# The use of Residential Gateways in Content Delivery Networking

F.T.H. den Hartog, B.L.G. Bastiaans, M.A. Blom, M.G.M. Pluijmaekers
TNO Telecom
P.O. Box 5050, 2600 GB Delft, The Netherlands
f.t.h.denhartog@telecom.tno.nl

R.D. van der Mei
Center for Mathematics and Computer Science
Kruislaan 413, 1098SJ Amsterdam, The Netherlands
Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081HV Amsterdam, The Netherlands

***Abstract*-We have investigated and compared various aspects of the storage of multimedia content at residential gateways (RG) in the home or at caches in the public network. From qualitative analyses of Personal Video Recording and Content Delivery Networking techniques, we conclude that service architectures incorporating storage on the RG as well as in the public network are viable options for the near future. We developed a performance model that calculates the availability of required bandwidth as a function of link bandwidths, cache sizes, access data rates, the number of users, the number of simultaneously multicast video streams, and various parameters that characterize content viewing and browsing behavior. The model is based on a realistic caching hierarchy, and is implemented in a tool that forms an excellent basis for the development of a Decision Support System for planning access networks in the context of rapidly evolving RG technology.**

## I. Introduction

In the coming years, more and more households will have access to a wide variety of communication services that demand high bandwidths and huge storage capacity in the networks involved. Examples are high-quality streaming, peer-to-peer multimedia entertainment and gaming. It is obvious that the commercial viability of these broadband services is greatly dependent on the user friendliness of the delivery, not just to the homes but all the way to the end-user equipment. For that purpose an intelligent telecommunication infrastructure will be required not only in the public domain but also within the home. It is expected that the Residential Gateway (RG) will play a crucial role. In this paper, the RG is defined as a device or a combination of devices that connects one or more access networks to one or more home networks and delivers services to the home environment [1]. Recent history shows that the required functionality and thus the complexity of the RG increases significantly with the diversity of the services delivered and with the bandwidth offered by the access network. However, any detailed vision on this increase in RG complexity and consequent technological development is still lacking. One of the issues concerns the role of the RG in a comprehensive network-, service-, and management architecture, and especially the consequences for local storage.

The discussion on where content could be stored best, at the home on an RG or in the public network, is complicated and confusing. Many business and regulation-related issues play a role, such as production costs, initial investment costs, applicable business models, privacy, and the distribution of tasks over the parties involved. But also technical constraints come into play: content protection (digital rights management), security, scalability, available bandwidth, Quality-of-Service (QoS), parallel recording ability, and the availability of Content Delivery Networking (CDN) techniques, such as IP Multicast and caching [2]. We have investigated the advantages and disadvantages of the storage of multimedia content at home or in the public network in various ways. This paper summarizes our main results. First, we look at a specific case of a service where local or network storage might be considered: Personal Video Recording (PVR). After describing the various CDN techniques that might be used in public networks as well as RGs, we then assess the relation between the technical properties of the storage infrastructure and the performance experienced by the end users. A performance model is laid out that calculates the availability of required bandwidth as a function of storage capacities, link bandwidths, number of users, etc., based on a hierarchical caching architecture including RGs. Also some simulation results are shown.

The work presented in this paper has been carried out in the collaborative project “Residential Gateway Environment” (www.rge.brabantbreedband.nl), which is partially funded by the Dutch Ministry of Economic Affairs.

## II. PVR at Home or in the Public Network

A PVR is an interactive recording device or function for television programs. Unlike the traditional video cassette recorder (VCR), it records the programs in a digital format (MPEG-1 or MPEG-2) on a hard disk. Currently, most PVR functionality is typically implemented in a high-end set-top-box type RG, equipped with a hard disk of sufficient capacity and digital recording hardware and software. In addition, a consumer normally subscribes to a service that provides an Electronic Program Guide (EPG) via an Internet connection. Here we will call this implementation a “Home PVR (HPVR)”, indicating that most functionality is present in equipment in the home. The PVR function can also be realized as a “network service”, in which case the content is stored on disk drives located in the public network. A consumer may have personal disk space on a network server that can be used for own recordings, or may have access to TV-programs that are stored by a service provider. We will call this implementation a “Network PVR (NPVR)”. In Fig. 1 the HPVR and NPVR architectures are schematically drawn.

The characteristics of HPVR and NPVR that we compared include the access network requirements, the customer

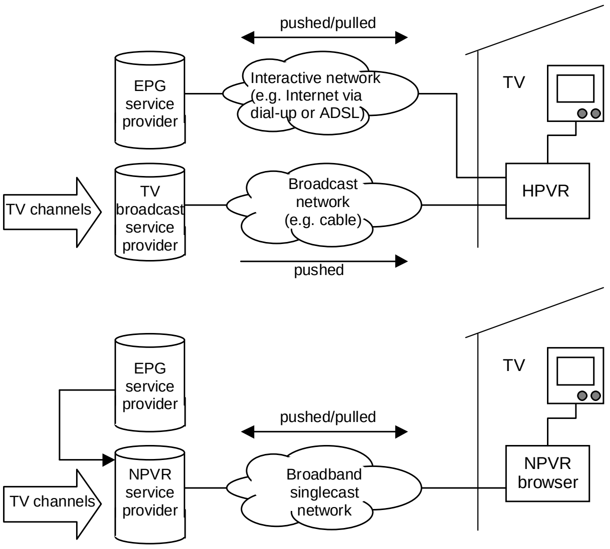


Fig. 1. Schematic overview of HPVR and NPVR setups.

premises equipment requirements, the parallel recording capabilities, the integrity of EPG meta data, the storage capacity, and the peer-to-peer networking functions. The results are described in [3] and are summarized below.

