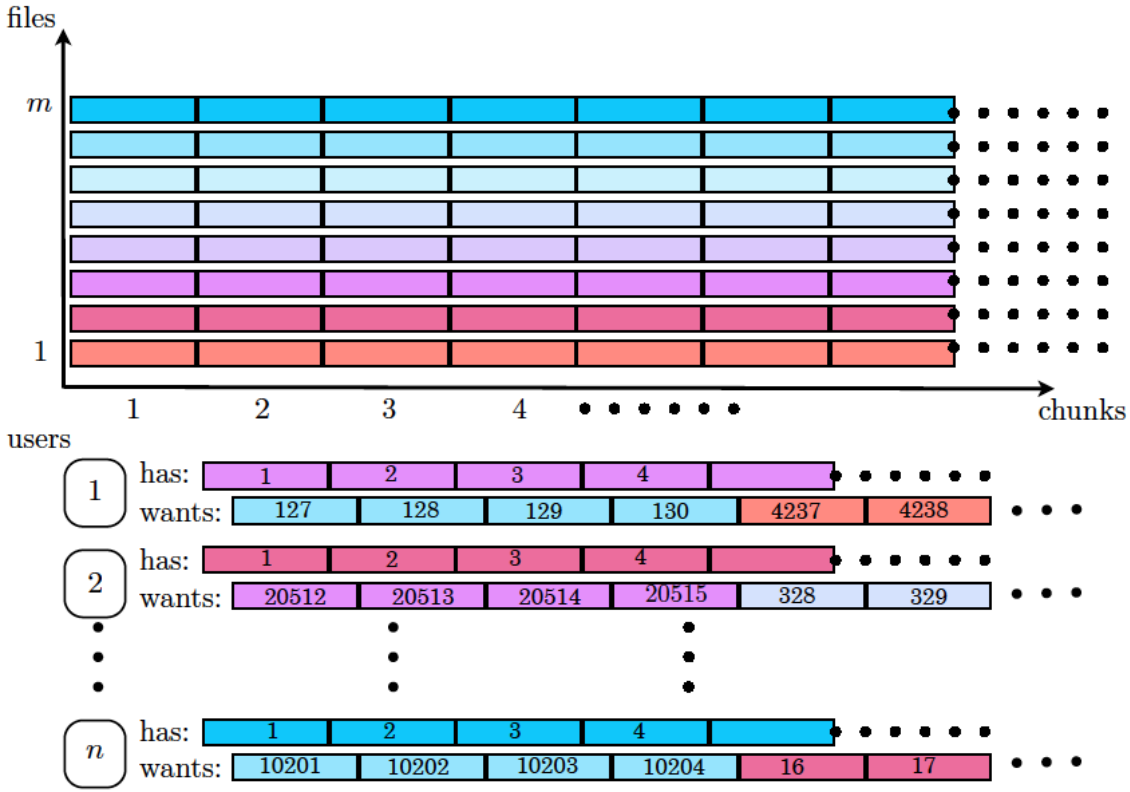
- For $m \gg nM$ there is nothing we can do (caching is ineffective!). This is the regime where asynchronous content reuse is negligible.

- Spatial multiplexing and coded multicasting do not cumulate (in fact, there is tension between the two approaches).

D2D Network with Random Demands and Random Caching



- Grid network (for analytical simplicity);
- Protocol model (as in the Gupta-Kumar model);



- An **artificial** model to model asynchronous content reuse and prevent "naive multicasting" (irrelevant for video on-demand);
- Files are formed by $L \to \infty$ packets.
- Users place random requests of sequences of $L' < \infty$ packets from library files, with uniformly distributed starting point.

Definition: Cache placement A feasible cache placement $\mathsf{G} = \{\mathcal{U}, \mathcal{F}, \mathcal{E}\}$ is a bipartite graph with "left" nodes $\mathcal{U}$, "right" nodes $\mathcal{F}$ and edges $\mathcal{E}$ such that $(u, f) \in \mathcal{E}$ indicates that file f is assigned to the cache of user u, such that the degree of each user node is $\leq M$.

Π_c is a probability mass function over $\mathcal{G}$, i.e., a particular cache placement $\mathsf{G} \in \mathcal{G}$ is assigned with probability $\Pi_c(\mathsf{G})$. $\Diamond$

Definition: Random requests At each request time (integer multiples of L'), each user $u \in \mathcal{U}$ makes a request to a segment of length L' of chunks from file $\mathsf{f}_u \in \mathcal{F}$, selected independently with probability P_r. The vector of current requests f is a random vector taking on values in $\mathcal{F}^n$, with product joint probability mass function $\mathbb{P}(\mathsf{f} = (f_1, \ldots, f_n)) = \prod_{i=1}^{n} P_r(f_i)$. $\Diamond$

Definition: Transmission policy The transmission policy Π_t is a rule to activate the D2D links in the network. Let $\mathcal{L}$ denote the set of all directed links. Let $\mathcal{A} \subseteq 2^{\mathcal{L}}$ denote the set of all feasible subsets of links (this is a subset of the power set of $\mathcal{L}$, formed by all independent sets in the network interference graph). Let $\mathsf{A} \subset \mathcal{A}$ denote a feasible set of simultaneously active links according to the protocol model. Then, Π_t is a conditional probability mass function over $\mathcal{A}$ given f (requests) and G (cache placement), assigning probability $\Pi_t(\mathsf{A}|\mathsf{f}, \mathsf{G})$ to $\mathsf{A} \in \mathcal{A}$. $\Diamond$

Definition: Useful received bits per slot For given P_r, Π_c and Π_t, and user $u \in \mathcal{U}$, the number of useful received information bits per slot unit time by user u at a given scheduling time is

$$T_u = \sum_{v:(u,v)\in \mathsf{A}} c_{u,v} 1\{\mathsf{f}_u \in \mathsf{G}(v)\}$$

where f_u denotes the file requested by user node u, $c_{u,v}$ denotes the rate of the link (u, v), and $\mathsf{G}(v)$ denotes the content of the cache of node v, i.e., the neighborhood of node v in the cache placement graph G. ◊

Definition: Number of nodes in outage The number of nodes in outage is the random variable

$$\mathsf{N}_o = \sum_{u\in\mathcal{U}} 1\{\mathbb{E}[T_u|\mathsf{f}, \mathsf{G}] = 0\}.$$

◊

Definition: Average outage probability The average (across the users) outage probability is given by

$$p_o = \frac{1}{n}\mathbb{E}[\mathsf{N}_o] = \frac{1}{n}\sum_{u\in\mathcal{U}} \mathbb{P}\left(\mathbb{E}[T_u|\mathsf{f}, \mathsf{G}] = 0\right).$$

◊

Definition: Max-min fairness throughput The minimum average user throughput is defined by

$$\overline{T}_{\min} = \min_{u \in \mathcal{U}} \left\{ \overline{T}_u \right\}.$$

◊

Definition: Throughput-Outage Tradeoff For given P_r, a throughput-outage pair (T, p) is *achievable* if there exists a cache placement Π_c and a transmission policy Π_t with outage probability $p_o \leq p$ and minimum per-user average throughput $\overline{T}_{\min} \geq T$.

