Motivation and Design of a Content Distribution Architecture for Rural Areas

Arvind Mahla, Deepak Martin, Ishani Ahuja, Quamar Niyaz, Aaditeshwar Seth

Dept. of Computer Science
Indian Institute of Technology Delhi, India

ABSTRACT

Decreasing prices of digital cameras, phones and video cameras have made these devices accessible to low-income communities and to development organizations working in these communities. As a result, the social media revolution around user generated content on YouTube and Facebook has found equivalents in rural areas; here, community radio, community video, and interactive voice based systems are used for social media instead of Internet websites to host user generated content. We observe that there is much benefit to be gained from sharing this content across different rural locations, but it is hard due to unavailable or flaky Internet access in these areas. In this paper, we analyze a 1000+ video content production and consumption dataset from a non-profit organization that specializes in participatory video production about agricultural best practices, and observe that solutions to connectivity in rural areas could greatly benefit from caching of content since much production and consumption tends to be local. Based on this insight, we propose a delay tolerant network architecture for content distribution, that recognizes content objects as first class entities cachable at different nodes in the network, and uses an always-on control channel on GPRS/EDGE connections to assist in the routing of data. Finally, we simulate the dataset in accordance with our architectural design to study the performance of different routing and caching algorithms in terms of delivery latency and other metrics for 6 different districts. We find that our proposed architecture is suitable to provide content distribution services with minimal investments in IT infrastructure, and we plan to do a small-scale field deployment shortly.

1. INTRODUCTION

People like to express themselves. We have seen how social media tools like YouTube, Flickr, and blogging have revolutionized creativity, knowledge sharing, and even political discourse in the world. Not known to many though, social media is also actively practiced in rural and low-income

Figure 1: Neeru, a community video reporter, gearing for a shoot. And a screening in progress.

areas of developing countries such as India. Non-profit organizations are known to train staff and volunteers in *video* recording, editing, and reporting and interviewing skills [1]. They use simple video cameras to put together films on local themes highlighting issues like corruption and domestic violence, document the culture by recording dramas and folk songs, and create educational programs on agricultural and health best practices [2]. The films are then screened in public gatherings in nearby villages, and also sold on CDs and DVDs, or relayed over local cable TV networks. The proliferation of mobile phones with reasonably good cameras have also helped many villagers turn into citizen journalists and report happenings from extremely remote regions [3]. Community radio is another popular medium in which radio station staff create relevant programs by engaging with communities [4]. Local production of content automatically makes it highly contextual, and restricting the range of broadcast to 10-15km makes it possible to supplement the broadcast with in-person interactions between communities and the people running the radio stations. In places where FM broadcast licenses are not available, a process termed as narrow-casting is used – public gatherings are organized in which programs are played over loud speakers and TV screens, and supplemented with interactive question-answer sessions [5].

As a result of this social media revolution in low-income communities made possible by ever decreasing prices of technology, large quantities of digital content is being produced at the grassroots. In India, the NGO Digital Green reports about communities trained by it across 5 states of having created 1,000+ agricultural videos in the last three years. The NGO Video Volunteers runs around 20 community video units and supports 50+ citizen reporters across the country. And India currently has almost 120 community radio stations with most stations producing over 5GB

of new audio content per month. There are many more organizations doing similar work and helping communities to document their lives through photographs, videos, and audio.

Informal conversations with staff working at these community video and radio stations reveal that the staff are eager to view content from other places, learn from it, and use it to improve their own programming. But in practice content is rarely shared across different locations. The lack of high bandwidth Internet access in most rural areas is cited as a significant impediment to content sharing. Although 2G GPRS/EDGE connectivity is available in large parts of rural India, 10-15KBps connectivity speeds are insufficient for multimedia content sharing [23]. Related work on faster long distance wireless links is too expensive to connect individual stations to the Internet [6]. Delay tolerant networking has been proposed as a viable low-cost alternative to synchronous connectivity, but most architecture designs are optimized for unicast transactional data transfer rather than for content sharing [7]. Novel network architectures are therefore required to facilitate content distribution in rural areas.

In this paper, we have analyzed a participatory video dataset from Digital Green, a non-profit organization that trains community members in agricultural video production. Our analysis shows that most content production and consumption is local, indicating that content caching techniques could be used to reduce Internet demand. We combine this idea with related work on delay tolerant networks to propose a new architecture specifically suited for content distribution application scenarios in rural areas. Following are the main characteristics of our proposed network architecture:

- *Content based network*: We recognize content objects as first-class entities. Primitives are provided to the application developer to publish and request for content objects, while the network takes responsibility to cache content, replicate content, locate the closest store currently holding the content, etc, completely transparent to the application. Similar to recent work on information centric networks for a clean slate design of the future Internet [8, 9, 10], we feel that working at the level of content objects will provide opportunities to make use of semantic information about the content to improve network performance.

- *Delay tolerance for bulk data transfers*: Other than critical news related content, most social media content typically allows for flexibility in terms of content delivery latencies. This motivates the use of mechanical-backhaul like techniques to provide connectivity at low-costs [7, 11]. We propose to use USB sticks and CDs/DVDs carried by people traveling between villages to move data between different locations.

- *Caching*: Given the locality of content production and consumption evidenced in the Digital Green dataset, caching is likely to provide benefit by serving content from local copies rather than acquire it repeatedly from content providers. We therefore allow infrastructure nodes in the network to cache content.

- *Control and data separation*: We leverage the more-or-less ubiquitous GPRS/EDGE connectivity to enable an

always-on control channel, and delegate bulk data transfers to delay tolerant channels. The control channel is used to make content download and upload requests, and assist in content routing by initializing routing tables at transit nodes in the network. This helps avoid typical delay tolerant networking scenarios where routing schemes are designed to operate without topology or link information [12].

