Blocking probability in a caching hierarchy network

Lu, Y; Kuipers, FA; Hartog, F; Van Mieghem, PFA

DOI
10.1109/CCNC.2011.5766594

Publication date
2011

Document Version
Accepted author manuscript

Published in
8th Annual IEEE Consumer Communications and Networking Conference 2011

Blocking probability in a caching hierarchy network

Yue Lu[1], Fernando Kuipers[1], Frank den Hartog[2], and Piet Van Mieghem[1]
[1] Delft University of Technology, {Y.Lu, F.A.Kuipers, P.F.A.VanMieghem}@tudelft.nl
[2] TNO, frank.denhartog@tno.nl

***Abstract*—We develop a performance model for the availability of a MobileTV service in a caching hierarchy network. The probability that bandwidth for the service is available is calculated as a function of channel popularity, the number of available channels, cache sizes, network configuration, and content viewing behavior. The system performance and the key factors that affect the performance are analyzed.**

I. Introduction

A content delivery network (CDN) is a system of computers containing copies of data placed at various points (caches) in a network so as to maximize content availability for clients. Caches are used to reduce bandwidth requirements, reduce server load, and improve the client response times for content.

Various caching architectures exist, such as hierarchical and distributed caching. Compared with hierarchical caching, distributed web caching achieves shorter times to send a document from the cache to the destination, and shows very good performance in well-interconnected areas without requiring intermediate cache levels [1]. However, the deployment of distributed caching on a large scale is difficult, due to large times to find a document in the caching architecture, high bandwidth use and administrative issues.

In the case of hierarchical caching, the original complete content sources are stored in a central server located at the highest level of the caching hierarchy. The content may be stored in caches placed at multiple levels in the hierarchical network, but also at the end user [2]. A hierarchical caching architecture is particularly beneficial when cooperating cache servers do not have high-speed connectivity. Notably, a MobileTV network via UMTS could be designed as a hierarchical architecture. In this case, popular objects can be efficiently diffused toward the demand. We show that an hierarchical architecture for the distribution of streaming content is realistic, by applying a new *caching hierarchy* model. The model is described in the next three sections of the paper. In Section V, the model is applied to a realistic use case. The MobileTV service investigated is a real-time live sports/news stream delivery service. It uses multicast to deliver the TV streams to save resources, which is what MBMS (Multimedia Broadcast and Multicast Services) [3] does nowadays. An analytical model for multicast TV without a pausing service has been studied in [4]. We consider that users can pause the TV stream, for instance when a call is coming in. Mobile Handover is considered as a new user arrival/departure in our model. We realize that such a caching hierarchy requires caches to be placed at the key access points in the network, which requires significant coordination among the participating cache servers. Current IP configurations of mobile networks do not yet allow for this.

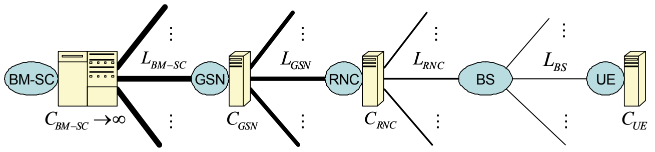

Fig. 1. Generic model for a caching architecture to deliver video streams.

II. The service architecture

Fig. 1 presents a generic model for a hierarchical caching architecture. It is based on a realistic UMTS network architecture for which we assume that caches can only be placed at easily accessible locations. In Fig. 1, $BM-SC$ represents the highest level of a caching hierarchy which acts as a MobileTV content data center; $GGSN/SGSN$ (referred to collectively as GPRS Support Node, GSN) represents the 2nd level at Regional Exchanges; RNC (Radio Network Controller) represents the 3rd level at Local Exchanges; the BS nodes are the $Base\ Stations$ which will not be equipped with a cache; and UE acts as User Equipment (i.e. Mobile Phone) at the lowest level of the hierarchy. C_i is the amount of cache disk space (in Mbits) at level i reserved for caching video streams, and L_i represents the capacity (Mbit/s) at level-i link available for the video streaming service, where $i \in \{BS, RNC, GSN, BM-SC\}$. C_{UE} represents the local cache space at an end-user, and L_{BS} represents the access data rate of an end-user, which should not be smaller than the video streaming rate v (Mbit/s). We use N_{RNC} and N_{GSN} to represent the number of branches connected to a RNC and to a GSN, respectively.