The throughput-outage achievable region $\mathcal{T}$ is the closure of all achievable throughput-outage pairs (T, p). In particular, we let $T^*(p) = \sup\{T : (T, p) \in \mathcal{T}\}$. ◊

Notice that $T^*(p)$ is the result of the following optimization problem (over Π_c, Π_t):

$$\begin{aligned} \text{maximize} \quad & \overline{T}_{\min} \\ \text{subject to} \quad & p_o \leq p, \end{aligned}$$

- $T^*(p)$ is non-decreasing in p.
- The range of feasible outage probability, in general, is an interval $[p_{o,\min}, 1]$ for some $p_{o,\min} \geq 0$.
- We say that an achievable point (T, p) dominates an achievable point (T', p') if $p \leq p'$ and $T \geq T'$.
- The Pareto boundary of $\mathcal{T}$ consists of all achievable points that are not dominated by other achievable points, i.e., it is given by $\{(T^*(p), p) : p \in [p_{o,\min}, 1]\}$.

Achievability Strategy: Clustering and Random Decentralized Caching

- Clustering: the network is divided into clusters of equal size, denoted by $g_c(m)$. Each user searches its desired content in its own cluster.

- Random Caching: each node independently caches M files according to a common probability distribution P_c^*.

Theorem 1: The caching distribution P_c^* maximizing the probability that any user $u \in \mathcal{U}$ finds its requested file inside its corresponding cluster is given by

$$P_c^*(f) = \left[1 - \frac{\nu}{z_f}\right]^+, \quad f = 1, \ldots, m,$$

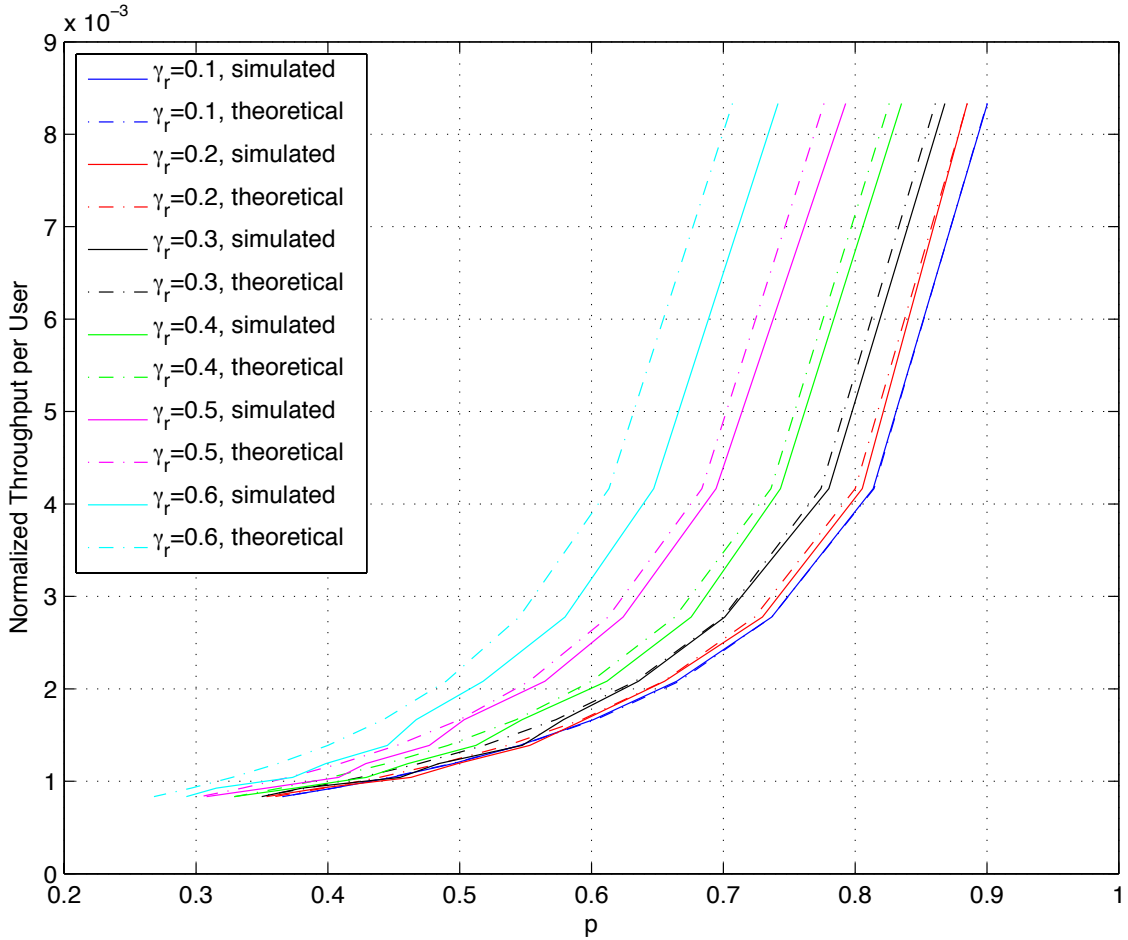
where $\nu = \frac{m^*-1}{\sum_{j=1}^{m^*} \frac{1}{z_j}}$, $z_j = P_r(j)^{\frac{1}{M(g_c(m)-1)-1}}$, and $m^* = \Theta\left(\min\{\frac{M}{\gamma_r} g_c(m), m\}\right)$. $\square$

Theorem 2: Assume $P_r(f) = \frac{f^{-\gamma_r}}{\sum_{j=1}^{m} \frac{1}{j^{\gamma_r}}}$ (Zipf demand distribution), let $\alpha \triangleq \frac{1-\gamma_r}{2-\gamma_r}$, and $\lim_{n\to\infty} \frac{m^\alpha}{n} = 0$. Then, the throughput-outage tradeoff achievable by random caching and clustering behaves as:

$$T(p) = \begin{cases} \frac{C}{K}\frac{M}{\rho_1 m} + o(1/m), & p = (1-\gamma_r)e^{\gamma_r - \rho_1} \\ \frac{CA}{K}\frac{M}{m(1-p)^{\frac{1}{1-\gamma_r}}} + o\left(\frac{1}{m(1-p)^{\frac{1}{1-\gamma_r}}}\right), & p = 1 - a\left(\frac{g_c(m)}{m}\right)^{1-\gamma_r} \\ \frac{CB}{K} m^{-\alpha} + o\left(m^{-\alpha}\right), & 1 - a\rho_2^{1-\gamma_r} m^{-\alpha} \le p \le 1 - ab^{1-\gamma_r} m^{-\alpha} \\ \frac{CD}{K} m^{-\alpha} + o\left(m^{-\alpha}\right), & 1 - ab^{1-\gamma_r} m^{-\alpha} \le p \le 1, \end{cases}$$

where we define $a = \gamma_r^{\gamma_r} M^{1-\gamma_r}$, $b = \left(\frac{1-\gamma_r}{a}\right)^{\frac{1}{2-\gamma_r}}$, $A \triangleq \gamma_r{}^{\frac{\gamma_r}{1-\gamma_r}}$, $B \triangleq \frac{a\rho_2^{1-\gamma_r}}{1+a\rho_2^{2-\gamma_r}}$, $D \triangleq \frac{ab^{1-\gamma_r}}{1+ab^{2-\gamma_r}}$ and where ρ_1 and ρ_2 are positive parameters satisfying $\rho_1 \ge \gamma_r$ and $\rho_2 \ge b$. The cluster size $g_c(m)$ is any function of m satisfying $g_c(m) = \omega\left(m^\alpha\right)$ and $g_c(m) \le \gamma_r m/M$. □

Not just the usual scaling law: We can pin down the constants too!!