We have implemented this architecture and plan to do a pilot deployment in January 2012. In this paper, we describe the architecture and then simulate the Digital Green dataset to study the performance of multicast, cache replacement and pre-fetching algorithms in the architecture. Our simulations reveal that even simple LRU-caching with modest cache sizes can bring about a reduction of up to 80% in the amount of content downloaded from the Internet, with delivery latencies that are no worse than simple postal delivery of content via DVDs.

Our vision is that such an architecture could either be deployed by various development organizations to help them distribute content to their field offices, or by local entrepreneurs to fetch content and develop small businesses around publishing DVDs or copying music and videos to cellphones of people. In both cases, the deployments could evolve into a public infrastructure much like the Internet, and be used to distribute grassroots social media content in rural hinterlands.

The rest of the paper is structured as follows. In the next section, we present a detailed analysis of the Digital Green dataset to understand production and consumption patterns for social media. In the following section, we show how observations from the dataset motivate a network architecture for distribution of social media content in rural areas. The dataset is then used to run simulations on the proposed architecture to evaluate various routing algorithms on real-world topologies. Finally, related work and ideas for future extension are presented.

2. DATASET ANALYSIS

Digital Green is a non-profit organization that trains communities to create agricultural videos on crop sowing, harvesting, etc best practices by specifically involving the local community in the video production process. The videos are then shown on television sets and projectors in group meetings of farmers from the same community. This participatory method of video creation and dissemination is claimed to ensure better diffusion of agricultural practices than other methods such as broadcast of videos made by experts sitting in agricultural universities and government agencies [13].

Digital Green shared with us comprehensive data of their content generation and distribution activities from Jan 2009 to March 2011. During this time, approximately 1,000 videos were produced across the states of Madhya Pradesh, Orissa, Bihar, Karnataka and Jharkhand. These videos were cumulatively screened over 20,000 times for over 35,000 farmers. Details of screening locations were recorded at the level of villages. In India, villages are grouped into blocks, blocks into districts, and districts into states; we also had data about the blocks and districts to which different villages belonged.

In Figure 2(a), we show the production rate of videos at the block level. The video consumption rate, indicated
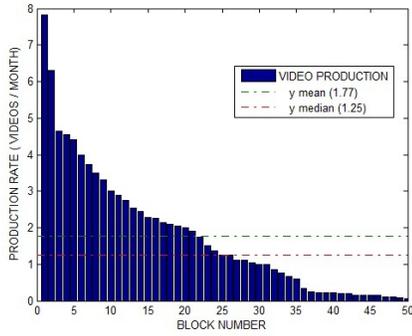
(a) Video production
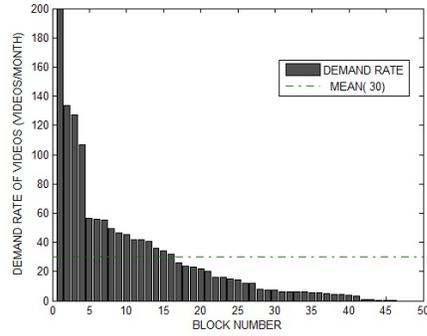
(b) Video consumption
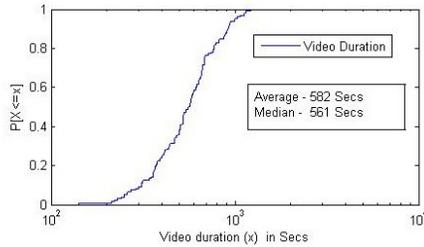
(c) Video size distribution
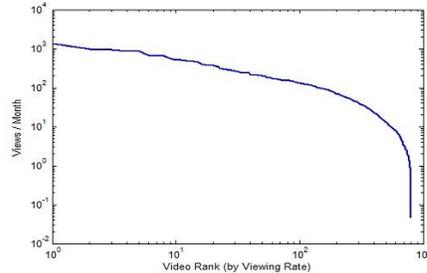
(d) Video popularity
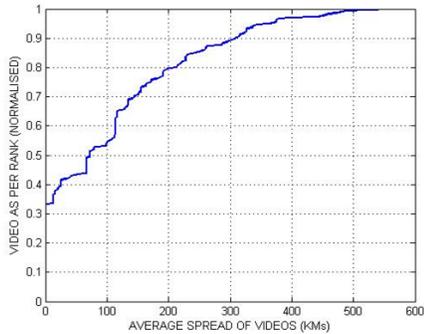
(e) Geographic spread
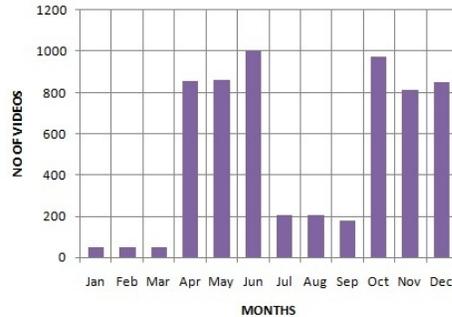
(f) Seasonality

Figure 2: Statistics from the Digital Green dataset

by the number of screenings of a video held in different places, is shown in Figure 2(b). Taking an average video length of 10min (or 50MB), indicated by the video size distribution in Figure 2(c), this translates into an upload demand of 100MB/month/block and a download demand of 1.5GB/month/block. Such high ratios of disparity between upload and download volumes has also been seen in other social media related systems [14]. Although the requirements for upload and download bandwidths are not much in absolute terms, the challenge for Digital Green is the poor quality of Internet connectivity in their areas of operation because of which they are not able to transfer videos of even a few megabytes [15]. They instead rely on different site operators to ship DVDs to the Digital Green headoffice in Delhi, where they aggregate videos received from all across the country and then send copies to everybody. An automated content distribution system would be of great value to them.