- The HPVR is a set-top-box or PC plug-in with one or two tuners and a hard disk drive with a capacity larger than 100 GB that can sustain only a limited number of simultaneous data streams to and from the disk. It also contains processing hardware for video coding/decoding and for the EPG control. The home equipment for the NPVR is a box or PC plug-in with a modem and processing hardware for video decoding only. The parallel recording capabilities for the NPVR have hardly any limitations, because resources are shared over a large number of subscribers, which keeps the relative costs low for providing extra channels.
- Integrity of metadata for the EPG is not guaranteed in the HPVR between successive daily downloads. The NPVR architecture results in higher integrity, because metadata resides in the server that has almost instantaneous access to these program data.
- The amount of storage sets a hard limit for HPVRs that will shift upwards with new generations of hard disk drives for new releases of boxes. In principle there is no storage capacity limitation for an NPVR.
- PVR users may want to share their content with others. For an HPVR it means that the content has to be transferred from one box to another. In the case of personal storage space being available on an NPVR, only the “pointer” to the content has to be exchanged.
- In the context of this paper, the most important difference between HPVR and NPVR is the type of connection to the home. An HPVR relies on the programs being offered on a broadcast medium (e.g. cable) combined with a narrowband interactive connection for the EPG. The NPVR, in contrast, needs a dedicated interactive broadband connection between the NPVR-server and the home. The penetration of broadband singlecast media is still limited, but growing rapidly [4].

It can be concluded that HPVRs and NPVRs will most probably coexist in the market for quite some time, since neither HPVRs nor NPVRs have compelling advantages or disadvantages. Service architectures incorporating storage in the public network as well as on the RG are therefore realistic options for the near future. This means that RGs might also obtain a significant role in balancing the load in a public network. The goal of the following sections is to analyze this role in more detail.

## III. Basic CDN Concepts and Architectures

Network- and server overload are serious threats to the success of broadband services like PVR. It might therefore be useful for RGs to support CDN techniques such as caching or IP Multicast, since RGs are part of the service delivery chain. Obviously, only the gateways with significant storage capacity and suitable application software can be used for local storage. In terms of the OSI-based ontology for the classification of RGs as laid out in [1], this concerns RGs of Type E and Type F only. The Type E or Type F RG will then act as a cache at the lowest level of a caching hierarchy. Two different basic concepts of caching techniques can be identified: explicitly configured proxy-caches and transparent proxy-caching with request interception and redirection. To understand how both mechanisms can be implemented in an RG, it is necessary to make a clear distinction between two separate functions that can reside in the RG: the switching and routing function on the one hand, and the storage or caching function on the other hand. The RG can therefore not be seen as a stand-alone cache, but more as an advanced switch/router combined with a cache, in one single physical device or distributed in the home network. This is depicted in Fig. 2 [5].

To store content on the RG, content requests must be directed to its cache. Explicit configuration of the RG as the proxy-cache for use by the end-user devices will cause content requests to be routed by the RG towards its own storage function in a straightforward manner. The RG will then act as a normal cache by serving the requested content to

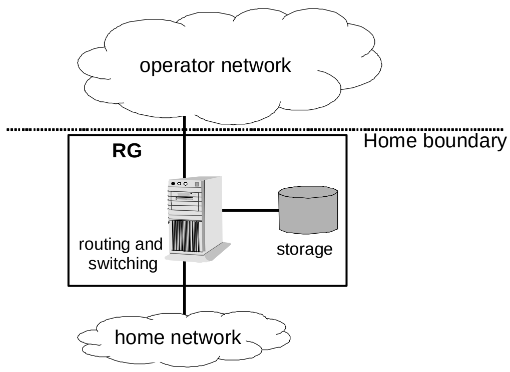


Fig. 2. Routing/switching and storage/caching function of RGs.

the end-user device, or by fetching the requested content from another cache or server first and then serving it to the end-user device. The cache can also be implemented in a way that is more transparent to the user. Of course, content requests should still be directed to the storage function of the RG for it to act as a forward proxy-cache. In this case, however, transparent interception and (re)direction of content requests is performed by the routing and switching functions of the RG.

Caching architectures can basically be divided into three categories [6]: hierarchical, distributed and hybrid architectures. In the case of a hierarchical caching architecture, caches are placed at multiple levels of the network. This is particularly beneficial when some cooperating cache servers do not have high-speed connectivity. In this case, popular objects can be efficiently diffused towards the demand. On the other hand, hierarchical caching has several drawbacks: (1) setting up a cache hierarchy requires caches to be placed at the key access points in the network, which requires significant coordination among the participating cache servers, (2) a caching hierarchy may introduce additional delays, (3) high-level caches may become performance bottlenecks, and (4) multiple copies of the same object may be stored at different cache levels, which is inefficient. In distributed caching systems there are only caches at the bottom level, and there are no intermediate caching levels. To decide from which cache to fetch an object, the caches then need to keep track of metadata information about the content of the caches. Consequently, with distributed caching most of the traffic flows through the low network levels, which are less congested, and no additional disk space is required from the intermediate levels. In a hybrid caching scheme, caches may cooperate with other caches at the same level or at a higher level using distributed caching.

