IP Anycast Architecture

Akash K Singh, PhD
IBM Corporation
Sacramento, USA

Abstract—This paper illustrates the methodology and architecture for network addressing and routing in which datagram packets routed through mathematical topological nearest node in a cluster of potential receivers that are being identified by equivalent destination address space. Mathematical framework is proposed to improve the Anycast usage.

Keywords- IP Anycast, Multicast, Mobile IPV6, Addressing, Routing

I. INTRODUCTION

IP Anycast has many attractive features for any service that involve the replication of multiple instances across the Internet. IP Anycast allows multiple instances of the same service to be "naturally" discovered, and requests for this service to be delivered to the closest instance. However, while briefly considered as an enabler for content delivery networks (CDNs) when they first emerged, IP Anycast was deemed infeasible in that environment. The main reasons for this decision were the lack of load awareness of IP Anycast and unwanted side effects of Internet routing changes on the IP Anycast mechanism. In this article we re-evaluate IP Anycast for CDNs by proposing a load-aware IP Anycast CDN architecture. Our architecture is prompted by recent developments in route control technology, as well as better understanding of the behavior of IP Anycast in operational settings. Our architecture makes use of route control mechanisms to take server and network load into account to realize load-aware Anycast. We show that the resulting redirection requirements can be formulated as a Generalized Assignment Problem and present practical algorithms that address these requirements while at the same time limiting connection disruptions that plague regular IP Anycast. We evaluate our algorithms through trace based simulation using traces obtained from a production CDN network.

Fig 1.0 Global deployment

II. ARCHITECTURE

We propose to implement anycast free of all the previously discussed limitations by means of address-translation capabilities provided by the Mobile IPv6 protocol. These capabilities have originally been introduced to enable communication with mobile nodes while they move among various networks. However, we demonstrate that one can also exploit these capabilities to implement anycasting. The general idea is to present an anycast group to its clients as a single mobile node. The anycast functionality is then implemented by informing each client that this (fictitious) mobile node has moved to the location of the actual anycast node the client is going to communicate with. Similar to what happens in mobile environments, announcing the movement causes the client to redirect all its traffic targeting the mobile node to the new location while keeping the movement transparent to the client applications. This effectively enables the anycast nodes to jointly service their clients via a single anycast address. The following section discusses some basic aspects of Mobile IPv6, which is the standard protocol designed for mobile communication. Then, we show how selected functions of Mobile IPv6 can be used to implement versatile anycast.

A. Mobile IPv6

Mobile IPv6 (MIPv6) consists of a set of extensions to the IPv6 protocol [19]. MIPv6 has been proposed to enable any *IPv6 mobile node* (MN) to be reached by any other *correspondent node* (CN), even if the MN is temporarily away from its usual location. MIPv6 assumes that each MN belongs to one home network, which contains at least one MIPv6-enabled router capable of serving as a *home agent* (HA). Such an HA acts as a representative for the MN while it is away. To allow one to reach an MN while it is away from home and connected to some visited network, MIPv6 distinguishes between two types of addresses that are assigned to MNs. The *home address* identifies an MN in its home network and never changes. An MN can always be reached at its home address. An MN can also have a *care-of address*, which is obtained from a visited network when the MN moves to that network. The care-of address represents the current physical network attachment of the MN and can change as the MN moves among various networks. The MN reports all its care-of addresses to it's HA. The goal of MIPv6 is to ensure uninterrupted communication with MNs via their home addresses and independently of their current

network attachment. To this end, MIPv6 provides two mechanisms to communicate with MNs that are away from home. The first mechanism is *tunneling*, by which the HA transparently tunnels the traffic targeting the home address of an MN to the care-of address of that node (see Figure 1a). The advantage of tunneling is that it is totally transparent to the CNs. Hence, no MIPv6 support is required from any node other than the MN and its HA. However, tunneling can also lead to two problems. First, if many MNs from the same home network are away, then their shared HA can become a bottleneck. Also, if the distance between an MN and its home network is large, then tunneling can introduce significant communication latency. These two problems are addressed by the second MIPv6 communication mechanism, called *route optimization*. It enables an MN to reveal its care-of address to any CN to allow direct communication (see Figure 1b). Revealing the care-of address causes the CN to create a translation binding between the home- and care-of addresses of an MN. The binding allows the CN to translate between home- and care-of address in the incoming and outgoing traffic, which enables the CN to communicate with the MN directly at its care-of address. This eliminates the latency introduced by tunneling, and offloads the HA. Route optimization is slightly less transparent than tunneling, as the IP layer at the CN is aware of the current physical attachment of the MN. However, that information is confined inside the IP layer, which effectively hides careof addresses from higher-level protocols such as TCP and UDP. As a consequence, these protocols use only the home address of an MN and the changes in the MN's location remain transparent to applications running on CNs.

We consider the following anycast field equations defined over an open bounded piece of network and /or feature space $\Omega \subset R^d$. They describe the dynamics of the mean anycast of each of p node populations.

$$\begin{cases} (\frac{d}{dt}+l_i)V_i(t,r) = \sum_{j=1}^{p}\int_\Omega J_{ij}(r,\bar{r})S[(V_j(t-\tau_{ij}(r,\bar{r}),\bar{r})-h_{|j})]d\bar{r} \\ \qquad\qquad + I_i^{ext}(r,t), \qquad t \geq 0, 1 \leq i \leq p, \\ V_i(t,r) = \phi_i(t,r) \qquad\qquad t \in [-T,0] \end{cases} \tag{1}$$

We give an interpretation of the various parameters and functions that appear in (1), Ω is finite piece of nodes and/or feature space and is represented as an open bounded set of R^d . The vector r and $\bar{r}$ represent points in Ω. The function $S: R \to (0,1)$ is the normalized sigmoid function:

$$S(z) = \frac{1}{1+e^{-z}} \tag{2}$$

It describes the relation between the input rate v_i of population i as a function of the packets potential, for example, $V_i = v_i = S[\sigma_i(V_i - h_i)]$. We note V the $p-$ dimensional vector $(V_1,...,V_p)$. The p function $\phi_i, i = 1,..., p,$ represent the initial conditions, see below. We note ϕ the $p-$ dimensional vector $(\phi_1,...,\phi_p)$. The p function $I_i^{ext}, i = 1,..., p,$ represent external factors from other network areas. We note I^{ext} the $p-$ dimensional vector $(I_1^{ext},...,I_p^{ext})$. The $p \times p$ matrix of functions $J = \{J_{ij}\}_{i,j=1,...,p}$ represents the connectivity between populations i and j, see below. The p real values $h_i, i = 1,..., p,$ determine the threshold of activity for each population, that is, the value of the nodes potential corresponding to 50% of the maximal activity. The p real positive values $\sigma_i, i = 1,..., p,$ determine the slopes of the sigmoids at the origin. Finally the p real positive values $l_i, i = 1,..., p,$ determine the speed at which each anycast node potential decreases exponentially toward its real value. We also introduce the function $S: R^p \to R^p,$ defined by $S(x) = [S(\sigma_1(x_1 - h_1)),..., S(\sigma_p - h_p))],$ and the diagonal $p \times p$ matrix $L_0 = diag(l_1,..., l_p)$. Is the intrinsic dynamics of the population given by the linear response of data transfer. $(\frac{d}{dt}+l_i)$ is replaced by $(\frac{d}{dt}+l_i)^2$ to use the alpha function response. We use $(\frac{d}{dt}+l_i)$ for simplicity although our analysis applies to more general intrinsic dynamics. For the sake of generality, the propagation delays are not assumed to be identical for all populations, hence they are described by a matrix $\tau(r,\bar{r})$ whose element $\tau_{ij}(r,\bar{r})$ is the propagation delay between population j at $\bar{r}$ and population i at r. The reason for this assumption is that it is still unclear from anycast if propagation delays are independent of the populations. We assume for technical reasons that τ is continuous, that is $\tau \in C^0(\bar{\Omega}^2, R_+^{p \times p})$. Moreover packet data indicate that τ is not a symmetric function i.e., $\tau_{ij}(r,\bar{r}) \neq \tau_{ij}(\bar{r},r),$ thus no assumption is made about this symmetry unless

otherwise stated. In order to compute the righthand side of (1), we need to know the node potential factor V on interval $[-T,0]$. The value of T is obtained by considering the maximal delay:

$$\tau_m = \max_{i,j(r,\bar{r}\in\Omega\times\Omega)} \tau_{i,j}(r,\bar{r}) \qquad (3)$$

Hence we choose $T = \tau_m$

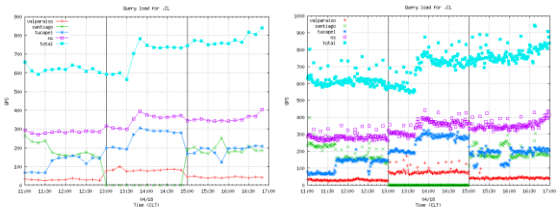

Fig 1.1 The query load per anycast and unicast node

B. Versatile Anycast

Our anycast implementation exploits the fact that Mobile IPv6 decouples home- and care-of addresses, effectively allowing for the traffic directed to the former to be transparently redirected to the latter. This comes close to the anycast communication model, in which traffic sent to the anycast address of an anycast group is routed to the interface of some anycast node within that group. Recall that our solution causes each anycast group to appear to its clients as an MN. The anycast address *X* of that group then becomes the home address of that fictitious MN. The addresses of anycast nodes within the group, in turn, act as care-of addresses to which the traffic can be redirected. By disclosing different care-of addresses to different clients, the anycast group can convince different clients that the MN has moved to different locations (see Figure 2). Note that the client's higher (transport and application) layers retain the illusion that they communicate with the one and only node holding address *X*, as the translation between homeand care-of addresses is confined in the network layer. We implement the above communication model in two steps. First, we make sure that any traffic targeting the anycast address reaches one given anycast node within the respective anycast group. Second, we enable that node to transparently handoff clients to other anycast nodes within the group. Realizing these two steps allows us to implement versatile anycast, as we explain next.

C. Anycast Address

Constructing an anycast group requires creating its anycast address first. Such an address should be independent of the group composition, as the composition may change at any moment. We achieve this independence in two stages. First, we allow the anycast address to be provided by any anycast node within the group, as IPv6 enables any node to generate new IP addresses and attach them to its network interface. Second, we ensure that the anycast address remains valid despite changes in the group composition by allowing it to be *taken over* by any other anycast node as necessary. We refer to the anycast node holding the anycast address of its group as a *contact node*. To enable the anycast group to move its anycast address at will, the contact node registers that address with its HA, which results in a secret key being shared between the contact node and the HA. The contact node shares that key with some *backup nodes* within the group so that each of them can impersonate the contact node. Impersonating enables each backup node to take over the anycast address once the contact node has left the group, which causes the HA to tunnel all the traffic targeting the anycast address to the backup node. Doing so preserves the reachability of the anycast address as all the traffic addressed to the anycast group keeps on reaching some anycast node. Although the anycast address is now stable, the performance of anycast communication might still turn out to be poor because extensive tunneling to the new contact node can overload the home agent and introduce communication latency. These limitations are addressed by route optimization wherein the care-of address of an MN is revealed to a CN, allowing for direct communication between them. Since each anycast group appears to its clients and HAs as a single
regular MN, it can also use route optimization, causing the clients to communicate directly with the contact node using its actual address. This results in the performance of anycast communication remaining optimal. Note that the anycast group can prevent the contact node from becoming a potential single point of failure by providing multiple anycast addresses and registering them in the DNS. In that case, different anycast addresses can be handled by different anycast nodes, each acting as a contact node for its respective anycast address. Since these addresses never change, they can safely be registered in the DNS for a long time. Further details of our anycast address implementation can be found in the accompanying technical report [31].

D. Mathematical Framework

A convenient functional setting for the non-delayed packet field equations is to use the space $F = L^2(\Omega, R^p)$ which is a Hilbert space endowed with the usual inner product:

$$\langle V, U \rangle_F = \sum_{i=1}^{p} \int_\Omega V_i(r) U_i(r) dr \qquad (1)$$

To give a meaning to (1), we defined the history space $C = C^0([-\tau_m, 0], F)$ with $\|\phi\| = \sup_{t\in[-\tau_m,0]} \|\phi(t)\| F$, which is the Banach phase space associated with equation (3). Using the notation $V_t(\theta) = V(t+\theta), \theta \in [-\tau_m, 0]$, we write (1) as

$$\begin{cases} \dot{V}(t) = -L_0 V(t) + L_1 S(V_t) + I^{ext}(t), \\ V_0 = \phi \in C, \end{cases} \quad (2)$$

Where

$$\begin{cases} L_1 : C \to F, \\ \phi \to \int_\Omega J(.,\bar{r})\phi(\bar{r}, -\tau(.,\bar{r}))d\bar{r} \end{cases}$$

Is the linear continuous operator satisfying $\|L_1\| \le \|J\|_{L^2(\Omega^2, R^{p\times p})}$. Notice that most of the papers on this subject assume Ω infinite, hence requiring $\tau_m = \infty$.

Proposition 1.0 If the following assumptions are satisfied.

1. $J \in L^2(\Omega^2, R^{p\times p})$,
2. The external current $I^{ext} \in C^0(R, F)$,
3. $\tau \in C^0(\overline{\Omega^2}, R_+^{p\times p}), \sup_{\overline{\Omega^2}} \tau \le \tau_m$.

Then for any $\phi \in C$, there exists a unique solution $V \in C^1([0,\infty), F) \cap C^0([-\tau_m, \infty, F)$ to (3)

Notice that this result gives existence on R_+, finite-time explosion is impossible for this delayed differential equation. Nevertheless, a particular solution could grow indefinitely, we now prove that this cannot happen.

E. Boundedness of Solutions

A valid model of neural networks should only feature bounded packet node potentials.

Theorem 1.0 All the trajectories are ultimately bounded by the same constant R if $I \equiv \max_{t\in R^+} \|I^{ext}(t)\|_F < \infty$.

Proof :Let us defined $f : R \times C \to R^+$ as

$$f(t, V_t) \overset{def}{=} \langle -L_0 V_t(0) + L_1 S(V_t) + I^{ext}(t), V(t) \rangle_F = \frac{1}{2}\frac{d\|V\|_F^2}{dt}$$

We note $l = \min_{i=1,\dots p} l_i$

$$f(t, V_t) \le -l\|V(t)\|_F^2 + (\sqrt{p|\Omega|}\|J\|_F + I)\|V(t)\|_F$$

Thus, if

$$\|V(t)\|_F \ge 2\frac{\sqrt{p|\Omega|}.\|J\|_F + I}{l} \overset{def}{=} R, f(t, V_t) \le -\frac{lR^2}{2} \overset{def}{=} -\delta < 0$$

Let us show that the open route of F of center 0 and radius R, B_R, is stable under the dynamics of equation. We know that $V(t)$ is defined for all $t \ge 0s$ and that $f < 0$ on ∂B_R, the boundary of B_R. We consider three cases for the initial condition V_0. If $\|V_0\|_C < R$ and set $T = \sup\{t \mid \forall s \in [0,t], V(s) \in \overline{B_R}\}$. Suppose that $T \in R$, then $V(T)$ is defined and belongs to $\overline{B_R}$, the closure of B_R, because $\overline{B_R}$ is closed, in effect to ∂B_R, we also have $\frac{d}{dt}\|V\|_F^2 |_{t=T} = f(T, V_T) \le -\delta < 0$ because $V(T) \in \partial B_R$. Thus we deduce that for $\varepsilon > 0$ and small enough, $V(T+\varepsilon) \in \overline{B_R}$ which contradicts the definition of T. Thus $T \notin R$ and $\overline{B_R}$ is stable.

Because f<0 on $\partial B_R, V(0) \in \partial B_R$ implies that $\forall t > 0, V(t) \in B_R$. Finally we consider the case $V(0) \in C\overline{B_R}$. Suppose that $\forall t > 0, V(t) \notin \overline{B_R}$, then $\forall t > 0, \frac{d}{dt}\|V\|_F^2 \le -2\delta$, thus $\|V(t)\|_F$ is monotonically decreasing and reaches the value of R in finite time when $V(t)$ reaches ∂B_R. This contradicts our assumption. Thus $\exists T > 0 \mid V(T) \in B_R$.

Proposition 1.1 : Let s and t be measured simple functions on X. for $E \varepsilon M$, define

$$\phi(E) = \int_E s\, d\mu \quad (1)$$

Then ϕ is a measure on M.

$$\int_X (s+t)d\mu = \int_X s\, d\mu + \int_X t d\mu \quad (2)$$

Proof : If s and if $E_1, E_2, \dots$ are disjoint members of M whose union is E, the countable additivity of μ shows that

$$\phi(E) = \sum_{i=1}^{n} \alpha_i \mu(A_i \cap E) = \sum_{i=1}^{n} \alpha_i \sum_{r=1}^{\infty} \mu(A_i \cap E_r)$$

$$= \sum_{r=1}^{\infty}\sum_{i=1}^{n} \alpha_i \mu(A_i \cap E_r) = \sum_{r=1}^{\infty} \phi(E_r)$$

Also, $\varphi(\phi)=0,$ so that φ is not identically ∞.

Next, let s be as before, let $\beta_1,...,\beta_m$ be the distinct values of t,and let $B_j=\{x:t(x)=\beta_j\}$ If $E_{ij}=A_i\cap B_j,$ the

$\int_{E_{ij}}(s+t)d\mu=(\alpha_i+\beta_j)\mu(E_{ij})$

and $\int_{E_{ij}}sd\mu+\int_{E_{ij}}td\mu=\alpha_i\mu(E_{ij})+\beta_j\mu(E_{ij})$

Thus (2) holds with E_{ij} in place of X. Since X is the disjoint union of the sets E_{ij} $(1\le i\le n,1\le j\le m),$ the first half of our proposition implies that (2) holds.

Theorem 1.1: If K is a compact set in the plane whose complement is connected, if f is a continuous complex function on K which is holomorphic in the interior of , and if $\varepsilon>0,$ then there exists a polynomial P such that $|f(z)=P(z)|<\varepsilon$ for all $z\varepsilon K$. If the interior of K is empty, then part of the hypothesis is vacuously satisfied, and the conclusion holds for every $f\varepsilon C(K)$. Note that K need to be connected.

Proof: By Tietze's theorem, f can be extended to a continuous function in the plane, with compact support. We fix one such extension and denote it again by f. For any $\delta>0,$ let $\omega(\delta)$ be the supremum of the numbers $|f(z_2)-f(z_1)|$ Where z_1 and z_2 are subject to the condition $|z_2-z_1|\le\delta$. Since f is uniformly continous, we have $\lim_{\delta\to0}\omega(\delta)=0$ (1) From now on, δ will be fixed. We shall prove that there is a polynomial P such that

$$|f(z)-P(z)|<10,000\ \omega(\delta)\quad(z\varepsilon K)\qquad(2)$$

By (1), this proves the theorem. Our first objective is the construction of a function $\Phi\varepsilon C_c^{'}(R^2),$ such that for all z

$$|f(z)-\Phi(z)|\le\omega(\delta),\qquad(3)$$

$$|(\partial\Phi)(z)|<\frac{2\omega(\delta)}{\delta},\qquad(4)$$

And

$$\Phi(z)=-\frac{1}{\pi}\iint_X\frac{(\partial\Phi)(\zeta)}{\zeta-z}d\zeta d\eta\qquad(\zeta=\xi+i\eta),\qquad(5)$$

Where X is the set of all points in the support of Φ whose distance from the complement of K does not δ. (Thus X contains no point which is "far within" K.) We construct Φ as the convolution of f with a smoothing function A. Put $a(r)=0$ if $r>\delta,$ put

$$a(r)=\frac{3}{\pi\delta^2}(1-\frac{r^2}{\delta^2})^2\qquad(0\le r\le\delta),\qquad(6)$$

And define

$$A(z)=a(|z|)\qquad(7)$$

For all complex z. It is clear that $A\varepsilon C_c^{'}(R^2)$. We claim that

$$\iint_{R^s}A=1,\qquad(8)$$

$$\iint_{R^2}\partial A=0,\qquad(9)$$

$$\iint_{R^3}|\partial A|=\frac{24}{15\delta}<\frac{2}{\delta},\qquad(10)$$

The constants are so adjusted in (6) that (8) holds. (Compute the integral in polar coordinates), (9) holds simply because A has compact support. To compute (10), express ∂A in polar coordinates, and note that

$$\partial A/\partial\theta=0,$$

$$\partial A/\partial r=-a^{'},$$

Now define

$$\Phi(z)=\iint_{R^2}f(z-\zeta)Ad\xi d\eta=\iint_{R^2}A(z-\zeta)f(\zeta)d\xi d\eta\qquad(11)$$

Since f and A have compact support, so does Φ. Since

$$\Phi(z)-f(z)$$

$$=\iint_{R^2}[f(z-\zeta)-f(z)]A(\xi)d\xi d\eta\quad(12)$$

And $A(\zeta)=0$ if $|\zeta|>\delta,$ (3) follows from (8). The difference quotients of A converge boundedly to the corresponding partial derivatives, since $A\varepsilon C_c^{'}(R^2)$. Hence the last expression in (11) may be differentiated under the integral sign, and we obtain

$$(\partial\Phi)(z) = \iint_{R^2} (\overline{\partial A})(z-\zeta) f(\zeta) d\xi d\eta$$
$$= \iint_{R^2} f(z-\zeta)(\partial A)(\zeta) d\xi d\eta$$
$$= \iint_{R^2} [f(z-\zeta) - f(z)](\partial A)(\zeta) d\xi d\eta \quad (13)$$

The last equality depends on (9). Now (10) and (13) give (4). If we write (13) with Φ_x and Φ_y in place of $\partial\Phi$, we see that Φ has continuous partial derivatives, if we can show that $\partial\Phi = 0$ in G, where G is the set of all $z \varepsilon K$ whose distance from the complement of K exceeds δ. We shall do this by showing that

$$\Phi(z) = f(z) \qquad (z \varepsilon G); \qquad (14)$$

Note that $\partial f = 0$ in G, since f is holomorphic there. Now if $z \varepsilon G$, then $z - \zeta$ is in the interior of K for all ζ with $|\zeta| < \delta$. The mean value property for harmonic functions therefore gives, by the first equation in (11),

$$\Phi(z) = \int_0^\delta a(r) r dr \int_0^{2\pi} f(z - re^{i\theta}) d\theta$$
$$= 2\pi f(z) \int_0^\delta a(r) r dr = f(z) \iint_{R^2} A = f(z) \quad (15)$$

For all $z \varepsilon G$, we have now proved (3), (4), and (5) The definition of X shows that X is compact and that X can be covered by finitely many open discs $D_1, ..., D_n$, of radius 2δ, whose centers are not in K. Since $S^2 - K$ is connected, the center of each D_j can be joined to ∞ by a polygonal path in $S^2 - K$. It follows that each D_j contains a compact connected set E_j, of diameter at least 2δ, so that $S^2 - E_j$ is connected and so that $K \cap E_j = \phi$. with $r = 2\delta$. There are functions $g_j \varepsilon H(S^2 - E_j)$ and constants b_j so that the inequalities.

$$|Q_j(\zeta, z)| < \frac{50}{\delta}, \qquad (16)$$

$$\left| Q_j(\zeta, z) - \frac{1}{z-\zeta} \right| < \frac{4,000\delta^2}{|z-\zeta|^2} \qquad (17)$$

Hold for $z \notin E_j$ and $\zeta \in D_j$, if

$$Q_j(\zeta, z) = g_j(z) + (\zeta - b_j) g_j^2(z) \qquad (18)$$

Let Ω be the complement of $E_1 \cup ... \cup E_n$. Then Ω is an open set which contains K. Put $X_1 = X \cap D_1$ and $X_j = (X \cap D_j) - (X_1 \cup ... \cup X_{j-1})$, for $2 \le j \le n$,

Define

$$R(\zeta, z) = Q_j(\zeta, z) \qquad (\zeta \varepsilon X_j, z \varepsilon \Omega) \qquad (19)$$

And

$$F(z) = \frac{1}{\pi} \iint_X (\partial\Phi)(\zeta) R(\zeta, z) d\zeta d\eta \qquad (20)$$
$$(z \ \varepsilon \ \Omega)$$

Since,

$$F(z) = \sum_{j=1} \frac{1}{\pi} \iint_{X_i} (\partial\Phi)(\zeta) Q_j(\zeta, z) d\xi d\eta, \qquad (21)$$

(18) shows that F is a finite linear combination of the functions g_j and g_j^2. Hence $F \varepsilon H(\Omega)$. By (20), (4), and (5) we have

$$|F(z) - \Phi(z)| < \frac{2\omega(\delta)}{\pi\delta} \iint_X \left| R(\zeta, z) - \frac{1}{z-\zeta} \right| d\xi d\eta \quad (z \ \varepsilon \ \Omega) \quad (22)$$

Observe that the inequalities (16) and (17) are valid with R in place of Q_j if $\zeta \ \varepsilon \ X$ and $z \ \varepsilon \ \Omega$. Now fix $z \ \varepsilon \ \Omega$., put $\zeta = z + \rho e^{i\theta}$, and estimate the integrand in (22) by (16) if $\rho < 4\delta$, by (17) if $4\delta \le \rho$. The integral in (22) is then seen to be less than the sum of

$$2\pi \int_0^{4\delta} \left(\frac{50}{\delta} + \frac{1}{\rho} \right) \rho d\rho = 808\pi\delta \qquad (23)$$

And

$$2\pi \int_{4\delta}^\infty \frac{4,000\delta^2}{\rho^2} \rho d\rho = 2,000\pi\delta. \qquad (24)$$

Hence (22) yields

$$|F(z) - \Phi(z)| < 6,000\omega(\delta) \qquad (z \ \varepsilon \ \Omega) \qquad (25)$$

Since $F \ \varepsilon \ H(\Omega)$, $K \subset \Omega$, and $S^2 - K$ is connected, Runge's theorem shows that F can be uniformly approximated on K by polynomials. Hence (3) and (25) show that (2) can be satisfied. This completes the proof.

Lemma 1.0 : Suppose $f \varepsilon C_c^{'}(R^2)$, the space of all continuously differentiable functions in the plane, with compact support. Put

$$\partial = \frac{1}{2}\left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right) \qquad (1)$$

Then the following "Cauchy formula" holds:

$$f(z) = -\frac{1}{\pi}\iint_{R^2}\frac{(\partial f)(\zeta)}{\zeta - z}d\xi d\eta$$

$$(\zeta = \xi + i\eta) \qquad (2)$$

Proof: This may be deduced from Green's theorem. However, here is a simple direct proof:

Put $\varphi(r,\theta) = f(z + re^{i\theta}),\ r > 0,\ \theta$ real

If $\zeta = z + re^{i\theta}$, the chain rule gives

$$(\partial f)(\zeta) = \frac{1}{2}e^{i\theta}\left[\frac{\partial}{\partial r} + \frac{i}{r}\frac{\partial}{\partial \theta}\right]\varphi(r,\theta) \qquad (3)$$

The right side of (2) is therefore equal to the limit, as $\varepsilon \to 0$, of

$$-\frac{1}{2}\int_{\varepsilon}^{\infty}\int_{0}^{2\pi}\left(\frac{\partial\varphi}{\partial r} + \frac{i}{r}\frac{\partial\varphi}{\partial\theta}\right)d\theta dr \qquad (4)$$

For each $r > 0, \varphi$ is periodic in θ, with period 2π. The integral of $\partial\varphi/\partial\theta$ is therefore 0, and (4) becomes

$$-\frac{1}{2\pi}\int_0^{2\pi}d\theta\int_{\varepsilon}^{\infty}\frac{\partial\varphi}{\partial r}dr = \frac{1}{2\pi}\int_0^{2\pi}\varphi(\varepsilon,\theta)d\theta \qquad (5)$$

As $\varepsilon \to 0,\ \varphi(\varepsilon,\theta) \to f(z)$ uniformly. This gives (2)

If $X^{\alpha} \in a$ and $X^{\beta} \in k[X_1,...X_n]$, then $X^{\alpha}X^{\beta} = X^{\alpha+\beta} \in a$, and so A satisfies the condition $(*)$. Conversely,

$$(\sum_{\alpha\in A}c_{\alpha}X^{\alpha})(\sum_{\beta\in\square^n}d_{\beta}X^{\beta}) = \sum_{\alpha,\beta}c_{\alpha}d_{\beta}X^{\alpha+\beta} \qquad (\textit{finite sums}),$$

and so if A satisfies $(*)$, then the subspace generated by the monomials $X^{\alpha}, \alpha \in a$, is an ideal. The proposition gives a classification of the monomial ideals in $k[X_1,...X_n]$: they are in one to one correspondence with the subsets A of $\square^n$ satisfying $(*)$. For example, the monomial ideals in $k[X]$ are exactly the ideals $(X^n),\ n \geq 1$, and the zero ideal (corresponding to the empty set A). We write $\langle X^{\alpha} \mid \alpha \in A\rangle$ for the ideal corresponding to A (subspace generated by the $X^{\alpha}, \alpha \in a$).