Figure 2(d) shows the Zipfian plot for the total number of views of different videos, in terms of the count of people who attended different screenings of the video. This is clearly a straight line for two orders of magnitude of video ranks, much like the power-law distributions seen in other social media systems [14]. However, there are perceptible cutoffs at the top and bottom of the distribution. This is interesting. The cutoff at the head of the distribution probably indicates that in the Digital Green system there are a few very popular videos but the videos have not overrun their lifetime as yet. The cutoff at the tail probably indicates that Digital Green purposely facilitates the viewing of certain niche content, unlike websites such as YouTube where niche content discovery is hard and therefore there is a longer tail.

We next study the geographical spread of videos, ie. how far from their production site are the videos viewed. Here

we only consider operations in the state of Madhya Pradesh since it is the most active in terms of content production and distribution, and has Digital Green services spread across 32 blocks. Figure 2(e) shows the ranked videos versus the 90%ile distance spread of the video. It can be seen that 33% of the videos are screened in the same block in which the video was produced. 80% of the videos have a geographical spread of 200 KMs, which is likely to span not more than two districts. The rest 20% of the videos do not have a spread of more than 600 KMs. This shows that content is highly localized. We will describe later how this localization property can be used to improve content distribution efficiencies through caching.

All the videos also had associated metadata indicating the ideal months in which the videos would be relevant. This tagging was probably done by Digital Green itself to help their staff select appropriate videos to screen at relevant times of the year. Figure 2(f) plots this relevancy tagging over time. There is clearly more emphasis placed during the Rabi and Kharif seasons. Kharif is a summer crop that is it is sown during April to June and harvested in the month of October, while Rabi is a winter crop sown in October and harvested in April. We will describe later how such semantic information about content can be used to improve content distribution efficiencies.

3. NETWORK ARCHITECTURE

We next use the observations made through dataset analysis listed in the previous section to come up with a network architecture specifically suited for social media content distribution in rural areas. We want to keep the following constraints and opportunities in mind:

- The common use-case for applications running on the network would be publish-subscribe applications, content broadcast, multicast, and downloads initiated by browsing content stores. This would cover typical requirements, such as those of Digital Green to distribute content to its field offices, for community radio stations to share content with other stations, and for citizen journalists to publish their content in a public space.

- The applications can be assumed to be tolerant to delays no worse than those experienced through postal transfer as if the content was shipped on DVDs. Options should be available for urgent data transfer though, if required.

- The infrastructure should provide data transfer capabilities to local rendezvous points in villages, such as kiosks or community radio/video stations. The last-mile distribution of content to community members can be assumed to happen through radio broadcast, or video screening in public gatherings, DVD distribution, copying of content to mobile phones over Bluetooth, etc. Similarly, gathering content from the grassroots can be assumed to happen when community members visit the station, or the station staff make field visits to interact with the community.

- 2G GPRS/EDGE connectivity is available in most rural locations, but it not suitable for large amounts of data transfers of the order of tens of megabytes.

- Content production and consumption can be assumed to be localized as a common case, as evidenced in the Digital Green dataset analysis.

Designing for the common use-case under the constraints given above logically leads us towards the network architecture we propose in this section. First, delay tolerance of applications and the inability of cellular networks to support large data rates implies that offline access will have to be a key feature of the architecture. The network should provide asynchronous functions for upload and download of data, and synchronization primitives to resolve local updates that were made while offline. We therefore choose a mechanical-backhaul solution using USB keys carried by people to ferry data across villages and cities, and a separate always-on control channel using cellular data services to issue data upload and download requests. Second, the focus on content distribution naturally encourages a content based architecture with functions to manipulate content objects as a whole, rather than individual packets. We therefore assume that the network is aware of content download and upload requests, publish-subscribe or multicast memberships, etc, and can transparently choose where to cache content objects, replicate them, track their availability, and identify optimal content download locations. We describe the proposed architecture in more detail in the rest of the section.

3.1 Entities
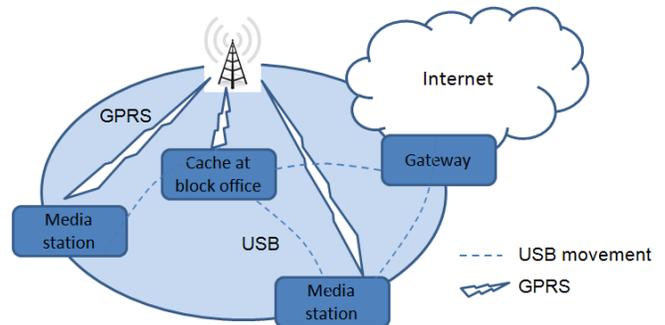
Figure 3: System design

Referring to Figure 3, following are the entities constituting the architecture:

- **Media station**: We use "media stations" as an alias for community radio or video stations, or any other endpoint such as a kiosk that runs applications to share content. Communication devices at media stations are assumed to carry two interfaces: an always-on GPRS/EDGE interface to serve as a control channel, and a delay tolerant interface through which content is moved using USB keys. Communication interfaces are generalizable to VSAT, dial-up, etc, but we do not mention them here for simplicity.

- **Gateway**: Gateways are devices that can be placed in the offices of non-profit organizations in cities, and are assumed to have always-on Internet access. Media stations register with one or more gateways, and issue content push/pull requests over their control channel. The gateways are assumed to be aware of the network topology in their geographical neighborhood, and possibly also aware of approximate movement schedules of people carrying USB keys. They then use optimization algorithms

to derive routes to move data to and from the media station and the Internet. They also keep track of multiple requests for the same content to be able to derive optimal replica locations and create multicast groups.

- **Caches**: Caches may run as modules on media stations and gateways, or even on independent nodes, and serve as content staging points. Figure 3 shows an example of an independent cache node located at a block office. This node serves to transit content moved by people carrying USB keys between village media stations and the block office, and between the block and district offices. Caches also maintain an always-on control connection so that gateways can update them with content routing information. For example, the cache nodes can be instructed to copy certain content objects onto appropriate USB keys according to the movement schedule of the keys.