We only focus on the service of delivering video streams. The cache sizes and the bandwidth as discussed in the following sections are dedicated to this service.

III. Model with assumptions

A. Description

We consider streaming video applications that start from the content data center and *multicast* to all households connected to the multicast tree. The end-user has the option of zapping between videos and pausing *once* during a video. Therefore, each node must be able to store the video currently being watched in a cache connected to that node. The maximum time that can be recorded depends on the cache size and the bandwidth required to deliver the video at a certain Quality of Service. A user may of course pause more often if a recording function on the UE allows him to do so and the UE contains enough storage capacity. This does not add extra blocking to the CDN as we model it. If the user pauses more than once but the storage capacity left at UE is not enough for recording the delayed stream, he will automatically leave from the service.

After pausing, the user can resume the same video, or zap to a different video. In the first case, the video is resumed in *unicast* mode from the closest cache which *can* provide the delayed video. In the second case, the end-user zaps to a different video, which is received via multicast.

No bandwidth is reserved for multicast or unicast, so multicast flows and unicast flows share the same bandwidth capacity. Hence, if too many users pause and use unicast, there will probably be insufficient bandwidth left for new multicast requests. Similarly, if users all have very different tastes and too many video/MobileTV channels are transmitted via multicast, then a new multicast or unicast request may face blocking.

We assume that the arrival processes of MobileTV users are Poissonian and the user viewing time is exponentionally distributed, based on regular TV user behaviors [4].

We use the following definitions for our model:

- *Switch time*: duration that the user is watching a MobileTV channel before zapping to a different MobileTV channel. It is exponentially distributed with mean $1/\mu_s$.
- *View time before pausing*: the period an end-user watches the video stream without pausing. This time is exponentially distributed with mean $1/\mu_v$.
- *View time before leaving*: the period an end-user continuously watches the video stream before leaving (excluding the pause time). This time is exponentially distributed with mean $1/\mu_L$.
- *Pause time* t_p: the duration of the pause. It is exponentially distributed with mean $1/\mu_p$. Immediately after the pause, the end-user can have 3 options: 1) switch to another multicast video, with probability P_s; 2) turn off the MobileTV and leave the system, with probability P_L; or 3) resume the same video stream, with probability $1 - P_s - P_L$.

We use λ_{TV} to represent how many users, attached to one BS, turn on their MobileTV service every second; and N_{TV} represents the average number of registered MobileTV users, attached to one BS. A registered user of our video streaming service is always in one of the following states:

- "off" state: the user is off-line.
- "M" state: the user has started watching the video and the video is delivered via multicast.
- "$Pause$" state: the user is pausing.
- "U_{BM-SC}","U_{GSN}","U_{RNC}","U_{UE}" (unicast) states: the user continues to watch the video after having paused, and the stream is unicast from cache $BM - SC$, GSN, RNC or UE, respectively.

After pausing, whether the stream will be unicast from UE, RNC, GSN or $BM - SC$ depends on the cache sizes dedicated to the video at each level and how long the end-user paused the stream.

We define P_i with $i \in \{BM - SC, GSN, RNC, UE\}$, as the probability that the video stream is unicast from a cache at level i after a pause. $T_i = C_i/L_i$ is defined as the time capacity of the cache at i reserved for a video stream, measured in the number of seconds of video that may be recorded, with $L_{UE} = v$. L_i/v is the maximum number of video streams that can be multicast in parallel at level i. We assume that the cache at $BM - SC$ always contains a copy of the complete video, i.e. $T_{BM-SC} = \infty$, and $T_{BM-SC} > T_{GSN} > T_{RNC} > T_{UE}$.