LEMMA 1.1. Let S be a subset of $\square^n$. The the ideal a generated by $X^{\alpha}, \alpha \in S$ is the monomial ideal corresponding to

$$A \overset{df}{=} \{\beta \in \square^n \mid \beta - \alpha \in \square^n, \quad some\ \alpha \in S\}$$

Thus, a monomial is in a if and only if it is divisible by one of the $X^{\alpha}, \alpha \in| S$

PROOF. Clearly A satisfies $(*)$, and $a \subset \langle X^{\beta} \mid \beta \in A\rangle$. Conversely, if $\beta \in A$, then $\beta - \alpha \in \square^n$ for some $\alpha \in S$, and $X^{\beta} = X^{\alpha}X^{\beta-\alpha} \in a$. The last statement follows from the fact that $X^{\alpha} \mid X^{\beta} \Leftrightarrow \beta - \alpha \in \square^n$. Let $A \subset \square^n$ satisfy $(*)$. From the geometry of A, it is clear that there is a finite set of elements $S = \{\alpha_1,...\alpha_s\}$ of A such that $A = \{\beta \in \square^n \mid \beta - \alpha_i \in \square^2,\ some\ \alpha_i \in S\}$ (The $\alpha_i{}'s$ are the corners of A) Moreover, $a \overset{df}{=} \langle X^{\alpha} \mid \alpha \in A\rangle$ is generated by the monomials $X^{\alpha_i}, \alpha_i \in S$.

DEFINITION 1.0. For a nonzero ideal a in $k[X_1,...,X_n]$, we let $(LT(a))$ be the ideal generated by $\{LT(f) \mid f \in a\}$

LEMMA 1.2 Let a be a nonzero ideal in $k[X_1,...,X_n]$; then $(LT(a))$ is a monomial ideal, and it equals $(LT(g_1),...,LT(g_n))$ for some $g_1,...,g_n \in a$.

PROOF. Since $(LT(a))$ can also be described as the ideal generated by the leading monomials (rather than the leading terms) of elements of a.

THEOREM 1.2. Every *ideal* a in $k[X_1,...,X_n]$ is finitely generated; more precisely, $a = (g_1,...,g_s)$ where $g_1,...,g_s$ are any elements of a whose leading terms generate $LT(a)$

PROOF. Let $f \in a$. On applying the division algorithm, we find

$f = a_1 g_1 + ... + a_s g_s + r, \quad a_i, r \in k[X_1, ..., X_n]$, where either $r = 0$ or no monomial occurring in it is divisible by any $LT(g_i)$. But $r = f - \sum a_i g_i \in a$, and therefore $LT(r) \in LT(a) = (LT(g_1), ..., LT(g_s))$, implies that every monomial occurring in r is divisible by one in $LT(g_i)$. Thus $r = 0$, and $g \in (g_1, ..., g_s)$.

DEFINITION 1.1. A finite subset $S = \{g_1, | ..., g_s\}$ of an ideal a is a standard ($(Gr\ddot{o}bner)$ bases for a if $(LT(g_1), ..., LT(g_s)) = LT(a)$. In other words, S is a standard basis if the leading term of every element of a is divisible by at least one of the leading terms of the g_i.

THEOREM 1.3 *The ring* $k[X_1, ..., X_n]$ *is Noetherian i.e., every ideal is finitely generated.*

PROOF. For $n = 1, \ k[X]$ is a principal ideal domain, which means that every ideal is generated by single element. We shall prove the theorem by induction on n . Note that the obvious map $k[X_1, ...X_{n-1}][X_n] \to k[X_1, ...X_n]$ is an isomorphism – this simply says that every polynomial f in n variables $X_1, ...X_n$ can be expressed uniquely as a polynomial in X_n with coefficients in $k[X_1, ..., X_n]$:

$$f(X_1, ...X_n) = a_0(X_1, ...X_{n-1}) X_n^r + ... + a_r(X_1, ...X_{n-1})$$

Thus the next lemma will complete the proof

LEMMA 1.3. If A is Noetherian, then so also is $A[X]$

PROOF. For a polynomial

$$f(X) = a_0 X^r + a_1 X^{r-1} + ... + a_r, \quad a_i \in A, \quad a_0 \neq 0,$$

r is called the degree of f , and a_0 is its leading coefficient. We call 0 the leading coefficient of the polynomial 0. Let a be an ideal in $A[X]$. The leading coefficients of the polynomials in a form an ideal a' in A, and since A is Noetherian, a' will be finitely generated. Let $g_1, ..., g_m$ be elements of a whose leading coefficients generate a' , and let r be the maximum degree of g_i . Now let $f \in a,$ and suppose f has degree $s > r$, say, $f = aX^s + ...$ Then $a \in a'$, and so we can write

$$a = \sum b_i a_i, \quad b_i \in A,$$

$a_i = leading\ coefficient\ of\ g_i$

Now

$f - \sum b_i g_i X^{s-r_i}, \quad r_i = \deg(g_i),$ has degree $< \deg(f)$. By continuing in this way, we find that $f \equiv f_t \mod(g_1, ...g_m)$ With f_t a polynomial of degree $t < r$. For each $d < r$, let a_d be the subset of A consisting of 0 and the leading coefficients of all polynomials in a of degree d; it is again an ideal in A . Let $g_{d,1}, ..., g_{d,m_d}$ be polynomials of degree d whose leading coefficients generate a_d . Then the same argument as above shows that any polynomial f_d in a of degree d can be written $f_d \equiv f_{d-1} \mod(g_{d,1}, ...g_{d,m_d})$ With f_{d-1} of degree $\leq d-1$. On applying this remark repeatedly we find that $f_t \in (g_{r-1,1}, ...g_{r-1,m_{r-1}}, ...g_{0,1}, ...g_{0,m_0})$ Hence

$$f_t \in (g_1, ...g_m g_{r-1,1}, ...g_{r-1,m_{r-1}}, ..., g_{0,1}, ..., g_{0,m_0})$$

and so the polynomials $g_1, ..., g_{0,m_0}$ generate a

F. Anycast Traffic Handoff

Our implementation of the anycast address ensures that all the client traffic reaches the contact node. However, this node should not handle all the traffic by itself. It therefore needs a mechanism that allows it to transparently handoff the traffic to other anycast nodes, which later may transparently hand it off again. We refer to the anycast node that hands off a client as a *donor*, and to the anycast node that takes over the client as an *acceptor*. Recall that address translation in MIPv6 is performed according to bindings created during MIPv6 route optimization. As we discussed in the previous section, anycast groups already exploit this mechanism to establish direct communication between contact nodes and their clients. However, since route optimizations are performed separately for each client, the anycast group can also use them to hand off individual clients between any pair of anycast nodes. To this end, the anycast group carefully mimics the signaling of a mobile node performing route optimization. Switching the network traffic alone might not be enough, as many applications communicate with their clients using stateful connections such as TCP. In that

case, the donor must provide the acceptor with the state of all the network connections opened by the client, so that the acceptor can continue to communicate with the client using these connections and does not reset them. Depending on the application, the same might hold for the application-level state of the client. Our anycast implementation provides anycast nodes with the ability to exchange all such state information as necessary. Further details of how this is done can be found in [31].

Fig 1.2 Query Load Distribution

III. IMPLEMENTATION CHALLENGES

We address a number of practical considerations for the algorithm design and parameter selection in real-world deployments in this section.

A. Demand Estimation

Our MIP formulation needs the demand for videos to compute placement. This, however, is not known a priori. Our approach is to use the recent history (e.g., the past 7 days) as a guide to future demand for the videos. We use this history as an input to our formulation. While history is available for existing videos, new videos are added to the library continually. Further, from our analysis, we find that many such newly added videos receive a significant number of requests. Hence we also need to address the problem of placement of new content into the system. While demand estimation for such new videos is an active area of research [2, 14] and beyond the scope of this paper, we use a simple estimation strategy. It is based on the observation that a significant number of the newly added videos belong to TV series, and that videos from a TV series exhibit similar demand patterns. Figure 4 presents the daily re- quest count for different episodes of a particular series show during one month. Although there is some vari- ation, we observe considerable similarity in the request volume for each episode of the series. For instance, on the day of release, episode 2 was requested around 7000 times, and episode 3 around 8700 times. In our system, we base our demand estimate for a new episode of a TV series on the requests for the previous week's episode of the same series (e.g., request pattern of episode 2 is used as demand estimate for episode 3). We show the effectiveness of our approach in Section 7.8. We use another simple estimation strategy for block- buster movies. From exogenous information [2], we assume that we are informed of a list of blockbuster movies (e.g., 1–3 movies each week). Then, we take the demand history of the most popular movie in the pre-vious week and use it as the predicted demand for the blockbuster movies that are released this week. Complementary caching: While TV shows and blockbuster movies account for the majority of requests for new videos — series episodes account for more than half of the requests for new releases — we still do not have a demand estimate for the remaining new videos (music videos, unpopular movies, etc.). Our current system uses a small LRU cache to handle load due to new releases for which we do not have estimates. This cache also handles unpredictable popularity surges to some videos (which is often why LRU caches are used).

B. Time-varying demand

We observe that the request pattern changes quite significantly over time, both in aggregate intensity and
its distribution over the individual items in the library. For instance, users typically make significantly more re-quests on Fridays and Saturdays, while the traffic mix during the peak intervals on those two weekend days are quite different. The bandwidth required to serve these requests will correspondingly vary when they are served from remote VHOs. The placement should be able to handle such change and still satisfy the link constraints throughout the entire time period that the placement would remain before it is re-evaluated. While accounting for link utilization at all times (e.g., each second during a 7-day interval) might guarantee that we never exceed the link constraint, it makes the problem computationally infeasible as the number of link bandwidth constraints (6) is proportional to the number of time slices in |T |. We find that the demand

during non-peak periods does not overload any links. Therefore, we identify a very small number of peak de- mand periods (typically, we use |T | = 2) for which to enforce the link constraints. Picking the size of time window to compute load is also critical. If we pick a small time window, we may not capture the representa-tive load and hence will not place videos appropriately. If we use a large window, we may considerably over-provision capacity for our MIP to become feasible.

C. Placement Update Frequence

Another consideration is the frequency of implement-ing a new placement using our algorithm. While updat-ing our placement more frequently allows the system to adapt to changing demands and correct for estima-tion errors more gracefully, each update incurs over- head, both in computing the new MIP solution and mi-grating the content.

One of the great successes of category theory in computer science has been the development of a "unified theory" of the constructions underlying denotational semantics. In the untyped λ -calculus, any term may appear in the function position of an application. This means that a model D of the λ -calculus must have the property that given a term t whose interpretation is $d \in D$, Also, the interpretation of a functional abstraction like $\lambda x.x$ is most conveniently defined as a function from $D \text{ to } D$, which must then be regarded as an element of *D*. Let $\psi : [D \to D] \to D$ be the function that picks out elements of *D* to represent elements of $[D \to D]$ and $\phi : D \to [D \to D]$ be the function that maps elements of *D* to functions of *D*. Since $\psi(f)$ is intended to represent the function f as an element of *D*, it makes sense to require that $\phi(\psi(f)) = f$, that is, $\psi o \psi = id_{[D \to D]}$ Furthermore, we often want to view every element of *D* as representing some function from *D to D* and require that elements representing the same function be equal – that is

$$\psi(\varphi(d)) = d$$

or

$$\psi o \phi = id_D$$

The latter condition is called extensionality. These conditions together imply that $\phi \text{ and } \psi$ are inverses--- that is, *D* is isomorphic to the space of functions from *D to D* that can be the interpretations of functional abstractions: $D \cong [D \to D]$ .Let us suppose we are working with the untyped $\lambda - calculus$, we need a solution ot the equation $D \cong A + [D \to D]$, where A is some predetermined domain containing interpretations for elements of *C*. Each element of *D* corresponds to either an element of *A* or an element of $[D \to D]$, with a tag. This equation can be solved by finding least fixed points of the function $F(X) = A + [X \to X]$ from domains to domains --- that is, finding domains *X* such that $X \cong A + [X \to X]$, and such that for any domain *Y* also satisfying this equation, there is an embedding of *X* to *Y* --- a pair of maps

$$X \underset{f^R}{\overset{f}{\square}} Y$$

Such that

$$f^R o f = id_X$$

$$f o f^R \subseteq id_Y$$

Where $f \subseteq g$ means that $f \text{ approximates } g$ in some ordering representing their information content. The key shift of perspective from the domain-theoretic to the more general category-theoretic approach lies in considering *F* not as a function on domains, but as a *functor* on a category of domains. Instead of a least fixed point of the function, *F*.

Definition 1.3: Let ***K*** be a category and $F : K \to K$ as a functor. A fixed point of *F* is a pair (A,a), where A is a ***K-object*** and $a : F(A) \to A$ is an isomorphism. A prefixed point of F is a pair (A,a), where A is a ***K-object*** and a is any arrow from F(A) to A

Definition 1.4 : An $\omega - chain$ in a category ***K*** is a diagram of the following form:

$$\Delta = D_o \xrightarrow{f_o} D_1 \xrightarrow{f_1} D_2 \xrightarrow{f_2} \cdots.$$

Recall that a cocone μ of an $\omega - chain\ \Delta$ is a *K*-object *X* and a collection of K –*arrows* $\{\mu_i : D_i \to X \mid i \geq 0\}$ such that $\mu_i = \mu_{i+1} o f_i$ for all $i \geq 0$. We sometimes write $\mu : \Delta \to X$ as a reminder of the arrangement of $\mu's$ components Similarly, a colimit $\mu : \Delta \to X$ is a cocone with the property that if $\nu : \Delta \to X'$ is also a cocone then there exists a unique mediating arrow $k : X \to X'$ such that for all $i \geq 0,, \nu_i = k o \mu_i$. Colimits of $\omega - chains$ are sometimes referred to as $\omega - co\lim its$. Dually, an $\omega^{op} - chain$ in ***K*** is a diagram of the following form:

$\Delta = D_o \xleftarrow{f_o} D_1 \xleftarrow{f_1} D_2 \xleftarrow{f_2} \cdots$ A cone $\mu : X \to \Delta$ of an $\omega^{op} - chain$ Δ is a ***K***-object X and a collection of **K**-arrows $\{\mu_i : D_i \mid i \geq 0\}$ such that for all $i \geq 0,\ \mu_i = f_i \, o \, \mu_{i+1}$. An ω^{op} -limit of an $\omega^{op} - chain$ Δ is a cone $\mu : X \to \Delta$ with the property that if $\nu : X^{'} \to \Delta$ is also a cone, then there exists a unique mediating arrow $k : X^{'} \to X$ such that for all $i \geq 0, \mu_i \, o \, k = \nu_i$. We write $\perp_k$ (or just $\perp$) for the distinguish initial object of ***K,*** when it has one, and $\perp \to A$ for the unique arrow from $\perp$ to each ***K***-object A. It is also convenient to write $\Delta^{-} = D_1 \xrightarrow{f_1} D_2 \xrightarrow{f_2} \cdots$ to denote all of Δ except D_o and f_0. By analogy, μ^{-} is $\{\mu_i \mid i \geq 1\}$. For the images of Δ and μ under ***F*** we write $F(\Delta) = F(D_o) \xrightarrow{F(f_o)} F(D_1) \xrightarrow{F(f_1)} F(D_2) \xrightarrow{F(f_2)} \cdots$ and $F(\mu) = \{F(\mu_i) \mid i \geq 0\}$

We write F^i for the ***i***-fold iterated composition of ***F*** – that is, $F^o(f) = f, F^1(f) = F(f), F^2(f) = F(F(f))$,etc. With these definitions we can state that every monitonic function on a complete lattice has a least fixed point:

Lemma 1.4. Let ***K*** be a category with initial object $\perp$ and let $F : K \to K$ be a functor. Define the $\omega - chain \Delta$ by

$\Delta = \perp \xrightarrow{!\perp \to F(\perp)} F(\perp) \xrightarrow{F(!\perp \to F(\perp))} F^2(\perp) \xrightarrow{F^2(!\perp \to F(\perp))} \cdots\cdots\cdots$

If both $\mu : \Delta \to D$ and $F(\mu) : F(\Delta) \to F(D)$ are colimits, then (D,d) is an intial F-algebra, where $d : F(D) \to D$ is the mediating arrow from $F(\mu)$ to the cocone μ^{-}

Theorem 1.4 Let a DAG G given in which each node is a random variable, and let a discrete conditional probability distribution of each node given values of its parents in G be specified. Then the product of these conditional distributions yields a joint probability distribution P of the variables, and (G,P) satisfies the Markov condition.

Proof. Order the nodes according to an ancestral ordering. Let $X_1, X_2, \ldots\ldots\ldots X_n$ be the resultant ordering. Next define.

$P(x_1, x_2, \ldots . x_n) = P(x_n \mid pa_n) P(x_{n-1} \mid Pa_{n-1}) \ldots$
$\ldots P(x_2 \mid pa_2) P(x_1 \mid pa_1),$

Where PA_i is the set of parents of X_i of in G and $P(x_i \mid pa_i)$ is the specified conditional probability distribution. First we show this does indeed yield a joint probability distribution. Clearly, $0 \leq P(x_1, x_2, \ldots x_n) \leq 1$ for all values of the variables. Therefore, to show we have a joint distribution, as the variables range through all their possible values, is equal to one. To that end, Specified conditional distributions are the conditional distributions they notationally represent in the joint distribution. Finally, we show the Markov condition is satisfied. To do this, we need show for $1 \leq k \leq n$ that
whenever

$P(pa_k) \neq 0, if \ \ P(nd_k \mid pa_k) \neq 0$
$and \ \ P(x_k \mid pa_k) \neq 0$
$then \ P(x_k \mid nd_k, pa_k) = P(x_k \mid pa_k),$

Where ND_k is the set of nondescendents of X_k of in G. Since $PA_k \subseteq ND_k$, we need only show $P(x_k \mid nd_k) = P(x_k \mid pa_k)$. First for a given k , order the nodes so that all and only nondescendents of X_k precede X_k in the ordering. Note that this ordering depends on k , whereas the ordering in the first part of the proof does not. Clearly then

$$ND_k = \{X_1, X_2, \ldots . X_{k-1}\}$$
$$Let$$
$$D_k = \{X_{k+1}, X_{k+2}, \ldots . X_n\}$$

follows $\sum_{d_k}$

We define the m^{th} *cyclotomic field to be the field* $Q[x]/(\Phi_m(x))$ Where $\Phi_m(x)$ is the m^{th} cyclotomic polynomial. $Q[x]/(\Phi_m(x))$ $\Phi_m(x)$ *has degree* $\varphi(m)$ *over* Q *since* $\Phi_m(x)$ has degree $\varphi(m)$. *The roots of* $\Phi_m(x)$ are just the primitive m^{th} roots of unity, so the complex embeddings of $Q[x]/(\Phi_m(x))$ *are simply the* $\varphi(m)$ *maps*

$\sigma_k : Q[x]/(\Phi_m(x)) \mapsto C,$
$1 \leq k \prec m, (k, m) = 1, \quad where$
$$\sigma_k(x) = \xi_m^k,$$

ξ_m being our fixed choice of primitive m^{th} root of unity. Note that $\xi_m^k \in Q(\xi_m)$ for every k; it follows that $Q(\xi_m) = Q(\xi_m^k)$ for all k relatively prime to m. In particular, the images of the σ_i coincide, so $Q[x]/(\Phi_m(x))$ *is Galois over* Q. *This means that we can write* $Q(\xi_m)$ *for* $Q[x]/(\Phi_m(x))$ *without much fear of ambiguity; we will do so from now on, the identification being* $\xi_m \mapsto x$. *One advantage of this is that one can easily talk about cyclotomic fields being extensions of one another,or intersections or compositums; all of these things take place considering them as subfield of* C. We now investigate some basic properties of cyclotomic fields. The first issue is whether or not they are all distinct; to determine this, we need to know which roots of unity lie in $Q(\xi_m)$.Note, for example, that if m is odd, then $-\xi_m$ is a $2m^{th}$ root of unity. We will show that this is the only way in which one can obtain any non-m^{th} roots of unity.

LEMMA 1.5 If m divides n, then $Q(\xi_m)$ *is contained in* $Q(\xi_n)$

PROOF. Since $\xi^{n/m} = \xi_m$, *we have* $\xi_m \in Q(\xi_n)$, *so the result is clear*

LEMMA 1.6 If m and n are relatively prime, then

$$Q(\xi_m, \xi_n) = Q(\xi_{nm})$$

and

$$Q(\xi_m) \cap Q(\xi_n) = Q$$

(Recall the $Q(\xi_m, \xi_n)$ is the compositum of $Q(\xi_m)$ *and* $Q(\xi_n)$)

PROOF. One checks easily that $\xi_m \xi_n$ is a primitive mn^{th} root of unity, so that

$$Q(\xi_{mn}) \subseteq Q(\xi_m, \xi_n)$$

$$[Q(\xi_m, \xi_n) : Q] \le [Q(\xi_m) : Q][Q(\xi_n : Q]$$

$$= \varphi(m)\varphi(n) = \varphi(mn);$$

Since $[Q(\xi_{mn}) : Q] = \varphi(mn);$ this implies that $Q(\xi_m, \xi_n) = Q(\xi_{nm})$ We know that $Q(\xi_m, \xi_n)$ has degree $\varphi(mn)$ over Q, so we must have

$$[Q(\xi_m, \xi_n) : Q(\xi_m)] = \varphi(n)$$

and

$$[Q(\xi_m, \xi_n) : Q(\xi_m)] = \varphi(m)$$

$$[Q(\xi_m) : Q(\xi_m) \cap Q(\xi_n)] \ge \varphi(m)$$

And thus that $Q(\xi_m) \cap Q(\xi_n) = Q$

PROPOSITION 1.2 For any m and n

$$Q(\xi_m, \xi_n) = Q(\xi_{[m,n]})$$

And

$$Q(\xi_m) \cap Q(\xi_n) = Q(\xi_{(m,n)});$$

here $[m,n]$ and (m,n) denote the least common multiple and the greatest common divisor of m and n, respectively.

PROOF. Write $m = p_1^{e_1} \ldots\ldots p_k^{e_k}$ *and* $p_1^{f_1} \ldots . p_k^{f_k}$ where the p_i are distinct primes. (We allow e_i *or* f_i to be zero)

$$Q(\xi_m) = Q(\xi_{p_1^{e_1}})Q(\xi_{p_2^{e_2}}) \ldots Q(\xi_{p_k^{e_k}})$$

and

$$Q(\xi_n) = Q(\xi_{p_1^{f_1}})Q(\xi_{p_2^{f_2}}) \ldots Q(\xi_{p_k^{f_k}})$$

Thus

$$\begin{aligned} Q(\xi_m, \xi_n) &= Q(\xi_{p_1^{e_1}}) \ldots\ldots\ldots Q(\xi_{p_2^{e_k}})Q(\xi_{p_1^{f_1}}) \ldots Q(\xi_{p_k^{f_k}}) \\ &= Q(\xi_{p_1^{e_1}})Q(\xi_{p_1^{f_1}}) \ldots Q(\xi_{p_k^{e_k}})Q(\xi_{p_k^{f_k}}) \\ &= Q(\xi_{p_1^{\max(e_1, f_1)}}) \ldots\ldots\ldots Q(\xi_{p_1^{\max(e_k, f_k)}}) \\ &= Q(\xi_{p_1^{\max(e_1, f_1)} \ldots\ldots\ldots p_1^{\max(e_k, f_k)}}) \\ &= Q(\xi_{[m,n]}); \end{aligned}$$

An entirely similar computation shows that $Q(\xi_m) \cap Q(\xi_n) = Q(\xi_{(m,n)})$

Mutual information measures the information transferred when x_i is sent and y_i is received, and is defined as

$$I(x_i, y_i) = \log_2 \frac{P(x_i / y_i)}{P(x_i)} \; bits \qquad (1)$$

In a noise-free channel, **each** y_i is uniquely connected to the corresponding x_i , and so they constitute an input –output pair (x_i, y_i) for which

$$P(x_i / y_j) = 1 \; and \; I(x_i, y_j) = \log_2 \frac{1}{P(x_i)} \quad \text{bits;}$$

that is, the transferred information is equal to the self-

information that corresponds to the input x_i In a very noisy channel, the output y_i and input x_i would be completely uncorrelated, and so $P(x_i/y_j)=P(x_i)$ and also $I(x_i,y_j)=0;$ that is, there is no transference of information. In general, a given channel will operate between these two extremes. The mutual information is defined between the input and the output of a given channel. An average of the calculation of the mutual information for all input-output pairs of a given channel is the average mutual information:

$$I(X,Y)=\sum_{i.j}P(x_i,y_j)I(x_i,y_j)=\sum_{i.j}P(x_i,y_j)\log_2\left[\frac{P(x_i/y_j}{P(x_i)}\right]$$

bits per symbol . This calculation is done over the input and output alphabets. The average mutual information. The following expressions are useful for modifying the mutual information expression:

$$P(x_i,y_j)=P(x_i/y_j)P(y_j)=P(y_j/x_i)P(x_i)$$

$$P(y_j)=\sum_i P(y_j/x_i)P(x_i)$$

$$P(x_i)=\sum_i P(x_i/y_j)P(y_j)$$

Then

$$I(X,Y)=\sum_{i.j}P(x_i,y_j)$$

$$=\sum_{i.j}P(x_i,y_j)\log_2\left[\frac{1}{P(x_i)}\right]$$

$$-\sum_{i.j}P(x_i,y_j)\log_2\left[\frac{1}{P(x_i/y_j)}\right]$$

$$\sum_{i.j}P(x_i,y_j)\log_2\left[\frac{1}{P(x_i)}\right]$$

$$=\sum_i\left[P(x_i/y_j)P(y_j)\right]\log_2\frac{1}{P(x_i)}$$

$$\sum_i P(x_i)\log_2\frac{1}{P(x_i)}=H(X)$$

$$I(X,Y)=H(X)-H(X/Y)$$

Where $H(X/Y)=\sum_{i,j}P(x_i,y_j)\log_2\frac{1}{P(x_i/y_j)}$

is usually called the equivocation. In a sense, the equivocation can be seen as the information lost in the noisy channel, and is a function of the backward conditional probability. The observation of an output symbol y_j provides $H(X)-H(X/Y)$ bits of information. This difference is the mutual information of the channel. *Mutual Information: Properties* Since

$$P(x_i/y_j)P(y_j)=P(y_j/x_i)P(x_i)$$

The mutual information fits the condition

$$I(X,Y)=I(Y,X)$$

And by interchanging input and output it is also true that

$$I(X,Y)=H(Y)-H(Y/X)$$

Where

$$H(Y)=\sum_j P(y_j)\log_2\frac{1}{P(y_j)}$$

This last entropy is usually called the noise entropy. Thus, the information transferred through the channel is the difference between the output entropy and the noise entropy. Alternatively, it can be said that the channel mutual information is the difference between the number of bits needed for determining a given input symbol before knowing the corresponding output symbol, and the number of bits needed for determining a given input symbol after knowing the corresponding output symbol

$$I(X,Y)=H(X)-H(X/Y)$$

As the channel mutual information expression is a difference between two quantities, it seems that this parameter can adopt negative values. However, and is spite of the fact that for some $y_j, H(X/y_j)$ can be larger than $H(X)$, this is not possible for the average value calculated over all the outputs:

$$\sum_{i,j}P(x_i,y_j)\log_2\frac{P(x_i/y_j)}{P(x_i)}=\sum_{i,j}P(x_i,y_j)\log_2\frac{P(x_i,y_j)}{P(x_i)P(y_j)}$$

Then

$$-I(X,Y)=\sum_{i,j}P(x_i,y_j)\frac{P(x_i)P(y_j)}{P(x_i,y_j)}\le 0$$

Because this expression is of the form

$$\sum_{i=1}^{M}P_i\log_2\left(\frac{Q_i}{P_i}\right)\le 0$$

The above expression can be applied due to the factor $P(x_i)P(y_j),$ which is the product of two probabilities, so that it behaves as the quantity Q_i, which in this expression is a dummy variable that fits the condition $\sum_i Q_i\le 1$. It can be concluded that

the average mutual information is a non-negative number. It can also be equal to zero, when the input and the output are independent of each other. A related entropy called the joint entropy is defined as

$$H(X,Y)=\sum_{i,j}P(x_i,y_j)\log_2\frac{1}{P(x_i,y_j)}$$

$$=\sum_{i,j}P(x_i,y_j)\log_2\frac{P(x_i)P(y_j)}{P(x_i,y_j)}$$

$$+\sum_{i,j}P(x_i,y_j)\log_2\frac{1}{P(x_i)P(y_j)}$$

Theorem 1.5: Entropies of the binary erasure channel (BEC) The BEC is defined with an alphabet of two inputs and three outputs, with symbol probabilities. $P(x_1)=\alpha$ *and* $P(x_2)=1-\alpha,$ and transition probabilities

$P(y_3/x_2)=1-p$ *and* $P(y_2/x_1)=0,$

and $P(y_3/x_1)=0$

and $P(y_1/x_2)=p$

and $P(y_3/x_2)=1-p$

Lemma 1.7. Given an arbitrary restricted time-discrete, amplitude-continuous channel whose restrictions are determined by sets F_n and whose density functions exhibit no dependence on the state s, let n be a fixed positive integer, and $p(x)$ an arbitrary probability density function on Euclidean *n*-space. $p(y\mid x)$ for the density $p_n(y_1,...,y_n\mid x_1,...x_n)$ and F *for* F_n. For any real number a, let

$$A=\left\{(x,y):\log\frac{p(y\mid x)}{p(y)}>a\right\}\qquad(1)$$

Then for each positive integer u, there is a code (u,n,λ) such that

$$\lambda\le ue^{-a}+P\{(X,Y)\notin A\}+P\{X\notin F\}\qquad(2)$$

Where

$P\{(X,Y)\in A\}=\int_A\cdots\int p(x,y)dxdy,\qquad p(x,y)=p(x)p(y\mid x)$

and

$$P\{X\in F\}=\int_F\cdots\int p(x)dx$$

Proof: A sequence $x^{(1)}\in F$ *such that*

$$P\left\{Y\in A_{x^1}\mid X=x^{(1)}\right\}\ge 1-\varepsilon$$

$$where\ A_x=\{y:(x,y)\varepsilon A\};$$

Choose the decoding set B_1 to be $A_{x^{(1)}}$. Having chosen $x^{(1)},.....,x^{(k-1)}$ and $B_1,...,B_{k-1}$, select $x^k\in F$ such that

$$P\left\{Y\in A_{x^{(k)}}-\bigcup_{i=1}^{k-1}B_i\mid X=x^{(k)}\right\}\ge 1-\varepsilon;$$

Set $B_k=A_{x^{(k)}}-\bigcup_{i=1}^{k-1}B_i$, If the process does not terminate in a finite number of steps, then the sequences $x^{(i)}$ and decoding sets $B_i,\ i=1,2,...,u,$ form the desired code. Thus assume that the process terminates after t steps. (Conceivably $t=0$). We will show $t\ge u$ by showing that $\varepsilon\le te^{-a}+P\{(X,Y)\notin A\}+P\{X\notin F\}$. We proceed as follows.