Comparison between formulas and simulation for the minimum throughput per user v.s. outage probability. The throughput is normalized by C (link rate), $m = 1000$, $n = 10000$, reuse factor $K = 4$, $\gamma_r \in [0.1, 0.6]$.

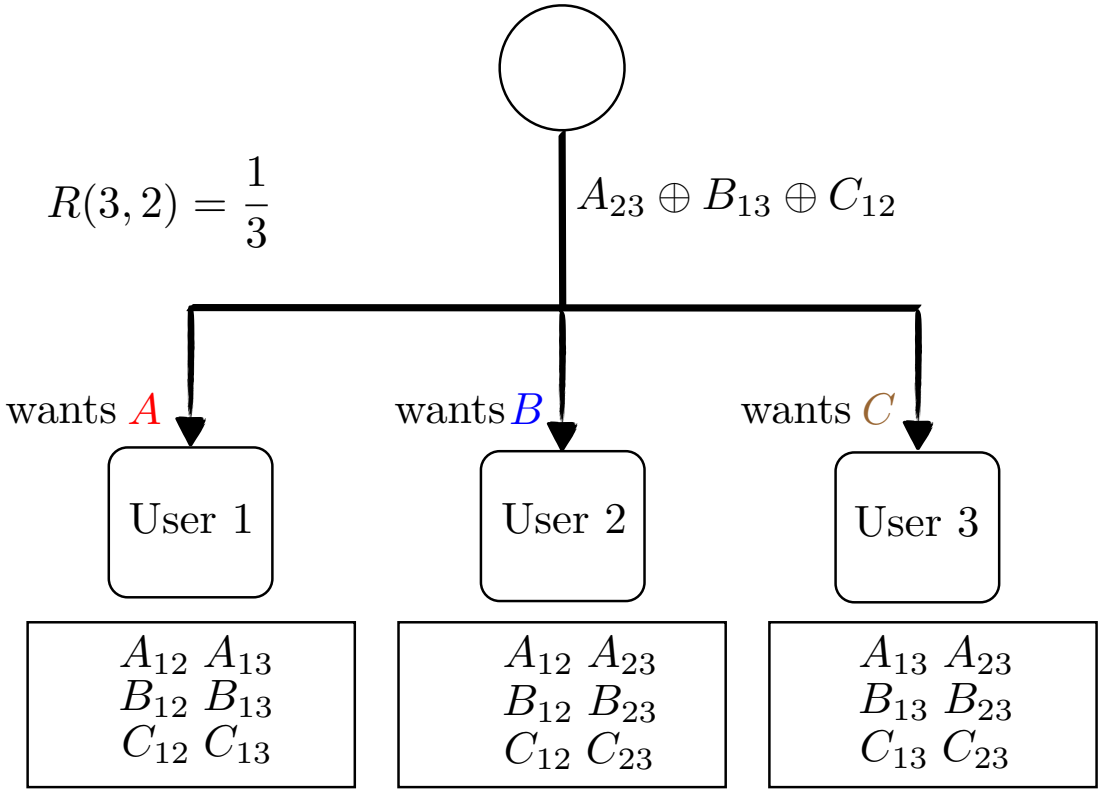
The Maddah-Ali and Niesen Scheme

- Definition of rate: number of equivalent file transmissions needed to deliver all demanded files to all users (zero outage, arbitrary demands).
- Maddah-Ali and Niesen show that the following rate is achievable:

$$R(M) = n\left(1 - \frac{M}{m}\right)\frac{1}{1 + \frac{Mn}{m}}$$

- The corresponding throughput scaling law is obtained as $T(M) = \frac{C}{R(M)}$ where C is the data rate of the multicast bottleneck link, and therefore.
- In the relevant regime of $nM \gg m$ we have again $T(M) = \Theta(\frac{m}{M})$.
- An information theoretic cut-set bound on the expanded compound channel corresponding to all possible (arbitrary) demands, shows that $R(M)$ is optimal within a bounded multiplicative factor.

Example: $n = m = 3$, $M = 2$



- Files are divided into three sub-packets of size $1/3$ each:

$$A = (A_{12}, A_{13}, A_{23}), \quad B = (B_{12}, B_{13}, B_{23}), \quad C = (C_{12}, C_{13}, C_{23})$$

The Ji, Caire and Molisch D2D scheme

- Same setting of Maddah-Ali and Niesen, but no "omniscient" central server.
- For a unit-area (squared) network, transmission range $r \geq \sqrt{2}$ a single transmission is received by all nodes (multicasting only).
- With $r < \sqrt{2}$, we can induce spatial spectrum reuse.