- **Content lookup service**: A content lookup service in the Internet is used to maintain a list of replica locations for each content object. Content objects are assigned a unique human-readable identifier by their publisher, for example *bihar.digitalgreen.org/2010/04/paddy67s* would be an identifier assigned by Digital Green to a video on paddy cultivation from Bihar. Content publishers then host authoritative content name servers containing records for content objects about their current replica sites, indexed on the content identifiers. Much like in DNS, a set of root name servers point to the authoritative servers and help resolve content lookup queries. We have not shown content providers and lookup servers in Figure 3 for simplicity.

Whenever a cache module decides to hold a content object, it registers the object with the lookup service. Similarly, whenever it purges an object, it unregisters the object. Thus, when a gateway receives a content download request, it consults the lookup service to obtain the current replica locations for the content. It then computes routes from these locations to the destination, chooses the most optimal route, and uses the control channel to initialize routing tables at various intermediate nodes.

- **Application servers**: Application servers run browsing or publish-subscribe applications on the Internet. The architecture enables a way to run these applications offline at the media stations. A key-value pair database is provided to applications; a master copy of the database is maintained on the Internet, a local copy of a relevant subset is maintained at media stations, and the two are synchronized completely transparent to the application developer. Synchronization can happen either in real time over the GPRS/EDGE control channel, or by packaging the update into a content object that is transferred over the USB data channel.

We are able to meet all the goals and constraints mentioned in the previous section through this architecture. The content-based design allows cache nodes and gateways to implement intelligent routing and multicast algorithms for optimal delivery. Delay tolerance provides low-cost connectivity to remote areas. A control channel helps provide up to date knowledge of the network topology to infrastructure nodes to help them make routing decisions. And a database synchronization service makes offline-online application development easier.

3.2 Operations

Fig. 4 describes the same ideas in a network-stack view. Applications can issue content pull requests which make their way to the gateway over the available control channel. The gateway queries the lookup service to identify a suitable replica location for the content object from where it can be served, and initializes routes for its transfer over the network of caches. The actual data transfer then happens in a delay tolerant manner over the network of nodes, first making its way to the cache, and from there to the media station. Data is segmented and reassembled (SAR) into bundles, similar to the bundling protocol proposed by the DTN (Delay Tolerant Networks) research group [16]. Finally, applications are designed to work in an offline manner off their local cached databases, which are periodically synchronized with a master database in the Internet.

Uploads are done in a similar manner. An upload notification is sent over the control channel to the content provider, which assigns a content-id and adds an entry in the content lookup service. This entry indicates that a copy of the content is available at the originating node itself. The content is also dispatched over the delay tolerant channel to publish it with the content provider on the Internet. Upon publication, application servers can be notified so that other nodes can discover the content and issue download requests for it. The requests can be served from either the originating media station, or by downloading a copy from the content publisher via intermediate nodes, or from any cache locations that may have the content cached with them.

In the next section, we study different routing and replication algorithms on this architecture by simulating the Digital Green dataset for content sharing.

3.3 Deployment and extensions

We have completed a testbed implementation of the content distribution architecture, and built a publish-subscribe application on it. We plan to deploy it in the field in January 2012, in affiliation with our partner NGOs – Digital Green and Video Volunteers. Digital Green plans to finally use the network to distribute content to its 400+ rural sites, where staff and community members will organize public screenings of the videos. Video Volunteers plans to use the network to deliver content to its remote community video units, and burn DVDs on-demand at these locations[1].

We are also making extensions to the architecture to use mobile phones at the last mile. With the availability of low-cost smartphones and tablets such as the Android IDEOS [17], the network will allow mobile devices to make content download and upload requests on GPRS, and indicate their closest cache site from where they can pick-up or upload content offline whenever they are in physical proximity to the site.

4. SIMULATION ANALYSIS

We designed a simulator to evaluate caching strategies on our proposed rural content distribution architecture. We assume broadband Internet connectivity at district headquar-

---

[1]As part of another project, we are also developing a DVD authoring tool that lets administrators select videos, advertisements, photographs, etc, for burning on to DVDs. Different menu templates such as overlaying of video thumbnails on maps, or displaying tag-clouds for menus, would be possible.
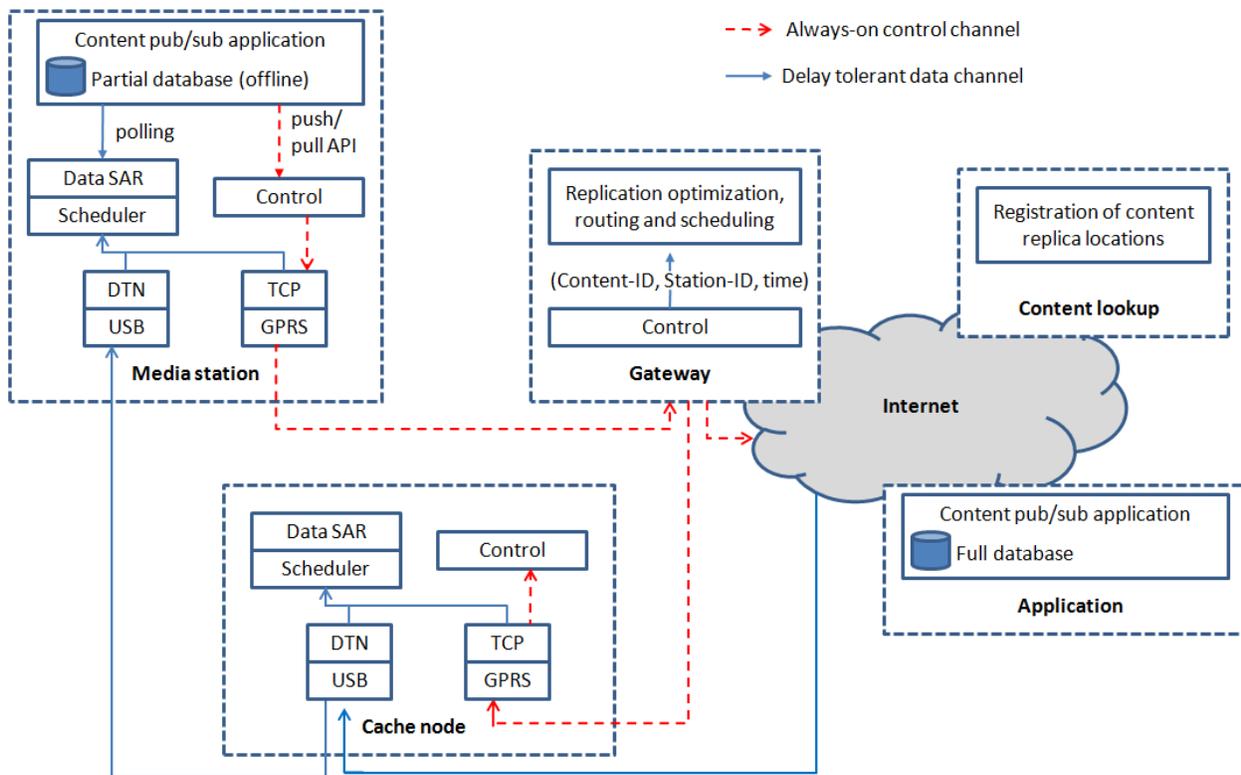
Figure 4: Network stack