Let

$$B=\bigcup_{j=1}^{t}B_j.\quad(If\ \ t=0,\ take\ \ B=\phi).\ Then$$

$$P\{(X,Y)\in A\}=\int_{(x,y)\in A}p(x,y)dx\,dy$$

$$=\int_x p(x)\int_{y\in A_x}p(y\mid x)dy\,dx$$

$$=\int_x p(x)\int_{y\in B\cap A_x}p(y\mid x)dy\,dx+\int_x p(x)$$

IV. EXPERIMENTAL DESIGN

We evaluate the performance of our scheme and study various "what-if" scenarios through detailed simulation experiments. We compare our scheme against existing alternatives of using a least recently used (LRU) or a least frequently used (LFU) cache replacement strategy.

A. Algorithms

2) Ideals. Let A be a ring. Recall that an *ideal a* in A is a subset such that a is subgroup of A regarded as a group under addition;

$$a\in a,r\in A\Rightarrow ra\in A$$

The ideal generated by a subset S of A is the intersection of all ideals A containing a ----- it is easy to verify that this is in fact an ideal, and that it consist of all finite sums of the form $\sum r_i s_i$ with $r_i\in A, s_i\in S$. When $S=\{s_1,.....,s_m\}$, we shall write $(s_1,.....,s_m)$ for the ideal it generates.

Let a and b be ideals in A. The set $\{a+b\mid a\in a,b\in b\}$ is an ideal, denoted by $a+b$

. The ideal generated by $\{ab \mid a \in a, b \in b\}$ is denoted by ab. Note that $ab \subset a \cap b$. Clearly ab consists of all finite sums $\sum a_i b_i$ with $a_i \in a$ and $b_i \in b$, and if $a = (a_1, ..., a_m)$ and $b = (b_1, ..., b_n)$, then $ab = (a_1 b_1, ..., a_i b_j, ..., a_m b_n)$ .Let a be an ideal of A. The set of cosets of a in A forms a ring A/a , and $a \mapsto a + a$ is a homomorphism $\phi : A \mapsto A/a$. The map $b \mapsto \phi^{-1}(b)$ is a one to one correspondence between the ideals of A/a and the ideals of A containing a An ideal p if *prime* if $p \neq A$ and $ab \in p \Rightarrow a \in p$ or $b \in p$. Thus p is prime if and only if A/p is nonzero and has the property that $ab = 0, \quad b \neq 0 \Rightarrow a = 0,$ i.e., A/p is an integral domain. An ideal m is *maximal* if $m \neq\mid A$ and there does not exist an ideal n contained strictly between m and A. Thus m is maximal if and only if A/m has no proper nonzero ideals, and so is a field. Note that m maximal $\Rightarrow$ m prime. The ideals of $A \times B$ are all of the form $a \times b$, with a and b ideals in A and B. To see this, note that if c is an ideal in $A \times B$ and $(a,b) \in c$, then $(a,0) = (a,b)(1,0) \in c$ and $(0,b) = (a,b)(0,1) \in c$. This shows that $c = a \times b$ with

$$a = \{a \mid (a,b) \in c \ \ some \ \ b \in b\}$$

and

$$b = \{b \mid (a,b) \in c \ \ some \ \ a \in a\}$$

Let A be a ring. An A-algebra is a ring B together with a homomorphism $i_B : A \to B$. A *homomorphism of* A -algebra $B \to C$ is a homomorphism of rings $\varphi : B \to C$ such that $\varphi(i_B(a)) = i_C(a)$ for all $a \in A$. An A-algebra B is said to be *finitely generated* (or of *finite-type* over A) if there exist elements $x_1, ..., x_n \in B$ such that every element of B can be expressed as a polynomial in the x_i with coefficients in $i(A)$, i.e., such that the homomorphism $A[X_1, ..., X_n] \to B$ sending X_i to x_i is surjective. A ring homomorphism $A \to B$ is *finite*, and B is finitely generated as an A-module. Let k be a field, and let A be a k -algebra. If $1 \neq 0$ in A, then the map $k \to A$ is injective, we can identify k with its image, i.e., we can regard k as a subring of A . If 1=0 in a ring R, the R is the zero ring, i.e., $R = \{0\}$.

Polynomial rings. Let k be a field. A *monomial* in $X_1, ..., X_n$ is an expression of the form $X_1^{a_1} ... X_n^{a_n}, \quad a_j \in N$. The *total degree* of the monomial is $\sum a_i$. We sometimes abbreviate it by X^α, $\alpha = (a_1, ..., a_n) \in \square^n$. The elements of the polynomial ring $k[X_1, ..., X_n]$ are finite sums $\sum c_{a_1 ... a_n} X_1^{a_1} ... X_n^{a_n}, \quad c_{a_1 ... a_n} \in k, \quad a_j \in \square$ With the obvious notions of equality, addition and multiplication. Thus the monomials from basis for $k[X_1, ..., X_n]$ as a k -vector space. The ring $k[X_1, ..., X_n]$ is an integral domain, and the only units in it are the nonzero constant polynomials. A polynomial $f(X_1, ..., X_n)$ is *irreducible* if it is nonconstant and has only the obvious factorizations, i.e., $f = gh \Rightarrow g$ or h is constant. **Division in** $k[X]$. The division algorithm allows us to divide a nonzero polynomial into another: let f and g be polynomials in $k[X]$ with $g \neq 0$; then there exist unique polynomials $q, r \in k[X]$ such that $f = qg + r$ with either $r = 0$ or deg r < deg g . Moreover, there is an algorithm for deciding whether $f \in (g)$, namely, find r and check whether it is zero. Moreover, the Euclidean algorithm allows to pass from finite set of generators for an ideal in $k[X]$ to a single generator by successively replacing each pair of generators with their greatest common divisor.

***(Pure)* lexicographic** *ordering (lex).* Here monomials are ordered by lexicographic(dictionary) order. More precisely, let $\alpha = (a_1, ...a_n)$ and $\beta = (b_1, ...b_n)$ be two elements of $\square^n$; then $\alpha > \beta$ *and* $X^\alpha > X^\beta$ (lexicographic ordering) if, in the vector difference $\alpha - \beta \in \square$, the left most nonzero entry is positive. For example, $XY^2 > Y^3Z^4; \quad X^3Y^2Z^4 > X^3Y^2Z$. Note that this isn't quite how the dictionary would order them: it would put *XXXYYZZZZ* after *XXXYYZ* . *Graded reverse lexicographic order (grevlex).* Here monomials are ordered by total degree, with ties broken by reverse lexicographic ordering. Thus, $\alpha > \beta$ if $\sum a_i > \sum b_i$, or $\sum a_i = \sum b_i$ and in

$\alpha - \beta$ the right most nonzero entry is negative. For example:

$X^4Y^4Z^7 > X^5Y^5Z^4$ *(total degree greater)*

$XY^5Z^2 > X^4YZ^3, \quad X^5YZ > X^4YZ^2$.

Orderings on $k[X_1,...X_n]$. Fix an ordering on the monomials in $k[X_1,...X_n]$. Then we can write an element f of $k[X_1,...X_n]$ in a canonical fashion, by re-ordering its elements in decreasing order. For example, we would write

$f = 4XY^2Z + 4Z^2 - 5X^3 + 7X^2Z^2$

as

$f = -5X^3 + 7X^2Z^2 + 4XY^2Z + 4Z^2 \quad (lex)$

or

$f = 4XY^2Z + 7X^2Z^2 - 5X^3 + 4Z^2 \quad (grevlex)$

Let $\sum a_\alpha X^\alpha \in k[X_1,...,X_n]$, in decreasing order:

$f = a_{\alpha_0} X^{\alpha_0} + {}_{\alpha_1} X^{\alpha_1} + ..., \qquad \alpha_0 > \alpha_1 > ..., \quad \alpha_0 \neq 0$

Then we define.

- The *multidegree* of f to be multdeg(f)= α_0 ;
- The *leading coefficient of* f to be *LC(* f *)*= a_{α_0} ;
- The *leading monomial of* f to be *LM(* f *)* = X^{α_0} ;
- The *leading term of* f to be *LT(* f *)* = $a_{\alpha_0} X^{\alpha_0}$

For the polynomial $f = 4XY^2Z + ...,$ the multidegree is (1,2,1), the leading coefficient is 4, the leading monomial is XY^2Z, and the leading term is $4XY^2Z$. **The division algorithm in $k[X_1,...X_n]$.**

Fix a monomial ordering in $\mathbb{Z}^2$. Suppose given a polynomial f and an ordered set $(g_1,...g_s)$ of polynomials; the division algorithm then constructs polynomials $a_1,...a_s$ and r such that $f = a_1g_1 + ... + a_sg_s + r$ Where either $r = 0$ or no monomial in r is divisible by any of $LT(g_1),...,LT(g_s)$ **Step 1:** If $LT(g_1) \mid LT(f)$, divide g_1 into f to get

$f = a_1g_1 + h, \qquad a_1 = \dfrac{LT(f)}{LT(g_1)} \in k[X_1,...,X_n]$

If $LT(g_1) \mid LT(h)$, repeat the process until $f = a_1g_1 + f_1$ (different a_1) with $LT(f_1)$ not divisible by $LT(g_1)$. Now divide g_2 into f_1, and so on, until $f = a_1g_1 + ... + a_sg_s + r_1$ With $LT(r_1)$ not divisible by any $LT(g_1),...LT(g_s)$

Step 2: Rewrite $r_1 = LT(r_1) + r_2$, and repeat Step 1 with r_2 for f :

$f = a_1g_1 + ... + a_sg_s + LT(r_1) + r_3$ (different a_i's) **Monomial ideals.** In general, an ideal a will contain a polynomial without containing the individual terms of the polynomial; for example, the ideal $a = (Y^2 - X^3)$ contains $Y^2 - X^3$ but not Y^2 or X^3.

DEFINITION 1.5. An ideal a is *monomial* if

$\sum c_\alpha X^\alpha \in a \Rightarrow X^\alpha \in a$

all α with $c_\alpha \neq 0$.

PROPOSITION 1.3. Let a be a *monomial ideal*, and let $A = \{\alpha \mid X^\alpha \in a\}$. Then A satisfies the condition $\alpha \in A, \ \beta \in \mathbb{Z}^n \Rightarrow \alpha + \beta \in$ $(*)$ And a is the k -subspace of $k[X_1,...,X_n]$ generated by the $X^\alpha, \alpha \in A$. Conversely, of A is a subset of $\mathbb{Z}^n$ satisfying $(*)$, then the k-subspace a of $k[X_1,...,X_n]$ generated by $\{X^\alpha \mid \alpha \in A\}$ is a monomial ideal.

PROOF. It is clear from its definition that a monomial ideal a is the k -subspace of $k[X_1,...,X_n]$ generated by the set of monomials it contains. If $X^\alpha \in a$ and $X^\beta \in k[X_1,...,X_n]$.

If a permutation is chosen uniformly and at random from the $n!$ possible permutations in S_n, then the counts $C_j^{(n)}$ of cycles of length j are dependent random variables. The joint distribution of $C^{(n)} = (C_1^{(n)},...,C_n^{(n)})$ follows from Cauchy's formula, and is given by

$$P[C^{(n)}=c]=\frac{1}{n!}N(n,c)=1\left\{\sum_{j=1}^{n}jc_j=n\right\}\prod_{j=1}^{n}(\frac{1}{j})^{c_j}\frac{1}{c_j!},$$

for $c\in\mathbb{Z}_+^n$.

Lemma1.7 For nonnegative integers $m_{1,\ldots,}m_n$,

$$E\left(\prod_{j=1}^{n}(C_j^{(n)})^{[m_j]}\right)=\left(\prod_{j=1}^{n}\left(\frac{1}{j}\right)^{m_j}\right)1\left\{\sum_{j=1}^{n}jm_j\le n\right\} \qquad (1.4)$$

Proof. This can be established directly by exploiting cancellation of the form $c_j^{[m_j]}/c_j^{!}=1/(c_j-m_j)!$ when $c_j\ge m_j$, which occurs between the ingredients in Cauchy's formula and the falling factorials in the moments. Write $m=\sum jm_j$. Then, with the first sum indexed by $c=(c_1,\ldots c_n)\in\mathbb{Z}_+^n$ and the last sum indexed by $d=(d_1,\ldots,d_n)\in\mathbb{Z}_+^n$ via the correspondence $d_j=c_j-m_j$, we have

$$\begin{aligned}E\left(\prod_{j=1}^{n}(C_j^{(n)})^{[m_j]}\right)&=\sum_{c}P[C^{(n)}=c]\prod_{j=1}^{n}(c_j)^{[m_j]}\\&=\sum_{c:c_j\ge m_j\ \text{for all}\ j}1\left\{\sum_{j=1}^{n}jc_j=n\right\}\prod_{j=1}^{n}\frac{(c_j)^{[m_j]}}{j^{c_j}c_j!}\\&=\prod_{j=1}^{n}\frac{1}{j^{m_j}}\sum_{d}1\left\{\sum_{j=1}^{n}jd_j=n-m\right\}\prod_{j=1}^{n}\frac{1}{j^{d_j}(d_j)!}\end{aligned}$$

This last sum simplifies to the indicator $1(m\le n)$, corresponding to the fact that if $n-m\ge 0$, then $d_j=0$ for $j>n-m$, and a random permutation in S_{n-m} must have some cycle structure $(d_1,\ldots,d_{n-m})$. The moments of $C_j^{(n)}$ follow immediately as

$$E(C_j^{(n)})^{[r]}=j^{-r}1\{jr\le n\} \qquad (1.2)$$

We note for future reference that (1.4) can also be written in the form

$$E\left(\prod_{j=1}^{n}(C_j^{(n)})^{[m_j]}\right)=E\left(\prod_{j=1}^{n}Z_j^{[m_j]}\right)1\left\{\sum_{j=1}^{n}jm_j\le n\right\}, \qquad (1.3)$$

Where the Z_j are independent Poisson-distribution random variables that satisfy $E(Z_j)=1/j$

The marginal distribution of cycle counts provides a formula for the joint distribution of the cycle counts C_j^n, we find the distribution of C_j^n using a combinatorial approach combined with the inclusion-exclusion formula. (1.1)

Lemma 1.8. For $1\le j\le n$,

$$P[C_j^{(n)}=k]=\frac{j^{-k}}{k!}\sum_{l=0}^{[n/j]-k}(-1)^l\frac{j^{-l}}{l!} \qquad (1.1)$$

Proof. Consider the set I of all possible cycles of length j, formed with elements chosen from $\{1,2,\ldots n\}$, so that $|I|=n^{[j]/j}$. For each $\alpha\in I$, consider the "property" G_α of having α; that is, G_α is the set of permutations $\pi\in S_n$ such that α is one of the cycles of π. We then have $|G_\alpha|=(n-j)!$, since the elements of $\{1,2,\ldots,n\}$ not in α must be permuted among themselves. To use the inclusion-exclusion formula we need to calculate the term S_r, which is the sum of the probabilities of the r-fold intersection of properties, summing over all sets of r distinct properties. There are two cases to consider. If the r properties are indexed by r cycles having no elements in common, then the intersection specifies how rj elements are moved by the permutation, and there are $(n-rj)!1(rj\le n)$ permutations in the intersection. There are $n^{[rj]}/(j^r r!)$ such intersections. For the other case, some two distinct properties name some element in common, so no permutation can have both these properties, and the r-fold intersection is empty. Thus

$$\begin{aligned}S_r&=(n-rj)!1(rj\le n)\\&\times\frac{n^{[rj]}}{j^r r!}\frac{1}{n!}=1(rj\le n)\frac{1}{j^r r!}\end{aligned}$$

Finally, the inclusion-exclusion series for the number of permutations having exactly k properties is

$$\sum_{l\ge 0}(-1)^l\binom{k+l}{l}S_{k+l},$$

Which simplifies to (1.1) Returning to the original hat-check problem, we substitute j=1 in (1.1) to obtain the distribution of the number of fixed points of a random permutation. For $k=0,1,\ldots,n$,

$$P[C_1^{(n)}=k]=\frac{1}{k!}\sum_{l=0}^{n-k}(-1)^l\frac{1}{l!}, \qquad (1.2)$$

and the moments of $C_1^{(n)}$ follow from (1.2) with $j=1$. In particular, for $n\ge 2$, the mean and variance of $C_1^{(n)}$ are both equal to 1. The joint distribution of $(C_1^{(n)},\ldots,C_b^{(n)})$ for any $1\le b\le n$ has an expression similar to (1.7); this too can be

derived by inclusion-exclusion. For any $c=(c_1,...,c_b)\in \square_+^b$ with $m=\sum ic_i$,

$$P[(C_1^{(n)},...,C_b^{(n)})=c]$$

$$=\left\{\prod_{i=1}^{b}\left(\frac{1}{i}\right)^{c_i}\frac{1}{c_i!}\right\}\sum_{l\geq 0\ with\ \sum il_i\leq n-m}(-1)^{l_1+...+l_b}\prod_{i=1}^{b}\left(\frac{1}{i}\right)^{l_i}\frac{1}{l_i!}$$

The joint moments of the first b counts $C_1^{(n)},...,C_b^{(n)}$ can be obtained directly from (1.2) and (1.3) by setting $m_{b+1}=...=m_n=0$

The limit distribution of cycle counts

It follows immediately from Lemma 1.2 that for each fixed j, as $n\to\infty$,

$$P[C_j^{(n)}=k]\to\frac{j^{-k}}{k!}e^{-1/j},\quad k=0,1,2,...,$$

So that $C_j^{(n)}$ converges in distribution to a random variable Z_j having a Poisson distribution with mean $1/j$; we use the notation $C_j^{(n)}\to_d Z_j$ where $Z_j\square P_o(1/j)$ to describe this. Infact, the limit random variables are independent.

Theorem 1.6 The process of cycle counts converges in distribution to a Poisson process of $\square$ with intensity j^{-1}. That is, as $n\to\infty$,

$$(C_1^{(n)},C_2^{(n)},...)\to_d(Z_1,Z_2,...)\qquad(1.1)$$

Where the $Z_j, j=1,2,...,$ are independent Poisson-distributed random variables with $E(Z_j)=\frac{1}{j}$

Proof. To establish the converges in distribution one shows that for each fixed $b\geq 1$, as $n\to\infty$,

$$P[(C_1^{(n)},...,C_b^{(n)})=c]\to P[(Z_1,...,Z_b)=c]$$

Error rates

The proof of Theorem says nothing about the rate of convergence. Elementary analysis can be used to estimate this rate when $b=1$. Using properties of alternating series with decreasing terms, for $k=0,1,...,n,$

$$\frac{1}{k!}\left(\frac{1}{(n-k+1)!}-\frac{1}{(n-k+2)!}\right)\leq\left|P[C_1^{(n)}=k]-P[Z_1=k]\right|$$

$$\leq\frac{1}{k!(n-k+1)!}$$

It follows that

$$\frac{2^{n+1}}{(n+1)!}\frac{n}{n+2}\leq\sum_{k=0}^{n}\left|P[C_1^{(n)}=k]-P[Z_1=k]\right|\leq\frac{2^{n+1}-1}{(n+1)!}\qquad(1.11)$$

Since

$$P[Z_1>n]=\frac{e^{-1}}{(n+1)!}\left(1+\frac{1}{n+2}+\frac{1}{(n+2)(n+3)}+...\right)<\frac{1}{(n+1)!},\qquad(1.3)$$

We see from (1.11) that the total variation distance between the distribution $L(C_1^{(n)})$ of $C_1^{(n)}$ and the distribution $L(Z_1)$ of Z_1

V. EXPERIMENTAL SETUP

We perform our experiments using a custom built simulator. By default, we use a 55-node network mod- eled from a backbone network of a deployed IPTV ser- vice. The network has 70+ bi-directional links connect- ing these locations. We assume that all these links have equal capacity; however, we vary the actual value to understand the trade-off between disk capacity and link bandwidth. Similarly, we focus on the scenario where all VHOs have equal disk space, but also present re- sults where VHOs have heterogeneous disk capacities. To simulate user requests, we use one month's worth of VoD request traces from a nationally deployed VoD service. This trace contains requests to various types of videos, including music videos and trailers, TV shows, and full-length movies. For simplicity, we map these videos to four different video lengths: 5 minutes, 30 minutes, 1 hour, and 2 hours and assume that we need 100 MB, 500 MB, 1 GB, and 2 GB respectively for stor-ing them on disk. We assume that all videos are of stan-dard definition and stream at 2 Mbps. We start with a baseline scenario of a backbone network with each link being 1 Gbps and the aggregate disk capacity across all VHOs being 2 times the entire library size. We then vary many of the parameters to understand the various trade-offs and sensitivity of the system. In our experiments, we use our MIP formulation to place the videos in the VHOs. Unless stated otherwise, we update our MIP-based placement every week using the video requests in the previous 7 days as history. We assume time windows of 1 hour each across two time slices to capture the link constraints. For comparison, we simulate three alternative approaches: • Random + LRU: For each video, we place one copy at a randomly chosen VHO. The rest of disk space in each VHO is used for LRU cache.

• Random + LFU: This is similar to Random + LRU, but uses LFU instead of LRU.

• Top-K + LRU: We replicate top K videos at every VHO. The remaining videos are assigned randomly to one location. The remaining disk space at each location is used for LRU cache. This is a highly simplified version of [18].

Due to the local cache replacement in all the alternative approaches, if a VHO does not have a local copy of a requested video, it is difficult in practice for the VHO to find which VHO is best to fetch the video from. In our experiments, we assume the existence of an Oracle that can tell us the nearest location with a copy of the video, which is the best case for caches in terms of minimizing the total bandwidth consumed for the transfer.

Establish the asymptotics of $\mathrm{P}\left[A_n(C^{(n)})\right]$ under conditions (A_0) and (B_{01}), where

$$A_n(C^{(n)}) = \bigcap_{1\le i\le n} \bigcap_{r_i'+1\le j\le r_i} \left\{C_{ij}^{(n)} = 0\right\},$$

and $\zeta_i = (r_i'/r_{id}) - 1 = O(i^{-g'})$ as $i \to \infty$, for some $g' > 0$. We start with the expression

$$P[A_n(C^{(n)})] = \frac{P[T_{0m}(Z') = n]}{P[T_{0m}(Z) = n]} \prod_{\substack{1\le i\le n \\ r_i'+1\le j\le r_i}} \left\{1 - \frac{\theta}{ir_i}(1+E_{i0})\right\} \qquad (1.1)$$

$$P[T_{0n}(Z') = n] = \frac{\theta d}{n} \exp\left\{\sum_{i\ge 1}[\log(1+i^{-1}\theta d) - i^{-1}\theta d]\right\} \left\{1 + O(n^{-1}\varphi'_{\{1,2,7\}}(n))\right\} \qquad (1.2)$$

and

$$P[T_{0n}(Z') = n] = \frac{\theta d}{n} \exp\left\{\sum_{i\ge 1}[\log(1+i^{-1}\theta d) - i^{-1}\theta d]\right\} \left\{1 + O(n^{-1}\varphi_{\{1,2,7\}}(n))\right\} \qquad (1.3)$$

Where $\varphi'_{\{1,2,7\}}(n)$ refers to the quantity derived from Z'. It thus follows that $P[A_n(C^{(n)})] \square Kn^{-\theta(1-d)}$ for a constant K, depending on Z and the r_i' and computable explicitly from (1.1) – (1.3), if Conditions (A_0) and (B_{01}) are satisfied and if $\zeta_i^* = O(i^{-g'})$ from some $g' > 0$, since, under these circumstances, both $n^{-1}\varphi'_{\{1,2,7\}}(n)$ and $n^{-1}\varphi_{\{1,2,7\}}(n)$ tend to zero as $n \to \infty$. In particular, for polynomials and square free polynomials, the relative error in this asymptotic approximation is of order n^{-1} if $g' > 1$.