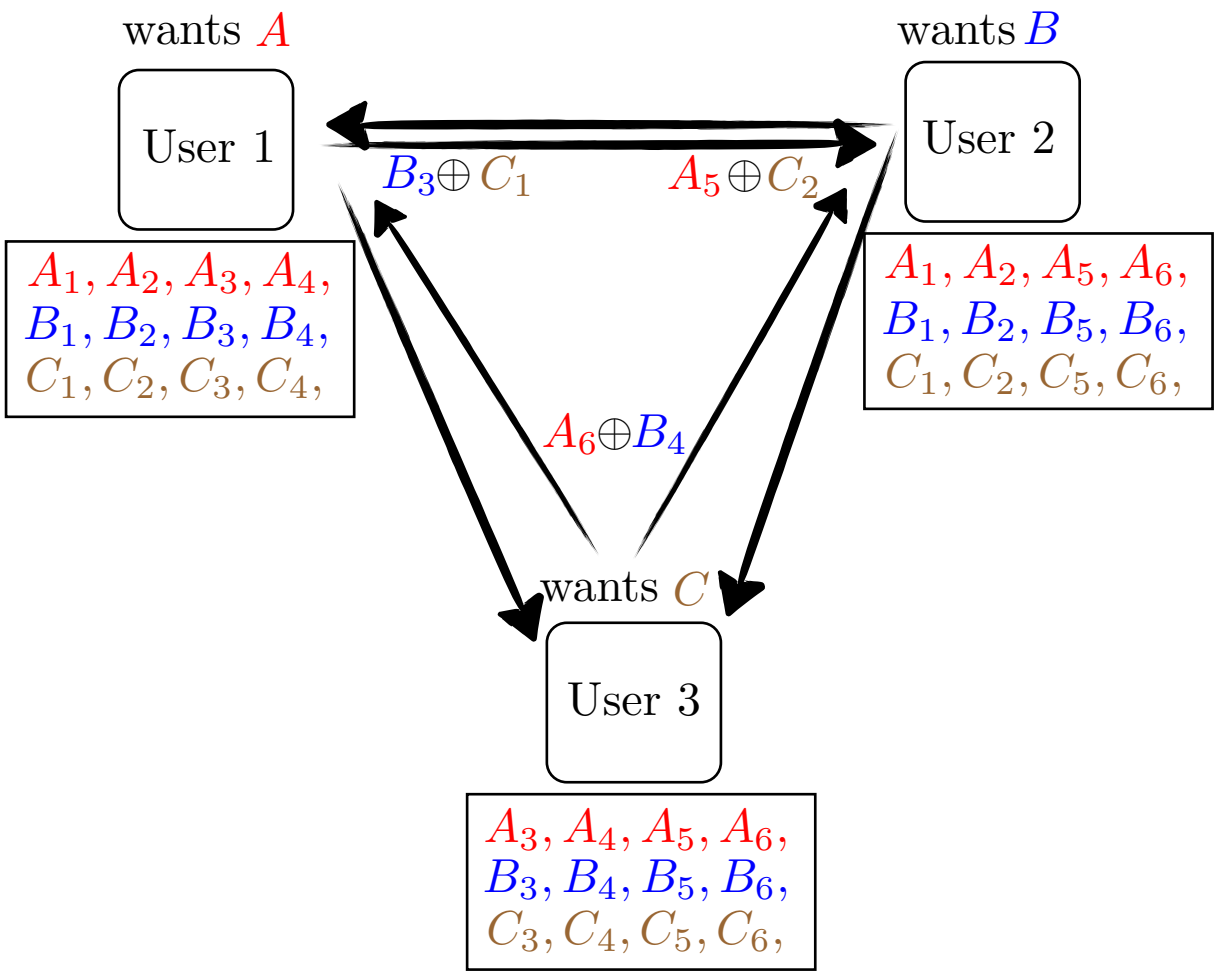
Theorem 3: With D2D transmission radius $r \geq \sqrt{2}$ and $t = \frac{Mn}{m} \in \mathbb{Z}^+$, the following rate is achievable:

$$R(M) = \frac{m}{M}\left(1 - \frac{M}{m}\right).$$

Moreover, when t is not an integer, the convex lower envelope of $R(M)$, seen as a function of $M \in [0 : m]$, is achievable. □

Notice that, as before, we obtain the throughput scaling law $T(M) = \Theta(\frac{M}{m})$ in the regime of $nM \gg m$.

Example: $n = m = 3$, $M = 2$: achievability



- We divide each packet of each file into 6 subpackets, and denote the subpackets of the j-th packet as $\{A_\ell : \ell = 1, \ldots, 6\}$, $\{B_\ell : \ell = 1, \ldots, 6\}$, and $\{C_\ell : \ell = 1, \ldots, 6\}$. The size of each subpacket is $F/6$. We let user u stores Z_u, $u = 1, 2, 3$, given as follows:

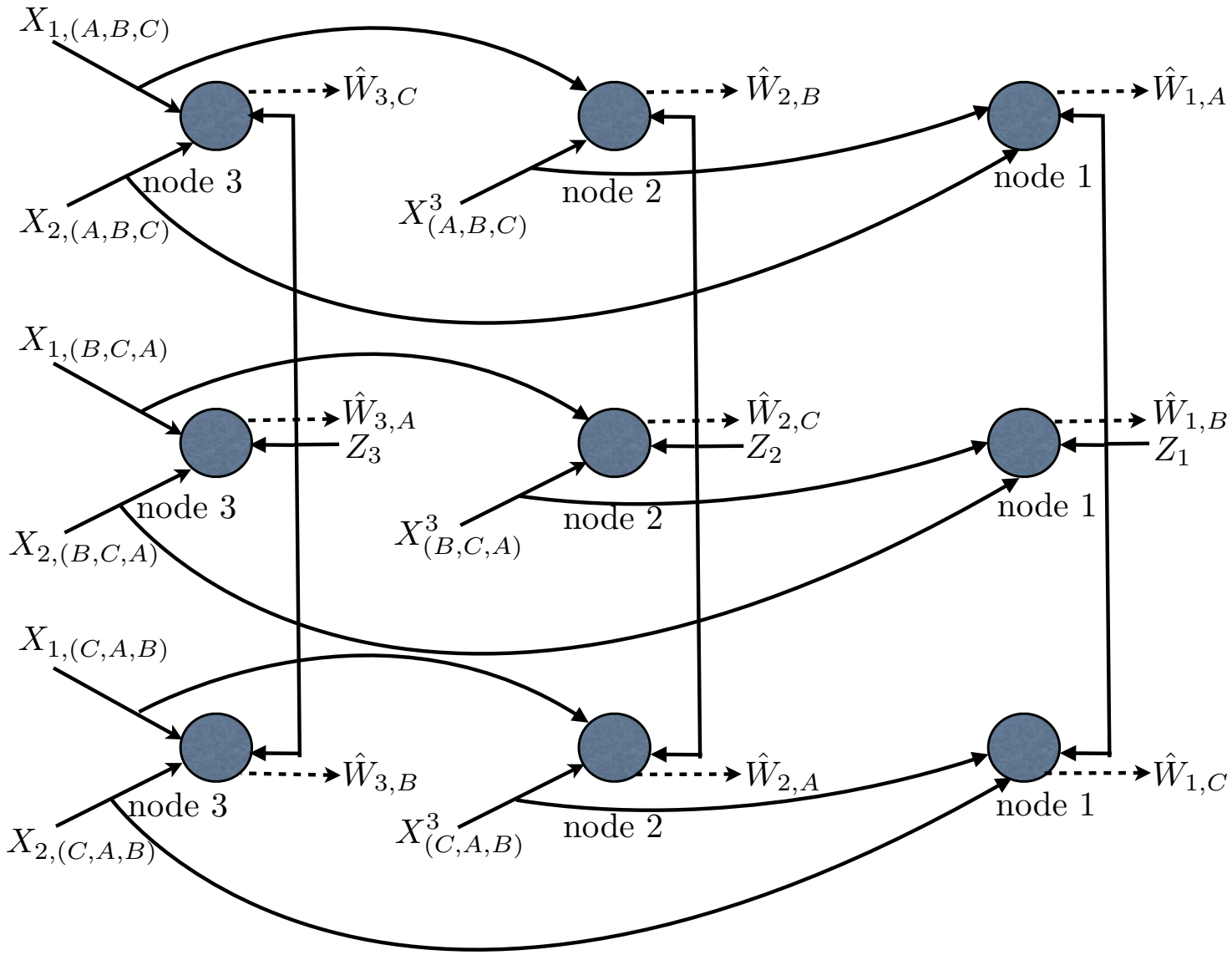
$$\begin{aligned} Z_1 =& (A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4, C_1, C_2, C_3, C_4), \\ Z_2 =& (A_1, A_2, A_5, A_6, B_1, B_2, B_5, B_6, C_1, C_2, C_5, C_6), \\ Z_3 =& (A_3, A_4, A_5, A_6, B_3, B_4, B_5, B_6, C_3, C_4, C_5, C_6), \end{aligned}$$

- Consider the demand $\mathsf{f} = (A, B, C)$. Since the request vector contains distinct files, specifying which segment of each file is requested (i.e., the vector s) is irrelevant and shall be omitted.