The probability that the pause time $t_p < T_{UE}$ is equal to $1 - \exp(-\mu_p T_{UE})$; and the probability that $T_{UE} < t_p \leq T_{RNC}$ is equal to $\exp(-\mu_p T_{UE}) - \exp(-\mu_p T_{RNC})$. Hence, P_i can be expressed in terms of the probability distribution of the duration of a pausing period:

$$
\begin{aligned}
&P_{UE}\text{=}(1\text{-}P_s\text{-}P_L)(1\text{-}\exp(\text{-}\mu_p T_{UE}))\\
&P_{RNC}\text{=}(1\text{-}P_s\text{-}P_L)(\exp(\text{-}\mu_p T_{UE})\text{-}\exp(\text{-}\mu_p T_{RNC}))\\
&P_{GSN}\text{=}(1\text{-}P_s\text{-}P_L)(\exp(\text{-}\mu_p T_{RNC})\text{-}\exp(\text{-}\mu_p T_{GSN}))\\
&P_{BM-SC}\text{=}(1\text{-}P_s\text{-}P_L)(\exp(\text{-}\mu_p T_{GSN})\text{-}\exp(\text{-}\mu_p T_{BM-SC}))
\end{aligned}
$$

When $T_{BM-SC} \to \infty$, we obtain $\sum P_i = 1 - P_s - P_L$ with $i \in \{BM - SC, GSN, RNC, UE\}$.

B. Markov Analysis

Changing states per individual end-user can be modeled as a continuous-time Markov chain with states and transition rates as depicted in Fig. 2.

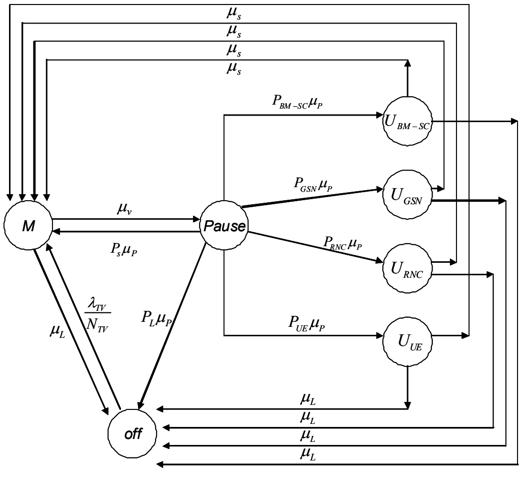

Fig. 2. Continuous-time Markov chain for state transitions.

The infinitesimal generator Q [5, pp.183] is explicitly given by Eq. (1), where the elements q_{xy} at row x and column y reflect a change from state x towards state y. We have $\sum_{y=1,x\neq y}^{7} q_{xy} = -q_{xx}$ [5, pp.181], where states 1 to 7 are respectively M, $Pause$, U_{BM-SC}, U_{GSN}, U_{RNC}, U_{UE}, and off.

$$Q = \begin{bmatrix} -(\mu_v+\mu_L) & \mu_v & 0 & 0 & 0 & 0 & \mu_L \\ P_s\mu_p & -\mu_p & P_{BM-SC}\mu_p & P_{GSN}\mu_p & P_{RNC}\mu_p & P_{UE}\mu_p & P_L\mu_p \\ \mu_s & 0 & -(\mu_s+\mu_L) & 0 & 0 & 0 & \mu_L \\ \mu_s & 0 & 0 & -(\mu_s+\mu_L) & 0 & 0 & \mu_L \\ \mu_s & 0 & 0 & 0 & -(\mu_s+\mu_L) & 0 & \mu_L \\ \mu_s & 0 & 0 & 0 & 0 & -(\mu_s+\mu_L) & \mu_L \\ \frac{\lambda_{TV}}{N_{TV}} & 0 & 0 & 0 & 0 & 0 & -\frac{\lambda_{TV}}{N_{TV}} \end{bmatrix} \tag{1}$$