For $0 \le b \le n/8$ and $n \ge n_0$, with n_0

$$d_{TV}(L(C[1,b]), L(Z[1,b])) \le d_{TV}(L(\hat{C}[1,b]), L(\hat{Z}[1,b])) \le \varepsilon_{\{7,7\}}(n,b),$$

Where $\varepsilon_{\{7,7\}}(n,b) = O(b/n)$ under Conditions $(A_0), (D_1)$ and (B_{11}) Since, by the Conditioning Relation,

$$L(\hat{C}[1,b] \mid T_{0b}(C) = l) = L(\hat{Z}[1,b] \mid T_{0b}(Z) = l),$$

It follows by direct calculation that

$$d_{TV}(L(\hat{C}[1,b]), L(\hat{Z}[1,b])) = d_{TV}(L(T_{0b}(C)), L(T_{0b}(Z))) = \max_A \sum_{r\in A} P[T_{0b}(Z) = r] \left\{1 - \frac{P[T_{bn}(Z) = n-r]}{P[T_{0n}(Z) = n]}\right\} \qquad (1.4)$$

Suppressing the argument Z from now on, we thus obtain

$$d_{TV}(L(\hat{C}[1,b]), L(\hat{Z}[1,b])) = \sum_{r\ge 0} P[T_{0b} = r]\left\{1 - \frac{P[T_{bn} = n-r]}{P[T_{0n} = n]}\right\}_+$$

$$\le \sum_{r>n/2} P[T_{0b} = r] + \sum_{r=0}^{[n/2]} \frac{P[T_{0b} = r]}{P[T_{0b} = n]} \times \left\{\sum_{s=0}^{n} P[T_{0b} = s](P[T_{bn} = n-s] - P[T_{bn} = n-r]\right\}_+$$

$$\le \sum_{r>n/2} P[T_{0b} = r] + \sum_{r=0}^{[n/2]} P[T_{0b} = r] \times \sum_{s=0}^{[n/2]} P[T_{0b} = s] \frac{\{P[T_{bn} = n-s] - P[T_{bn} = n-r]\}}{P[T_{0n} = n]} + \sum_{s=0}^{[n/2]} P[T_{0b} = r] \sum_{s=[n/2]+1}^{n} P[T = s]P[T_{bn} = n-s]/P[T_{0n} = n]$$

The first sum is at most $2n^{-1}ET_{0b}$; the third is bound by

$$(\max_{n/2<s\le n} P[T_{0b} = s])/P[T_{0n} = n] \le \frac{2\varepsilon_{\{10.5(1)\}}(n/2,b)}{n} \frac{3n}{\theta P_\theta[0,1]},$$

$$\frac{3n}{\theta P_\theta[0,1]} 4n^{-2}\phi^*_{\{10.8\}}(n)\sum_{r=0}^{[n/2]} P[T_{0b}=r]\sum_{s=0}^{[n/2]} P[T_{0b}=s]\frac{1}{2}|r-s|$$

$$\leq \frac{12\phi^*_{\{10.8\}}(n)}{\theta P_\theta[0,1]}\frac{ET_{0b}}{n}$$

Hence we may take

$$\varepsilon_{\{7,7\}}(n,b) = 2n^{-1}ET_{0b}(Z)\left\{1+\frac{6\phi^*_{\{10.8\}}(n)}{\theta P_\theta[0,1]}\right\}P$$

$$+\frac{6}{\theta P_\theta[0,1]}\varepsilon_{\{10.5(1)\}}(n/2,b) \qquad (1.5)$$

Required order under Conditions $(A_0),(D_1)$ and (B_{11}), if $S(\infty)<\infty$. If not, $\phi^*_{\{10.8\}}(n)$ can be replaced by $\phi^*_{\{10.11\}}(n)$ in the above, which has the required order, without the restriction on the r_i implied by $S(\infty)<\infty$. Examining the Conditions $(A_0),(D_1)$ and (B_{11}), it is perhaps surprising to find that (B_{11}) is required instead of just (B_{01}); that is, that we should need $\sum_{l\geq 2} l\varepsilon_{il} = O(i^{-a_1})$ to hold for some $a_1>1$. A first observation is that a similar problem arises with the rate of decay of ε_{i1} as well. For this reason, n_1 is replaced by $\tilde{n}_1$. This makes it possible to replace condition (A_1) by the weaker pair of conditions (A_0) and (D_1) in the eventual assumptions needed for $\varepsilon_{\{7,7\}}(n,b)$ to be of order $O(b/n)$; the decay rate requirement of order $i^{-1-\gamma}$ is shifted from ε_{i1} itself to its first difference. This is needed to obtain the right approximation error for the random mappings example. However, since all the classical applications make far more stringent assumptions about the $\varepsilon_{i1}, l\geq 2$, than are made in (B_{11}). The critical point of the proof is seen where the initial estimate of the difference $P[T^{(m)}_{bn}=s]-P[T^{(m)}_{bn}=s+1]$. The factor $\varepsilon_{\{10.10\}}(n)$, which should be small, contains a far tail element from $\tilde{n}_1$ of the form $\phi^\theta_1(n)+u^*_1(n)$, which is only small if $a_1>1$, being otherwise of order $O(n^{1-a_1+\delta})$ for any $\delta>0$, since $a_2>1$ is in any case assumed. For $s\geq n/2$, this gives rise to a contribution of order $O(n^{-1-a_1+\delta})$ in the estimate of the difference $P[T_{bn}=s]-P[T_{bn}=s+1]$, which, in the remainder of the proof, is translated into a contribution of order $O(tn^{-1-a_1+\delta})$ for differences of the form $P[T_{bn}=s]-P[T_{bn}=s+1]$, finally leading to a contribution of order $bn^{-a_1+\delta}$ for any $\delta>0$ in $\varepsilon_{\{7.7\}}(n,b)$. Some improvement would seem to be possible, defining the function g by $g(w)=1_{\{w=s\}}-1_{\{w=s+t\}}$, differences that are of the form $P[T_{bn}=s]-P[T_{bn}=s+t]$ can be directly estimated, at a cost of only a single contribution of the form $\phi^\theta_1(n)+u^*_1(n)$. Then, iterating the cycle, in which one estimate of a difference in point probabilities is improved to an estimate of smaller order, a bound of the form

$$\left|P[T_{bn}=s]-P[T_{bn}=s+t]\right| = O(n^{-2}t+n^{-1-a_1+\delta})$$

for any $\delta>0$ could perhaps be attained, leading to a final error estimate in order $O(bn^{-1}+n^{-a_1+\delta})$ for any $\delta>0$, to replace $\varepsilon_{\{7.7\}}(n,b)$. This would be of the ideal order $O(b/n)$ for large enough b, but would still be coarser for small b.

With b and n as in the previous section, we wish to show that

$$\left|d_{TV}(L(C[1,b]),L(Z[1,b]))-\frac{1}{2}(n+1)^{-1}|1-\theta|E|T_{0b}-ET_{0b}|\right|$$

$$\leq \varepsilon_{\{7,8\}}(n,b),$$

Where $\varepsilon_{\{7.8\}}(n,b)=O(n^{-1}b[n^{-1}b+n^{-\beta_{12}+\delta}])$ for any $\delta>0$ under Conditions $(A_0),(D_1)$ and (B_{12}), with β_{12}. The proof uses sharper estimates. As before, we begin with the formula

$$d_{TV}(L(\tilde{C}[1,b]),L(\tilde{Z}[1,b]))$$

$$=\sum_{r\geq 0}P[T_{0b}=r]\left\{1-\frac{P[T_{bn}=n-r]}{P[T_{0n}=n]}\right\}_+$$

Now we observe that

$$\left|\sum_{r\geq 0} P[T_{0b}=r]\left\{1-\frac{P[T_{bn}=n-r]}{P[T_{0n}=n]}\right\}_+ - \sum_{r=0}^{[n/2]}\frac{P[T_{0b}=r]}{P[T_{0n}=n]}\right|$$
$$\times\left|\sum_{s=[n/2]+1}^{n} P[T_{0b}=s](P[T_{bn}=n-s]-P[T_{bn}=n-r])\right|$$
$$\leq 4n^{-2}ET_{0b}^2+(\max_{n/2<s\leq n} P[T_{0b}=s])/P[T_{0n}=n]$$
$$+P[T_{0b}>n/2]$$
$$\leq 8n^{-2}ET_{0b}^2+\frac{3\varepsilon_{\{10.5(2)\}}(n/2,b)}{\theta P_\theta[0,1]}, \qquad (1.1)$$

We have

$$\Big|\sum_{r=0}^{[n/2]}\frac{P[T_{0b}=r]}{P[T_{0n}=n]}$$
$$\times\Big(\left\{\sum_{s=0}^{[n/2]} P[T_{0b}=s](P[T_{bn}=n-s]-P[T_{bn}=n-r]\right\}_+$$
$$-\left\{\sum_{s=0}^{[n/2]} P[T_{0b}=s]\frac{(s-r)(1-\theta)}{n+1}P[T_{0n}=n]\right\}_+\Big)\Big|$$

$$\leq\frac{1}{n^2P[T_{0n}=n]}\sum_{r\geq 0}P[T_{0b}=r]\sum_{s\geq 0}P[T_{0b}=s]|s-r|$$
$$\times\left\{\varepsilon_{\{10.14\}}(n,b)+2(r\vee s)|1-\theta|n^{-1}\left\{K_0\theta+4\phi^*_{\{10.8\}}(n)\right\}\right\}$$
$$\leq\frac{6}{\theta nP_\theta[0,1]}ET_{0b}\varepsilon_{\{10.14\}}(n,b)$$
$$+4|1-\theta|n^{-2}ET_{0b}^2\left\{K_0\theta+4\phi^*_{\{10.8\}}(n)\right\}$$
$$\left(\frac{3}{\theta nP_\theta[0,1]}\right)\Big\}, \qquad (1.2)$$

The approximation in (1.2) is further simplified by noting that

$$\sum_{r=0}^{[n/2]}P[T_{0b}=r]\left|\left\{\sum_{s=0}^{[n/2]}P[T_{0b}=s]\frac{(s-r)(1-\theta)}{n+1}\right\}_+\right.$$
$$\left.-\left\{\sum_{s=0}P[T_{0b}=s]\frac{(s-r)(1-\theta)}{n+1}\right\}_+\right|$$
$$\leq\sum_{r=0}^{[n/2]}P[T_{0b}=r]\sum_{s>[n/2]}P[T_{0b}=s]\frac{(s-r)|1-\theta|}{n+1}$$
$$\leq|1-\theta|n^{-1}E(T_{0b}1\{T_{0b}>n/2\})\leq 2|1-\theta|n^{-2}ET_{0b}^2, \qquad (1.3)$$

and then by observing that

$$\sum_{r>[n/2]}P[T_{0b}=r]\left\{\sum_{s\geq 0}P[T_{0b}=s]\frac{(s-r)(1-\theta)}{n+1}\right\}$$
$$\leq n^{-1}|1-\theta|(ET_{0b}P[T_{0b}>n/2]+E(T_{0b}1\{T_{0b}>n/2\}))$$
$$\leq 4|1-\theta|n^{-2}ET_{0b}^2 \qquad (1.4)$$

Combining the contributions of (1.2) –(1.3), we thus find tha

$$\Big|\ d_{TV}(L(\tilde{C}[1,b]),L(\tilde{Z}[1,b]))$$
$$-(n+1)^{-1}\sum_{r\geq 0}P[T_{0b}=r]\left\{\sum_{s\geq 0}P[T_{0b}=s](s-r)(1-\theta)\right\}_+\Big|$$
$$\leq\varepsilon_{\{7.8\}}(n,b)$$
$$=\frac{3}{\theta P_\theta[0,1]}\left\{\varepsilon_{\{10.5(2)\}}(n/2,b)+2n^{-1}ET_{0b}\varepsilon_{\{10.14\}}(n,b)\right\}$$
$$+2n^{-2}ET_{0b}^2\left\{4+3|1-\theta|+\frac{24|1-\theta|\phi^*_{\{10.8\}}(n)}{\theta P_\theta[0,1]}\right\} \qquad (1.5)$$

The quantity $\varepsilon_{\{7.8\}}(n,b)$ is seen to be of the order claimed under Conditions $(A_0),(D_1)$ and (B_{12}), provided that $S(\infty)<\infty$; this supplementary condition can be removed if $\phi^*_{\{10.8\}}(n)$ is replaced by $\phi^*_{\{10.11\}}(n)$ in the definition of $\varepsilon_{\{7.8\}}(n,b)$, has the required order without the restriction on the r_i implied by assuming that $S(\infty)<\infty$. Finally, a direct calculation now shows that

$$\sum_{r\geq 0}P[T_{0b}=r]\left\{\sum_{s\geq 0}P[T_{0b}=s](s-r)(1-\theta)\right\}_+$$
$$=\frac{1}{2}|1-\theta|E|T_{0b}-ET_{0b}|$$

A. MIP Scheme Performance Evaluation

In the first experiment, we solve an MIP instance and place the videos according to that solution. Then, we play out the request log based on the solution. For each week, we construct a new parameter set based on previous week's demand history and recompute a new MIP instance. We use a link capacity of 1 Gbps for MIP constraints. The aggregate disk space is around 2 Figure 6: Aggregate bandwidth across all links, averaged over five minutes. times the entire library size. Of this, around 5% of the disk space at each VHO is used as an LRU cache. We compare our scheme with the three alternatives using the same disk space. For the top-K, we experimented with both K=10, and K=100. We present the results only for K=100, as K=10 was highly similar to Random + LRU. We use the first nine days' requests to warm up

the caches and run the tests using the remaining three weeks of requests.

Maximum Link Bandwidth. We identify the maximum link usage across all links at each time instant and show how it varies over the three-week period in Figure 5. We observe that, for the same amount of disk space, our proposed scheme can satisfy all requests using signif- icantly lower peak bandwidth. Specifically, the max- imum bandwidth needed for our case is 1364 Mbps, while the maximum value for Random+LRU is 2400 Mbps, 2366 Mbps for Random+LFU, and 2938 Mbps for Top-100+LRU. Note that the maximum value for our scheme is slightly larger than 1 Gbps, which is the link capacity provided for the MIP instance. This is be- cause each week introduces new videos, some of which we do not have a good estimate. While the small LRU cache helps absorb some of the errors in estimation, we believe a more sophisticated estimation strategy will help even further. We confirmed this through exper- iments assuming perfect knowledge of traffic pattern: the maximum bandwidth in that case always stayed within the constraint of 1 Gbps (See Table 3). *Total Bytes Transferred.* We calculate the total amount of network transfer where each video transfer is weighted by the video size and hop count. A good placement scheme will result in a small value because most of the requests would be satisfied locally or by nearby neigh- bors. We present the results in Figure 6. We calculate the aggregate transfers across all links and calculate the average over five minute intervals. We see similar trends to what was observed for the peak bandwidth. Our scheme consistently transfers fewer bytes compared to the other caching based schemes. LRU and LFU per- form almost identically. Surprisingly, Top-100 + LRU results in a higher peak utilization and total bytes trans- ferred. We attribute this to the fact that video popular- ity does not have a very high skew; even the less popular videos incur significant load. With the Top-100 videos occupying significant storage, there is less space for the LRU cache, and hence the performance becomes worse.

To analyze this further, we present the break-up of disk utilization in each VHO based on one solution to our MIP formulation in Figure 7. We characterize the top 100 videos as highly popular, the next 20% of videos as medium popular, and the remaining as un- popular. The highly popular videos occupy a relatively small port ion of the total disk space, while the medium popular videos occupy a significant proportion of the total disk space in the system (e.g., more than 30%). We also present the number of copies for each of the top 2000 videos in one of our MIP solutions (Figure 8). We observe that our solution intelligently places more copies for popular videos. This is to avoid remotely fetching frequently requested videos, which not only in-creases the overall cost (i.e., byte transfer), but also leads to link capacity violations. However, in our so- lution, even highly popular videos are not replicated everywhere (e.g., less than 30 VHOs have a copy of the 10th most popular video). On the other hand, we ob- serve that more than 1500 videos have multiple copies in the entire system. These two figures indicate that medium popular videos result in significant load and have to be dealt with intelligently. A given movie needs only a few copies— anywhere from 2 copies to 10 copies — but together these videos consume significant space. As a result, our solution carefully distributes copies of these videos across the VHOs. Unfortunately, caching schemes will have difficulty dealing with medium pop- ular videos, unless the cache size is sufficiently large.

Comparative LRU Cache Performance. We performed a simple experiment to understand the performance of a dynamic LRU cache replacement strategy. The ag- gregate disk space across all locations is around twice the entire library size while each location has the same disk space (and equal to the disk used in the MIP ex-periments). More than half of the space in each VHO was reserved for the LRU cache. We present the re- sults in Figure 9. As is clear from the figure, not only does the cache cycle, a large number of videos are not cachable because all the space in the cache is currently being used. Almost 20% of the requests could not be cached locally due to this. All this results in around 60% of requests being served by remote offices.

Other results. We ran other experiments, which we only summarize here for lack of space. We repeated the ex-periments with the aggregate disk space being 5 times the library size. We find that our approach still results in lower aggregate and peak bandwidth, although the difference between our scheme and the other approaches is smaller. We also experimented with the case of infi- nite link bandwidth to compare with the unconstrained link case [3]. Then, the maximum link bandwidth used by such a solution sometimes grows to more than twice the link bandwidth our scheme needs. In general, since a solution to the unconstrained problem does not have a limit on link usage, the maximum link usage can grow arbitrarily large, while our scheme finds the best trade-off, given the link and disk constraints.

Example 1.0. Consider the point $O=(0,...,0)\in\square^n$. For an arbitrary vector r, the coordinates of the point $x=O+r$ are equal to the respective coordinates of the vector $r: x=(x^1,...x^n)$ and $r=(x^1,...,x^n)$. The vector r such as in the example is called the position vector or the radius vector of the point x. (Or, in greater detail: r is the radius-vector of x w.r.t an origin O). Points are frequently specified by their radius-vectors. This presupposes the choice of O as the "standard origin". Let us summarize. We have

considered $\mathbb{R}^n$ and interpreted its elements in two ways: as points and as vectors. Hence we may say that we leading with the two copies of $\mathbb{R}^n$: $\mathbb{R}^n$ = {points}, $\mathbb{R}^n$ = {vectors}

Operations with vectors: multiplication by a number, addition. Operations with points and vectors: adding a vector to a point (giving a point), subtracting two points (giving a vector). $\mathbb{R}^n$ treated in this way is called an *n-dimensional affine space*. *(*An "abstract" affine space is a pair of sets , the set of points and the set of vectors so that the operations as above are defined axiomatically). Notice that vectors in an affine space are also known as "free vectors". Intuitively, they are not fixed at points and "float freely" in space. From $\mathbb{R}^n$ considered as an affine space we can precede in two opposite directions: $\mathbb{R}^n$ as an Euclidean space $\Leftarrow$ $\mathbb{R}^n$ as an affine space $\Rightarrow$ $\mathbb{R}^n$ as a manifold.Going to the left means introducing some extra structure which will make the geometry richer. Going to the right means forgetting about part of the affine structure; going further in this direction will lead us to the so-called "smooth (or differentiable) manifolds". The theory of differential forms does not require any extra geometry. So our natural direction is to the right. The Euclidean structure, however, is useful for examples and applications. So let us say a few words about it:

Remark 1.0. *Euclidean geometry.* In $\mathbb{R}^n$ considered as an affine space we can already do a good deal of geometry. For example, we can consider lines and planes, and quadric surfaces like an ellipsoid. However, we cannot discuss such things as "lengths", "angles" or "areas" and "volumes". To be able to do so, we have to introduce some more definitions, making $\mathbb{R}^n$ a Euclidean space. Namely, we define the length of a vector $a = (a^1, ..., a^n)$ to be

$$|a| := \sqrt{(a^1)^2 + ... + (a^n)^2} \qquad (1)$$

After that we can also define distances between points as follows:

$$d(A, B) := \left|\overrightarrow{AB}\right| \qquad (2)$$

One can check that the distance so defined possesses natural properties that we expect: is it always non-negative and equals zero only for coinciding points; the distance from A to B is the same as that from B to A (symmetry); also, for three points, A, B and C, we have $d(A,B) \le d(A,C) + d(C,B)$ (the "triangle inequality"). To define angles, we first introduce the scalar product of two vectors

$$(a, b) := a^1 b^1 + ... + a^n b^n \qquad (3)$$

Thus $|a| = \sqrt{(a,a)}$. The scalar product is also denote by dot: $a.b = (a,b)$, and hence is often referred to as the "dot product" . Now, for nonzero vectors, we define the angle between them by the equality

$$\cos\alpha := \frac{(a,b)}{|a||b|} \qquad (4)$$

The angle itself is defined up to an integral multiple of 2π . For this definition to be consistent we have to ensure that the r.h.s. of (4) does not exceed 1 by the absolute value. This follows from the inequality

$$(a,b)^2 \le |a|^2 |b|^2 \qquad (5)$$

known as the Cauchy–Bunyakovsky–Schwarz inequality (various combinations of these three names are applied in different books). One of the ways of proving (5) is to consider the scalar square of the linear combination $a + tb$, where $t \in R$. As $(a + tb, a + tb) \ge 0$ is a quadratic polynomial in t which is never negative, its discriminant must be less or equal zero. Writing this explicitly yields (5). The triangle inequality for distances also follows from the inequality (5).

Example 1.1. Consider the function $f(x) = x^i$ (the i-th coordinate). The linear function dx^i (the differential of x^i) applied to an arbitrary vector h is simply h^i .From these examples follows that we can rewrite df as

$$df = \frac{\partial f}{\partial x^1} dx^1 + ... + \frac{\partial f}{\partial x^n} dx^n, \qquad (1)$$

which is the standard form. Once again: the partial derivatives in (1) are just the coefficients (depending on x); $dx^1, dx^2, ...$ are linear functions giving on an arbitrary vector h its coordinates $h^1, h^2, ...,$ respectively. Hence

$$df(x)(h) = \partial_{hf(x)} = \frac{\partial f}{\partial x^1} h^1 + ... + \frac{\partial f}{\partial x^n} h^n, \qquad (2)$$

Theorem 1.7. Suppose we have a parametrized curve $t \mapsto x(t)$ passing through $x_0 \in \mathbb{R}^n$ at $t = t_0$ and with the velocity vector $x(t_0) = \upsilon$ Then

$$\frac{df(x(t))}{dt}(t_0) = \partial_\upsilon f(x_0) = df(x_0)(\upsilon) \qquad (1)$$

Proof. Indeed, consider a small increment of the parameter $t: t_0 \mapsto t_0 + \Delta t$, Where $\Delta t \mapsto 0$. On the other hand, we have $f(x_0 + h) - f(x_0) = df(x_0)(h) + \beta(h)|h|$ for an arbitrary vector h , where $\beta(h) \to 0$ when $h \to 0$. Combining it together, for the increment of $f(x(t))$ we obtain

$$f(x(t_0 + \Delta t) - f(x_0)$$
$$= df(x_0)(\upsilon.\Delta t + \alpha(\Delta t)\Delta t)$$
$$+\beta(\upsilon.\Delta t + \alpha(\Delta t)\Delta t).|\upsilon\Delta t + \alpha(\Delta t)\Delta t|$$
$$= df(x_0)(\upsilon).\Delta t + \gamma(\Delta t)\Delta t$$

For a certain $\gamma(\Delta t)$ such that $\gamma(\Delta t) \to 0$ when $\Delta t \to 0$ (we used the linearity of $df(x_0)$). By the definition, this means that the derivative of $f(x(t))$ at $t = t_0$ is exactly $df(x_0)(\upsilon)$. The statement of the theorem can be expressed by a simple formula:

$$\frac{df(x(t))}{dt} = \frac{\partial f}{\partial x^1}x^1 + ... + \frac{\partial f}{\partial x^n}x^n \qquad (2)$$

To calculate the value Of df at a point x_0 on a given vector υ one can take an arbitrary curve passing Through x_0 at t_0 with υ as the velocity vector at t_0 and calculate the usual derivative of $f(x(t))$ at $t = t_0$.

Theorem 1.8. For functions $f, g: U \to \mathbb{R}$, $U \subset \mathbb{R}^n$,

$$d(f+g) = df + dg \qquad (1)$$
$$d(fg) = df.g + f.dg \qquad (2)$$

Proof. Consider an arbitrary point x_0 and an arbitrary vector υ stretching from it. Let a curve $x(t)$ be such that $x(t_0) = x_0$ and $x(t_0) = \upsilon$.
Hence

$$d(f+g)(x_0)(\upsilon) = \frac{d}{dt}(f(x(t)) + g(x(t)))$$

at $t = t_0$ and

$$d(fg)(x_0)(\upsilon) = \frac{d}{dt}(f(x(t))g(x(t)))$$

at $t = t_0$ Formulae (1) and (2) then immediately follow from the corresponding formulae for the usual derivative Now, almost without change the theory generalizes to functions taking values in $\mathbb{R}^m$ instead of $\mathbb{R}$. The only difference is that now the differential of a map $F: U \to \mathbb{R}^m$ at a point x will be a linear function taking vectors in $\mathbb{R}^n$ to vectors in $\mathbb{R}^m$ (instead of $\mathbb{R}$) . For an arbitrary vector $h \in \mathbb{R}^n$,

$$F(x+h) = F(x) + dF(x)(h) + \beta(h)|h| \qquad (3)$$

Where $\beta(h) \to 0$ when $h \to 0$. We have $dF = (dF^1, ..., dF^m)$ and

$$dF = \frac{\partial F}{\partial x^1}dx^1 + ... + \frac{\partial F}{\partial x^n}dx^n$$
$$= \begin{pmatrix} \frac{\partial F^1}{\partial x^1} & \cdots & \frac{\partial F^1}{\partial x^n} \\ \cdots & \cdots & \cdots \\ \frac{\partial F^m}{\partial x^1} & \cdots & \frac{\partial F^m}{\partial x^n} \end{pmatrix} \begin{pmatrix} dx^1 \\ \cdots \\ dx^n \end{pmatrix} \qquad (4)$$

In this matrix notation we have to write vectors as vector-columns.

Theorem 1.9. For an arbitrary parametrized curve $x(t)$ in $\mathbb{R}^n$, the differential of a map $F: U \to \mathbb{R}^m$ (where $U \subset \mathbb{R}^n$) maps the velocity vector $\dot{x}(t)$ to the velocity vector of the curve $F(x(t))$ in $\mathbb{R}^m$:

$$\frac{dF(x(t))}{dt} = dF(x(t))(\dot{x}(t)) \qquad (1)$$

Proof. By the definition of the velocity vector,

$$x(t + \Delta t) = x(t) + \dot{x}(t).\Delta t + \alpha(\Delta t)\Delta t \qquad (2)$$

Where $\alpha(\Delta t) \to 0$ when $\Delta t \to 0$. By the definition of the differential,

$$F(x+h) = F(x) + dF(x)(h) + \beta(h)|h \qquad (3)|$$

Where $\beta(h) \to 0$ when $h \to 0$. we obtain

$$F(x(t + \Delta t)) = F(x + \underbrace{\dot{x}(t).\Delta t + \alpha(\Delta t)\Delta t}_{h})$$

$$= F(x) + dF(x)(\dot{x}(t)\Delta t + \alpha(\Delta t)\Delta t) +$$

$$\beta(\dot{x}(t)\Delta t + \alpha(\Delta t)\Delta t).|\dot{x}(t)\Delta t + \alpha(\Delta t)\Delta t|$$

$$= F(x) + dF(x)(\dot{x}(t)\Delta t + \gamma(\Delta t)\Delta t$$

For some $\gamma(\Delta t)\to 0$ when $\Delta t\to 0$. This precisely means that $dF(x)\dot{x}(t)$ is the velocity vector of $F(x)$. As every vector attached to a point can be viewed as the velocity vector of some curve passing through this point, this theorem gives a clear geometric picture of dF as a linear map on vectors.

Theorem 1.10 Suppose we have two maps $F:U\to V$ and $G:V\to W,$ where $U\subset \mathbb{R}^n, V\subset \mathbb{R}^m, W\subset \mathbb{R}^p$ (open domains). Let $F:x\mapsto y=F(x)$. Then the differential of the composite map $GoF:U\to W$ is the composition of the differentials of F and G:

$$d(GoF)(x)=dG(y)odF(x) \qquad (4)$$

Proof. We can use the description of the differential .Consider a curve $x(t)$ in $\mathbb{R}^n$ with the velocity vector $\dot{x}$. Basically, we need to know to which vector in $\mathbb{R}^p$ it is taken by $d(GoF)$. the curve $(GoF)(x(t)=G(F(x(t))$. By the same theorem, it equals the image under dG of the Anycast Flow vector to the curve $F(x(t))$ in $\mathbb{R}^m$. Applying the theorem once again, we see that the velocity vector to the curve $F(x(t))$ is the image under dF of the vector $\dot{x}(t)$. Hence $d(GoF)(\dot{x})=dG(dF(\dot{x}))$ for an arbitrary vector $\dot{x}$.