ters, but only control channel availability at the village and block offices. Thus, district headquarters can be considered as gateway nodes for the block and village media stations in the district. We next describe various steps we took to create network topologies and USB-key movement schedules for the simulation experiments.

4.1 Topology construction

The dataset did not contain the latitude and longitude information of various sites. We were able to obtain latitude and longitude information for the districts and blocks from Google maps. However, since villages were not always marked, we created village locations randomly scattered around the block headquarters within a radius of 15 KM.

It was hard to extract any road connectivity information from the Internet maps. We therefore laid out roads according to a simple heuristic. We connected all blocks to their district headquarter, and connected two blocks together if the pair of blocks and the district headquarter formed an obtuse triangle. This would roughly mimic a road network that tries to maintain short distances between blocks and districts, much like what is noticed in practice. To connect villages to each other, we used a threshold of 3 KM, ie. locations within 3 KM of each other were connected directly. We then connected villages to their parent blocks, or to neighboring blocks, if none of their neighboring villages were connected. This would mimic clusters of villages sharing a single road to block headquarters, similar to what is seen in practice. For all calculations, we used the Eu-

clidian distance between pairs of locations to estimate their geographical separation.

We assume that all data transfers across district headquarters will happen over the Internet, and hence we do not simulate any USB-key movements across districts. This also simplifies our simulations because we can now examine different content distribution algorithms separately within each district rather than over the entire Digital Green network topology.

4.2 Movement schedules for USB keys

We assume that USB keys would be carried by staff and community members traveling between different villages and blocks on buses and other forms of public transport.

- Bus services were assumed to run between blocks, and between blocks and districts, at a frequency of 4-7 times per day in one direction. Buses would start after 7am in the morning and run up to 7pm in the night. The USB keys would begin their return journey after spending 2-3 hours at the destination.

- Movement between villages, and between villages and blocks, was assumed to happen using ad hoc means of road transport at a time chosen randomly between 7am to 7pm.

- At most one journey was assumed to happen between a pair of nodes (village-village, village-block, block-block, block-district) in a day, and at most four journeys per week. This is to model the case that multiple staff members from the same site would not travel on the same day, or frequently over a week.

This overall pattern mimicked the movement of Digital Green staff as described qualitatively by them, and also evidenced by us during field trips to even other NGOs for various projects.

4.3 Request generation

The video production and screening times in the Digital Green database were used to generate content upload and download requests on the always-available control channel. The content was then assumed to move on USB keys carried by staff traveling to or from the content destination.

4.4 Algorithms simulated

We implemented the following four content distribution algorithms. Each assumed knowledge of the expected delays on various links [12], and hence we were able to use Dijkstra's algorithm to find shortest paths.

- **Unicast**: Each content download request was routed independently to the destination from the nearest cache where the content was available at the time of request. Caches retained content transiting through them based on different cache replacement policies.

- **Multicast**: While a content object is in transit during unicast, another request may arrive for the same content. As an optimization on unicast, a multicast group is formed at this time to also include non-cache locations to serve content that may be in transit through them, rather than fetch it from some upstream cache node as in unicast. A multicast group would thus dynamically evolve for each content object.

  We are currently improving this method by probabilistically creating longer-living multicast groups based on the similarity of content requests made by nodes in the past.

- **Multicast with pre-fetching**: During multicast, caches with a high graph degree are also selected for replication of content. This exercises a simple heuristic that nodes with a higher degree are more likely to lie on shortest paths to other nodes, and hence content pre-fetched at these locations may have a better chance of getting utilized for download. The heuristic we used is to pre-fetch content at the top-k highest degree cache nodes.

- **Optimal multicast**: To serve as a lowerbound on content delivery latency, we assumed future knowledge of content demand and formed multicast groups on a per-content basis. A source-based multicast tree was then computed by merging shortest paths from all requesting nodes to the source, and the tree was populated with content upon the very first request from any group-member for the content. The content is also persisted in the cache for the lifetime of the multicast group.

For each algorithm, we studied two cache replacement policies: LRU, and a seasonality based policy. In the seasonality based policy, content was retained in the cache if its metadata indicated it to be relevant during that month; an LRU policy was followed for the remaining content.

4.5 Simulation results

We present here simulation results for six districts only, which are representative of a wide variety of topologies expected to arise in the field. Figure 5 shows these topologies with solid lines representing road connectivity between district and block headquarters, and dash lines signifying connectivity between blocks.

- **Dewas district**: A degenerate tree topology with a single leaf-node to provide connectivity at a cluster of villages away from the gateway.