- In the coded delivery phase (see figure) user 1 multicasts $B_3 + C_1$ (useful to both user 2 and 3), user 2 multicasts $A_5 + C_2$ (useful to both users 1 and 3) and user 3 multicasts $A_6 + B_4$ (useful to both users 1 and 2).

- It follows that $R(2) = R_1 + R_2 + R_3 = \frac{1}{6} \cdot 3 = \frac{1}{2}$ is achievable.

Example: $n = m = 3$, $M = 2$: outer bound



- Z_u denotes the cached symbols at user $u = 1, 2, 3$, $X_{u,\mathsf{f}}$ denotes the transmitted message from user u in correspondence of demand f, and $\hat{W}_{u,f}$ is the decoded message at user u relative to file f.

- Considering user 3, from the cut that separates $(X_{1,(A,B,C)}, X_{2,(A,B,C)}, X_{1,(B,C,A)}$ and $(\hat{W}_{3,C}, \hat{W}_{3,A}, \hat{W}_{3,B})$, and by using the fact that the sum of the entropies of the received messages and the entropy of the side information (cache symbols) cannot be smaller than the number of requested *information bits*, we obtain

$$\sum_{s=1}^{\frac{L}{L'}} \Big(R^{\mathrm{T}}_{1,s,(A,B,C)} + R^{\mathrm{T}}_{2,s,(A,B,C)} + R^{\mathrm{T}}_{1,s,(B,C,A)} + R^{\mathrm{T}}_{2,s,(B,C,A)} + R^{\mathrm{T}}_{1,s,(C,A,B)} + R^{\mathrm{T}}_{2,s,(C,A,B)} \Big) + MFL \geq 3FL' \cdot L/L'.$$

- Similar inequalities are obtained by permuting the indices (corresponding cuts for the other users).

- By summing the corresponding inequalities and dividing all terms by 2, we obtain

$$\sum_{s=1}^{\frac{L}{L'}} \Big(R^{\mathrm{T}}_{1,s,(A,B,C)} + R^{\mathrm{T}}_{2,s,(A,B,C)} + R^{\mathrm{T}}_{3,s,(A,B,C)} + R^{\mathrm{T}}_{1,s,(B,C,A)} + R^{\mathrm{T}}_{2,s,(B,C,A)} + R^{\mathrm{T}}_{3,s,(B,C,A)} + R^{\mathrm{T}}_{1,s,(C,A,B)} + R^{\mathrm{T}}_{2,s,(C,A,B)} + R^{\mathrm{T}}_{3,s,(C,A,B)} \Big) + \frac{3}{2} MFL \geq \frac{9}{2} FL.$$

- Since we are interested in minimizing the worst-case rate, the sum $R^{T}_{1,s,\mathrm{f}} + R^{T}_{2,s,\mathrm{f}} + R^{T}_{3,s,\mathrm{f}}$ must yields the same min-max value R^{T} for any s and f. This yields the bound

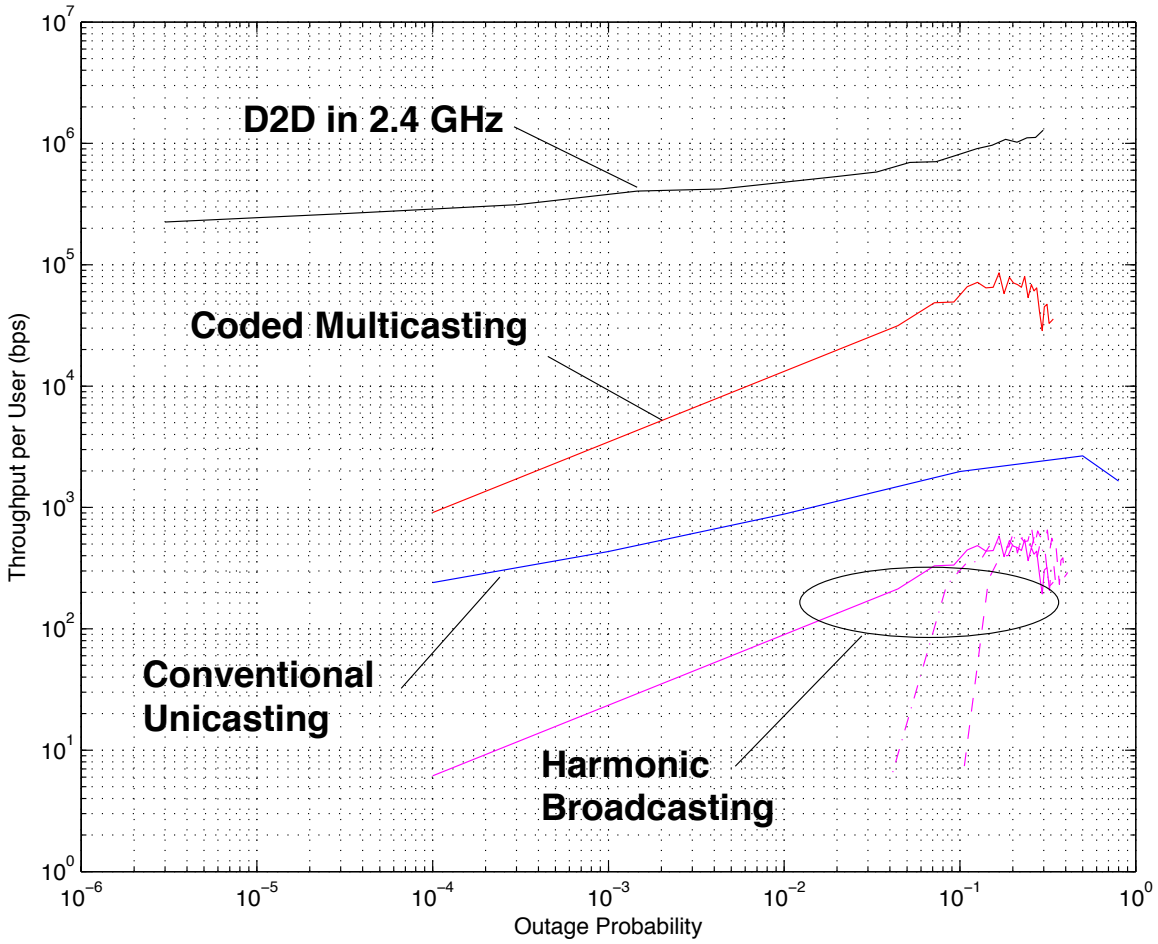
$$\frac{3L}{L'} R^{\mathrm{T}} \geq \frac{9}{2} FL - \frac{3}{2} MFL.$$

- Finally, by definition of rate $R(M)$, we have that $R(M) = R^T/(FL')$. Therefore, dividing both sides by $3FL$, we obtain that the best possible achievable rate must satisfy

$$R^*(M) \geq \frac{3}{2} - \frac{1}{2}M.$$

- In the example of this section, for $M = 2$ we obtain $R^*(2) \geq \frac{1}{2}$ (in this case the achievability scheme given before is information theoretically optimal).