Corollary 1.0. If we denote coordinates in $\mathbb{R}^n$ by $(x^1,...,x^n)$ and in $\mathbb{R}^m$ by $(y^1,...,y^m)$, and write

$$dF=\frac{\partial F}{\partial x^1}dx^1+...+\frac{\partial F}{\partial x^n}dx^n \qquad (1)$$

$$dG=\frac{\partial G}{\partial y^1}dy^1+...+\frac{\partial G}{\partial y^n}dy^n, \qquad (2)$$

Then the chain rule can be expressed as follows:

$$d(GoF)=\frac{\partial G}{\partial y^1}dF^1+...+\frac{\partial G}{\partial y^m}dF^m, \qquad (3)$$

Where dF^i are taken from (1). In other words, to get $d(GoF)$ we have to substitute into (2) the expression for $dy^i=dF^i$ from (3). This can also be expressed by the following matrix formula:

$$d(GoF)=\begin{pmatrix}\frac{\partial G^1}{\partial y^1}&\dots&\frac{\partial G^1}{\partial y^m}\\ \dots&\dots&\dots\\ \frac{\partial G^p}{\partial y^1}&\dots&\frac{\partial G^p}{\partial y^m}\end{pmatrix}\begin{pmatrix}\frac{\partial F^1}{\partial x^1}&\dots&\frac{\partial F^1}{\partial x^n}\\ \dots&\dots&\dots\\ \frac{\partial F^m}{\partial x^1}&\dots&\frac{\partial F^m}{\partial x^n}\end{pmatrix}\begin{pmatrix}dx^1\\ \dots\\ dx^n\end{pmatrix} \qquad (4)$$

i.e., if dG and dF are expressed by matrices of partial derivatives, then $d(GoF)$ is expressed by the product of these matrices. This is often written as

$$\begin{pmatrix}\frac{\partial z^1}{\partial x^1}&\dots&\frac{\partial z^1}{\partial x^n}\\ \dots&\dots&\dots\\ \frac{\partial z^p}{\partial x^1}&\dots&\frac{\partial z^p}{\partial x^n}\end{pmatrix}=\begin{pmatrix}\frac{\partial z^1}{\partial y^1}&\dots&\frac{\partial z^1}{\partial y^m}\\ \dots&\dots&\dots\\ \frac{\partial z^p}{\partial y^1}&\dots&\frac{\partial z^p}{\partial y^m}\end{pmatrix}$$

$$\begin{pmatrix}\frac{\partial y^1}{\partial x^1}&\dots&\frac{\partial y^1}{\partial x^n}\\ \dots&\dots&\dots\\ \frac{\partial y^m}{\partial x^1}&\dots&\frac{\partial y^m}{\partial x^n}\end{pmatrix}, \qquad (5)$$

Or

$$\frac{\partial z^\mu}{\partial x^a}=\sum_{i=1}^{m}\frac{\partial z^\mu}{\partial y^i}\frac{\partial y^i}{\partial x^a}, \qquad (6)$$

Where it is assumed that the dependence of $y\in\mathbb{R}^m$ on $x\in\mathbb{R}^n$ is given by the map F , the dependence of $z\in\mathbb{R}^p$ on $y\in\mathbb{R}^m$ is given by the map $G,$ and the dependence of $z\in\mathbb{R}^p$ on $x\in\mathbb{R}^n$ is given by the composition GoF .

Definition 1.6. Consider an open domain $U\subset\mathbb{R}^n$. Consider also another copy of $\mathbb{R}^n$, denoted for distinction $\mathbb{R}^n_y$, with the standard coordinates $(y^1...y^n)$. A system of coordinates in the open domain U is given by a map $F:V\to U,$ where $V\subset\mathbb{R}^n_y$ is an open domain of $\mathbb{R}^n_y$, such that the following three conditions are satisfied :

(1) F is smooth;

(2) F is invertible;

(3) $F^{-1}:U\to V$ is also smooth

The coordinates of a point $x\in U$ in this system are the standard coordinates of $F^{-1}(x)\in\mathbb{R}^n_y$

In other words,

$$F:(y^1...,y^n)\mapsto x=x(y^1...,y^n) \qquad (1)$$

Here the variables $(y^1...,y^n)$ are the "new" coordinates of the point x

Example 1.2. Consider a curve in $\mathbb{R}^2$ specified in polar coordinates as

$$x(t):r=r(t),\varphi=\varphi(t) \qquad (1)$$

We can simply use the chain rule. The map $t\mapsto x(t)$ can be considered as the composition of the maps $t\mapsto(r(t),\varphi(t)),(r,\varphi)\mapsto x(r,\varphi)$. Then, by the chain rule, we have

$$\dot{x}=\frac{dx}{dt}=\frac{\partial x}{\partial r}\frac{dr}{dt}+\frac{\partial x}{\partial\varphi}\frac{d\varphi}{dt}=\frac{\partial x}{\partial r}\dot{r}+\frac{\partial x}{\partial\varphi}\dot{\varphi} \qquad (2)$$

Here $\dot{r}$ and $\dot{\varphi}$ are scalar coefficients depending on t, whence the partial derivatives $\partial x/\partial r,\partial x/\partial\varphi$ are vectors depending on point in $\mathbb{R}^2$. We can compare this with the formula in the "standard" coordinates: $\dot{x}=e_1\dot{x}+e_2\dot{y}$. Consider the vectors $\partial x/\partial r,\partial x/\partial\varphi$. Explicitly we have

$$\frac{\partial x}{\partial r}=(\cos\varphi,\sin\varphi) \qquad (3)$$

$$\frac{\partial x}{\partial\varphi}=(-r\sin\varphi,r\cos\varphi) \qquad (4)$$

From where it follows that these vectors make a basis at all points except for the origin (where $r=0$). It is instructive to sketch a picture, drawing vectors corresponding to a point as starting from that point. Notice that $\partial x/\partial r,\partial x/\partial\varphi$ are, respectively, the velocity vectors for the curves $r\mapsto x(r,\varphi)$ $(\varphi=\varphi_0\ fixed)$ and $\varphi\mapsto x(r,\varphi)$ $(r=r_0\ fixed)$. We can conclude that for an arbitrary curve given in polar coordinates the velocity vector will have components $(\dot{r},\dot{\varphi})$ if as a basis we take $e_r:=\partial x/\partial r,e_\varphi:=\partial x/\partial\varphi$:

$$\dot{x}=e_r\dot{r}+e_\varphi\dot{\varphi} \qquad (5)$$

A characteristic feature of the basis e_r,e_φ is that it is not "constant" but depends on point. Vectors "stuck to points" when we consider curvilinear coordinates.

Proposition 1.3. The velocity vector has the same appearance in all coordinate systems.

Proof. Follows directly from the chain rule and the transformation law for the basis e_i .In particular, the elements of the basis $e_i=\partial x/\partial x^i$ (originally, a formal notation) can be understood directly as the velocity vectors of the coordinate lines $x^i\mapsto x(x^1,...,x^n)$ (all coordinates but x^i are fixed). Since we now know how to handle velocities in arbitrary coordinates, the best way to treat the differential of a map $F:\mathbb{R}^n\to\mathbb{R}^m$ is by its action on the velocity vectors. By definition, we set

$$dF(x_0):\frac{dx(t)}{dt}(t_0)\mapsto\frac{dF(x(t))}{dt}(t_0) \qquad (1)$$

Now $dF(x_0)$ is a linear map that takes vectors attached to a point $x_0\in\mathbb{R}^n$ to vectors attached to the point $F(x)\in\mathbb{R}^m$

$$dF=\frac{\partial F}{\partial x^1}dx^1+...+\frac{\partial F}{\partial x^n}dx^n$$

$$(e_1,...,e_m)\begin{pmatrix}\frac{\partial F^1}{\partial x^1}\cdots\frac{\partial F^1}{\partial x^n}\\ \cdots\ \cdots\ \cdots\\ \frac{\partial F^m}{\partial x^1}\cdots\frac{\partial F^m}{\partial x^n}\end{pmatrix}\begin{pmatrix}dx^1\\ \cdots\\ dx^n\end{pmatrix}, \qquad (2)$$

In particular, for the differential of a function we always have

$$df=\frac{\partial f}{\partial x^1}dx^1+...+\frac{\partial f}{\partial x^n}dx^n, \qquad (3)$$

Where x^i are arbitrary coordinates. The form of the differential does not change when we perform a change of coordinates.

Example 1.3 Consider a 1-form in $\mathbb{R}^2$ given in the standard coordinates:

$A=-ydx+xdy$ In the polar coordinates we will have $x=r\cos\varphi,y=r\sin\varphi$, hence

$$dx=\cos\varphi dr-r\sin\varphi d\varphi$$

$$dy=\sin\varphi dr+r\cos\varphi d\varphi$$

Substituting into A, we get

$$A=-r\sin\varphi(\cos\varphi dr-r\sin\varphi d\varphi)$$

$$+r\cos\varphi(\sin\varphi dr+r\cos\varphi d\varphi)$$

$$=r^2(\sin^2\varphi+\cos^2\varphi)d\varphi=r^2d\varphi$$

Hence $A=r^2d\varphi$ is the formula for A in the polar coordinates. In particular, we see that this is again a 1-form, a linear combination of the differentials of coordinates with functions as coefficients. Secondly, in a more conceptual way, we can define a 1-form in

a domain U as a linear function on vectors at every point of U :

$$\omega(\upsilon) = \omega_1\upsilon^1 + \ldots + \omega_n\upsilon^n, \qquad (1)$$

If $\upsilon = \sum e_i\upsilon^i$, where $e_i = \partial x / \partial x^i$. Recall that the differentials of functions were defined as linear functions on vectors (at every point), and

$$dx^i(e_j) = dx^i\left(\frac{\partial x}{\partial x^j}\right) = \delta^i_j \qquad (2)$$

at every point x.

Theorem 1.9. For arbitrary 1-form ω and path γ, the integral $\int_\gamma \omega$ does not change if we change parametrization of γ provide the orientation remains the same.

Proof: Consider $\left\langle \omega(x(t)), \frac{dx}{dt'} \right\rangle$ and $\left\langle \omega(x(t(t'))), \frac{dx}{dt'} \right\rangle$ As

$$\left\langle \omega(x(t(t'))), \frac{dx}{dt'} \right\rangle = \left| \left\langle \omega(x(t(t'))), \frac{dx}{dt'} \right\rangle . \frac{dt}{dt'} \right|,$$

B. Storage and Bandwidth Tradeoffs

To understand the trade-off between storage and band-width, we identify how much disk space is needed to find a feasible solution to the MIP, given the link capacity. In Figure 10, we show the feasibility region (where we can serve all the requests without violating disk and link constraints) when we vary the link capacity. Note that the minimum aggregate disk space in the system must be as large as the entire library size, to store at least one copy of each movie (the bottom line in Fig- ure 10). When each link has a capacity of 0.5 Gbps, and all VHOs have the same amount of disk (denoted by "Uniform Disk"), we need at least 5 times more disk than what is needed to store one copy of each movie. We also observe that if we increase link capacity, then we can satisfy all the requests with much smaller disk. We also consider the case where there are three dif- ferent types of VHOs. Based on the number of sub- scribers at individual VHOs, we identify 12 large VHOs, 19 medium VHOs, and 24 small VHOs. In our exper- iments, a large VHO has twice the disk of a medium VHO, which in turn has twice the disk of a small VHO. The middle line in Figure 10 corresponds to the case of non-uniform VHOs. We observe that compared to the uniform VHO case, we need significantly smaller aggre- gate disk space to satisfy all the requests. Specifically, with 0.5 Gbps links, the total disk we need is less than 3 times the entire library size (vs. 5 for the uniform VHO case). This is because the majority of requests originate from those large VHOs and some of the medium VHOs. With a larger disk, these VHOs can serve more videos locally. Not surprisingly, as we increase the link ca- pacity, the gap between uniform and non-uniform cases decreases and converges to the library size.

C. Scalability

In the next set of experiments, we vary the library size and request load and investigate how the system resource requirement varies accordingly. In the first ex- periment, we fix the library size and increase the num- ber of requests, while maintaining the same popularity distribution as the trace, which can be approximated by a combination of zipf with an exponential drop-off [8]. We also maintain the diurnal patterns as we scale the request intensity. In Figure 11, we plot the minimum capacity for each link to serve all requests without vi- olating constraints. We show three cases with different disk capacities at each VHO. Obviously, we need larger link capacity with smaller disk or more requests (e.g., larger user base). However, we observe that the growth rate for link capacity is slightly smaller than the growth rate of traffic intensity. This is because a local copy of a highly popular video is able to handle many requests without using the network. We next experiment with growth in the VoD library. We analyzed the logs by Cha et al. [8] and found that the skew parameter stays similar even for the larger li- braries (e.g., 250K videos). We simplified the process of trace generation by only sampling from a zipf distribu-tion rather than a combination of zipf and exponential drop-off. We set the disk size at each VHO such that the aggregate disk space is around 3 times the library size. For ease of comparison, we use the same total number of requests regardless of the library size. We present the results in Figure 12 where we see two in- teresting trends. First the link bandwidth needed for a feasible solution (with 3x disk) drops as the library size increases. This is due to the combination of increased disk space and distribution of requests to videos. Sec- ond, we see that the number of active flows in the peak (i.e., total number of requests for videos) goes up with library size. This is because as the number of videos increases, for the same total number of requests, the distribution of requests is more dispersed.

D. Caching subsets

In this experiment we examine the effect of complementary caching on the performance of the MIP solution. We vary the amount of cache as a percentage of the disk space at each VHO and add that space to each VHO. We run the experiment for one week's worth of requests and measure the peak link utilization and the average aggregate bytes transferred. The results are shown in Figure 13. As

expected, both the peak and the aggregate decrease with increasing cache size. The reduction is significant as we go from no cache to 5% cache. The reduction, however, is not as significant as we increase the amount of cache further. This result shows that while a cache is important to handle errors in estimation or sudden changes in popularity, it is more important to get the placement correct.

E. Topology

We investigate how different topologies affect the ca-pacity required to meet all requests. In addition to the backbone network used in the previous sections, we consider two hypothetical networks: tree and full mesh (where each pair of nodes has a direct link). We also consider an artificial topology generated by combining Sprint and Abovenet topologies from the RocketFuel traces [17]. In each experiment, we use the same amount of aggregate disk across all VHOs, set at 3x the library size. In Table 2, we present our experimental results. As expected, we observe that with more links, we can serve all requests with lower link capacity. For instance, 1 Gbps capacity for each link is more than sufficient in the backbone case, while we need more than 2 Gbps for the tree topology. A thorough understanding of the im-pact of topology is an interesting topic of future work.

F. Time-varying request / response

As discussed in Section 6.2, we use only a small num-ber of time windows over which we evaluate the peak demand of videos requested and examine if the place- ment ensures we remain within the link capacity con- straints throughout the week. We performed experi- ments to understand the trade-offs on choosing the time window by varying it from 1 second to 1 day. Results are shown in Table 3. Using the peak request demand for 1-second time windows, we find that a feasible solu- tion exists for the MIP when each link is 0.5 Gbps. We also observe that the maximum link utilization during the corresponding time window is 0.5 Gbps. However, outside the peak window, some of the links are loaded up to 0.85 Gbps. This is because the MIP solution considers the link constraints only during the 1-second window, and due to highly varying request mix, the re-quest pattern during the window is not representative. As a result, the placement solution is not able to sat-isfy the link constraint outside of the window. Similar conclusions apply for 1-minute windows. On the other hand, with 1-day time windows, a feasible solution re-quires 2 Gbps links. But we observe that all links al- ways carry less than 1 Gbps of traffic during the entire 7-day period. Thus, 1-day windows lead to a significant over-estimation of the required link bandwidth. With 1-hour windows, the feasible solution requires 1 Gbps links. Correspondingly, the maximum link bandwidth over the entire 7-day period is also less than 1 Gbps. Thus, 1-hour windows seem to give the best trade-off between accurate estimation of the peak window and actual link utilization.

Running the placement often allows us to handle demand changes or estimation errors gracefully. However, each iteration incurs computational and transfer over- heads. We experimented with different placement fre- quencies to see how they affect performance. We show the maximum link bandwidth usage, total data transfer, and the fraction of requests served locally for the last two weeks. We consider both video size and number of hops to calculate total data trans- fer, as in (2). We do not use the complementary LRU- cache here. We observe that if we update the place- ment once in two weeks, then the maximum bandwidth grows significantly. This is because with less frequent updates, the error in demand estimation accumulates over time, as the current placement does not adapt to changes in the demand pattern. We observe that com- pared to weekly updates, daily updates only modestly improves the maximum bandwidth usage or miss ratio. However, by utilizing the most recent request history information, we can achieve around 10% improvement in terms of total data transfer. Using a 14-day history with weekly placement updates, we did not find any meaningful differences compared to a 7-day history. In Table 4, we quantify the error in our estimation of demand for new videos by presenting the performance when we have perfect knowledge. When we have perfect knowledge, our MIP-based solution always maintains the link utilization below capacity (< 1 Gbps), serves all the requests while using less total network band- width, and serves a majority of requests locally. On the other hand, without any estimation for new videos, we observe that the maximum bandwidth grows to over 8+ times the link capacity, and the resulting placement results in lots of remote transfers. Our simple estima- tion strategy, while not perfect, allows for performance comparable to when we have perfect knowledge. Cost of placement updates: One aspect to con- sider when determining the frequency of updates is the network transfer cost due to video migration for a new placement. We can slightly modify equation (9), such that we consider the cost of migration based on the pre- vious mapping (refer Section 5.2.2). In our experiments with this modified objective term, we find that around 2.5K videos need to be transferred between two place-ments. We argue that this is a small cost compared to the number of requests (e.g., 100Ks per day) and hence is quite manageable. In practice, we can even lower the update costs by piggybacking on requests. That is, when a new placement requires a particular VHO i to store video m, i can wait until a user requests m, fetch it and store a copy in the pinned portion of disk. We plan to investigate this aspect further in the future.

G. Generalized Overlay Models

3.1.1. Routing Overlay Model. The generalized routing overlay model is mainly due to RON [Andersen et al. 2001]. The purpose of a routing overlay is to provide an improved routing performance over that provided in the Internet. It does so by finding the best paths to destinations and quickly detecting failures to route around them. The overlay
nodes perform probes to its neighboring nodes in the overlay. These probes help detect whether an overlay link has failed and help the overlay node to measure performance statistics to each overlay node. To perform routing, the overlay network executes a routing protocol (usually link state) between the nodes in the overlay network. This allows the nodes to select best paths (based on some metrics, including latency, throughput, loss rate, and bandwidth) and route around failures in the network as they occur. The routing overlay generally performs better than the native layer routing because it is able to choose paths that are not available at the native layer (due to various reasons including policy restrictions and physical restrictions) and detect failures earlier than the native layer.

Service Overlay Model. The generalized service overlay model is mainly due to SON [Duan et al. 2003]. A service overlay network is deployed by a third-party service provider to provide an added service to its customers. The service provided is generally QoS, but may also be resilient routing, content delivery, and security. The service
overlay is usually provided some support (for a price) for its operation from the ISPs. For example, a QoS-enhancing service overlay network may purchase bandwidth and delivery guarantees for its traffic from ISPs [Duan et al. 2003].

Security Overlay Model. The generalized security overlay model (Figure 9) is due to SOS [Keromytis et al. 2002] and Mayday [Andersen 2003]. The primary purpose of a security overlay is to provide DoS-resistant communication for its participants. A security overlay typically utilizes two primary services for its operation (i) an anonymous routing service that hides the overlay traffic and the location of overlay nodes to prevent attacks on the nodes and internode traffic and (ii) a filtering service around the protected target to allow only overlay traffic through. These are usually combined with a user authentication service at the edges of the overlay network to identify legitimate users of the protection service. Depending on the type of authentication used, the routing service used, and the filtering parameters used, the overlay can provide different levels of protection for different types of users.

H. Advanced Generalized Overlay Topology Models

One of the first considerations in designing an overlay network is to choose an overlay topology to connect the various overlay nodes. The overlay topology chosen is generally application-specific to suit the particular requirements of the service it is designed to provide. The topology chosen has a direct impact on the performance, scalability and overhead, security, and failure resistance of the overlay network. In this section, we discuss topology design issues for different kinds of overlay networks.

Topology construction for routing overlays: Routing overlays are designed to improve the resiliency and performance over the underlying Internet paths. In routing overlays, the overlay topology and the routing protocol have a direct impact on the failure resistance and recovery of the overlay network [Li and Mohapatra 2004a]. One overlay topology design characteristic that directly affects failure resistance is path diversity. Path diversity can be considered in two layers: (1) overlay layer and (2) physical layer. Overlay-layer path diversity ensures that overlay-level paths share as few overlay-level links as possible. However, this does not necessarily translate into diversity at the physical layer (see Figure 11). If diverse overlay paths share the same physical links, the failure of the shared physical link breaks both the paths simultaneously. Therefore, building overlay networks with physical diversity provides more resistance to failures and improves recovery. Among the routing overlays, RON uses overlay-layer path diversity where it forms a full-mesh among all nodes that monitor their connectivity to every other overlay node. This enables a RON node to quickly detect path outages and to choose the best possible alternate path to reach a remote destination. On the other hand, the full-mesh design is not scalable and has a large operational overhead of $O(N2)$ where N is the number of overlay nodes in the system. One proposal to reduce this overhead is to randomly interconnect overlay nodes bound by certain degree constraints [Chu et al. 2000; Li and Mohapatra 2004a] but this comes with a tradeoff of reducing the degree of connectivity and alternate paths. An alternative approach for selection of overlay nodes in a physical topology-aware manner is proposed in Han et al. [2005]. In this work, the authors utilize offline analysis of a large quantity of traceroute and ping measurement data for this purpose. From the measurements, they calculate path diversity and latency as metrics to choose the best placement of overlay nodes. Path diversity is calculated as the number of overlapping nodes between the indirect overlay path through an overlay node and the direct native layer path. If two paths through the same node have a high correlation, the paths are assumed to be part of the same cluster. Based on this clustering scheme, a simple heuristic to choose the number and placement of overlay nodes is to choose one node at

random from each cluster. A similar clustering is applied for latency measurements between source destination pairs through an overlay node. The final heuristic uses latency measurements to pick the desired set of nodes from the set of nodes chosen by the path diversity heuristic. Similar ideas are explored in Cui et al. [2002], where the authors explore the problem of assigning backup paths with the aim of minimizing the joint probability of failure between the primary and backup paths. This probability is minimized by selecting backup paths with minimal correlation at the physical level to the primary path. Another underlying topology-aware topology construction scheme was proposed in Qiu et al. [2003]. In this work, the authors describe a distributed binning scheme to take advantage of the physical proximity between the nodes in an overlay. In this scheme, overlay nodes partition themselves into "bins" based on ping measurements to certain landmarks, for example, DNS servers. Nodes that are physically close to each other group themselves into bins (clusters). Clusters that are proximate to each other can then be clustered together to form a higher-order cluster, and so on. This binning scheme can be used to build an overlay topology with a better routing performance than random node selection. For example, a simple strategy used by the authors to build such an overlay is to have an overlay node pick half of its neighbors from the nodes closest to itself (approximated by picking them at random from its bin) and the other half at random (to maintain connectivity). Even this simple scheme was shown to perform better than constructing a random overlay network. The relationship between overlay topology and its performance is formalized in Zhang et al. [2006]. In particular, the authors identify three graph-theoretic metrics for the design of highly efficient routing overlay topologies: (i) characteristic path length (CPL); (ii) average cut size; and (iii) the weighted node degree sum. The first, characteristic path length (CPL) is defined as the median of the means of the shortest path lengths connecting each vertex to all other vertices. A small value of CPL provides a better routing performance because the path lengths to be traversed are smaller. The average cut size is a measure of the path diversity available in a graph. A larger cut size implies a more richly connected graph, and hence better performance. While the CPL and average cut size most directly affect routing performance, a third metric, weighted node degree sum (WNDS), is required to compensate for the difference in the utilization levels of links between nodes with different degrees. WNDS assigns larger weights for smaller degrees. Hence, a smaller value of WNDS corresponds to a richly connected graph, and thus a better performance. Based on these metrics, the authors propose a heuristic for node selection and topology design that aims to find a subgraph with small CPL and WNDS but a large cut size. Failure recovery is an important consideration in the topology construction of a routing overlay. When a failure occurs, the overlay chooses a new overlay-level route in an attempt to route around the failure. The success of routing around the failure depends on two factors: (i) the availability of alternate routes and (ii) the quality of available alternate routes. The availability of alternate routes depends directly on the topology used to build the overlay. The full mesh, for example, provides the maximum number of alternate routes, and hence a better success of failure recovery [Li and Mohapatra 2004a]. The quality of the available alternate routes depends on the correlation in the physical links comprising the alternate overlay paths with the physical links comprising the failed primary path. If there is a high correlation between the paths, the probability that the alternate path is also affected by the same failure as the primary path is high. This highlights the importance of minimum correlation between native layer paths during the construction of the overlay network. Han et al. [2005] develop heuristics for the construction of overlay networks with maximum path diversity (and hence minimum correlation) through a knowledge of the native layer topology. Similar ideas are explored by Cui et al. [2002], where the authors explore the problem of assigning backup paths with the aim of minimizing the joint probability of failure between the primary and backup paths. This probability is minimized by selecting backup paths with minimal correlation at the physical level to the primary path. The importance of native-layer topology awareness in overlay construction is also highlighted in Li and Mohapatra [2004a]. The authors show that topology-aware approaches have comparable
failure recovery ratios (the ratio of paths recovered to total number of failures) and recovered path penalties (the added penalty due to the selection of a less optimal recovery path) to the optimum cases at much less routing overhead when compared to full mesh.

Topology construction for service overlays: QoS-enhancing overlays like SON [Duan et al. 2003] and QRON [Li and Mohapatra 2004b] aim to provide service and bandwidth guarantees to applications. For this purpose, the generalized SON overlay for QoS requires the provisioning of bandwidth-guaranteed tunnels between overlay nodes. The actual topology to be chosen is less evident in this case, but it stands to argue that good end-to-end latency and scalability would be beneficial in SON design. In QRON [Li and Mohapatra 2004b] the authors propose the construction of a global-scale SON to provide QoS guarantees, hence scalability becomes an important consideration. The topology is constructed such that nodes within the same domain are fully meshed, and there is at least one tunnel between two neighboring domains. To improve the scalability of this architecture, the authors propose a hierarchical naming scheme which logically groups overlay nodes into disparate clusters (see Section 2.3).

Vieira and Liebeherr [2004b] propose methods to guide the topology design of a large-scale SON. They aim to minimize the costs associated with the interconnection of overlay nodes across multiple ISPs while providing the best access to end users. The problem is formulated as an optimization problem and proven to be NP-hard. The authors proposemultiple heuristics to approximate the optimal solution. Fan and Ammar [2006] study the problem of dynamically reconfiguring SON topologies to suit communication requirements. Such a reconfiguration can allow the SON to better adapt to changing communication requirements, but with an associated cost. The authors aim
to minimize the overall cost, which includes the cost of delivering traffic over the network and the cost of reconfiguring the overlay. The problem is shown to be NP-hard, and the authors propose several approximations for it.

Topology construction for security overlays: In security overlays (Section 3.1.3) traffic is tunneled through a series of overlay nodes to reach a protected destination. The primary objective of the overlay design is to provide DoS-resistant communication service to its users and to protect the overlay nodes from DoS attacks. Latency and improving end-to-end performance are not part of the overlay design. For example, the SOS [Keromytis et al. 2002] overlay topology follows a Chord ring [Stoica et al. 2003] which is vital to the DoS-resistant service provided by SOS. The SOS provides an effective solution to DoS defence, but the tradeoff is that the end-to-end latency takes a hit and is increased by a margin of 5 to 8 times over unicast latency. In WebSOS, Stavrou et al. [2005], implement, in addition to Chord, a CAN topology [Ratnasamy et al. 2001] with comparable performance metrics. In FONet [Kurian and Sarac 2007], we move away from using the circuitous routing used in previous proposals, to providing DoS defence using the SON model of service. A third-party OSP deploys overlay nodes with bandwidth guarantees to protect inter-FONet traffic from DoS attacks. Additionally, individual overlay nodes are protected against DoS attacks via the filtering of undesired traffic. By avoiding the need for circuitous routing and protecting the overlay
nodes from attack, FONet improves on the end-to-end performance of applications using these overlays without compromising security. In addition to DoS attacks, security overlays can also be vulnerable to compromise of overlay nodes. Since overlay nodes are generally made up of end systems deployed by users rather than core routers deployed by ISPs, they are in general more vulnerable to malicious attacks and intrusion. Some authors have explored mechanisms to enhance overlay networks with protection against such attacks. Walters et al. [2010],
employ data mining techniques to detect outliers in data reported by overlay nodes.2 The reasoning behind the proposed technique is that a malicious insider will have difficulty in lying consistently to (i) every other node (spatial outliers) and (ii) over time (temporal outliers). The problem of ensuring Byzantine resiliency and intrusion tolerance has received more attention in P2P overlays [Johansen et al. 2006; Sit and Morris 2002; Singh et al. 2004; Castro et al. 2002]. Wang [2005] and Wang et al. [2005] analyze the vulnerability of structured overlay topologies to two types of intrusion attacks: (i) penetration attacks that aim to directly attack a protected server by compromising nodes in the path to the protected server and (ii) proxy depletion attacks that aim to disable the overlay (proxy) network by compromising all nodes in it. The authors demonstrate that without added protection measures like proxy migration and reconfiguration,3 the overlay network can be vulnerable to penetration attacks. The vulnerability to penetration attacks is linear to the depth of the overlay topology that is, the number of nodes to traverse to reach the target. In proxy depletion attacks, the overlay topology plays an important part in its resiliency.
Specifically the authors observe that topologies with a lower vertex degree and balanced connectivity overall showed better resiliency to depletion attacks. With these requirements, Chord, with its high connectivity, is shown to perform poorly against depletion attacks. Topologies like CAN [Ratnasamy et al. 2001] and the de Brujin graph
with lower degrees and well-distributed connectivity are expected to exhibit better resiliency to depletion attacks [Frechette 2005]. For DoS attacks on the nodes, they demonstrate that overlay networks can provide scalable resistance to large-scale DoS attacks. The size of the overlay topology has a linear impact on the volume of attack
the overlay can withstand. In contrast to the results on latency observed by SOS and WebSOS [Stavrou et al. 2005], the authors also contend that the overlay network can *improve* the end-to-end performance of the end user due to the presence of long-lived TCP connections between overlay nodes. For a detailed explanation of the results, the reader is referred to Wang [2005].