We define a state vector $S(t)$: $\{M$, $Pause$, U_{BM-SC}, U_{GSN}, U_{RNC}, U_{UE}, $off\}$ with $\sum_{s=1}^{7} S_s(t) = 1$ (total law of probability), and with $\lim_{t\to\infty} S(t) = \pi$. Thus, the steady-state (row) vector π is a solution of $\pi Q = 0$. In the following, the components of π are indexed as π_M, π_{Pause}, $\pi_{U_{BM-SC}}$, $\pi_{U_{GSN}}$, $\pi_{U_{RNC}}$, $\pi_{U_{UE}}$, and π_{off}.

IV. Blocking

Our analysis assumes that user events are independent.

A. Definition of blocking probability $B(k)$

In steady state, $B(k)$ denotes *the probability that an individual user cannot get access to the service of his choice* when he requests a video channel k, or when he resumes it after pausing. We define $B_m(k)$ as the end-to-end (E2E) *multicast request blocking*, and $B_{u;i}$ as the E2E *unicast request blocking* if the end user retrieves the delayed video data from cache at level i after the pause. According to the total law of probability $\Pr[Blocking] = \sum_{s=1}^{7} \Pr[Blocking|state]\cdot\Pr[state]$, we have

$$\begin{aligned} B(k) &= [P_s B_m(k) + \sum_i P_i B_{u;i}]\cdot\pi_{Pause} \\ &\quad + B_m(k)\cdot(1-\pi_{Pause}) \end{aligned} \tag{2}$$

where $i \in \{BM-SC, GSN, RNC, UE\}$.

B. Computation of $B_m(k)$

To compute $B_m(k)$, we introduce two new probability functions $P(k)$ and $B_{Engset}(k)$. $P(k)$ is the probability that channel k is "on" and $B_{Engset}(k)$ is the probability that the link L_{RNC} is consumed by m multicast channels other than the requested channel k. The bandwidth bottleneck of the system is at L_{RNC}.

For our computation of $B_{Engset}(k)$ we refer to [4] with $\lambda_k = \lambda_{k\_L_{RNC}}$, $u_k = u_{k\_L_{RNC}}$ and $\mu_k = \mu_{k\_L_{RNC}}$ (in Eqs. 4 and 8), and m is the number of available channels for multicast, which can be expressed as $m = \max\{1, \frac{L_{RNC}}{v} - N_{TV}(\pi_{U_{BM-SC}} + \pi_{U_{GSN}} + \pi_{U_{RNC}})\}$. Then, we have

$$B_m(k) = (1-P(k))B_{Engset}(k) \tag{3}$$

where $P(k)$ and $B_{Engset}(k)$ can be deduced based on our methods discussed in [4]. Here a multicast user pausing the video is not considered as leaving in order to assure content availability. A user can only leave the multicast network in state $Pause$ and state M. After the pause, resuming the channel k from the local cache UE is not considered as leaving from the network. Hence, the leaving rate from the multicast channel k after the $Pause$ is the sum of the leaving rates to all states (except to state U_{UE}), which is equal to $(1-P_{UE})\mu_P$. The leaving rate from the multicast channel k after the state M is the sum of the leaving rates to state M and to state off (when switching to another channel and turning off the system), which is equal to $\mu_s + \mu_L$. Thus, the users' leaving rate u_k is equal to the number of channel k viewers at link L_{RNC} ($N_{TV}\alpha_k$) multiplied by the mean leaving rate of a multicast user ($\pi_M(\mu_s+\mu_L)+\pi_{Pause}(1-P_{UE})\mu_P$), where α_k represents the popularity of video channel k. In other words, an end-user chooses channel k with probability α_k. Similarly, we can compute the users' arrival rate λ_k based on Fig. 2:

$$\begin{aligned} \frac{\lambda_k}{N_{TV}} &= (\pi_M+\pi_{U_{BM-SC}}+\pi_{U_{GSN}}+\pi_{U_{RNC}}+\pi_{U_{UE}})\cdot \\ &\quad (1\text{-}\alpha_k)\mu_s\alpha_k + \pi_{Pause}(1\text{-}\alpha_k)P_s\mu_p\alpha_k + \pi_{off}\frac{\lambda_{TV}}{N_{TV}}\alpha_k \\ \frac{u_k}{N_{TV}} &= \pi_M\alpha_k(\mu_s+\mu_L)+\pi_{Pause}\alpha_k(1-P_{UE})\mu_P \end{aligned} \tag{4}$$