How do these schemes compare in practice?



(details in [M. Ji, GC, A. F. Molisch, arXiv:1305.5216])

Conclusions

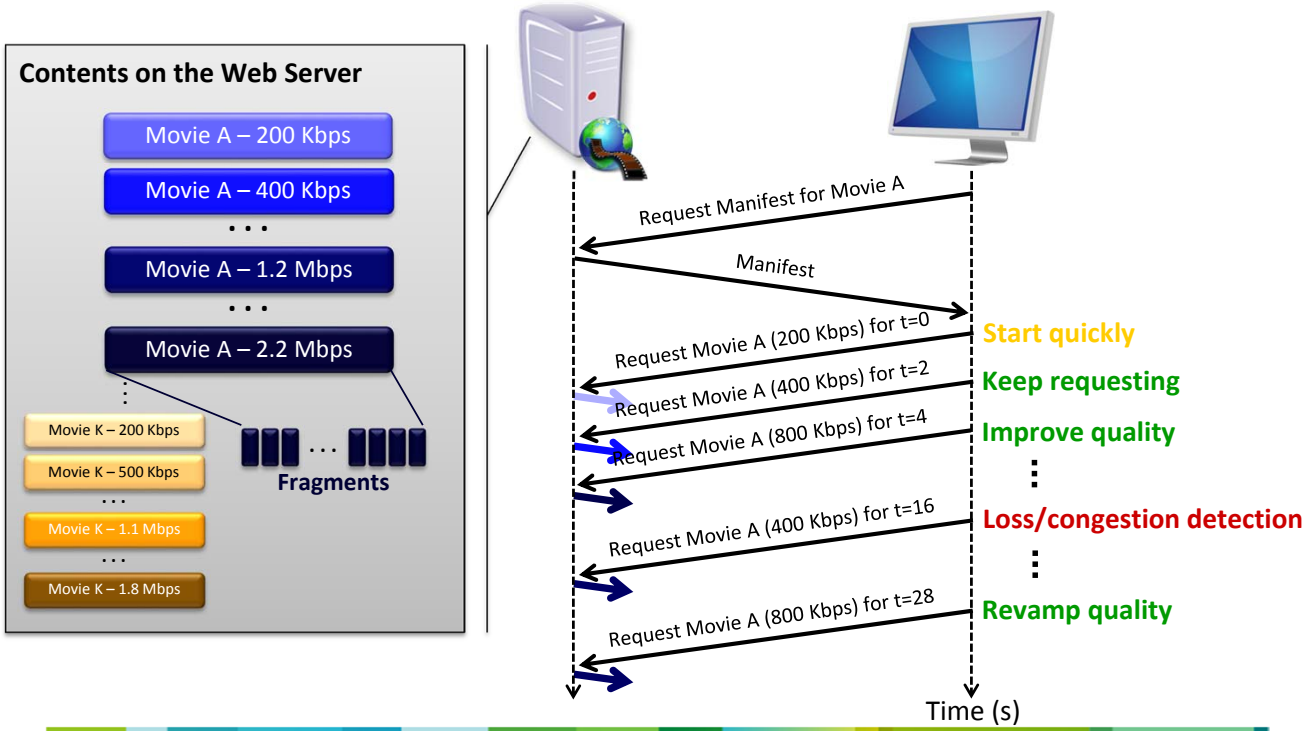
- Exploiting the asynchronous content reuse is key for achieving the required 100x.
- Caching at the wireless edge has a great potential, since it relaxes the constraints on the backhaul (expensive network component).
- Femto-Caching (helper nodes), and D2D Caching (caching at the user devices).
- Good news for LTE operators: new use of the macro-cellular base stations at off-peak times.
- **Caching achieves "Moore's Law" for bandwidth:** $T = \Theta(M/m)$**.**

Thank You

DASH (Dynamic Adaptive Streaming over HTTP)

Adaptive Streaming over HTTP

Multi-Bitrate Encoding and Other Concepts



- Microsoft Smooth Streaming (Silverlight).
- Apple HTTP Live Streaming.
- 3GPP Dynamic Adaptive Streaming over HTTP (DASH).

System model and problem statement

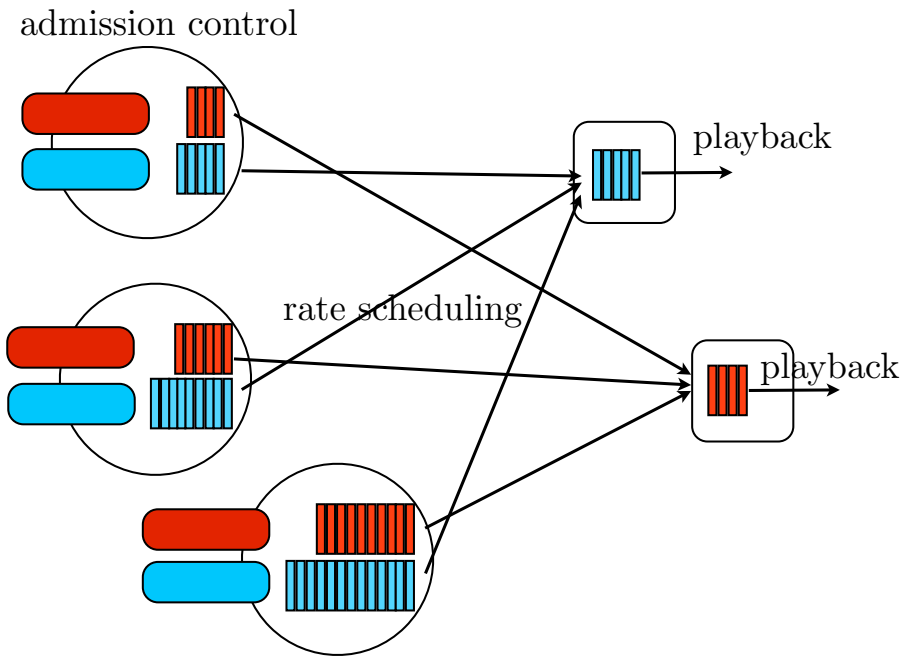
- The network is defined by a bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{H}, \mathcal{E})$.
- Edges $(h, u) \in \mathcal{E}$ when there exists a potential transmission link between $h \in \mathcal{H}$ and $u \in \mathcal{U}$.
- Users $u \in \mathcal{U}$ request video files f_u from a library of possible files $\mathcal{F}$.
- Video files are formed by sequences of chunks of fixed playback duration $T_{\text{chunk}} = (\#\text{ frames per GOP})/\eta$ (η is the frame rate).
- Playback starts after a short pre-buffering time T_u (expressed in multiples of T_{chunk}).
- Problem: schedule the chunk transmission such that for each $u \in \mathcal{U}$ and time $t = T_u, T_u + 1, T_u + 2, \ldots$