I. Overlay Routing

The feasibility of overlay routing in improving the policy-based network layer in routing performance has been validated by several experimental [Anderson et al. 1999, 2001; Rahul et al. 2006; Zhang et al. 2006] and analytical results [Zhang et al. 2006; Qiu et al. 2003]. Much work has gone into enhancing overlay routing with an aim to improving its performance, enhancing its scalability, and reducing its overhead. Figure 12 shows an overview of some of the relevant work. In this section, we group these works under three topics, as follows:

Performance: The end-to-end performance of the overlay network depends on (i) routing protocols and path selection algorithms; (ii) path selection metrics; and (iii) minimal hit-time after failures.

Routing protocol and path selection: The most common routing and path selection approach in overlay routing is a link-state-based proactive approach [Andersen et al. 2001]. In this approach, we assume knowledge of global topology and link-state information. The shortest path is chosen (using Dijkstra's algorithm) for each flow-based on
the desired routing metric. Another approach is a link-state, based reactive approach proposed in Zhu et al. [2006]. In the reactive approach, link-state advertisements and global knowledge are assumed as in the proactive case. The difference is that the one chosen initially as the best one is maintained for the subsequent flows, unless the existing
path is no longer suited to provide certain performance guarantees. The authors contend that the reactive routing scheme leads to amore stable overlay routing scheme with fewer path changes. A recent study presented in Rahul et al. [2006] suggests that overlay paths typically have very high persistence (in the order of hours), suggesting that a reactive approach can be well-suited for most overlay needs without sacrificing performance. In QRON [Li and Mohapatra 2004b], the authors suggest two alternative path-selection algorithms that can provide load balancing in addition to satisfying the performance requirements. Another routing approach is the feedback-based approach proposed in Zhao et al. [2003]. In the feedback-based approach, each overlay node maintains a small number (usually two) of backup routes to every other overlay node in its routing table. When the overlay node detects that the primary path is lossy or not available, it switches to one of its backup routes. The backup routes are disjoint at the overlay level (not necessarily at the physical level), and hence have a reasonable probability of being available even if the primary path fails [Li and Mohapatra 2004a].

Metrics: The metrics used for path selection depend on the type of application the overlay intends to support. Choosing the correct metric is important to ensure the best performance for the routing process. The most common metric used is latency as proposed by RON [Andersen et al. 2001]. Latency is well-suited for most network applications, and is the most easily measured via path probes. RON additionally proposed two other metrics: path loss and throughput. Path loss is more difficult to measure as it has to be estimated from the two-way path loss probability of a probe packet. A simplifying assumption is that the bidirection loss is equally divided in both directions. Throughput can be estimated by using the TCP throughput equation based on the observed latency and loss rate. Zhu et al. [2006], propose the use of available (overlay) bandwidth as a metric for path selection. Available overlay bandwidth is defined as the minimum available bandwidth of all physical links comprising the overlay link.
The authors argue that latency, loss rate, and throughput are not directly indicative of traffic load in the path (latency depends primarily on propagation latency rather than traffic load, losses occur only after congestion has already happened, while throughput as measured, in RON is the TCP throughput which depends on factors like flow size and advertised window [Zhu et al. 2006]). However, available bandwidth is not easily measured, and estimation techniques have to be used which can incur added load in the network. Amir et al. [2005] propose a two-metric routing decision for VoIP applications. In VoIP, the goal is to maximize the number of packets that arrive with a certain threshold for playback at the receiver. Packets that arrive after the threshold are useless, but limited loss of packets can be tolerated. The metric proposed depends on both loss rate and latency of the link, and is called expected latency. In QRON, Li and Mohapatra [2004b] suggest two alternate metrics to use in conjunction with Dikjstra's shortest-path algorithm. Since, in QRON, path selection aims to satisfy the QoS requirement, the main criterion used is available bandwidth. Thus both metrics proposed in QRON are dependent on the available bandwidth and additionally on the available computational capacity of the nodes in the overlay.

Reducing hit time during failure recovery: We define hit time as the time period after the failure of a native link comprising an overlay path during which there is no data flow between a source-destination pair that was using the overlay path. Note that our definition assumes that the overlay *always* finds a new path, and is more general than the usual definition of hit time that defines it for a single overlay link [Seetharaman and Ammar 2006]. The more general definition helps us to account for multipath overlays and other techniques for reducing hit time. In the generalized routing overlay, hit time is comprised of the time to detect the fault, the route convergence time during which all the nodes in the network are aware of the fault and a new route is calculated, and the time taken to switch to the new route. In single-hop indirection overlays [Gummadi et al. 2004], since there is no routing convergence, the hit time is comprised of the time to detect the failure and the time to switch to the new route. Finally, in multipath overlays [Andersen et al. 2003], assuming that there is at least one redundant route between the source and destination that is active, the hit time is zero. In general routing overlays, the hit time depends directly on the frequency of active probing between the nodes. However, as shown by several authors [Keralapura et al. 2004; Seetharaman and Ammar 2006], higher probe rates lead to an increase in negative interactions between overlays and native traffic, referred to as

route flapping. An improved awareness of the native-layer routing process at the overlay layer can reduce the number of route flaps. Multipath overlays [Andersen et al. 2003] provide an answer to this problem at the cost of increased network traffic. Mesh routing is used in these overlays to add redundant packets into the network by duplicating traffic along multiple redundant routes. Multipath routing can be used in conjunction with general routing overlays to tradeoff between the hit-time and the redundancy required in the network.

Scalability and overhead: We consider overhead and scalability together because of their close correlation. In RON Andersen et al. [2001], state that the RON overlay is scalable to around 50 nodes. This limit in scalability is caused by the overhead introduced by the overlay operation. There are three components that make up this overhead:
(i) probing or ping overhead between overlay nodes; (ii) link-state broadcasts to announce up-to-date link state; and (iii) the computational overhead required at an overlay node to process state and data traffic. We ignore the computational overhead because it is not necessarily an overlay design problem (although as done in QRON [Li andMohapatra 2004b], residual computational capacity of the nodes can be considered in the routing process). In the generalized routing overlay model, the probing overhead is generally unavoidable (except in SOSR [Gummadi et al. 2004]). In link-state routing protocols (we do not consider feedback-based approaches because, as per Li and Mohapatra [2004a] they are less scalable than link-state approaches) the link state advertisements are also required. The overhead imposed by these two factors becomes excessive in RON due to the full-mesh overlay topology used. It has been shown that a 50-node full-mesh overlay introduces about 30 Kbps routing overhead [Andersen et al. 2001]. There have been two general approaches to solving this scalability problem: (i) approaches that deal with flat topologies and (ii) approaches that deal with hierarchical topologies. Flat topologies like the full-mesh are not scalable, hence several other topologies have been proposed by researchers [Li and Mohapatra 2004a]. Li and Mohapatra [2004a] analyze several such topologies for their routing overhead and conclude that there are several topologies available which, unlike full-mesh, can scale linearly with overlay size. The impact of these alternate topologies on routing performance is quantified in Rewaskar and Kaur [2004]. They observe through largescale Internet measurements that reducing the degree of connectivity of the overlay topology by a factor of 2 reduces the overhead by a factor of nearly 4. However, this reduction in degree affects the availability of paths with lower latency and loss rate than the default path by a 40% and 30% percent, respectively. Similar observations for probing duration show that doubling the probing duration reduces overhead by half while generating stale routing information for 10% and 30% for latency and loss rates, respectively. The tradeoff is apparent, while full mesh provides the best performance, alternative topologies with a lesser degree of connectivity can provide acceptable results for most requirements. Another approach to reducing the overhead would be to reduce the amount of monitoring required to maintain the overlay network. The task of monitoring O(N2) paths can be reduced to the task of monitoring k (k is approximately O(log N)) linearly independent paths [Chen et al. 2004]. A detailed discussion of the algorithms is beyond the scope of this survey, and we avoid further discussion. The overlay topologies considered so far have been flat, requiring every overlay node to have a complete global picture to perform overlay routing. A viable alternative is to use a hierarchical topology (Figure 13). The hierarchical approach is an axial method to enhance the scalability of routing protocols (e.g., OSPF) in the Internet. Hierarchical methods depend on the ability to "aggregate" routing information without incurring significant penalties associated with the loss of information. These ideas are extended to overlay networks in QRON [Li and Mohapatra 2004b]. The authors propose a hierarchical organization with nodes organized into clusters based on their proximity. Clusters are further organized into higher-level clusters, and so on, with Level-1 clusters forming the highest level of aggregation. Link state advertisements are made only within the cluster, and gateway nodes in each cluster aggregate the local information to form a full-mesh topology connecting them. A similar clustering approach is proposed in Kostic and Vahdat [2002]. They observe that the hierarchical approach exponentially improves network overhead and scalability, while its performance penalty is within 15% of the optimal. A third approach to improving the scalability of overlay networks is the single-hop overlay indirection approach [Andersen et al. 2001; Gummadi et al. 2004; Han et al. 2005]. We discussed this approach in detail previously in Section 2.4.

J. Overlay and Nonoverlay Traffic

An important consideration in the operation of routing overlay networks is their interactions with each other and nonoverlay traffic. Since different overlays and background traffic share many of the same network resources, they may compete for these resources with each other. There are two different types of interactions: (i) interactions between overlay routing and underlay routing and (ii) interactions between different overlay routing schemes. For an example of how the interaction between overlay and underlay routing can affect both layers negatively, consider Figures 14, 15, and 16 (Note: In these figures, the native layer or underlay is at the bottom, and the overlay layer is at the top). In Figure 14,

when the native link BD (at the bottom) fails, the overlay recovers and switches from path ADE to path ACGHE. Some time later, the native layer recovers from the failure and chooses a new path ABCD to route from A to D (Figure 15). The overlay layer, due to its probing, detects a new, shorter path to E and hence switches back to ADE. Suppose that this switch causes the native link BC to be overloaded. If traffic engineering is present in the domain, it may switch to AC instead of ABC in the original routing tables (Figure 16). The overlay now detects that path ACE is more advantageous than ADE, and hence switches to ACE. These route flaps are caused due to the lack of interaction between the different layers during recovery. Several authors have explored these interactions and how to accommodate for them in overlay architectures. Qiu et al. [2003] study interactions between different overlay and underlay routing in the absence of traffic engineering (i.e., the underlying network-level routing remains the same). They observe that in the absence of traffic engineering (TE), different routing schemes can interact well with each other. Overlay routing can provide nearly optimal latency at the expense of added network cost and link utilization. The goal of traffic engineering, however, is to reduce network costs by varying network-level routing to changes in traffic conditions. In this context, both overlay routing and traffic engineering continually adapt to each other to minimize their respective cost functions. The authors separately study this effect with traffic engineering provided by an OSPF route optimizer and an MPLS optimizer. In the OSPF case, they observe that there is a significant performance degradation, so much so that the nonoptimized case outperforms the optimized case. MPLS, based traffic engineering on the other hand performs significantly better than the OSPF case. The authors contend that the MPLS optimizer has much more fine-grained control over overlay traffic, as opposed to the OSPF optimizer which allows it to adjust its routing matrix more effectively. Liu et al. [2005] further explore the effects of overlay routing on traffic engineering. They model the interaction between the conflicting objectives of overlay routing and traffic engineering as a noncooperative, nonzero sum two-player game.4 The authors demonstrate that when modeled this way the game has a stable and unique Nash equilibrium point.5 A discussion of the Nash equilibrium is beyond the scope of this article; but note that Nash equilibrium is not an efficient state for either player. The selfish behavior of overlay routing and its interaction with traffic engineering as
a result degrades the performance of regular users and the underlay network as a whole. Keralapura et al. [2004], examine the interactions between overlay routing and traffic engineering in the presence of unexpected events like failures. They identify that overlay routing violates two basic assumptions made by ISPs in their traffic engineering policies: (i) traffic demand is relatively constant over shorter periods of time and (ii) changes in the path within a domain do not impact traffic demands. This leads to frequent oscillations in routing and makes traffic matrices more dynamic and difficult to predict. Traffic engineering is also responsible for implementing load-balancing policies of the ISP. Overlay routing can bypass these load-balancing requirements violating the ISPs load-balancing intent. Another consideration is the case of a single overlay that spans multiple AS domains. The effects of a failure of a physical link in one of the domains can cause the overlay to switch its paths, affecting the load on links in other domains. This is an undesired effect and can lead to oscillations in domains due to an event in another domain.

Seetharaman and Ammar [2006] study the behavior of networks in which the overlay layer and the native layer operate independently of each other. The problem arises due to lack of coordination in the failure-recovery mechanisms of the two layers. As described in Section 3.1.1, a routing overlay uses probe messages to detect failures (often quicker than the native layer) and finds its own alternate path. This independence results in a dual rerouting at the two layers, and hence to oscillations. Completely avoiding this recovery process is also not ideal and can lead to partitioned overlays. Hence, the authors suggest an improved "awareness" of the underlay recovery at the overlay layer. The overlay layer suppresses or delays its own rerouting process in deference to the recovery process at the native layer. The authors demonstrate that, in this way, the number of oscillations due to dual rerouting are reduced. However, the tradeoff is the time required to recover from the failure at the overlay layer, which may now
depend on the speed at which the underlay recovers. Seetharaman et al. [2007] suggest preemptive strategies for each layer that try to prevent the other layers from needing to make a readjustment which could potentially lead to oscillations. For the overlay layer, this amounts to making available bandwidth measurements on the native-layer
links and limiting overlay bandwidth consumption to be less than the available bandwidth. For the native layer, the strategy takes into account the fact that overlay routing will choose paths with the lowest latencies. So during its TE calculations, the native layer tries to ensure that the hop-count of paths between source-destination pairs stays
within a small threshold of its previous value. Keralapura et al. [2005] examine the interactions that occur when multiple overlays coexist. Since each overlay takes independent routing decisions without knowledge of the other, oscillations in network load and routes are both possible. The authors identify three conditions for such oscillations : (i) a failure or path degradation event which triggers the recovery event in overlays; (ii) a shared link between the two overlays; and (iii) correlation between probe periods

of the overlays. The aggressiveness of each overlay plays an important part in the probability that two overlays will get synchronized and go into oscillations. Generally, an increase in aggressiveness (defined as the the ratio of probe timeout and probe interval) translates to a higher probability of synchronization. Oscillations are detrimental to both overlay and nonoverlay traffic. A careful consideration of the impact of a new overlay needs to be done before deployment. Overlays potentially can also benefit from a common probing layer similar to that suggested by Nakao et al [2003].

VI. LOAD AWARE ANYCAST CDN ARCHITECTURE

In this section we first describe the workings of a load-aware anycast CDN and briefly discuss the pros and cons of this approach vis-a-vis more conventional CDN architectures. We also give an informal description of the load-balancing algorithm required for our approach before describing it more formally in later sections.

A. Load aware anycast CDN

Figure 1 shows a simplified view of a load-aware anycast CDN. We assume a single autonomous system (AS) in which IP anycast is used to reach a set of CDN nodes distributed within the AS. For simplicity we show two such CDN nodes, *A* and *B* in Figure 1. In the rest of this article, we use the terms "CDN node" and "content server" interchangeably. We further assume that the AS in question has a large footprint in the country or region in which it will be providing CDN service; for example, in the US, Tier-1 ISPs have this kind of footprint.1 Our article investigates synergistic benefits of having control over the PEs of a CDN. We note that these assumptions are both practical, and, more importantly, a recent study of IP anycast [Ballani et al. 2006] has shown this to be the ideal type of deployment to ensure good proximity properties.2 Figure 1 also shows the route controller component that is central to our approach [Van der Merwe et al. 2006; Verkaik et al. 2007]. The route controller activates routes with provider edge (PE) routers in the CDN provider network. As described in Van der Merwe et al. [2006], this mechanism involves pre-installed MPLS tunnels routes for a destination IP address (an anycast address in our case) from each PE to every other PE. Thus, to activate a route from a given PE *PEi* to another PE *PEj*, the controller only needs to signal *PEi* to start using an appropriate MPLS label. In particular, route change does not involve any other routers and in this sense is an atomic operation. The route controller can use this mechanism to influence the anycast routes selected by the ingress PEs. For example, in Figure 1, to direct packets entering through PE *PE*1 to the CDN node B, the route controller would signal *PE*1 to activate the MPLS tunnel from *PE*1 to *PE*5; to send these packets to node A instead, the route controller would similarly activate the tunnel from *PE*1 to *PE*0. For our purposes, the route controller takes as inputs, ingress load from the PEs at the edge of the network, server load from the CDN nodes for which it is performing redirection, and the cost matrix of reaching a given CDN server from a given PE to compute the routes in accordance with the algorithms described in Section 3. The load-aware anycast CDN then functions as follows (with reference to Figure 1). All CDN nodes that are configured to serve the same content (*A* and *B*), advertise the same IP anycast address into the network via BGP (respectively through *PE*0 and *PE*5). *PE*0 and *PE*5 in turn advertise the anycast address to the route controller, which is responsible to advertise the (appropriate) route to all other PEs in the network (*PE*1 to *PE*4). These PEs in turn advertise the route via eBGP sessions with peering routers (*PEa* to *PEd*) in neighboring networks so that the anycast address becomes reachable throughout the Internet (in the figure represented by access networks *I* and *II*). Request traffic for content on a CDN node will follow the reverse path. Thus, a request will come from an access network, and enter the CDN provider network via one of the ingress routers *PE*1 to *PE*4. In the simple setup depicted in Figure 1, such request traffic will then be forwarded to either *PE*0 or *PE*5 en-route to one of the CDN nodes. Based on the two load feeds (ingress PE load and server load) provided to the route controller, it can decide which ingress PE (*PE*1 to *PE*4) to direct to which egress PE (*PE*0 or *PE*5). By assigning different PEs to appropriate CDN nodes, the route controller can minimize the network costs of processing the demand and distributed the load among the CDN nodes. In summary, our approach utilizes the BGP-based proximity property of IP anycast to deliver clients packets to nearest ingress PEs. These external portions of the paths of anycast packets are determined purely by inter-AS BGP routes. Once packets enter the provider network, it is the route controller that decides where these packets will be delivered through mapping ingress PEs to content servers. The route controller makes these decisions taking into account both network proximity of the internal routes and server loads.

Let p be a rational prime and let $K = \mathbb{Q}(\zeta_p)$. We write ζ for ζ_p or this section. Recall that K has degree $\varphi(p) = p-1$ over $\mathbb{Q}$. We wish to show that $O_K = \mathbb{Z}[\zeta]$. Note that ζ is a root of $x^p - 1$, and thus is an algebraic integer; since O_K is a ring we have that $\mathbb{Z}[\zeta] \subseteq O_K$. We give a proof without assuming unique factorization of ideals. We begin

with some norm and trace computations. Let j be an integer. If j is not divisible by p, then ζ^j is a primitive p^{th} root of unity, and thus its conjugates are $\zeta, \zeta^2, ..., \zeta^{p-1}$. Therefore

$$Tr_{K/\mathbb{Q}}(\zeta^j) = \zeta + \zeta^2 + ... + \zeta^{p-1} = \Phi_p(\zeta) - 1 = -1$$

If p does divide j, then $\zeta^j = 1$, so it has only the one conjugate 1, and $Tr_{K/\mathbb{Q}}(\zeta^j) = p - 1$ By linearity of the trace, we find that

$$Tr_{K/\mathbb{Q}}(1-\zeta) = Tr_{K/\mathbb{Q}}(1-\zeta^2) = ...$$
$$= Tr_{K/\mathbb{Q}}(1-\zeta^{p-1}) = p$$

We also need to compute the norm of $1-\zeta$. For this, we use the factorization

$$x^{p-1} + x^{p-2} + ... + 1 = \Phi_p(x)$$
$$= (x-\zeta)(x-\zeta^2)...(x-\zeta^{p-1});$$

Plugging in $x = 1$ shows that

$$p = (1-\zeta)(1-\zeta^2)...(1-\zeta^{p-1})$$

Since the $(1-\zeta^j)$ are the conjugates of $(1-\zeta)$, this shows that $N_{K/\mathbb{Q}}(1-\zeta) = p$ The key result for determining the ring of integers O_K is the following.

LEMMA 1.9

$$(1-\zeta)O_K \cap \mathbb{Z} = p\mathbb{Z}$$

Proof. We saw above that p is a multiple of $(1-\zeta)$ in O_K, so the inclusion $(1-\zeta)O_K \cap \mathbb{Z} \supseteq p\mathbb{Z}$ is immediate. Suppose now that the inclusion is strict. Since $(1-\zeta)O_K \cap \mathbb{Z}$ is an ideal of $\mathbb{Z}$ containing $p\mathbb{Z}$ and $p\mathbb{Z}$ is a maximal ideal of $\mathbb{Z}$, we must have $(1-\zeta)O_K \cap \mathbb{Z} = \mathbb{Z}$ Thus we can write $1 = \alpha(1-\zeta)$ For some $\alpha \in O_K$. That is, $1-\zeta$ is a unit in O_K.

COROLLARY 1.1 For any $\alpha \in O_K$, $Tr_{K/\mathbb{Q}}((1-\zeta)\alpha) \in p\mathbb{Z}$

PROOF. We have

$$Tr_{K/\mathbb{Q}}((1-\zeta)\alpha) = \sigma_1((1-\zeta)\alpha) + ... + \sigma_{p-1}((1-\zeta)\alpha)$$
$$= \sigma_1(1-\zeta)\sigma_1(\alpha) + ... + \sigma_{p-1}(1-\zeta)\sigma_{p-1}(\alpha)$$
$$= (1-\zeta)\sigma_1(\alpha) + ... + (1-\zeta^{p-1})\sigma_{p-1}(\alpha)$$

Where the σ_i are the complex embeddings of K (which we are really viewing as automorphisms of K) with the usual ordering. Furthermore, $1-\zeta^j$ is a multiple of $1-\zeta$ in O_K for every $j \neq 0$. Thus $Tr_{K/\mathbb{Q}}(\alpha(1-\zeta)) \in (1-\zeta)O_K$ Since the trace is also a rational integer.

PROPOSITION 1.4 Let p be a prime number and let $K = \mathbb{Q}(\zeta_p)$ be the p^{th} cyclotomic field. Then $O_K = \mathbb{Z}[\zeta_p] \cong \mathbb{Z}[x]/(\Phi_p(x))$; Thus $1, \zeta_p, ..., \zeta_p^{p-2}$ is an integral basis for O_K.

PROOF. Let $\alpha \in O_K$ and write

$\alpha = a_0 + a_1\zeta + ... + a_{p-2}\zeta^{p-2}$ With $a_i \in \mathbb{Q}$. Then

$$\alpha(1-\zeta) = a_0(1-\zeta) + a_1(\zeta - \zeta^2) + ...$$
$$+ a_{p-2}(\zeta^{p-2} - \zeta^{p-1})$$

By the linearity of the trace and our above calculations we find that $Tr_{K/\mathbb{Q}}(\alpha(1-\zeta)) = pa_0$ We also have

$Tr_{K/\mathbb{Q}}(\alpha(1-\zeta)) \in p\mathbb{Z}$, so $a_0 \in \mathbb{Z}$ Next consider the algebraic integer

$(\alpha - a_0)\zeta^{-1} = a_1 + a_2\zeta + ... + a_{p-2}\zeta^{p-3}$; This is an algebraic integer since $\zeta^{-1} = \zeta^{p-1}$ is. The same argument as above shows that $a_1 \in \mathbb{Z}$, and continuing in this way we find that all of the a_i are in $\mathbb{Z}$. This completes the proof.

Example 1.4 Let $K = \mathbb{Q}$, then the local ring $\mathbb{Z}_{(p)}$ is simply the subring of $\mathbb{Q}$ of rational numbers with denominator relatively prime to p. Note that this ring $\mathbb{Z}_{(p)}$ is not the ring $\mathbb{Z}_p$ of p-adic integers; to get $\mathbb{Z}_p$ one must complete $\mathbb{Z}_{(p)}$. The usefulness of $O_{K,p}$ comes from the fact that it has a particularly simple ideal structure. Let a be any proper ideal of $O_{K,p}$ and consider the ideal $a \cap O_K$ of O_K. We claim that $a = (a \cap O_K)O_{K,p}$; That is, that a is generated by the elements of a in $a \cap O_K$. It is clear from the definition of an ideal that $a \supseteq (a \cap O_K)O_{K,p}$. To prove the other inclusion, let α be any element of a. Then we can write $\alpha = \beta/\gamma$ where $\beta \in O_K$ and $\gamma \notin p$. In particular, $\beta \in a$ (since $\beta/\gamma \in a$ and a is an

ideal), so $\beta \in O_K$ and $\gamma \notin p$. so $\beta \in a \cap O_K$. Since $1/\gamma \in O_{K,p}$, this implies that $\alpha = \beta/\gamma \in (a \cap O_K)O_{K,p}$, as claimed.We can use this fact to determine all of the ideals of $O_{K,p}$. Let a be any ideal of $O_{K,p}$ and consider the ideal factorization of $a \cap O_K$ in O_K. write it as $a \cap O_K = p^n b$ For some n and some ideal b, relatively prime to p. we claim first that $bO_{K,p} = O_{K,p}$. We now find that

$a = (a \cap O_K)O_{K,p} = p^n bO_{K,p} = p^n O_{K,p}$ Since $bO_{K,p}$. Thus every ideal of $O_{K,p}$ has the form $p^n O_{K,p}$ for some n; it follows immediately that $O_{K,p}$ is noetherian. It is also now clear that $p^n O_{K,p}$ is the unique non-zero prime ideal in $O_{K,p}$. Furthermore, the inclusion $O_K \mapsto O_{K,p}/pO_{K,p}$ Since $pO_{K,p} \cap O_K = p$, this map is also surjection, since the residue class of $\alpha/\beta \in O_{K,p}$ (with $\alpha \in O_K$ and $\beta \notin p$) is the image of $\alpha\beta^{-1}$ in $O_{K/p}$, which makes sense since β is invertible in $O_{K/p}$. Thus the map is an isomorphism. In particular, it is now abundantly clear that every non-zero prime ideal of $O_{K,p}$ is maximal. To show that $O_{K,p}$ is a Dedekind domain, it remains to show that it is integrally closed in K. So let $\gamma \in K$ be a root of a polynomial with coefficients in $O_{K,p}$; write this polynomial as $x^m + \frac{\alpha_{m-1}}{\beta_{m-1}}x^{m-1} + \ldots + \frac{\alpha_0}{\beta_0}$ With $\alpha_i \in O_K$ and $\beta_i \in O_{K-p}$. Set $\beta = \beta_0\beta_1\ldots\beta_{m-1}$. Multiplying by β^m we find that $\beta\gamma$ is the root of a monic polynomial with coefficients in O_K. Thus $\beta\gamma \in O_K$; since $\beta \notin p$, we have $\beta\gamma/\beta = \gamma \in O_{K,p}$. Thus $O_{K,p}$ is integrally close in K.

COROLLARY 1.2. Let K be a number field of degree n and let α be in O_K then $N'_{K/\square}(\alpha O_K) = \left|N_{K/\square}(\alpha)\right|$

PROOF. We assume a bit more Galois theory than usual for this proof. Assume first that $K/\square$ is Galois. Let σ be an element of $Gal(K/\square)$. It is clear that $\sigma(O_K)/\sigma(\alpha) \cong O_{K/\alpha}$; since $\sigma(O_K) = O_K$, this shows that $N'_{K/\square}(\sigma(\alpha)O_K) = N'_{K/\square}(\alpha O_K)$. Taking the product over all $\sigma \in Gal(K/\square)$, we have $N'_{K/\square}(N_{K/\square}(\alpha)O_K) = N'_{K/\square}(\alpha O_K)^n$ Since $N_{K/\square}(\alpha)$ is a rational integer and O_K is a free $\square$ -module of rank n,

$O_K / N_{K/\square}(\alpha)O_K$ Will have order $N_{K/\square}(\alpha)^n$; therefore

$$N'_{K/\square}(N_{K/\square}(\alpha)O_K) = N_{K/\square}(\alpha O_K)^n$$

This completes the proof. In the general case, let L be the Galois closure of K and set $[L:K] = m$.