For Web caching, Rodriguez et al. [7] performed a numerical analysis of hierarchical, distributed and hybrid caching. From this work it can be concluded that distributed caching achieves shorter transmission times (i.e. the time to send a document from the cache to the destination) than hierarchical caching, and has very good performance in well inter-connected areas without requiring intermediate cache levels. However, the deployment of distributed caching on a large scale encounters problems, such as large connection times (i.e. the time that a request travels to hit a document in the caching hierarchy), high bandwidth use and administrative issues. We therefore concentrated our work on the use of RGs in hierarchical caching architectures.

To eliminate unnecessary content multiplication in the network as much as possible, the use of multicast techniques such as IP Multicast [8] is strongly recommended for content distribution when a large number of end users view the same content at the same time. Examples are rugby matches (live and recorded), other live events (e.g. royal weddings, annual report of company results), rewinds of TV programs, scheduled movies and music videos, and so on. Whereas with proxy-caching, content multiplication takes place at the application layer, with IP Multicast the multiplication takes place at the IP layer. It enables sources to send one copy of certain content to a group address, reaching all receivers who have subscribed to that group. The content is distributed to the receivers through a so-called distribution tree, and the content is only copied onto branches that lead to network regions where receivers are present that have subscribed to that group. This way, the same content need not be carried over the shared part of the network multiple times, one time for each receiver, as is the case with unicast operation, consequently resulting in less server load. IP Multicast is also far more efficient than broadcasting, because broadcast content is copied onto every branch regardless of the presence of receivers. On the other hand, IP Multicast is not suited for on demand services such as video-on-demand.

Although the RG is regarded as the lowest level of the caching hierarchy, it might still act as the IP Multicast leaf-router to the end-user devices. In that case, it should support User Datagram Protocol (UDP) traffic, a host-to-router protocol such as the Internet Group Management Protocol (IGMP), and a router-to-router protocol such as Protocol Independent Multicast (PIM) [5].

## IV. Performance Model

### A. *Generic Network Model*

The performance model as laid out in this section is based on a caching architecture for which we assume that storage can only be placed at existing accessible locations, i.e. at locations where network equipment is already present. Fig. 3 describes a generic network architecture for both

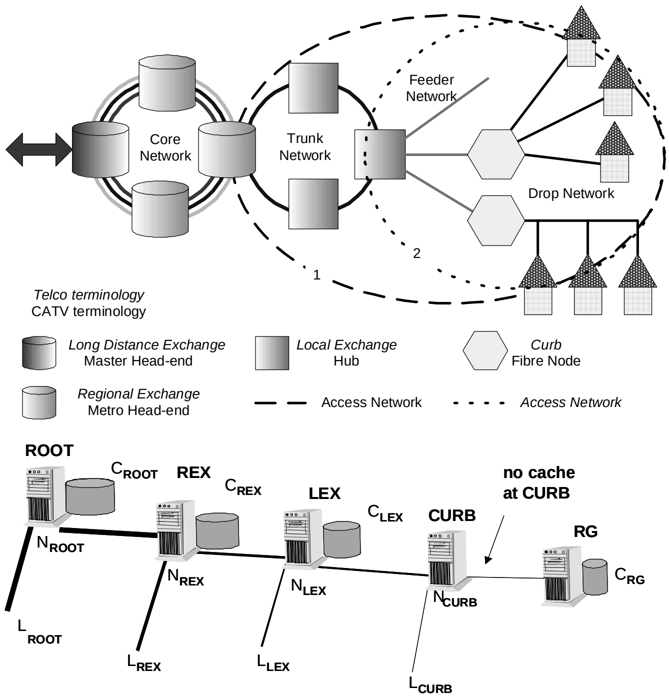


Fig. 3. Top: Generic network architecture [9]. Bottom: Generic model for the access and local storage architecture.

telecommunications and cable TV companies [9]. The model has been developed in such a way that on the one hand it covers the main factors that have an impact on performance, while on the other hand the model is simple enough to allow simple, fast but accurate performance predictions and to provide insight in the impact of the system parameters on the performance.

The households may be connected to the curb nodes via different access technologies. We consider three different classes of connectivity:

- *low bandwidth class*: copper-connected narrowband users with PSTN/ISDN connectivity (up to 128 kbps),
- *medium bandwidth class*: copper-connected broadband users with xDSL or cable connectivity (2 Mbps), and
- *high bandwidth class*: fiber-connected broadband users with LAN connectivity (10 Mbps).

We evaluate three different geographical scenarios by varying the mix of low, medium and high bandwidth users: countryside (where the majority of households have low-bandwidth connectivity), city district (with a mixture of bandwidths), and new estate (where most households have high-bandwidth connectivity). The mixture of access technologies is defined as follows: for $k\in\{low, medium, high\}$, $f_k$ is the fraction of households that have class-$k$ connectivity.

In this paper we focus on two applications: video streaming and WWW browsing. We assume that the available amount of cache disk space at each hierarchical level is strictly partitioned in a video part and a WWW part. The video part is further partitioned in per-class areas, whereas for WWW per-class caching seems to be less realistic and is therefore not considered. Define, for $i\in\{ROOT, REX, LEX, RG\}$, and for $k\in\{high, medium, low\}$,

- $C_i$ is the total amount of cache space at level $i$,
- $C_{i,k}^{(video)}$ is the amount of cache disk space reserved for caching class-$k$ video streams at level $i$,
- $C_i^{(WWW)}$ is the amount of cache disk space reserved for caching WWW traffic at level $i$.