Variable Bite-Rate coding and video quality levels

- Each chunk of file f is encoded at multiple quality levels $m \in \{1, \ldots, N_f\}$.
- Without loss of generality, we let $D_f(m,t)$ and $B_f(m,t)$ denote the video quality index and the number of bits per chunk of file f and chunk t.
- Quality decisions: at every chunk time t, choose the quality mode $m_u(t)$ for all requesting users $u \in \mathcal{U}$.
- Letting $R_{hu}(t)$ denote the source coding rate (bit per chunk) of chunk t received by user u from helper h, we have the source-coding rate constraint:

$$\sum_{h \in \mathcal{N}(u) \cap \mathcal{H}(f_u)} R_{hu}(t) = B_{f_u}(m_u(t), t), \quad \forall\, (h,u) \in \mathcal{E}.$$

Helpers transmission queues



- We assume that each helper node has transmission queues pointing at its served users $u \in \mathcal{N}(h)$.
- The evolution of the transmission queues is given by:

$$Q_{hu}(t+1) = \max\{Q_{hu}(t) - n\mu_{hu}(t), 0\} + R_{hu}(t), \quad \forall\ (h,u) \in \mathcal{E}.$$

Modeling the PHY as a deterministic slowly-varying network

- We "collapse" the PHY into the network long-term average achievable rate region $\mathcal{R}(t)$.
- By definition, $\mathcal{R}(t)$ is a convex bounded region of $\mathbb{R}_+^{|\mathcal{E}|}$.
- $\mathcal{R}(t)$ is (slowly) varying with t because of non-ergodic phenomena, such as users joining or leaving the system or user mobility.
- Example: intra-cell orthogonal access, treating inter-cell interference as noise:

$$\sum_{u \in \mathcal{N}(h)} \frac{\mu_{hu}(t)}{C_{hu}(t)} \leq 1, \quad \forall\, h \in \mathcal{H},$$

where

$$C_{hu}(t) = \mathbb{E}\left[\log\left(1 + \frac{P_h g_{hu}(t)|a_{hu}|^2}{1 + \sum_{h' \neq h} P_{h'} g_{h'u}(t)|a_{h'u}|^2}\right)\right].$$

(This corresponds to FDMA/TDMA orthogonal sharing of the downlink).

Network Utility Maximization

- Define time-averaged quantities as: $\overline{x} := \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[x(\tau)]$.
- Optimization Problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{u\in\mathcal{U}} \phi_u(\overline{D}_u) \\ \text{subject to} \quad & \overline{Q}_{hu} < \infty \; \forall \; (h,u) \in \mathcal{E} \\ & \alpha(t) \in A_{\omega(t)} \; \forall \; t, \end{aligned}$$

- Utility functions: $\phi_u(\cdot)$ are concave and non-decreasing functions.

- Network state:

$$\omega(t) = \{g_{hu}(t), D_{f_u}(\cdot, t), B_{f_u}(\cdot, t) : \forall\ (h, u) \in \mathcal{E}\}\,.$$

- Control actions: $\alpha(t) = \{\mathbf{R}(t), \boldsymbol{\mu}(t), \{m_u(t) : u \in \mathcal{U}\}\}$.

- Feasible set $A_{\omega(t)}$ defined by the source coding rate constraints and by the PHY rate region $\mathcal{R}(t)$.

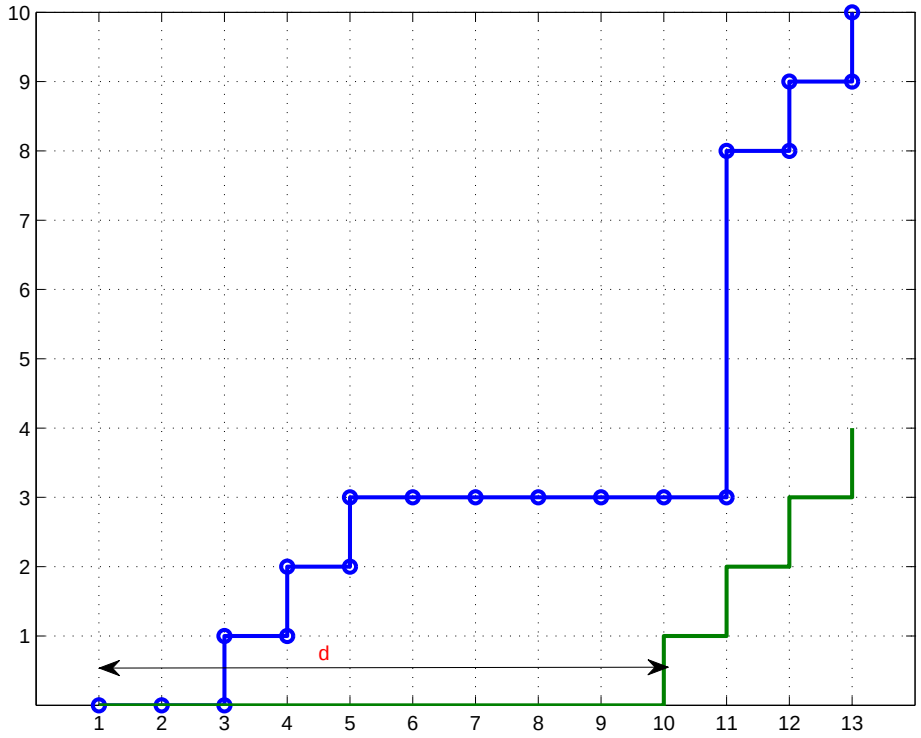
Dynamic policy via DPP

- We used the classical method of Lyapunov Drift Plus Penalty (DPP).
- The problem decomposes naturally into three decentralized subproblems: admission control, transmission scheduling, and greedy objective function maximization.
- In summary:
  1. Each user u decides from which helper $h \in \mathcal{N}(u)$ to request the next chunk, and at which quality.
  2. Each helper h decides to which user $u \in \mathcal{N}(h)$ to send for the whole chunk.
- The resulting scheme is a generalization of DASH to multiuser networks.
- Provably near-optimal (through control parameter) on a per-sample path basis (arbitrary evolution of $\omega(t)$).