C. Computation of $B_{u;i}$

$B_{u;i}$ is defined as the probability that an arbitrary attempt to jump to the unicast state U_i is blocked because the amount of required bandwidth is not available at one of the links over the unicast connection. For instance, if a unicast user can retrieve data from cache $BM-SC$ without blocking ($B_{u;BM-SC} = 0$), that means the request cannot be blocked at all levels ($BM-SC$, GSN, and RNC). We assume that, if an end-user resumes the stream from his local cache C_{UE} after the pause, this attempt will never be blocked.

Based on the definition of $B_{u;i}$ and the assumption that the blocking at different levels are independent, we have

$$\begin{aligned} &B_{u;BM-SC} = 1 - \prod_{i\in\{BM-SC,GSN,RNC\}}(1-B_{u\_L_i}) \\ &B_{u;GSN} = 1 - \prod_{i\in\{GSN,RNC\}}(1-B_{u\_L_i}) \\ &B_{u;RNC} = B_{u\_L_{RNC}} \\ &B_{u;UE} = 0 \end{aligned} \tag{5}$$

where

$$B_{u\_L_i} = \sum_{j=1}^{\min\{K,\lfloor L_i/v\rfloor\}} \pi_{j\_L_i} B_{u\_L_i\_j} \tag{6}$$

$B_{u\_L_i}$ represents the mean probability that a unicast request is blocked at level i. $\pi_{j\_L_i}$ represents the probability that j

video streams are multicast at link L_i in steady state, and K represents the amount of available video streams. $B_{u\_L_i\_j}$ represents the blocking probability of a unicast request at link L_i when j video streams are multicast at link L_i.

1) Computation of $\pi_{j\_L_i}$: The probability that j positions are occupied by j MobileTV channels when there are K available video channels for this system can be deduced as follows. According to [5, pp.18] and [6], the binomial probability generating function of $\pi_{j\_L_i}$ is

$$\varphi(z) = \sum_{j=0}^{\infty} \pi_{j\_L_i} z^j = \prod_{k=1}^{K} q_{k\_L_i} + p_{k\_L_i} z \tag{7}$$

where we always have $j < \min\{K, \lfloor L_i/v \rfloor\}$, $p_{k\_L_i} = 1 - e^{-\frac{\lambda_{k\_L_i}}{\mu_{k\_L_i}}}$ and $q_{k\_L_i} = 1 - p_{k\_L_i}$ for a particular link L_i.

$\lambda_{k\_L_i}$ is the users' multicast requests arrival rate at video channel k (which is also the arrival rate of channel k) at link L_i. $\mu_{k\_L_i}$ is the leaving rate of the channel k from link L_i (which can be computed via an $M/G/\infty$ model), and $u_{k\_L_i}$ is the channel k users' leaving rate at link L_i.

When computing $u_{k\_L_{GSN}}$, state transitions from $Pause$ to U_{UE} and to U_{RNC} are not considered as leaving from link L_{GSN}. Similarly, we can also compute $u_{k\_L_{BM-SC}}$.