*B. Model for Streaming Video*

We only consider streaming video applications that are initially started from the root node and *multicast* to all households connected to the tree (this excludes video-on-demand). The end user has the option of zapping between movies and pausing multiple times during a movie. Therefore, each node must be able to store the movie currently being watched in the cache connected to that node. Clearly, the maximum time that can be recorded depends on the cache size and the bandwidth required to broadcast the movie at a certain Quality-of-Service (QoS) level. After pausing, the user can resume the same movie, or zap to a different movie. In the first case, the movie is resumed in *unicast* mode from the most nearby cache that has stored the delayed movie. In the second case, the end user zaps to a different movie, which is received via multicast. For the user behavior model, we use the following definitions:

- The *switch time* is the length of time the user is watching a movie before zapping to a different movie. We assume it to be exponentially distributed with mean $1/\mu_s$.
- The *view time* is the period an end user watches a video stream without pausing. After the view time has expired, the end user pauses. We assume that this time is exponentially distributed with mean $1/\mu_v$.
- The *pause time* is the duration of the pause. We assume it to be exponentially distributed with mean $1/\mu_p$.

An end user is always in one of the following states:

- *multicast state* (“M”): the user has started watching the movie;
- *pausing state* (“P”): the user is pausing;
- *unicast states* (“$U_{ROOT}$”,…, “$U_{RG}$”): the user watches the movie after having paused, the stream is unicast from cache ROOT, REX, LEX and RG, respectively.

The bandwidth consumption of an end user for a given link in the tree model depends on the actual state of that user, which may change over time. In the M state the user receives the video signal via multicast, all the way from ROOT. In the P state the user receives no signal. In any of the unicast states bandwidth is only consumed on each of the downstream links between the cache and the end user.

The impact of the cache sizes is implicitly taken into account as follows: for $i\in\{ROOT, REX, LEX, RG\}$, $p_i$ is the probability that the movie is unicast from a cache at level $i$ after a pause. $p_0$ is the probability that the end user switches to a multicast movie immediately after a pause. The dynamics of changing states per individual end user can then be modeled as a continuous-time Markov chain (CTMC) with state space S := {M, P, $U_{ROOT}$, $U_{REX}$, $U_{LEX}$, $U_{RG}$}, where the transition rates are as depicted in Fig. 4.

For each class of end users, the values of $p_i$ depend on the cache sizes dedicated to video. Define, for $k\in\{low, medium, high\}$, $L_{CURB,k}$ as the access data rate of a class-$k$ customer. We assume that the bandwidth required by each user of class $k$ is equal to the access data rate. Moreover, define $T_{i,k}^{(video)}$ as the capacity of a cache $i$ reserved for class-$k$ video streams, measured in the number of seconds of video that may be recorded. Then, with $Z$ the number of movies that are multicast in parallel, $T_{i,k}^{(video)} = C_{i,k}^{(video)} / Z\, L_{CURB,k}$. For

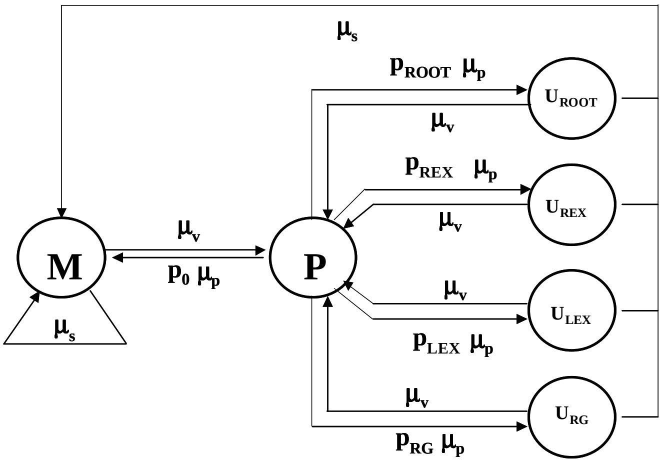


Fig. 4. Continuous-time Markov chain for state transitions of the end user watching and pausing streaming video.

ease of the discussion, we assume that for each level $i$ the video cache partitioning is proportional to the access data rates of the classes. We emphasize that this assumption is made to avoid unnecessary and superfluous notational conventions, and is not essential for the complexity of the analysis. Consequently $T_{i,k}^{(video)}$ is the same for each $k$. Hence, $p_i$ is also the same for each $k$, and can be expressed in terms of the probability distribution of the duration of a pausing period as follows. For $k \in \{high, medium, low\}$,

$$p_{RG} = (1 - p_0)(1 - \exp(-\mu_p T_{RG,k}^{(video)})),$$

$$p_{LEX} = (1 - p_0)(\exp(-\mu_p T_{RG,k}^{(video)}) - \exp(-\mu_p T_{LEX,k}^{(video)})),$$

$$p_{REX} = (1 - p_0)(\exp(-\mu_p T_{LEX,k}^{(video)}) - \exp(-\mu_p T_{REX,k}^{(video)})),$$

$$p_{ROOT} = (1 - p_0)(\exp(-\mu_p T_{REX,k}^{(video)}) - \exp(-\mu_p T_{ROOT,k}^{(video)})). \quad (1)$$

We assume that the cache at the root node always contains a copy of the complete movie, i.e. $C_{ROOT,k}^{(video)} = \infty$ and hence, $T_{ROOT,k}^{(video)} = \infty$ for each class $k$. The stationary probability $\pi_R$ that the system is in state R, for $R \in S$, can now be determined by solving the local balancing equations for the CTMC.