B. CDN Objectives and Benefits

We can summarize the goals of the architecture described above as follows: (i) To utilize the natural IP anycast proximity properties to reduce the distance traffic is carried towards the CDN's ISP; (ii) To react to overload conditions on CDN servers by steering traffic to alternative CDN servers; (iii) To minimize the disruption of traffic that results when ongoing sessions are being re-mapped to alternative CDN servers. Note that this means that "load-balancing" per server is not a specific goal of the algorithm: while CDN servers are operating within acceptable engineering loads, the algorithm should not attempt to balance the load. On the other hand, when overload conditions are reached, the system should react to deal with that, while not compromising proximity. A major advantage of our approach over DNS-based redirection systems is that the actual eyeball request is being redirected, as opposed to the local-DNS request in the case of DNS-based redirection. Further, with load-aware anycast, any redirection changes take effect very quickly, because PEs immediately start to route packets based on their updated routing table. In contrast, DNS caching by clients (despite short TTLs) typically results in some delay before redirection changes have an effect. The granularity of load distribution offered by our route control approach is at the PE level. For large tier-1 ISPs the number of PEs is typically in the high hundreds to low thousands. A possible concern for our approach is whether PE granularity will be sufficiently fine grained to adjust load in cases of congestion. Our results in Section 5 indicate that even with PE-level granularity we can achieve significant performance benefits in practice. Obviously, with enough capacity, no load balancing would ever be required. However, a practical platform needs to have load-balancing ability to cope with unexpected events such as flash crowd and node failures, and to flexibly react to even

more gradual demand changes because building up physical capacity of the platform is a very coarse-grain procedure. Our experiments will show that our architecture can achieve effective load balancing even under constrained resource provisioning. Before we describe and evaluate redirection algorithms that fulfill these goals, we briefly describe two other CDN-related functions enabled by our architecture that are not further elaborated upon in this article.

C. Long Lived Sessions

Despite increased distribution of rich media content via the Internet, the average Web object size remains relatively small [King 2006]. This means that download sessions for such Web objects will be relatively short lived with little chance of being impacted by any anycast re-mappings in our architecture. The same is, however, not true for long-lived sessions, for example, streaming or large file download [Van der Merwe et al. 2002]. (Both of these expectations are validated with our analysis of connections disruption count in Section 5.) In our architecture, we deal with this by making use of an additional application level redirection mechanisms *after* a particular CDN node has been selected via our load-aware IP Anycast redirection. This interaction is depicted in Figure 2. As before an eyeball will perform a DNS request that will be resolved to an IP Anycast address (*i* and *ii*). The eyeball will attempt to request the content using this address (*iii*), however, the CDN node will respond with an application level redirect (*iv*) [Van derMerwe et al. 2003] containing a unicast IP address associated with this CDN node, which the eyeball will use to retrieve the content (*v*). This unicast address is associated only with this CDN node, and the eyeball will therefore continue to be serviced by the same node regardless of routing changes along the way. While the additional overhead associated with application level redirection is clearly unacceptable when downloading small Web objects, it is less of a concern for long-lived sessions where the startup overhead is amortized. In parallel work, we proposed an alternative approach to handle extremely large downloads using anycast, without relying on HTTP redirection [Al-Qudah et al. 2009]. Instead, the approach in Al-Qudah et al. [2009] recovers from an disruption by reissuing the HTTP request for the remainder of the object using a range HTTP request. The CDN could then trigger these disruptions intentionally to switch the user to a different server mid-stream if the conditions change. However, that approach requires a browser extension. Recently, some CDNs started moving into utility (also known as cloud) computing arena, by deploying applications at the CDN nodes. In this environment, applications often form long-lived sessions that encompass multiple HTTP requests, with individual requests requiring the entire session state to execute correctly. Commercial application servers, including bothWeblogic and Websphere, allow servers to form a wide-area cluster where each server in the cluster can obtain the session state after successfully receiving any HTTP request in a session. Based on this feature, our approach for using anycast for request redirection can apply to this emerging CDN environment.

D. Network Congestion

As previously described, the load-aware CDN architecture only takes server load into account in terms of being "load-aware". (In other words, the approach uses network load information in order to effect the server load, but does not attempt to steer traffic away from network hotspots). The Route Control architecture, however, does allow for such traffic steering [Van der Merwe et al. 2006]. For example, outgoing congested peering links can be avoided by redirecting response traffic on the PE connecting to the CDN node (e.g., *PE*0 in Figure 1), while incoming congested peering links can be avoided by exchanging BGP Multi-Exit Discriminator (MED) attributes with appropriate peers [Van der Merwe et al. 2006]. We leave the full development of these mechanisms for future work.

E. Schemes and Metrics

We experiment with the following schemes and compare the performance.

—*Trace Playback* (PB). In this scheme we replayed all requests in the trace without any modification of server mappings. In other words, (PB) reflects the current CDN routing configuration.

—*Simple Anycast* (SAC). This is "native" Anycast, which represents an idealized proximity routing scheme, where each request is served at the geographically closest server.

—*Simple Load Balancing* (SLB). This scheme employs anycast to minimize the difference in load among all servers without considering the cost.

—*Advanced Load Balancing, Always* (ALB-A). This scheme always attempts to find a minimum cost mapping as described in Section 3.2.

—*ALB, On-overload* (ALB-O). This scheme aims to minimize connection disruptions as described in Section 3.3. Specifically, it normally only reassigns PEs currently mapped to overloaded servers and performs full remapping only if the cost reduction from full remapping would exceed 70%. In SAC, each PE is statically mapped to a server, and there is no change in the mappings across the entire experiment run. SLB and ALB-A recalculate the mappings every

_ seconds (the remapping interval). The initial _ value that we used to evaluate the different algorithms is 120 seconds. Later, in Section 5.5, we examine various values of _. We utilize the following metrics for performance comparison.

—*Server load.* We use the *number of concurrent requests* and *service data rate* at each server as measures of server load. A desirable scheme should keep the number below the capacity limit all the time.
—*Request air-miles.* We examine the *average miles* a request traverses within the CDN provider network before reaching a server as a proximity metric of content delivery within the CDN's ISP. A small value for this metric denotes small network link usage in practice.
—*Disrupted Connections and Over-Capacity Requests.* Another metric of redirection scheme quality is the *number of disrupted connections* due to remapping. Disruption occurs when a PE is remapped from server *A* to server *B*; the ongoing connections arriving from the PE may be disconnected because *B* may not have the connection information. Finally, we use the *number of over-capacity requests* as a metric to compare the ability of different schemes to prevent server overloading. A request is counted as over-capacity if it arrives at a server with existing concurrent requests already at or over the physical capacity limit. With our redirection scheme, a request may use a server different from the one used in the trace, and its response time may change, for example, depending on the server
load or capacity. In our experiments, we assume that the response time of each request is the same as the one in the trace no matter which server processes it as a result of our algorithms.

F. Rearchitecting Network

Research to circumvent current Internet limitations can be divided into those advocating a completely new architecture (clean-slate), and those defending an evolutionary approach due to incremental deployability concerns. From a research perspective, clean-slate design does not presume clean-slate deployment and aims at innovation through questioning fundamentals. A key question is to what extent a new paradigm thinking 'out-of-the-TCP/IP-box' for the future network is really necessary, e.g., as packet switching was to circuit switching in the 70's. The reasoning is based on the large scale use of the Internet for dissemination of data [15]. Tons of connected devices are generating and consuming content, without caring about the actual data source as long as integrity and authenticity are assured [17]. We can also observe this shift toward information-centric networking in the momentum of service oriented architectures (SOA) and infrastructures (SOI), XML routers, deep packet inspection (DPI), content delivery networks (CDN) and P2P overlay technologies. A common issue is the necessity to manage a huge quantity of data items, which is a quite different task than reaching a particular host. In today's Internet, forwarding decisions are made not only by IP routers, but also by middleboxes, VLAN switches, MPLS routers, DPIs, load balancers, mesh routing nodes and other cross-layer approaches. Moving down data-centric functions to the lower networking layers could be in tune with the trend in access and backbone technologies represented by the coupling of the dominant Ethernet access protocol and label switched all optical transport networks. More than an endless discussion around clean-slate design and actual network (r)evolution deployment, what we really need for the future Internetworking is 1) 'clean-slate thinking' beyond the TCP/IP heritage to foster innovation through questioning paradigms; and 2) feasibility work on an information-oriented infrastructure capable of supporting the actual and future demands over the network of networks.

G. Information Oriented Internetworking

Until recently, research in a new generation Internet has prompted architectural proposals (e.g., TRIAD, FARA, Plutarch, UIP, IPNL, HIP, ROFL) that mainly aimed at solving the host reachability problem by providing more flexible, expressive, and comprehensive naming and addressing frameworks than the Internet hierarchical IP address space. A move towards information interconnection can be observed in recent projects addressing the future Internet such as PSIRP [8], 4Ward [12], Trilogy, ICT's FIRE and other activities within EU FP7 and NSF FIND. Data-centric architectural proposals have started to emerge (e.g., DOA, i3, DONA, Haggle) and are similar in spirit to 'peer-to-peer', 'content-delivery', 'sensor' and 'delay-tolerant' networks.

H. Information Centric

In information/content/data - oriented/centric networks, the flow of messages is driven by the nodes that have expressed their interest and the information identifiers of the messages, rather than by explicit destination host interface names (IP addresses) assigned by senders. Reachability of destinations is not anymore delimited by topological information but by the notion of information scope [21]. Having the data location hidden makes the semantics of what defines a sender or receiver of data less relevant than the data itself, intuitively providing enhanced security (e.g., DDoS mitigation) and bridging connectivity challenged underlying networks (e.g., DTN). The publish/subscribe paradigm [11] is a promising trend to instantiate the so sought modern communication API [10] for information-centric systems. Pub/sub systems have been widely studied and employed for specific event-dissemination applications and have appealing characteristics like spatial and temporal decoupling [11]. In Internet-scale topic-based pub/sub internetworking [8, 20, 21], topics are unique information identifiers at different layers in the architecture accommodating different granularities and semantics (e.g., messages, channels, documents) to support every type of communications (e.g., transactional, interactive, etc.). The suitability and

benefits of moving the pub/sub layer downwards into the networking stack is one of the challenging objectives of interest-driven architectures where naming, routing, forwarding and addressing get fresh semantics (see Table 1). In the envisioned internetworking service bus, information objects are first-class citizens introducing a new global unmanaged namespace. A form of publication metadata information is required to enable the self-authentication of the data, fragmentation, scope delimitation, inter-domain policies, in-network management, caching, and so on [8]. A global namespace for data items enables caching capabilities for every type of communications. In comparison, caching over TCP/IP is costly and application-specific. In case of non-mutable information objects caching becomes trivial, whereas for streaming applications, caching can be seen as long in-network buffers. Hence, the architecture natively plays the role of current CDNs and avoids redundant traffic over network links [1]. Furthermore, a new namespace for information objects could easily accommodate multi- , any-, con- and unicast types of communication in addition to novel forms of network coding to increase the network's efficiency and resilience.

I. Reference Architectures

In this section, we briefly introduce the basics of two recent design choices from the EU FP7 Publish/Subscribe Internetworking Routing Paradigm (PSIRP) project [8] we selected as reference architectures. The RTFM architecture [20] gets its name from the functional building blocks that are recursively applied. The rendezvous (R) is in charge of matching subscriptions to publications and information scoping. The topology (T) management creates and maintains (sub-optimal) delivery trees used for traffic forwarding, acting both proactively (optimization) and re-actively (on-demand). The forwarding (F) functions perform the actual datagram delivery based on label switching techniques. Finally, mediation (M) refers to the node-tonode physical data transmission. A high-level operational overview of the RTFM could be as follows. After a node subscribes to a publication, a distributed rendezvous system (e.g. a type of DHT or semihierarchical solution as in DONA [17]) must first find a copy of the publication's metadata. Using the distributed rendezvous structure to route to a copy of the wanted data, the topology management systems are expected to gather enough information to identify the delivery trees needed to forward the actual data to the subscriber(s). Note that the RTF functions are not necessary co-located in nodes and are distributed and recursive in nature. In the black box rendezvous based networking approach [21], the key idea is to regard the network as a collection of black boxes based on a set of recursive rendezvous functions. The boxes operate in trusted domains hiding their internal topology and exposing outwards only labels and interest definitions. Recursivity [9] and scoped information layers are pivotal architectural patterns with a major goal: scalability. With the same goal but at a lower layer, efficient data structures enabling the data-centric networking functions (e.g., switching, label processing, caching) are called for to achieve the challenging scalability requirements of information-oriented networks heavily based on virtually 'unlimited' set of flat identifiers.

J. Flat Labels

The overall picture of an information-oriented network architecture is complex and deserves very detailed discussions spanning multiple disciplines (internetworking, network management, semantic layer, etc.). However, there is a common challenge in any data-oriented paradigm: the need to take switching decisions at wire speed (Gbps) based on a large universe of flat (non-topological, non-aggregatable) identifiers (e.g., 256-bit cryptographic hash values). Related work relying on flat labels includes ROFL [7], a proposal for Internet-scale routing on flat host identifiers based on neat DHT constructs. In our work we focus on flat identifiers with fundamentally different architectural principles (see Table 1). DONA [17] employs flat self-certifying labels for data objects operated by find/register primitives over IP networks, whereas our work is more ambitious and could run on top of L2 and L3. For the sake of generality and the objectives of this paper, we use the term flat label for data identifiers or any topology independent packet header forwarding identifiers.

K. Publish – Subscribe Switch

The Publish/Subscribe Switch (SPSwitch) is an abstract switching element that relays messages through strict portforwarding operations. In a more elaborated design, the SPSwitch performs more complex actions like label switching or querying the cache system. For the purposes of this work, it is enough to consider the generic problem of having to take switching decisions based on large flat identifiers (labels). Note that output destinations (ports) are not just limited to physical port-in/out interfaces but should be regarded as generic outputs, including also local processes, virtual ports, recursive operations, and cache systems. In the SPSwitch representation of Figure 1, each possible message output is represented by a Bloom filter [3], forming our first p-bank switching approach (x 4.3) and a reference switching model for an enhanced data structure (x 4.4).

L. Probabilistic Data Structure

Given the huge space and flatness of the information identifiers, our intuition is that Bloom filters and other compact hash-based data structures will play a fundamental role as efficient data aggregators in any information-centric architecture. Basically, a Bloom filter (BF) [3] is a space-efficient hashing-based data structure that answers set membership queries with some probability of being wrong (false positive rate). BFs are useful whenever you have a set of elements and space is an issue. Then, an approximate representation like a BF may be a powerful alternative if the effects of false positives can be managed. The performance of a BF does not depend at all on the size of the items but on the ratio memory=elements. Therefore, hashing-based data structures are an ideal room to handle the large set of flat identifiers. We refer to the large literature on BFs [3, 4, 6, 16] for details and mathematical background. Bloom filters are commonly used in IP forwarding and other widely studied networking applications (e.g., caches, P2P, measurement, packet classification) [6]. We expect increasingly more useful applications of BFs and its derivatives in new data-intense networking proposals (e.g., Internet accountability [2], flow management [4], credential-based network security [22], IP multicast revisited [19]) with strict performance and memory requirements. The authors of [13] briefly sketched the idea of aggregating active IP multicast addresses per output interface to achieve scalability. Due to space limitation we do not compare the SPSwitch design with existing hardware designs for fast networking. We are aware that compact hash tables and hashing functions are a daily aid in IP networking. However, there are notable operational differences and challenges (e.g., longest IP prefix vs. long flat identifier matching).

M. False Positives

It is important to place emphasis on the bounded effect of false positives in data-centric interest-driven architectures. First, the pub/sub paradigm inherently tolerates false positives, since datagrams corresponding to non subscribed items do not progress in the network and do not create forwarding states. Moreover, end-nodes will only process explicitly subscribed pieces of information. Second, with support for opportunistic caching, copies of data can be used to fulfill possible future requests of close by subscribers. Finally, packets forwarded due to false positives are not propagated over many hops due to the large label space and the decreasing probability of consecutive false positives.

VII. NAMING AND SCALABILITY

A. Explicit Aggregation

Hierarchical names help scalability by reducing the size and update-rate of the routing tables. While this is well-known, it is instructive to walk through the semantics of hierarchical routing in more detail. Consider a name of the form com.CNN.headlines. In terms of routing semantics, this name means that if you follow routing entries to com, you are guaranteed to find an entry for com.CNN somewhere along your path, and similarly if you follow routing entries to com.CNN you will eventually find an entry for com.CNN.headlines. These semantics, not any other details about hierarchies, are what enable scaling. In terms of global uniqueness, each entry in the hierarchy is not guaranteed to be unique, but each prefix is (i.e., com, com.CNN, and com.CNN.headlines are all globally unique). Because of this, one cannot route to arbitrary fragments (i.e., headlines, or CNN.headlines) but must look for prefixes to be assured you are routing towards the right entity. It is the lack of global uniqueness of fragments, not the need for aggregation, that drives the use of longest-prefix-match. The common assumption is that aggregation is impossible with flat naming.12 It is true that names do not build in hierarchy, so aggregation does not simply fall out of the naming structure itself, but it turns out that this lack of inherent hierarchy provides greater flexibility in aggregation. Given a set of globally unique names (say, A, B, C, etc., each of the form P:L), one can construct “explicit aggregation” by using concatenations of the form A.B.C.13 The semantics of such concatenations are that when following routing entries for A, you 12Simple scaling estimates in [6] suggest that routing on flat names without aggregation is easily within reach of today’s technologies, so the entire question of aggregation may be moot. However, for the purposes of this paper we will assume aggregation is necessary. 13There is nothing special in self-certifying names that allows this to happen; explicit aggregation can be applied to any naming system. Our only point is that explicit aggregation is more flexible than inherent aggregation. will eventually find one for B; and that when following routing entries for B, you will eventually find an entry for C. We will call this the aggregation invariant. There are two questions to address: How do you route using concatenations? The routing table consists of individual names A, B, etc., and when confronted with concatenated name A.B.C.D, the router searches for the deepest match (as depicted in Figure 2) and forwards the message accordingly. There are two advantages over longest-prefix match: the algorithm has the potential to be simpler to implement, and the aggregation does not affect the structure of the routing table (which is just a set of flat names and their associated outgoing port).14 Instead, all of the aggregation occurs on the

naming side, not on the routing side.15 How do you know when you can use a concatenation? When naming an object, one thinks about two things: the identifier (which is the name itself) and one or more fetch-terms which we can use to retrieve the object (which, in our case, consist of various concatenated names). These fetch-terms can be included in metadata associated with the name (and signed by the principal of the object) and when asking for the object the request can use these fetch-terms (rather than just the name); the fetch-terms enable the routing system to more easily find the object. It is the responsibility of the principal of an object to not sign any concatenation unless the aggregation invariant holds. This invariant might hold because of administrative relationships (as in DNS names), because of economic relationships (contracts with a CDN), or the organization of a particular piece of content (such as chunks of a large file). This form of aggregation is far more flexible than a strict hierarchy, and several forms of aggregation involving the same object can coexist simultaneously. This last point is important. Note that in in hierarchies, the object com.CNN.headlines can only fall under the aggregates com and com.CNN. In contrast, with explicit aggregation an object A can fall under an arbitrarily large number of concatenations, say C.B.A and D.F.A.

B. Semantics , Cache Consistency

Consistency: Consistency Semantics for Cached Objects

Objects cached within a content distribution network need different levels of consistency guarantees depending on their characteristics and user preferences. For instance, users may be willing to receive slightly outdated versions of objects such as news stories but are likely to demand the most up-to-date versions of "critical" objects such as financial information and sports scores. Typically, the stronger the desired consistency guarantee for an object, the higher the overheads of consistency maintenance. For reasons of flexibility and efficiency, rather than providing a single consistency semantics to all cached objects, a CDN should allow the consistency semantics to be tailored to each object or a group of related objects. One possible approach for doing so is to employ _-consistency semantics [21]. _-consistency requires that a cached version of an object is never out-of-date by more than _ time units with its server version. The value of _ determines the nature of the provided guarantee — the larger the value of _, the weaker the consistency guarantee (since the object could be out of date by up to _ time units at any instant). An advantage of _-consistency is that it provides a quantitatively characterizable guarantee by virtue of providing an upper bound on the amount by which a cached object could be stale (unlike certain mechanisms that only provide qualitative guarantees). Another advantage is the flexibility of choosing a different value of _ for each object, allowing the guarantee to be tailored on a per-object basis.1 Finally, strong consistency — a guarantee that a cached object is never out-of-date with the server version —is a special case of _-consistency with_ = 0Due to the above advantages, in this paper, we assume a CDN that provides _-consistency semantics. Next, we present a consistency mechanism to provide _-consistency and then discuss its implementation in a CDN.

C. Cache Consistency for CDNs

A consistency mechanism employed by a CDN should satisfy two key requirements: (i) *scalability:* the approach should scale to a large number of proxies employed by the CDN and should impose low overheads on the origin servers and proxies, and (ii) *flexibility:* the approach should support different levels of consistency guarantees. We now present a cache consistency mechanism that satisfies these requirements. Our approach is based on a generalization of *leases* [11]. In the original leases approach [11], the server grants a lease to each request from a proxy. The lease denotes the interval of time during which the server agrees to notify the proxy if the object is modified. After the expiration of the lease, the proxy must send a message requesting a lease renewal. More formally, a lease is a tuple fO; p; dg maintained by the server, where the server agrees to notify proxy p of all updates to an object O during time interval d. The leases approach has two drawbacks from the perspective of a CDN. First, leases provide strong consistency semantics by virtue of notifying a proxy of *all* updates to an object. As argued earlier, not all objects cached within a CDN need such stringent guarantees. Second, leases require the server to maintain state for each proxy caching an object; the resulting state space overhead can be excessive for large CDNs. Thus, leases do not scale well to busy
servers and large CDNs. To alleviate these drawbacks, we generalize leases along two dimensions: 1. We add a *rate parameter* _ to leases that indicates the rate, 1=_, at which the server agrees to notify a proxy of updates to an object. This enhancement allows a server to relax the consistency semantics provided by leases from strong consistency to _-consistency — a proxy is notified of updates at most once every _ time units (instead of after every update) and no later than _ time units after an update. Using _ = 0 reverts to the original leases approach (i.e., strong to provide weaker consistency guarantees (and correspondingly reduces the number of notifications sent to a proxy). 2. We allow a server to grant a single lease collectively to a group of proxies, instead of issuing a separate lease to each individual proxy.3 For each cached object, the proxy group

designates an invalidation proxy, referred to as the *leader*, that is responsible for all lease-related interactions with the server. The leader of a group manages the lease on behalf
of all the proxies in the group. Since a leader is selected per object no single proxy becomes the bottleneck. Moreover, the server only notifies the leader upon an update to the object; the leader is then responsible for propagating this notification to other proxies in the group that are caching the object. Such an approach has two significant advantages: (i) it reduces the the amount of state maintained at a server (by using a single lease to represent a proxy group instead of an individual proxy); and (ii) it reduces the number of notifications that need to be sent by the server (by offloading some of notification burden to leader proxies).

We refer to the resulting approach as *cooperative leases*. Formally, a cooperative lease is a tuple fO; G; L; d;_g where the server agrees to notify the leader L representing proxy group G of any updates to the object O once every _ time units for an interval d. While leases is a pure server-based approach to cache consistency, cooperative leases require both the server and the proxy (especially the leader) to participate in consistency maintenance. Hence this approach is more scalable when compared to original leases, and thus, more suited to CDN environments.

D. System Model for CSNs

Before discussing the implementation of cooperative leases in CDNs, we present the system model assumed in this paper. A content distribution network is defined to be a collection of proxies that cache content stored on origin servers. For the purposes of maintaining consistency, proxies within the CDN are assumed to be partitioned into non-overlapping groups referred to as *regions* (issues in doing so are beyond the scope of this paper). Proxies within a region are assumed to cooperate with one another for maintaining consistency of cached objects. Cooperative consistency *is orthogonal to cooperative caching* — whereas the latter involves sharing of cached data to service user requests, the former involves cooperation solely for maintaining consistency of data cached by proxies 3In addition, it is also possible for a lease to collectively represent multiple objects. Techniques for doing so are studied in [24]. within a region. Further, the organization of proxies into regions is limited to consistency maintenance; a different overlay topology can be used for exchanging data and meta-data within the CDN. Each proxy in a region is assumed to maintain a directory of mappings between the cached object and its corresponding leader (and
possibly other information required by the CDN). Several directory schemes such as hint caches [20] and bloom filters [7] have been proposed to efficiently maintain such information. Another approach is to use a simple consistent hashing [14] based scheme to determine the mapping between an object and the proxy that acts as the leader. Here a hashing function is used to hash on both the unique object identifier and the list of proxy identifiers to determine the best match. Although this approach reduces the flexibility in assigning the leader for an object, it reduces the space and the message exchange overhead. Any of the above schemes suffices for our purpose.

E. Simulation

We have designed an event-based simulator to evaluate the efficacy of cooperative leases. The simulator simulates one or more proxy regions within a CDN. Each proxy is assumed to receive requests from a large number of clients. Cache hits are serviced using locally cached data. Cache misses involve a remote fetch and are serviced by fetching the object from the leader (if one exists) or Trace Num Duration Unique Num from the server. The directory maintained by the proxy is used to make this decision. Our simulator supports all policies discussed in Section 3 for leader selection, server notifications, lease renewals and rate computations. Our experiments assume that each proxy maintains a disk-based cache to store objects. We assume each proxy cache is infinitely large — a practical assumption, since disk capacities today are in tens of gigabytes and a typical proxy can employ multiple disks. Data retrievals from disk (i.e., cache hits) are modeled using an empirically derived disk model with a fixed OS overhead added to each request. For cache misses, data retrieval over the network are modeled using the round trip time, available network bandwidth and the object size. The network latency and bandwidth between proxies and leaders is assumed to be 75ms and 500KB/s, while that between proxies and origin servers is 250ms and 250 KB/s. Although actual network latencies and bandwidths vary with network conditions, the use of this simple network model suffices for our purpose (due to our focus on consistency maintenance rather than end-user performance). Due to space constraints, we present only results for a single region; we performed experiments with multiple regions to verify that each region behaves similarly to other regions from the perspective of consistency maintenance (see [18]). Unless noted otherwise, our experiments assume a default region size of 10 proxies and a lease duration of 30 minutes. We also assume that a leader always caches a copy of the object and this copy is updated upon a modification.

F. Workload Characteristics

The workload for our experiments is generated using traces from actual proxies, each containing several hundred thousand requests. We use two different traces for our study; the characteristics of these traces

are shown in Table 2. The same set of traces are used for our simulations as well as our prototype evaluation (which employs trace replay). Each request in the trace provides information such as the time of the request, the requested URL, the size of the object, the client ID, etc. We use the client ID to map each request in the trace to a proxy in the region—all requests from a client are mapped to the same proxy. To determine when objects were modified, we considered using the last modified times as reported in the trace. However, these values were not always available. Since the modification times are crucial for evaluating cache consistency mechanisms, we employ an empirically derived model to generate modification times. Based on observations in [1, 13], we assume that 90% of all web objects change very infrequently (i.e., have an average lifetime of 60 days). We assume that 7% of all objects are mutable (i.e., have an average lifetime of 20 days) and the remaining 3% objects are very mutable (i.e., have a lifetime of 5 days). We partition all objects in the trace into these three categories and generate write requests and last modified times using exponentially distributed lifetimes. Although the average lifetimes are in days, given the high variance in the modification times there were numer- 6ous writes within the sampling duration of the trace. The number of synthetic writes generated for each trace is shown in Table 2. In practice the server will rely on a publishing system or a database trigger to detect a modification, the details of which are beyond the scope of the paper. Next, we describe our experimental results.

G. Network Model

The network model we are presenting in this section is largely based on the CCN as described by Van Jacobson in [7]. We have however added several aspects to it so that it better corresponds to the requirements highlighted in 2.3.