Handling the playback buffer

- Our problem formulation has completely neglected the users' playback buffer.
- We handle the playback buffer through a reasonable heuristic approach (pre-buffering/re-buffering and chunk skipping).
- Example: chunk arrival process

1	2	3	4	5	6	7	8	9	10	11	12	13
3	4	5	11	6	8	9	10	12	13	16	15	14



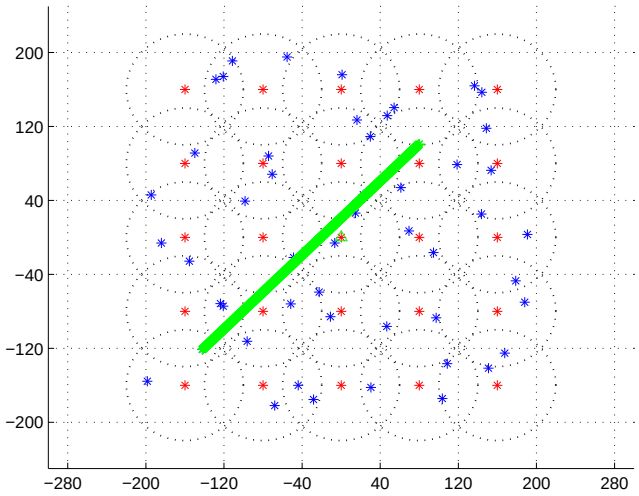
- Chunk availability and chunk consumption.
- The pre-buffering time T_u should be larger than the max chunk delivery delay.

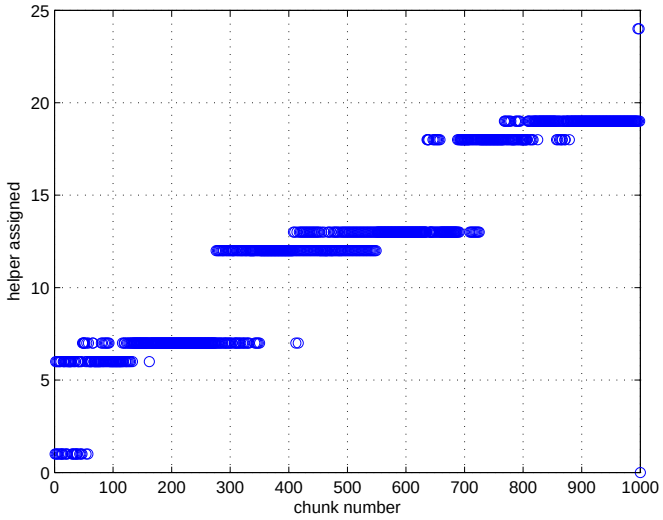
Chunk skipping and pre-buffering/re-buffering

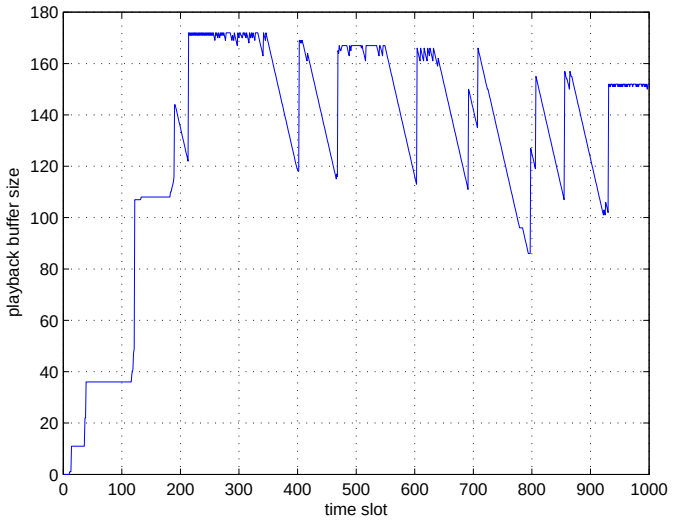
- If a "late" chunk is "blocking" a large increase in the playable buffer, then the chunk is skipped.
- Skipping decisions depend on a threshold on the promised increase in the playable buffer.
- The pre-buffering delay T_u is decided at each user u in a decentralized manner, by monitoring the max delivery delay observed in a sliding window.
- Each user keeps track of its max observed delivery delay in a window, such that in the case of a stall event, the re-buffering time is set as a function of updated observed max delay.
- Details can be found in:

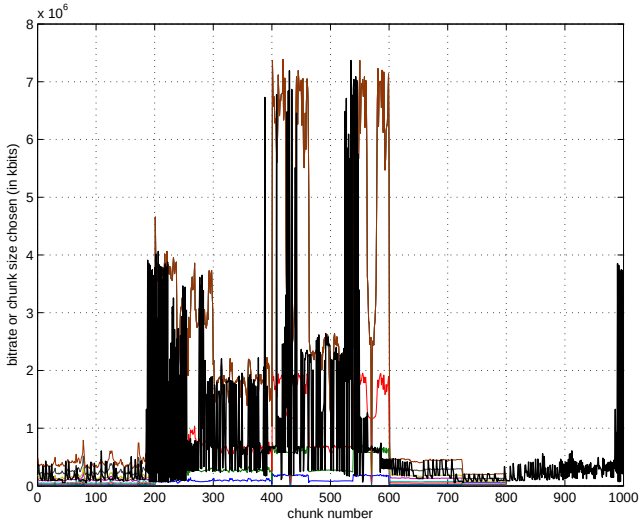
  D. Bethanabhotla, G. Caire and M. J. Neely, "Joint Transmission Scheduling and Congestion Control for Adaptive Video Streaming in Small-Cell Networks," ArXiv:1304.8083 (submitted to IEEE Trans. on Comm., 2013).

Mobility experiment with VBR coded video









Improvements and generalizations

- We considered a "pull" strategy with single per-user request queue in order to avoid out-of-order chunks.
- We considered a multi-cell MU-MIMO PHY, inspired by 802.11ac wave-2 (or future massive MIMO small cells at mm-wave bands).
- Details can be found in:

  [D. Bethanabhotla, GG and M. J. Neely, "Adaptive Video Streaming in MU-MIMO Networks," arXiv:1401.6476]

- Interesting solved issue: how to do efficient rate scheduling on the PHY in an efficient way in a multi-cell MU-MIMO network.

Conclusions

- Exploiting the asynchronous content reuse of wireless data killer apps is key for achieving the required 100x.
- Caching at the wireless edge has a great potential, since it relaxes the constraints on the backhaul (expensive network component).
- Femto-Caching (helper nodes), and D2D Caching (caching at the user devices).
- We have developed a NUM framework for adaptive dynamic streaming in Femto-Caching networks.
- Good news for LTE operators: new use of the macro-cellular base stations at off-peak times.
- Good news for Small-Cell/Enterprise WiFi manufacturers: Femto-Caching helpers density $\approx$ user density.

Thank You