The performance of the access and local storage architecture depends on the aggregated behavior of the different end users. The main performance metric in this model is therefore the blocking probability, defined as the probability that an arbitrary attempt to jump to any unicast state is blocked because the amount of required bandwidth is not available at one of the links over the unicast connection. The user behavior model indicates that the amount of bandwidth needed by an end user may be increased when an end user jumps from the multicast state to a unicast state, or an end user jumps from a unicast state to a higher-level unicast state (requiring bandwidth on a higher-level link). Each of the $Z$ movies is multicast from the ROOT node at each of the three quality classes continuously, so that the capacity $L_i^{(unicast)}$ available for unicast traffic at a level-$i$ link (assuming no influence from WWW traffic) is equal to $L_i - Z(L_{CURB,low} + L_{CURB,medium} + L_{CURB,high}), i \in \{ROOT, REX, LEX\}$.

The per-link blocking probabilities for video, $q_{i,k}^{(video)}$, are defined as the probability that an arbitrary jump to a unicast mode of a class-$k$ connection is blocked on link $L_i$. They can be calculated using a blocking model with $n$ channels and with $K$ customer classes ($k$=1,…,$K$, with $K$=3 in this paper) as shown in Fig. 5.

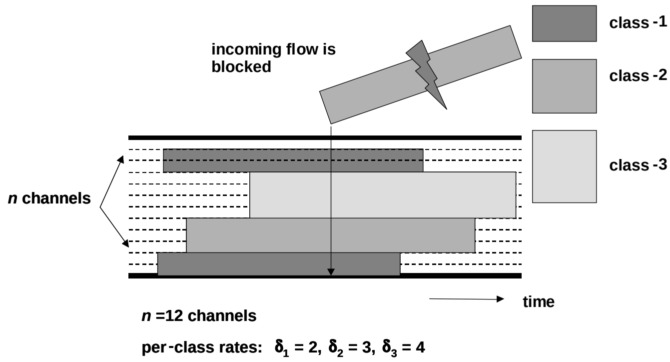


Fig. 5. Illustration of the multi-class blocking model.

Class-$k$ flows arrive according to an independent Poisson arrival process with rate $\lambda_k$, and require $\delta_k$ channels simultaneously, for the average duration of $\beta_k$ time units. If the number of available channels is less than $\delta_k$ upon arrival of a class-$k$ customer, then that flow is blocked. For the determination of $\delta_k$, it is assumed that the access link rates $L_{i,k}$ can (approximately) be expressed as integer multiples of some common rate $r$ (e.g. 64 kbps). This results in $\delta_{i,k} = L_{CURB,k} / r$. The arrival rate $\lambda_{i,k}$ can be calculated by realizing that a class-$k$ link at LEX level transports the combined traffic of $N_{CURB}$ households, out of which a fraction $f_k\alpha_{video}$ is actively watching video (with $\alpha_{video}$ the fraction of households that are watching video during the busy hour). A customer only generates a new flow (i.e., an arrival) when it makes a jump in the underlying CTMC from state P to state $U_{ROOT}$, $U_{REX}$ or $U_{LEX}$. Hence, it is easy to verify that, for $k \in \{high, medium, low\}$, and with similar reasoning for the higher links,

$$\lambda_{LEX,k} = \alpha_{video} N_{CURB} f_k \pi_P \mu_p (p_{ROOT} + p_{REX} + p_{LEX}),$$

$$\lambda_{REX,k} = \alpha_{video} N_{CURB} N_{LEX} f_k \pi_P \mu_p (p_{ROOT} + p_{REX}),$$

$$\lambda_{ROOT,k} = \alpha_{video} N_{CURB} N_{LEX} N_{REX} f_k \pi_P \mu_p p_{ROOT} . \quad (2)$$

In this model each of the branches of the network depcted in Fig. 3 is assumed has the same arrival process. We emphasize that this assumption is made for ease of the discussion, but does not complicate the analysis of the model. The “holding time” of a flow is ended when the underlying CTMC jumps from any unicast state to either state M or P. Hence, for $k \in \{high, medium, low\}$, the mean holding times are given by $\beta_{NRC,k} = 1 / (\mu_s + \mu_v)$ (see Fig. 4). $\zeta(l)$ denotes the steady-state probability that $l$ “channels” are occupied. $\zeta(l)$, $l$ = 0,1, …,$n$, can be calculated from $\lambda_k$, $\delta_k$, and $\beta_k$ with the Kaufmann-Roberts recursion [10,11]. The blocking probability $q_k$ of flows of class $k$ can then be expressed in terms of the variables $\zeta(l)$, $l = 0,1, \ldots,n$, as follows: for $k=1,\ldots,K$,

$$q_k = \sum_{l=n-\delta_k+1}^{n} \zeta(l) \quad (3)$$

### C. Model for WWW browsing

WWW browsing performance is known to be a highly complex interplay between many factors, such as the maximum window size, the network round-trip time, TCP slow start, congestion avoidance, the maximum siegment size, the HTTP version, amongst many others (cf., e.g., [14] for details). This makes a detailed performance analysis for WWW browsing extremely complicated. For network planning purposes, however, such detailed packet-level models are simplified by considering flow-level performance models (cf., e.g., [15]). To this end, we assume that each user alternates between two states: ON-periods, during which the end user is downloading a Web document, and OFF-periods, during which the end user is reading the downloaded Web document (see Fig. 6). We adopt the connectivity classes from the video model. Let $\tau^{(k)}_{ON}$ and $\tau^{(k)}_{OFF}$ be the mean durations of the ON- and OFF-periods for class-$k$ customers, respectively. The download rate of class-$k$ customers is assumed to be $L_{CURB,k}$, for $k\in\{low, medium, high\}$. Let $B_D$ denote the mean document size, and $t_R$ denote the mean read time. We expect that the read time is independent of the access data rate. Then, for $k\in\{high, medium, low\}$, $\tau^{(k)}_{ON} = B_D / L_{CURB,k}$ and $\tau^{(k)}_{OFF} = t_R$. Define $\lambda_k^{(WWW)}$ as the average arrival rate of WWW requests per individual user for $k\in\{high, medium, low\}$. Then $\lambda_k^{(WWW)} = 1 / (\tau^{(k)}_{ON} + \tau^{(k)}_{OFF})$.