According to Fig. 1 and 2, we have $\lambda_{k\_L_{RNC}} = \lambda_k$, $u_{k\_L_{RNC}} = u_k$ (see (4)); and both $\lambda_{k\_L_i}$ and $u_{k\_L_i}$ at other levels ($BM-SC$ and GSN) can be expressed as

$$\begin{aligned}
\lambda_{k\_L_{GSN}} &= N_{RNC}\lambda_{k\_L_{RNC}} \\
\lambda_{k\_L_{BM-SC}} &= N_{RNC}N_{GSN}\lambda_{k\_L_{RNC}} \\
u_{k\_L_{GSN}} &= N_{RNC}N_{TV}\pi_M\alpha_k(\mu_s + \mu_L) + N_{RNC}N_{TV}\cdot \\
&\quad \pi_{Pause}\alpha_k(1 - P_{UE} - P_{RNC})\mu_P \\
u_{k\_L_{BM-SC}} &= N_{RNC}N_{GSN}N_{TV}\pi_M\alpha_k(\mu_s{+}\mu_L){+}N_{RNC}N_{GSN}\cdot \\
&\quad N_{TV}\pi_{Pause}\alpha_k(1 - P_{UE} - P_{RNC} - P_{GSN})\mu_P \\
\mu_{k\_L_i} &= \lambda_{k\_L_i}/\exp(\frac{\lambda_{k\_L_i}}{u_{k\_L_i}} - 1) \qquad (8)
\end{aligned}$$

2) Computation of $B_{u\_L_i\_j}$: We use $L_{i\_j}^{(unicast)} = L_i - jv$ to represent the bandwidth for unicast when there are j multicast videos at level i, which corresponds to $\lfloor L_{i\_j}^{(unicast)}/v \rfloor = n_{i\_j}$ available unicast servers at level i. We assume that this unicast service at level i can be modeled as an $M/M/n$ queuing system. Based on [5, pp. 277] we have

$$B_{u\_L_i\_j} = \Pr[N_s \geq n_{i\_j}] = \frac{\Pr[N_s = 0]}{n_{i\_j}!(1 - \frac{\lambda_i}{n_{i\_j}\beta_i})} \frac{\lambda_i^{n_{i\_j}}}{\beta_i^{n_{i\_j}}} \tag{9}$$

where

$$\frac{1}{\Pr[N_s = 0]} = \sum_{a=0}^{n_{i\_j}-1} \frac{\lambda_i^a}{a!\beta_i^a} + \frac{\lambda_i^{n_{i\_j}}}{n_{i\_j}!\beta_i^{n_{i\_j}}} \frac{1}{1 - \frac{\lambda_i}{n_{i\_j}!\beta_i}}$$

λ_i is the arrival rate of unicast requests at L_i, and β_i is the leaving rate of unicast transmissions at link L_i.

For the arrival rate λ_{RNC}, we know that the unicast request has to pass through link L_{RNC} no matter from which higher level ($BM-SC$, GSN, or RNC) the end user retrieves the delayed video after the pause. For the leaving rate β_{RNC}, we know that one unicast transmission at link L_{RNC} can be considered as leaving when a unicast stream from $BM-SC$ or GSN or RNC leaves. Similarly, λ_{GSN}, λ_{BM-SC}, β_{GSN} and β_{BM-SC} can also be computed.

$$\begin{aligned}
\lambda_{RNC} &= N_{TV}\pi_{Pause}\mu_P(P_{BM-SC} + P_{GSN} + P_{RNC}) \\
\lambda_{GSN} &= N_{RNC}N_{TV}\pi_{Pause}\mu_P(P_{BM-SC} + P_{GSN}) \\
\lambda_{BM-SC} &= N_{RNC}N_{GSN}N_{TV}\pi_{Pause}\mu_P P_{BM-SC} \\
\beta_{RNC} &= N_{TV}(\pi_{U_{RNC}}{+}\pi_{U_{GSN}}{+}\pi_{U_{BM-SC}})(\mu_s{+}\mu_L) \\
\beta_{GSN} &= N_{RNC}N_{TV}(\pi_{U_{GSN}}{+}\pi_{U_{BM-SC}})(\mu_s{+}\mu_L) \\
\beta_{BM-SC} &= N_{RNC}N_{GSN}N_{TV}\pi_{U_{BM-SC}}(\mu_s{+}\mu_L)
\end{aligned}$$

Finally, by substituting expression (7) for $\pi_{j\_L_i}$ and that of $B_{u\_L_i\_j}$ in (9) into (6) and further into (5), $B_{u;i}$ for different cases of i is found. Furthermore, we can compute its mean unicast request blocking probability, $E[B_{u;i}] = \sum_i P_i B_{u;i}$ where $i \in \{BM-SC, GSN, RNC, UE\}$.