- **Rajgarh district**: A two-leaf tree topology to provide connectivity at two clusters, both away from the gateway.

- **Guna district**: A two-node mesh topology with one gateway in the mesh.

- **Keonjhar district**: A three-node mesh topology with one gateway in the mesh.

- **Dindori and Barwani districts**: A typical multi-node mesh topology with one gateway in the mesh.

We mainly study three metrics to evaluate our architecture: delivery latency, the volume of content downloaded from the Internet at the gateway, and the volume of content kept in transit on USB keys. Simulations are run with different cache sizes and number of caches. The number of caches is indicated as a percentage of the total number of nodes in the network; note that block headquarters are also included in this count and are assigned caches in preference to caches at village nodes.

4.5.1 Improvement with caching at gateways

Table 1: Download requirements at gateway

District	GB/month w/o cache	GB/month with 1GB cache	Improvement (%)
Dewas	8.73	1.13 ± 0.05	87.07
Rajgarh	7.94	4.70 ± 0.12	40.82
Guna	8.51	3.79 ± 0.18	55.44
Keonjhar	20.07	3.71 ± 0.3	81.50
Dindori	17.97	2.59 ± 0.18	85.59
Barwani	2.42	0.25 ± 0.01	89.73

Table 1 shows the improvement possible with 1GB cache sizes at the gateways and block nodes. This improvement is clearly because most content production and consumption is local to a district, hence caches can save considerable download bandwidth at the gateways. Although the exact numbers for appropriate cache sizes will change with the volume of data downloaded, the insight here is that a modest cache size of 1GB compared to monthly download volumes of 10GB is sufficient to cause up to an 80% improvement in download bandwidth requirements.

4.5.2 Delivery latencies within districts

We next study the average delivery latencies for content transfer within different districts. For simplicity, we do not consider any download or queueing delays at the gateway. Figure 5 shows the relative performance of unicast, multicast, and the optimal-multicast algorithms with LRU cache replacement for the six districts we have examined. Results for cache sizes of 1GB are shown here.

(a) Dewas: Topology

(b) Rajgarh: Topology

(c) Guna: Topology

(d) Devas: Average latency

(e) Rajgarh: Average latency

(f) Guna: Average latency

(g) Keonjhar: Topology

(h) Dindori: Topology

(i) Barwani: Topology

(j) Keonjhar: Average latency

(k) Dindori: Average latency

(l) Barwani: Average latency
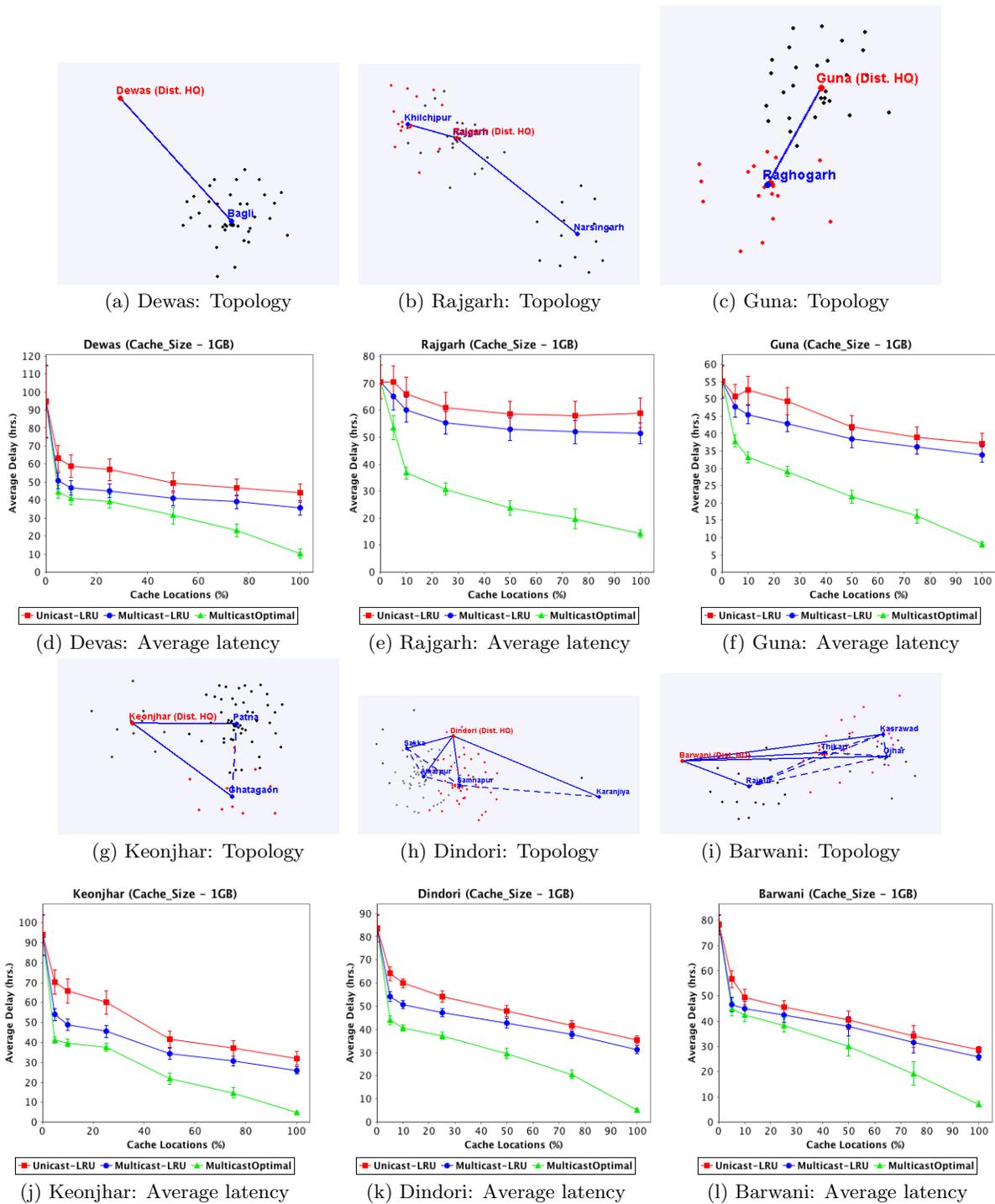
Figure 5: Simulation results for average data delivery latency