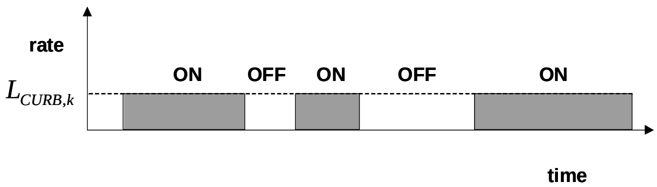


Fig. 6. User behavior model for Web browsing.

Similar to the video model, we consider an activity factor $\alpha_{WWW}$ representing the fraction of users that are active during a certain busy hour.

The main performance metric is the probability $q_k^{(WWW)}$ that an arbitrary class-$k$ flow is blocked because insufficient bandwidth is available. To calculate $q_k^{(WWW)}$, we consider two types of Web objects: static objects that may be cached, and dynamic objects that are not cacheable (e.g. personalized data). $g_{static}$ denotes the fraction of objects that is static, and $g_{dynamic}$ the fraction of objects that is dynamic. To assess the bandwidth use generated by an individual end user, consider an arbitrary static Web document requested by an end user. It is first checked whether the document is cached at the RG cache. If not, then it is checked whether the document is cached at the LEX cache, and if not so, the REX cache is checked. Therefore, for $i\in\{RG, LEX, REX\}$, we define $\varphi_i^{(WWW)}$ as the probability that an arbitrary static Web document is cached at level $i$, and for $i\in\{RG, LEX, REX, ROOT\}$, $\varphi_i$ is the fraction of requests for static Web documents that is not cached at any levels lower than $i$. The probabilities $\varphi_i$ determine the amount of traffic at the different hierarchical links, and hence, have an impact on the blocking probabilities. They can be quantified as:

$$\begin{aligned}
\varphi_{RG} &= 1,\\
\varphi_{LEX} &= 1 - \varphi_{RG}^{(cache)},\\
\varphi_{REX} &= (1 - \varphi_{LEX}^{(cache)})(1 - \varphi_{RG}^{(cache)}),\\
\varphi_{ROOT} &= (1 - \varphi_{REX}^{(cache)})(1 - \varphi_{LEX}^{(cache)})(1 - \varphi_{RG}^{(cache)}). \quad (4)
\end{aligned}$$

In general, the effectiveness of a cache depends on the local caching policy that is implemented. Typical examples are classical Least Recently Used and Least Frequently Used caching policies. Evidently, the effectiveness of a cache also depends on the cache size. In that context, interesting results have been found by Arlitt et al. [12], who analyzed the cache efficiency as a function of the cache size for different local caching policies. The results show that in general, the effectiveness of a cache depends only weakly on the choice of the local caching policies, unless a highly sub-optimal local policy is chosen. Based on these observations, we approximate the cache hit rate for static objects as a function of the cache size with the curve of Fig. 7.

To analyze the amount of traffic generated at each of the hierarchical links, we define $h_i$, with $i\in\{LEX, REX, ROOT\}$, as the fraction of requests for Web documents that is downloaded over link $L_i$. Then it can be readily verified that:

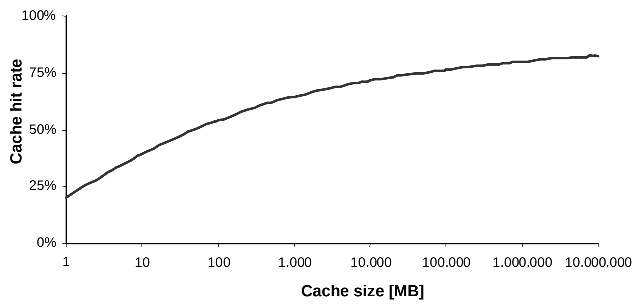


Fig. 7. Approximation of the cache hit rate for static objects as a function of the cache size.

$$\begin{aligned}
h_{LEX} &= g_{dynamic} + g_{static}\cdot\varphi_{LEX},\\
h_{REX} &= g_{dynamic} + g_{static}\cdot\varphi_{REX},\\
h_{ROOT} &= g_{dynamic} + g_{static}\cdot\varphi_{ROOT}. \quad (5)
\end{aligned}$$

The available amount of bandwidth at each link is shared between video traffic and WWW traffic. In case no priority scheme is implemented, the blocking probabilities $q_k^{(WWW)}$ at each link can be calculated by applying the Kaufmann-

Roberts recursion with $K$=6 classes (three video classes, and three WWW classes). The numbers of "channels" per link are similar to those discussed in the video model. Using similar arguments, the arrival rates of each of the three WWW quality classes is, for $k \in \{low, medium, high\}$,

$$\lambda_{LEX,k} = \lambda_k^{(WWW)} \alpha_{WWW} N_{CURB} f_k h_{LEX},$$
$$\lambda_{REX,k} = \lambda_k^{(WWW)} \alpha_{WWW} N_{CURB} N_{LEX} f_k h_{REX},$$
$$\lambda_{ROOT,k} = \lambda_k^{(WWW)} \alpha_{WWW} N_{CURB} N_{LEX} N_{REX} f_k h_{ROOT}. \quad (6)$$

The holding time is, for $k \in \{low, medium, high\}$, equal to $\tau^{(k)}_{ON}$, whereas the number of "channels" required by each of the class-$k$ flows is similar to those in the video model.