V. Experiment

We focus on a typical UMTS MobileTV network architecture for which we have assumed parameter values as shown in Table I. The cache sizes are based on current smart phone specifications and reasonable capital expenditure for the network operator. The average pause time, which can be considered as the average call duration, is set to 107.1 seconds based on the measurement study in [7].

TABLE I
Parameter values in our case study

$v = 250$ Kbit/s (for 3.5G)	$K = 12$
$N_{TV} = 600$	$C_{BM-SC} = 10$ TBytes
$N_{RNC} = 250$	$C_{GSN} = 800$ GBytes
$N_{GSN} = 3$	$C_{RNC} = 20$ GBytes
$\lambda_{TV}/N_{TV} = 1/600$	$C_{UE} = 1$ GBytes
$1/\mu_p = 107.1$ seconds	$L_{RNC} = 2$ Mbit/s
$1/\mu_s = 1/\mu_v = 1/\mu_L = 300$ s	$L_{GSN} = 70$ Mbit/s
$P_s = P_L = 1/100$	$L_{BM-SC} = 155$ Mbit/s

In a MobileTV system users do not view a TV channel for a long time. Thus, the probability that a user wants to pause multiple times is low. The channel popularity distribution α_k for the channels is obtained from a Dutch market survey. Channel 1 has a popularity of 15.1% and channel 23 of 0.2%. The first 23 TV channels cover in total 94.1%. The remaining channels uniformly share a popularity of 5.9%.

Fig. 3 plots the overall blocking probability $B(k)$ as function of the channel index k, for different values of K and L_{RNC}. The results show that the less popular TV channels (higher channel index) have a slightly lower probability to be blocked than the more popular channels, because the user arrival rate decreases slower than the user leaving rate, with increasing channel index k. Furthermore, Fig. 3 indicates that for $K = 12$, a 2.5 Mbit/s bandwidth between the RNC and the Base Station is enough to support a MobileTV service with end-to-end blocking probability around 1% for all TV

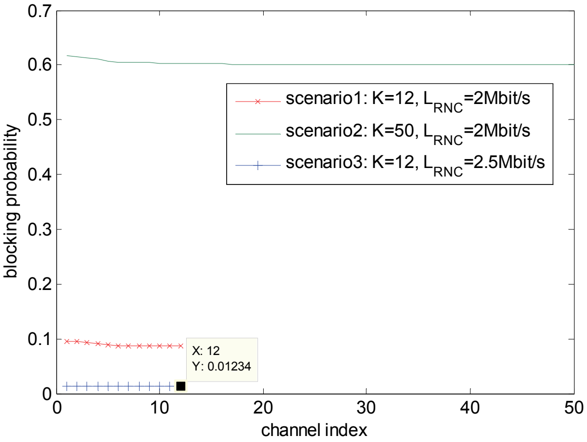

Fig. 3. The overall blocking probablity (2) as a function of the TV channel index k (channel 1 is the most popular TV channel), with $C_{UE} = 1$ GBytes.

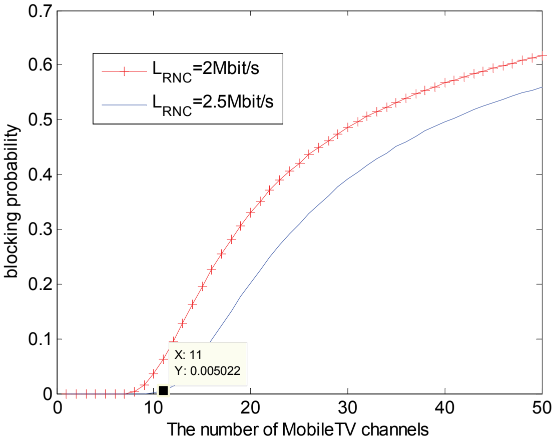

Fig. 4. The overall blocking probablity (2) as a function of the number of TV channels K in (6), for $k = 1$ and $C_{UE} = 1$ GBytes.