We observe that caching with unicast at the block headquarters in most districts (low %ages of number of caches), especially at Dewas, immediately results in a substantial improvement because of the inherently hierarchical topology – content gets located at the block level itself. With higher percentages of cache nodes, more and more improvement is noticed in mesh-like topologies than the tree topologies of Dewas and Rajgarh. Closer examination reveals that this is because short-circuited routes between villages begin to be used more often in mesh topologies rather than move content up and down the hierarchy. Overall, caching improves unicast delivery times by up to 50% in several cases, and at least by 15% otherwise, making a strong case to use caches at intermediate nodes.

Comparing unicast and multicast with the optimal, we notice that there is large scope for improvement of between

50% to 80% in various districts. Better heuristics can certainly be developed. For example, we can observe the sets of nodes that have similar interests and pre-fetch content accordingly for them. We also noticed in a few cases that local content produced around blocks some times evicts cached content that arrived from adjacent blocks. This could be improved by allowing the eviction of local content more easily than the eviction of remote content, potentially by ranking content based on the distance it has traveled. We are currently investigating such schemes more closely.

All the above stated results used the LRU cache replacement policy. We did not observe any noticeable difference with seasonality based cache replacement [18]. A closer examination of the data revealed that even though Digital Green had indicated preferential viewing of content according to the relevancy periods of content, but much popular content was actually screened all the year round while unpopular content was screened locally over a single round and not requested subsequently. We believe that this may have been done by Digital Green staff to sustain community interest in video screenings throughout the year; other datasets may follow a pattern more closely aligned with the metadata of content.

We also ran simulations for varying cache sizes, but did not notice any significant improvements beyond 1GB [18]. Upon examining which particular videos end up occupying the cache, we noticed that some popular videos were perennially present, while many other videos were evicted within a few months. We again believe this to be an artifact of Digital Green's screening strategy, that they make a round of neighboring villages to screen the same set of videos, and then repeat the round with a different set of videos. Caching therefore works well with small cache sizes because of the temporal locality of content demand.

4.5.3 Improvements with pre-fetching

Pre-fetching did not seem to offer significant improvements in latency across different districts, but in some cases it did somewhat improve the amount of data kept in transit on USB keys [18]. Closer examination of hit-rates at the high-degree locations identified for pre-fetching revealed that these nodes were often already included in the multicast tree, or sometimes the pre-fetched content got evicted from their cache before it got a chance to get requested.

We also analyzed the autocorrelation coefficient of requests per day for a few popular videos to understand if current popularity could serve as an indicator for future popularity, and thus be used to trigger pre-fetching for that content. However coefficients never went beyond 0.3 at lags of up to a week, and dropped even further beyond that. This indicates that popularity based predictors for content pre-fetching may not work well either.

4.6 Discussion

Insights from our simulation results can be improved with a closer examination of the data. Several data-specific aspects seem to influence the results, such as the strategies followed by Digital Green to select which videos to screen where and when. This also makes the dataset different from typical social media website datasets where users discover content on their own, albeit some times assisted by featured content on the front page, whereas here the selections were largely made by Digital Green staff based on their intuitions of what content may be liked by the people. Similarly, the geographical topologies dictate the extent to which short-circuiting of routes may happen, and therefore modeling the topology would be useful to determine the benefit that can be gleaned from cache placements at village nodes. Overall though, the simulations do illustrate the benefits of caching and multicast, and encourage further investigation.

5. RELATED WORK

Several pieces of work have discussed network systems to provide connectivity in rural areas. Sonesh, et al [6] developed a new MAC scheme to enable WiFi links to operate efficiently at long distances, and did a large deployment in India with a video conferencing application running on the infrastructure. We adopt a different approach of using delay tolerant networking principles to keep costs low. Systems like Daknet [19], KioskNet [7], and vLink [20] are also delay tolerance based and use mechanical backhaul such as cars and buses to ferry data between villages and cities, but are architected to largely target unicast transactional applications. On the other hand, our design is more suitable for content distribution applications. Formalizing the use of physical media such as CDs and DVDs to transport data was also proposed in systems such as PostmanNet [21]; our system is however grounded in a real application context and hence a step beyond abstract architectures. Further, we are unique in our approach of using GPRS/EDGE connectivity to serve as a control channel, which distinguishes our work from other architectures based on DTN [11, 22, 16].

Several application development frameworks aim to provide Internet access over flaky network connections [23], [24], and [25], but work in a online-offline manner rather than over multiple hops of disconnection. We too provide database synchronization primitives to make it easier to develop applications, but we are unique in our approach to synchronize namespaces of key-value pairs and to build it as part of a larger framework that operates on generic disconnection scenarios with multiple hops.