We also considered video traffic having strict priority over WWW traffic: WWW traffic can only use bandwidth not used by video traffic. Then, the WWW blocking probability is calculated by weighing the blocking probabilities per channel with the distribution of channels available for WWW. The latter follows directly from the steady-state probability distribution of the number of channels occupied by video.

Finally, off-line batch processing traffic such as email has been taken into account. Its average throughput can be easily calculated (results not shown), assuming that only video and WWW browsing make use of the caching infrastructure, and that the off-line batch processing services always have lowest priority.

## V. Simulation

We have developed a tool in Microsoft Excel / Visual Basic, called Cache Capacity Calculator (C3), which calculates blocking probabilities based on the model described in the previous section. The input parameters are:

- Access rates corresponding to the class-$k$ households (taken from section IV.A)
- Fractions of the households that have a class-$k$ connectivity
- Bandwidths $L_1$, $L_2$ and $L_3$ of the links between ROOT and REX, REX and LEX, and LEX and CURB, respectively
- Number of branches connected to each node
- Cache sizes for video and WWW for each node
- Total number of users connected to the ROOT node
- Fraction of the total number of users that is watching a video or browsing a website during busy hours
- Video pause time, view time and switch time
- Number of simultaneous multicast video-streams Z
- Average size of web pages and customer reading time
- A toggle for setting the priority for video streams.

For the links $L_1$, $L_2$ and $L_3$, the blocking probability $q_{total,k}$ for video and WWW are approximated for each of the three classes. Assuming that the per-link blocking probabilities are independent, $q_{total,k}$ can be expressed as:

$$q_{total,k} = 1 - (1 - q_{L1,k})(1 - q_{L2,k})(1 - q_{L3,k}) \quad (7)$$

As an illustration of the capabilities of C3, two examples are presented here. Fig. 8 shows the effect of the number of users in the network on the overall blocking probabilities for video and Web browsing for the three connectivity classes. The input parameters used in the simulation are:

- 60% of the households have ISDN/PSTN connectivity, 30% ADSL/Cable and 10% fiber.
- The bandwidths for the other links: $L_1$ = 155 Mbps, $L_2$ = 155 Mbps, and $L_3$ = 100 Mbps.
- The ROOT has 13 branches, the REX 100, the LEX 30, and the CURB 50.
- Both the video cache sizes and the web caches are: $C_{ROOT}$ = 1 TB, $C_{REX}$ = 10 GB, $C_{LEX}$ = 5GB, $C_{RG}$ = 128 MB.
- 5% of the users are watching video and 7% are browsing.
- The video watching behavior parameters: $1/\mu_s$ = 1800 s, $1/\mu_v$ = 1800 s, $1/\mu_p$ = 300 s, $p_0$ = 1%.
- The number of simultaneously multicast movies: $Z$ = 5.
- Average size of web page $B_D$ = 150 kB, read time $t_R$ = 20 s.
- Priority for video is set.

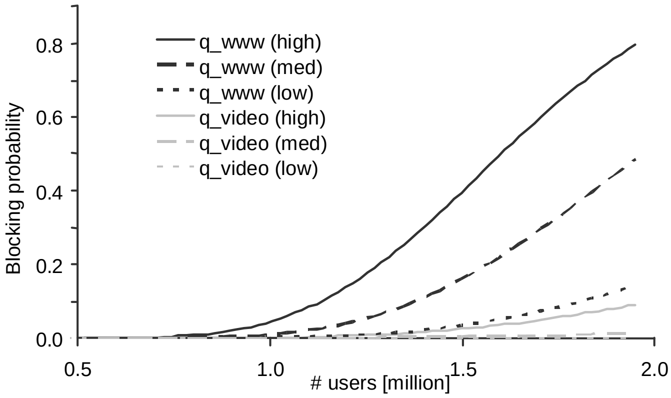


Fig. 8. Blocking probabilities as a function of the number of users in the network.

From the figure it can be seen that a larger number of users will result in a higher link use and therefore a higher blocking probability. Because of the priority for video, the blocking probability for WWW is larger than for video for each of the classes. When a video stream in a high connectivity class is blocked, there still might be some space left for a medium or low user in the remaining bandwidth. That is why the blocking probability is always lower for lower-class users. The maximum allowed number of users given a certain blocking probability can be read from the figure. For example, for WWW class high and a maximum allowed blocking probability of 10%, the maximum number of users on the network is about 1.2 million.

Fig. 9 shows the blocking probabilities for the different classes as a function of RG cache size. The total number of users is 2 million. The other parameters are the same as for Fig. 8, except for the priority for video, which is not set. The simulation shows that the overall blocking probability decreases for increasing RG cache size. Increasing the RG cache size from 0 to $C_{REX}$ relieves the bottleneck in $L_1$ because of a higher hit rate for Web pages in the RG. When the RG cache size is increased beyond $C_{REX}$, this results in part of the data that was cached in the ROOT now being cached on the RG. Therefore, the traffic at $L_1$ decreases and the overall blocking probability decreases until $L_1$ does not contribute to the over-all blocking probability anymore.