channel viewers. Without a caching architecture, L_{RNC} would dictate a $K < 10$, because only then $Kv < L_{RNC}$. Thus, this is where the advantage of caching starts playing a role. In addition, we can observe that the user's request is more likely to be blocked if there are more TV channels (larger K), and this factor significantly affects the overall quality. This is illustrated in Fig. 4, which illustrates the blocking probability as function of the number of available TV channels K for two different L_{RNC}. With the same configuration and parameter settings as shown in Table I, we also computed the mean unicast blocking probability $E[B_{u;i}]$ as a function of the local cache size at the end-user C_{UE}, ranging from 1 MB to 200 MB. We found that the mean unicast blocking probability decreases exponentially with the local cache size at the end user. Moreover, we observed that if an end-user wants to resume the video stream succesfully after the pause with probability $> 99\%$, his local cache size C_{UE} should be at least 16 MB, and a local cache size of ≥ 31 MB is enough to guarantee a unicast service availability of 99.99% after the pause. This easily matches current mobile phone specifications and the spare storage capacity might even support multiple pauses.

VI. Summary and Conclusions

In this paper, a new blocking model for a hierarchical caching multimedia delivery system is derived, inspired by related work on home gateway caching in [2]. Multicast and unicast request blocking have been analyzed separately. Our model allows to compute how many MobileTV channels can be supported for the overall blocking probability not to exceed a certain threshold. Furthermore, we can compute the required local cache size to assure that an end-user can successfully resume the streaming with a certain probability after the pause. Our results can be used to analyze the blocking in existing stream caching systems and to predict the system behavior of new service deployments.

Applying our model to a realistic MobileTV use case, we found that the number of available TV channels strongly affects the overall quality. Apparently, the less popular channels are still popular enough to end up with many unicast streams, and as such annihilate the efficiency advantage we hoped to create by multicasting the popular channels using a hierarchical caching architecture. When we increase the bandwidth between RNC and the $Base\ Station$ with 25%, the maximum amount of channels that could be supported increased with 38% (with the cache sizes given in Table I), and the advantage of having a CDN then becomes significant. Our example calculations further indicate that the local cache size requirements are easily matched by today's mobile phones.

Our model can also be adapted and dimensioned to scenarios with higher link capacity (such as foreseen for LTE networks) and more (HD)TV channels, and to random access operations.

References

[1] P. Rodriguez, C. Spanner, and E.W. Biersack, "Web caching architectures: hierarchical and distributed caching", 4th International Web Caching Workshop, San Diego, 1999.

[2] F.T.H. den Hartog, B.L.G. Bastiaans, M.A. Blom, M.G.M. Pluijmaekers, R.D. van der Mei, "The use of Residential Gateways in Content Delivery Networking", ATNAC, Sydney, Australia, December, 2004.

[3] 3GPP, MBMS Architecture and Functional Description, Technical Specification, TS 23.246, Release 6, June 2006.

[4] Y. Lu, F.A. Kuipers, M. Janic, and P. Van Mieghem, "E2E blocking probability of IPTV and P2PTV", IFIP Networking, Singapore, 2008.

[5] P. Van Mieghem, *Performance Analysis of communications Networks and Systems*, Cambridge University Press, 2006.

[6] J. Karvo, J. Virtamo, S. Aalto, O. Martikainen, "Blocking of dynamic multicast connections in a single link", IEEE BROADNETS, Stuttgart, Germany, 1998.

[7] J. Holub, J. G. Beerends, and R. Smid, "A Dependence between Average Call Duration and Voice Transmission Quality: Measurement and Applications", Wireless Telecommunications Symposium, Pomona, California, May, 2004.