Finally, we also contrast our architecture with efforts along a clean slate redesign of the Internet. We have drawn many insights and techniques from existing work on content based networking. Koponen, et al [10] assign opaque names to content objects and rely on a DHT-based registry to keep track of locations at which content objects are available. Our approach of using a content lookup service is similar, although we choose to use human-readable names for content objects and develop a DNS-like lookup service. They also suggest adding a shim layer between TCP and IP for backward compatibility to convert the existing network into a content based network; we however design a new architecture from scratch and use our own DTN-like bundling protocol instead of TCP/IP for content transfer. Jacobson, et al [8] also suggest the use of human readable names for identifying content, and depart from traditional *host centric* thinking about networks to a new *content centric* design. They propose a novel interest dissemination broadcast so that data can flow along the reverse direction taken by interest messages; this eliminates the need to assign a host destination to packets or even to have any knowledge of the underlying network topology; instead packet destinations can be content names. Interest broadcast however fails on delay tolerant networks having multiple interfaces; topology knowledge is required at this point, for example, to let the

network know whether to send data on USB keys, whether to use VSAT at its scheduled hours, or to use GPRS/EDGE at night hours when data transfers are cheaper. To handle such cases, we realized that we could not get away with ignoring the network topology, and hence we use a control channel to specifically initialize routes for content bundles. Jokela, et al [9] leave the content discovery and routing problems open, and propose to embed into the content header a Bloom filter of links over which the content is to be forwarded. Their solution is appropriate for the Internet context where fast switching of content bundles is required. In our case, since link transfer latencies and the time available to make switching decisions is typically much more, and we also assume the availability of a control channel, we choose to work with the traditional forwarding-table approach without adding source route information to the content bundles.

6. CONCLUSIONS

In this paper, we make a case for deploying delay tolerant content distribution networks in rural areas. First, we observe that significant amounts of digital content is produced in remote rural areas and there is a need to share this content with other locations. Second, we analyze a dataset of content production and consumption patterns and observe significant spatial locality in content accesses. Third, we design a content distribution architecture that is delay tolerant and able to make use of caching opportunities to distribute content. Fourth, we simulate different content routing and caching algorithms in line with our proposed architectural framework, and evaluate their performance for different network topologies. We discover that with modest cache-size to monthly download-volume ratios of 1:10, our proposed design is able to achieve up to 80% improvements in bandwidth requirements. Further, the delivery latencies guaranteed by our architecture are of the same order as postal delivery delays when content is shipped across on DVDs.

We argue that development organizations interested in disseminating digital content in rural areas, or to their rural site offices, should seriously consider using such a network architecture. The additional IT investment required is minimal – their field staff already have access to desktops and laptops which can run the network libraries, and hence the only additional investment is in procuring USB keys for their staff. It is possible though that different realizations of the architecture may be appropriate in different deployment contexts; for example, if field staff do not have access to laptops or smartphones to run applications, it may be worth considering an IVR system to make content requests and ship SD-cards with the requested content copied on to them. Nonetheless, the main advantage of the approach we are advocating is to automate content delivery to enable customized service to the end points. Manual selection of content and shipping of DVDs is painstaking, error prone, and not scalable. We believe that if a few development organizations take the initiative and set up a small network, it may even attract local entrepreneurs to grow the network by making a business out of fetching content to burn DVDs, or copy it on to SD cards of phones for a small fee. Copyright and piracy of content will be a significant challenge in such a public network though, and we can examine the problem once the incentives of different players in the prospective ecosystem become clearer. Nonetheless, the ability to record a song by your child, or film incidents of corruption and vi-olation in your village, and have them viewed all across the country, can indeed bring a strong feeling of empowerment that most rural communities have not experienced so far.

7. REFERENCES

[1] www.videovolunteers.org
[2] www.digitalgreen.org
[3] indiaunheard.videovolunteers.org
[4] A. Seth, "Community Radio 101," http://www.crindia.in, 2011.
[5] S. Bailur, "The Complexities of Community Participation in ICT for Development Projects," Proc. Social Implications of Computers in Developing Countries, 2007.
[6] R. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer, "WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks," Proc. USENIX NSDI, 2007.
[7] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, "Low-cost Communication for Rural Internet Kiosks Using Mechanical Backhaul," Proc. ACM Mobicom, 2006.
[8] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking Named Content," Proc. CoNEXT, 2009.
[9] P. Jokela, A. Zahemszky, S. Arianfar, P. Nikander, and C. Esteve, "LIPSIN: Line speed Publish/Subscribe Inter-Networking," Proc. ACM Sigcomm, 2009.
[10] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A Data-Oriented (and beyond) Network Architecture," Proc. ACM Sigcomm, 2007.
[11] Kevin Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," Proc. ACM Sigcomm, 2003.
[12] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," Proc. ACM SIGCOMM, 2004.
[13] R. Gandhi, R. Veeraraghavan, K. Toyama, and V. Ramprasad, "Digital Green: Participatory Video for Agricultural Extension," Proc. ICTD, 2007.
[14] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," Proc. IMC, 2007.
[15] R. Gandhi, "Personal communication," 2009.
[16] Delay Tolerant Networking Research Group, www.dtnrg.org
[17] J. Ford, "$80 Android Phone Sells Like Hotcakes in Kenya," singularityhub.com, 2011.
[18] A. Mahla, "Content Distribution Network for Rural Areas: Architecture and Simulation," MTech thesis, IIT Delhi, 2011.
[19] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking Connectivity in Developing Nations," IEEE Computer, 37(1):78-83, 2004.
[20] Teterless Computing Lab, University of Waterloo, "VLink," http://blizzard.cs.uwaterloo.ca/tetherless/index.php/VLink, 2009.
[21] R. Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, and A. Krishnamurthy, "Turning the Postal System into a Generic Digital Communication Mechanism," Proc. ACM SIGCOMM, 2004.
[22] J. Scott, P. Hui, J. Crowcroft, and C. Diot , "Haggle: A Networking Architecture Designed Around Mobile Users," Proc. IFIP WONS, 2006.
[23] A. Seth, M. Zaharia, S. Bhattacharya, and S. Keshav, "Policy Oriented Architecture for Opportunistic Communication on Heterogeneous Wireless Networks," Manuscript, University of Waterloo, 2006.
[24] R. Luk, M. Zaharia, M. Ho, B. Levine, and P. Aoki, "ICTD for Healthcare in Ghana: Two Parallel Case Studies," Proc. ICTD, 2009.
[25] S. Shah and A. Joshi, "COCO: A Web-Based Data Tracking Architecture for Challenged Network Environments," Proc. ACM Dev, 2010.