The main addition is the *Event Packet* object, which consists in a slightly modified and unsolicited Data Packet presented in the following paragraph. We are also changing the functioning of the FIB from a simple forward plane to a more intelligent forwarding engine able to favor some faces depending on their characteristics (bandwidth, cost, etc.).

H. Event Packets

A new type of packet is introduced in our architecture to cope with specific needs of vehicular networks: the Event Packet. In the scenario we are considering (presented in section 2.3) the mobile nodes are vehicles equipped with several sensors that are able to send measurement to nearby vehicles and to the infrastructure. In case of an emergency, such as when the vehicle detects an accident or some hazard on the road, it is expected to warn as soon as possible the other vehicles as well as the relevant safety services, located behind the infrastructure. In such a case, the vehicle should send an unsolicited message to the VANET, to be routed to other potentially affected vehicles. This is somehow contradictory with the CCN paradigm which requires that any content should only be produced or forwarded as an answer to a previously generated interest. It may also lead to malicious exploitation of the network resources in case a rogue node purposely sends large amounts of Event Packets to generate a Denial of Service (DoS). This risk may be easily mitigated by the neighboring nodes for instance through the limitation of the rate at which they accept the packets from the rogue node, ultimately dropping all the received packets. The Event Packet has exactly the same structure as the Data Packet presented on Fig. 3, but features an additional field which we call Expiry Time (ExpT). It is presented on Fig. 4. This field plays a similar role as the IP–TTL value representing the number of hops before a packet is dropped, but contrary to it the Event Packet ExpT indicates the time after which the packet should not be forwarded anymore *and* should be deleted from the Event Tables where it is still stored. The value of ExpT is not a system constant: it can – and should – be adapted in function of the type of event and the respective application. For instance, a“traffic jam” event lasting several days is probably overkill. Also, each node is responsible for honoring or not the overall housekeeping policy: for instance it may also delete an Event Packet before its expiry time if it does not have enough resources to keep it in its Events Table. The propagation of an Event Packet is explained on Fig. 5. This diagram represents mobile nodes that are in the vicinity of a specific event that is of interest for other mobile nodes that are too far away to detect that event. We are especially focusing on node C. The plain arrows represent existing adhoc links between neighbors. This links are weighted with their respective cost. We assume here that this cost represents the travel time between two nodes (including processing time on both sides), that all the nodes stay within range of each other for the whole duration of this short scenario and that the Expiry Time of the event is greater than this duration. It is also possible for a node to have several links with one of its neighbor if both share enough compatible network interfaces. For instance, it is the case of node B that shares two different ad-hoc channels with node R1. Nodes A and B are both sufficiently close to the event that occurred at $t = 0$ to generate an Event Message accordingly. For the purpose of this demonstration, let’s assume that these two *identical* event packets are generated simultaneously at $t = t0$. The sequential unfolding of the events is as follow: at $t = 1$, both A, R1 and R2 receive the same event packet, from different sources. B doesn’t forward the message over its more costly link with R2: it knows from its FIB that it is not necessary since a more efficient channel exists. A receives it from B but drops it immediately since he already has the same message in his Events Table. R1

got the event from A and R2 from B. At t = 2, R3 receives the message from R1 and C from R1. Finally, at t = 3, R3 drops the duplicate he got from R2 and C the one he got from R3.

I. Packet Forwarding

Since the route followed by packets is not known in advance, there is neither a simple deterministic way to predict the amount of time required for any packet to reach one interested node, nor any method to optimize the network with respect to specific application requirements such as high bandwidth or low latency. Since different physical or logical interfaces serve as support for CCN's faces, these faces feature different characteristics. In the specific case of VANETs applications, especially the one related to safety, it is advisable to either dedicate some of the resources to routing safety oriented messages or to implement prioritization algorithms. Our assumption consists in allowing a node to favor some faces over others, depending on the requirements of the application and more generally the overall context. To that extend, the original CCN's FIB is modified to take into account the characteristics of the exit faces. For an urgent message, the face using the protocol with the lowest latency will be used instead of the other ones. This is however only a local optimization and as such may sometimes be counterproductive in a self-organizing environment, but this limitation is mitigated when most of the nodes are able to use the same communication interfaces including the one used for routing priority messages.

J. Anycast Routing

Intra-Domain. Standard intra-domain unicast routing algorithms, whether distance-vector or link-state, are naturally amenable to routing anycast. As described in [31], for link-state protocols such as OSPF, the only modification required is that IPvN routers also advertise a high-cost "link" to the corresponding anycast address. This high cost is necessary to prevent routers from attempting to route *through* an anycast address. Note that from these link state advertisements, an IPvN router can easily identify every other IPvN router within its domain. With distance-vector protocols such as RIP [32], anycast routing merely requires that an IPvN router advertise a distance of zero to its anycast address; standard distance-vector then ensures that every router will discover the next hop to its closest IPvN router. Note that here, unlike link-state routing, an IPvN router cannot easily identify other IPvN routers. An alternate approach to both the above is simply to have an IPvN router indicate this in its standard unicast route advertisement by, for example, explicitly listing its anycast address. Because intra-domain routing algorithms build a complete (*i.e.*, nonaggregated) routing table, this makes anycast routing trivial – a router merely checks its unicast routing table for the closest anycast-addressed router. This involves a small modification to existing intra-domain routing algorithms but makes it trivial for IPvN routers to discover one another. As described in Section 3.3, this knowledge enables very simple intra-domain virtual topology construction. 22While the remainder of this paper will assume that the intradomainprotocol does allow IPvN routers to discover one anotherwe stress that this is merely a simplification and in no way a necessary requirement. In its absence (*i.e.*, for domains that use unmodified RIP [32]), the intra-domain virtual topology construction will merely have to implement some additional discovery mechanisms

Inter-Domain, option 1: non-aggregatable addresses, global routes. One approach to supporting inter-domain anycast is to designate a portion of the regular unicast address space to serve as anycast addresses and require that ISPs propagate route advertisements for anycast addresses in their inter-domain routing protocols. This approach is certainly implementable even today – as suggested by [28,31], a designated portion of the unicast address space could be assigned to anycast and propagating these routes in BGP would require a change in policy but not mechanism on the part of ISPs – and yet there is, with one exception [30], little deployment of global IP anycast. One reason for this narrow adoption is concern over the scalability of such an approach, particularly under RFC1546's fully general and dynamic IP anycast service model. Anycast addresses, as described above, are not aggregatable and must hence be advertised individually by routing protocols and lead to routing state that grows in direct proportion to the number of anycast groups. However, for our proposed use of anycast, scalability is unlikely to be a concern. Recall that a single anycast address is needed to serve each new generation of IP. Given the cost and effort for an ISP to roll out a new generation of routers, we imagine that the number of simultaneous attempts to deploy different IP versions is likely to be very small (ideally one) and will not lead to a problematic growth in routing state. Moreover, unlike the more commonly advocated uses of anycast (server selection, *etc.*), here the consumers of anycast addresses are not arbitrary endusers but rather the ISPs themselves who have an incentive to use these addresses sparingly. To further ensure this, ISPs might even charge to route anycast. Instead, our concern with this approach is that it requires that all ISPs eventually support the propagation of anycast routes. While this seems like a not unreasonable hope given that the only change required is a simple modification to policy, we would rather not rely on this assumption and hence explore alternate approaches.

Inter-Domain, option 2: aggregatable addresses, default routes. To address the poor scaling of traditional anycast
architectures, Katabi *et al.* propose GIA [31]. In GIA, scalability is achieved by introducing the notion of a "home" ISP domain
associated with an anycast group. GIA still allocates anycast its own portion of the IP address space – all addresses prefixed by a well-known "Anycast Indicator" sequence of bits. However the remaining address bits are drawn from the unicast address space of the home domain. This allows for simple "default" routes; a router with no anycast routing entry for a given address can look up the home domain's prefix in its unicast routing table and forward the packet towards the home domain. GIA requires that the home domain include at least one member of the anycast group and hence this ensures the packet will reach a group member although not necessarily the closest. For more optimized anycast routes, Katabi *et al.* propose an extension to BGP whereby border routers can initiate searches for nearby members of an anycast group. While GIA offers an elegant solution to scalable anycast, its deployment requires modifying the border routers at client domains. Given the current lack of deployment of GIA by ISPs, and to satisfy our stated required assumption of no global participation (Section 2), we present here an approach to anycast that requires no change by non-participant ISPs. We stress however that our proposal is somewhat motivated by expediency and open to eventual replacement by GIA (or a similar design) and/or the use of a limited number of non-aggregatable addresses as described above. Our proposal, along the lines described in [33], is to avoid (at least for now) introducing a special type of anycast address and instead just reuse a piece of the existing unicast address space. We borrow the basic insight behind GIA and advocate that anycast addresses be allocated from the unicast address space of a "default" ISP (*e.g.*, the first ISP to initiate deployment of IPvN) and IPvN routers are configured to advertise the anycast address in their IGP as described earlier. Additional ISPs that adopt IPvN also configure their IPvN routers to advertise the same anycast address internally. Standard unicast routing will deliver anycast packets to the closest IPvN router along the path from the source to the default ISP. For example, in Figure 2, ISPs Q and D deploy IPvN and D is the default domain; anycast packets from domains X and Y terminate in domain D while those from Z reach Q. To widen their reach, nondefault domains can peer with neighboring domains to advertise their anycast route. For example, in Figure 2, Q can peer with Y to advertise its path for the anycast address in question; Y's packets will then be delivered to Q rather than D. Thus the final picture in our proposal is not unlike that using non-aggregatable addresses. The key difference is that our use of a default ISP allows us to transition to that final picture through the *optional and independent* participation of ISPs. Even with no cooperation from non-IPvN domains, the above scheme will route anycast correctly, although imperfectly in terms of proximity to, IPvN routers. Here, the use of inter-domain advertising is an optimization that leads to more improved anycasting. A potential failing of our approach is that the default provider owns the anycast address and receives a larger than normal share of IPvN traffic. Ideally though, this could incite other ISPs to pursue inter-domain advertising of anycast addresses. Given its practicality, our discussion from here on will assume the the use of anycast addresses rooted in default ISPs.

K. IPvn packet Formation

Sections 3.1 and 3.2 described how IPvN packets are steered to IPvN routers. We now describe how these IPvN routers cooperate to form a "virtual" IPvN network, or vN-Bone, overlaid on an Internet where IPv(N-1) is ubiquitously deployed. Before delving into the details of our mechanisms, we make two observations: the first is that, unlike the case of network redirection which must be ubiquitously supported (whether explicitly as in option 1 for inter-domain anycast, or implicitly as in option 2 for inter-domain anycast, as described in Section 3.2) by both participant and non-participant providers, vN-Bones are implemented entirely by participant ISPs and hence this design space is much less constrained. Indeed, many of the techniques from the literature on overlays and testbeds [18, 20, 21, 21, 23] could likely find use here and, as such, our proposals are best viewed as one set of candidate solutions. The second observation is that virtual networks that span multiple ISPs are not new. Networks like the MBone [25] and 6Bone were all pioneering efforts in this respect. These networks relied greatly on manual configuration and, while the solutions we present do automate much of the topology construction and maintenance process, we readily accept that many ISPs might, as in the past, simply choose to configure their networks by hand. There are two main components to a virtual network:

1. virtual topology construction, and
2. routing over this virtual network

Note that because we do not assume ubiquitous deployment even within a participant ISP, each of the above must be addressed at both the intra and inter-domain level.

L. Topology Construction

The first component – vN-Bone construction – is fairly straightforward as it largely builds off the connectivity information revealed by the underlying IPv(N-1) routing protocols. For example, the IPv(N-1) intra-domain routing, whether link-state or distancevector (and assuming the anycast extensions

described in the previous section), ensures that every IPvN router has complete knowledge of the set of IPvN routers within its domain.3 The intradomain vN-Bone topology can then be constructed through simple rules such as: every IPvN router picks its *k* closest IPvN routers as neighbors on the vN-Bone. In the event that such rules leads to partitions, these can be easily detected and repaired because every router has complete knowledge of all other IPvN routers. At the inter-domain level, the most likely scenario is that ISPs set up inter-domain tunnels based on their peering policies. In the absence of such configuration, a newly joined ISP could reuse the anycast mechanism as the initial bootstrap by which to discover at least one other ISP that currently supports IPvN; having done this, the new ISP can discover additional neighbors through the interdomain vN-Bone routing (described below).4 For preventing partitions of the inter-domain vN-Bone topology, one simple approach is for every domain to ensure that it is connected (either directly or indirectly) to the "default" provider of the anycast address. Finally, as deployment spreads, the vN-Bone topology should evolve to be congruent with the underlying physical topology. This is easily achieved using the connectivity information revealed by the v(N-1) routing protocols at the intra and inter-domain levels.

M. VNs Addressing

The issue of routing is closely tied to that of host addressing. There are at least three aspects to addressing that to 3Recall our discussion in Section 3.2 about how such global knowledge can be achieved even in distance-vector protocols like RIP with one minor modification. In the absence of this modification, intra-domain vN-Bone construction over RIP would have to be implemented along the lines of the inter-domain vN-Bone construction; *i.e.*, through explicit neighbor discovery leveraging anycast for the initial bootstrap. 4Note that this use of anycast is only possible for a new ISP that isn't yet actively advertising the anycast. Otherwise, the anycast route would simply loop back to the initiator. be considered: (1) the format or structure of addresses, (2) address allocation and, (3) advertising addresses into the routing fabric.

In today's Internet, the allocation and advertisement of an endhost's IPv4 address is handled by its local access provider. If future IPvN architectures adopt a similar model then supporting universal access raises the question of how an endhost might obtain an IPvN address if its access provider does not yet support IPvN. A possible solution, along the lines proposed in RFC 3056 [34], is to have the endhost assign itself a unique IPvN address. This can be done, for example, by using one address bit to indicate such "self addressing" and deriving the remaining IPvN address bits from the endhost's unique IPv(N-1) address. Note that these self-addresses are very likely temporary and such endhosts will have to relabel if and when their access providers do adopt IPvN. This leaves us then with the question of how such temporary IPvN addresses are advertised and routed on. We explore this question in detail in the discussion on routing that follows. Finally, we note that this need or self-addressing arises in the case where address assignment is handled by an endhost's local provider. More generally however, we place no particular constraints on the addressing structure or allocation policy a next-generation IPvN may adopt.

Routing. In considering routing on this virtual network, we have to do so at two levels:

• routing between IPvN routers on the vN-Bone, and
• routing between any two IPvN endhosts

The two issues are closely related – given the ability to route between IPvN routers, routing between two IPvN endhosts is primarily a question of how we find the appropriate ingress and egress IPvN routers for a given source and destination IPvN endhosts. Note that most discussions of routing, whether application layer as in overlays, or at the IPv4 network layer, need not distinguish between the two cases above. The current network layer assumes that an endhost's ingress or egress router is simply its access router and hence this distinction in unnecessary. Unfortunately, our need for universal client access under partial IPvN deployment makes this assumption invalid (at the IPvN layer). Proposed overlay-based routing systems [9, 20] on the other hand, assume some form of higher-layer (*e.g.*, DNS) or out-of-band translation between an endhost identifier and its "attachment" point in the overlay. This translation can be invoked prior to communication between two endhosts and hence the issues of routing between endhosts is easily mapped to that of routing between two overlay routers. In our case however, there are a number of reasons why we might not want to make a similar assumption. First, endhosts are not assigned explicit attachment points in the vN-Bone. Moreover, an endhost might have different attachment points depending on the network location of the endhost it is communicating with and these attachment points will change as deployment spreads. Most importantly, this option raises issues similar to those with application-level redirection (Section 2.2) – given our self-imposed reluctance to assume the introduction of new services, it isn't clear who can effect this translation or how, because doing so would require intimate knowledge of the state of IPvN deployment.

Between routers:. The topology construction in Section 3.3.1 described the global vN-Bone as composed of intra-domain vNBone topologies interconnected by inter-domain tunnels. Given this topology, establishing routes between IPvN routers is achieved by IPvN routing protocols and will thus depend on the specifics of a particular IPvN. The space of possible routing solutions here is fairly

unconstrained as the participant nodes are all IPvN routers. In the discussion that follows, we assume the existence of separate intra and inter-domain IPvN routing protocols but assume no specific routing algorithm. For simplicity, we use the notation BG-319

N. Forwarding

We now briefly review the end-to-end data path taken by a packet. Assume IPv(N-1) is the current ubiquitously deployed version of IP, IPvN is the next generation IP and all IPvN routers form a virtual vN-Bone. We use $An-1$ to denote the IPv(N-1) anycast address assigned to the deployment of IPvN. Then, end-to-end forwarding of an IPvN packet works as follows:

• the source S encapsulates the IPvN packet in an IPv(N-1) header with destination $An-1$.

• using anycast, the packet is forwarded over legacy IPv(N- 1) routers to the closest IPvN router, R1.

• R1 strips off the IPv(N-1) header, processes the packet as needed, looks up the next hop (R2) to the destination using the vN-Bone forwarding tables, and forwards the packet to R2, once again encapsulating the packet in an IPv(N-1) header if required.

• this is repeated until the packet reaches the egress IPvN router which tunnels the packet through to the destination.

In addition, the source, either through configuration or an ARPlike protocol, discovers whether its first hop router supports IPvN
and, if so, does not encapsulate the packet. Similarly, every intermediate router will only invoke encapsulation if its next hop IPvN router is not an immediate (*i.e.*, physical layer) neighbor. Thus, as deployment spreads, the use of IPv(N-1) is gradually phased out.

O. Source Specific Multicast

The previous section presented an overall framework for evolvability based on the use of IP Anycast. In this section, we take IP Multicast as an example of a new IP service and work through its deployment under this framework. In so doing, we quite deliberately do not attempt to innovate on the details of the multicast protocols themselves; instead we take existing standards and describe how our framework might support their deployment. We focus our discussion on the deployment of source-specific multicast. A detailed description of deploying any-source multicast which uses a somewhat larger suite of protocols (IGMP, MSDP, MBGP, PIM-SM and PIM-DM), while we believe would follow along similar lines, is beyond the most masochistic tendencies of the authors.

Source Specific Multicast. Source Specific Multicast (SSM), a restricted form of the more general IP Multicast service [38],
provides one-to-many packet delivery between a designated source node and zero or more receivers [39]. As defined by RFC 3569 [40] and Holbrook [29, 41], source-specific multicast is implemented through the combined use of the Internet Gateway Multicast Protocol (IGMP) [42] and a reduced form of Sparse Mode PIM (Protocol Independent Multicast), denoted PIM-SSM. Through IGMP, a Designated Router (DR) on a local network tracks group membership on each of its network interfaces and participates in the wide-area multicast routing on behalf of the endhosts on its network. PIMSSM is then used to construct a tree rooted at the source DR to all receivers' DRs. For simplicity, we use endhosts to mean their DRs and focus only on the mechanics of the wide-area routing. In SSM, a multicast group, called a *channel*, is defined by the combination (S,G) of a multicast group address (G) and the unicast address (S) of the source. The SSM receiver interface supports joining and leaving a channel (subscribe(S,G),unsubscribe(S,G)). A subscribe results in a JOIN message being routed toward S, setting up routing state for the new receiver at every point along the path until the JOIN messages hits a router on the distribution tree. unsubscribe operations trigger PRUNE messages that tear down routing state in a similar manner. To multicast to the group, packets from the source are forwarded down the distribution tree using reverse-path forwarding. We now detail how SSM would operate using our framework from the previous section. We assume IPv4 is the ubiquitously deployed IP and use IPvM to denote a next generation IP that supports PIM-SSM. *Am* denotes the anycast address allocated for the deployment of IPvM and vM-Bone the IPvM virtual topology. Note that a single vM-Bone is reused by all multicast groups.

• To use IPvM, a client C first checks whether its local ISP supports IPvM. If not, then all IPvM packets will be encapsulated
in IPv4 packets as described below. Otherwise, IPvM packets are transmitted natively. In this example, we assume C's access provider does not support IPvM.

• To join channel (S,G) client C transmits a JOIN message shown as packet (P1) in Figure 5. Note that in practice, the source address denoted as *C* would be the IP address of C's Designated Router rather than C itself. Anycast routing delivers this PIM JOIN message to R1, the IPvM closest to C.

• R1 strips off the outer IPv4 header, and adds (G,S,*C*) to its multicast forwarding table. If this is R1's first multicast routing
entry for (G,S) then R1 looks up the next hop, say R2, to destination *S* on the vM-Bone, encapsulates the packet as (P2) in Figure 5 and unicasts P2 to R2. Otherwise, the JOIN operation is terminated since R1 is already on the delivery tree for (G,S). Once again,

if R1 and R2 are immediate neighbors then encapsulation is bypassed.

• The above is repeated until the packet hits an IPvM router already on the distribution tree for (G,S) or the egress IPvM router Rn in which case Rn unicasts the packet P3 from Figure 5 to S which sets up state (G,S,Rn). Again, in practice this state is stored at S's DR and not S itself. In fact, S would have no knowledge of the membership of G. Note that S's DR must know to decapsulate the packet. However as described in [RFC 2326], support for encapsulation is already required of DRs to handle packets tunneled to and from Rendezvous Points.

• Finally, to multicast to group G, S transmits a data packet to group G. The packet is picked up by S's Designated Router and forwarded through the (S,G) tree constructed as above. The point to note in the above exercise is the manner in which our general framework enables SSM to be universally accessible (*i.e.*, to all endhosts) despite partial deployment of IPvM and support for PIM-SSM. At every step in the tree construction and multicast forwarding, IPvM and PIM-SSM can run "natively" if possible and, if not, the general techniques of tunneling and anycast forwarding bridge the gap to the next island of IPvM support.

P. Authors and Affiliations

Dr Akash Singh is working with IBM Corporation as an IT Architect and has been designing Mission Critical System and Service Solutions; He has published papers in IEEE and other International Conferences and Journals.

He joined IBM in Jul 2003 as a IT Architect which conducts research and design of High Performance Smart Grid Services and Systems and design mission critical architecture for High Performance Computing Platform and Computational Intelligence and High Speed Communication systems. He is a member of IEEE (Institute for Electrical and Electronics Engineers), the AAAI (Association for the Advancement of Artificial Intelligence) and the AACR (American Association for Cancer Research). He is the recipient of numerous awards from World Congress in Computer Science, Computer Engineering and Applied Computing 2010, 2011, and IP Multimedia System 2008 and Billing and Roaming 2008. He is active research in the field of Artificial Intelligence and advancement in Medical Systems. He is in Industry for 18 Years where he performed various role to provide the Leadership in Information Technology and Cutting edge Technology.

VIII. REFERENCES

[1] Dynamics and Control of Large Electric Power Systems. Ilic, M. and Zaborszky, J. John Wiley & Sons, Inc. © 2000, p. 756.

[2] Modeling and Evaluation of Intrusion Tolerant Systems Based on Dynamic Diversity Backups. Meng, K. et al. Proceedings of the 2009 International Symposium on Information Processing (ISIP'09). Huangshan, P. R. China, August 21-23, 2009, pp. 101–104

[3] Characterizing Intrusion Tolerant Systems Using A State Transition Model. Gong, F. et al., April 24, 2010.

[4] Energy Assurance Daily, September 27, 2007. U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability, Infrastructure Security and Energy Restoration Division. April 25, 2010.

[5] CENTIBOTS Large Scale Robot Teams. Konoledge, Kurt et al. Artificial Intelligence Center, SRI International, Menlo Park, CA 2003.

[6] Handling Communication Restrictions and Team Formation in Congestion Games, Agogino, A. and Tumer, K. Journal of Autonomous Agents and Multi Agent Systems, 13(1):97–115, 2006.

[7] Robotics and Autonomous Systems Research, School of Mechanical, Industrial and Manufacturing Engineering, College of Engineering, Oregon State University

[8] D. Dietrich, D. Bruckner, G. Zucker, and P. Palensky, "Communication and computation in buildings: A short introduction and overview," *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3577–3584, Nov. 2010.

[9] V. C. Gungor and F. C. Lambert, "A survey on communication networks for electric system automation," *Comput. Networks*, vol. 50, pp. 877–897, May 2006.

[10] S. Paudyal, C. Canizares, and K. Bhattacharya, "Optimal operation of distribution feeders in smart grids," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4495–4503, Oct. 2011.

[11] D. M. Laverty, D. J. Morrow, R. Best, and P. A. Crossley, "Telecommunications for smart grid: Backhaul solutions for the distribution network," in *Proc. IEEE Power and Energy Society General Meeting*, Jul. 25–29, 2010, pp. 1–6.

[12] L. Wenpeng, D. Sharp, and S. Lancashire, "Smart grid communication network capacity planning for power utilities," in *Proc. IEEE PES, Transmission Distrib. Conf. Expo.*, Apr. 19–22, 2010, pp. 1–4.

[13] Y. Peizhong, A. Iwayemi, and C. Zhou, "Developing ZigBee deployment
guideline under WiFi interference for smart grid applications," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 110–120, Mar. 2011.

[14] C. Gezer and C. Buratti, "A ZigBee smart energy implementation for energy efficient buildings," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, May 15–18, 2011, pp. 1–5.

[15] R. P. Lewis, P. Igic, and Z. Zhongfu, "Assessment of communication methods for smart electricity metering in the U.K.," in *Proc. IEEE PES/IAS Conf. Sustainable Alternative Energy (SAE)*, Sep. 2009, pp. 1–4.

[16] A. Yarali, "Wireless mesh networking technology for commercial and industrial customers," in *Proc. Elect. Comput. Eng., CCECE*,May 1–4, 2008, pp. 000047–000052.

[17] M. Y. Zhai, "Transmission characteristics of low-voltage distribution networks in China under the smart grids environment," *IEEE Trans. Power Delivery*, vol. 26, no. 1, pp. 173–180, Jan. 2011.

[18] V. Paruchuri, A. Durresi, and M. Ramesh, "Securing powerline communications," in *Proc. IEEE Int. Symp. Power Line Commun. Appl., (ISPLC)*, Apr. 2–4, 2008, pp. 64–69.

[19] Q.Yang, J. A. Barria, and T. C. Green, "Communication infrastructures for distributed control of power distribution networks," *IEEE Trans. Ind. Inform.*, vol. 7, no. 2, pp. 316–327, May 2011.

[20] T. Sauter and M. Lobashov, "End-to-end communication architecture for smart grids," *IEEE Trans. Ind. Electron.*, vol. 58, no. 4, pp. 1218–1228, Apr. 2011.

[21] K. Moslehi and R. Kumar, "Smart grid—A reliability perspective," *Innovative Smart Grid Technologies (ISGT)*, pp. 1–8, Jan. 19–21, 2010.

[22] Southern Company Services, Inc., "Comments request for information on smart grid communications requirements," Jul. 2010

[23] R. Bo and F. Li, "Probabilistic LMP forecasting considering load uncertainty," *IEEE Trans. Power Syst.*, vol. 24, pp. 1279–1289, Aug. 2009.

[24] *Power Line Communications*, H. Ferreira, L. Lampe, J. Newbury, and T. Swart (Editors), Eds. New York: Wiley, 2010.

[25] G. Bumiller, "Single frequency network technology for fast ad hoc communication networks over power lines," WiKu-Wissenschaftsverlag Dr. Stein 2010.
[31] G. Bumiller, L. Lampe, and H. Hrasnica, "Power line communications for large-scale control and automation systems," *IEEE Commun. Mag.*, vol. 48, no. 4, pp. 106–113, Apr. 2010.
[32] M. Biagi and L. Lampe, "Location assisted routing techniques for power line communication in smart grids," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, 2010, pp. 274–278.
[33] J. Sanchez, P. Ruiz, and R. Marin-Perez, "Beacon-less geographic routing made partical: Challenges, design guidelines and protocols," *IEEE Commun. Mag.*, vol. 47, no. 8, pp. 85–91, Aug. 2009.
[34] N. Bressan, L. Bazzaco, N. Bui, P. Casari, L. Vangelista, and M. Zorzi, "The deployment of a smart monitoring system using wireless sensors and actuators networks," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, 2010, pp. 49–54.
[35] S. Dawson-Haggerty, A. Tavakoli, and D. Culler, "Hydro: A hybrid routing protocol for low-power and lossy networks," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, 2010, pp. 268–273.
[36] S. Goldfisher and S. J. Tanabe, "IEEE 1901 access system: An overview of its uniqueness and motivation," *IEEE Commun. Mag.*, vol. 48, no. 10, pp. 150–157, Oct. 2010.
[37] V. C. Gungor, D. Sahin, T. Kocak, and S. Ergüt, "Smart grid communications and networking," Türk Telekom, Tech. Rep. 11316-01, Apr 2011.