## VI. Conclusions

We have investigated various aspects of the storage of multimedia content both at home and in the public network. More specifically, we looked at the influence of local web- and video caching at the residential gateway on Quality-of-Service in relation to network parameters and user behavior. The services we studied were streaming video, Web browsing, and off-line batch processing such as e-mail.

We first performed a qualitative analysis of the currently developed and available solutions for Personal Video Recording and the Content Delivery Networking techniques suitable for RGs. It can be concluded that both Home PVRs and Network PVRs have advantages and disadvantages,

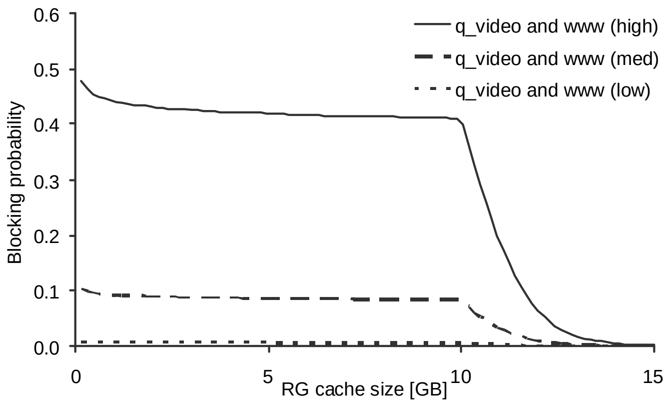


Fig. 9. Blocking probabilities as a function of $C_{RG}$

and will most probably coexist in the market. Service architectures incorporating storage in the public network as well as on the RG are therefore realistic options for the near future. RGs then should include a switching/routing function as well as a storage or caching function in order to perform as an explicitly configured proxy-cache or a transparent proxy-cache with request interception and redirection at the lowest level of a caching hierarchy. In the case that the RG supports IP Multicast, it can also act as the leaf-router to the end-user devices.

We have developed a performance model that calculates the probability that required bandwidth is available as a function of link bandwidths, cache sizes, access data rates, the number of users, the number of simultaneously multicast video streams, and various parameters that characterize watching and browsing behavior. The model is based on a realistic caching hierarchy. As far as we know, this is the first time that a performance model for a caching hierarchy has been developed that focuses on the blocking probability as the main performance metric, takes into account various multimedia services, and includes the RG. How the blocking probability translates to user-perceived QoS is a subject of further study. Other interesting extensions of the model would be the inclusion of peer-to-peer applications, the possibility of switching connectivity class, proactive caching, and applications that accelerate video streaming. A good overview of caching strategies suitable for streaming media is given in [13].

The algorithms have been implemented in a user-friendly tool, called C3, that can be used by network planners to answer "what-if" questions regarding capacity and performance in anticipated or hypothetical scenarios. As such, the tool forms an excellent basis for the development of a Decision Support System for planning access networks and local storage in the context of the rapidly evolving RGs.

## REFERENCES


[1] F.T.H. den Hartog, M. Balm, C.M. de Jong, and J.J.B. Kwaaitaal, "Convergence of Residential Gateway technology", *IEEE Commun. Mag.*, vol. 42, pp. 138-143, May 2004.

[2] F.T.H. den Hartog, N.H.G. Baken, D.V. Keyson, J.J.B. Kwaaitaal, and W.A.M. Snijders, "Tackling the complexity of Residential Gateways in an unbundling value chain", in *Proc. International Symposium on Services and Local Access (ISSLS) 2004*, Edinburgh, 2004.

[3] M.G.M. Pluijmaekers, R. Mies, and W.A.M. Snijders, "Personal Video Recorder: home or network function?", Deliverable D4.1; TSIT 1021 Residential Gateway Environment, 2003.

[4] M. Balm et al.,"Impact of access networks migration on Residential Gateways", Deliverable D2.2; TSIT 1021 Residential Gateway Environment, 2003.

[5] M. Pluijmaekers, "Content Delivery Networking techniques and the Residential Gateway", Deliverable D4.2; TSIT 1021 Residential Gateway Environment, 2004.

[6] J. Wang, "A survey of Web caching schemes for the Internet", *ACM SIGCOMM Computer Communication review*, vol. 29, pp. 36-46, October 1999.

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

[8] S. Deering, "Host extensions for IP Multicasting", IETF RFC 1112, August 1989.

[9] W.A.M. Snijders, H.P.A. van den Boom, P. van Wijk, and C.M. de Jong, "Access in the Netherlands in 2002", Deliverable D2.1; TSIT 1021 Residential Gateway Environment, 2002.

[10] J.S. Kaufman, "Blocking in a shared resource environment", *IEEE Transactions on Communications*, vol. 29, pp. 1474-1481, 1981.

[11] J.W. Roberts, "A service system with heterogeneous user requirements", *Performance of Data Communication Systems and their Applications*, pp. 423-431, 1981.

[12] M. Arlitt, R. Friedrich, T. Jin, "Performance evaluation of Web proxy cache replacement policies", *Performance Evaluation*, vol. 39, pp. 149-164, 2000.

[13] J. Liu and J. Xu, "Proxy caching for media streaming over the Internet", *IEEE Commun. Mag.*, vol. 42, pp. 88-94, August 2004.

[14] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. Computer Communication Review, 28(4), pp. 303-314, October 1998.

[15] S. Ben Fredj et al. (2001). Statistical bandwidth sharing: a study of congetsion at flow level. Proceedings ACM Sigcomm, pp. 111-122.