Science of Continuous Media Application Design in Wireless Networks of Mobile Devices

Shahram Ghandeharizadeh
Department of Computer Science
University of Southern California
Los Angeles, California 90089-0781

Abstract

Display of continuous media using self-organizing ad hoc networks of wireless communication systems will potentially be used in a variety of applications. Example deployments might include disaster relief missions, conferences, and university campuses to name a few. Challenges of these environments include their mobility, and unpredictable network bandwidth and loss characteristics. This paper explores a three step science of design for applications that manipulate continuous media. This science strives to satisfy the requirements of an application. It consists of a number of principles that impact the design of algorithms. These principles guide a system designer towards parameterized algorithms that treat network bandwidth and storage as one.

1 Introduction

Continuous media, audio and video clips, are essential to numerous applications such as distance education, collaborative scientific environments, homeland security, and entertainment. With distance education, a user at a remote location may access and view multimedia instructional material, e.g., SuperNet [5] and DakNet [24]. Similarly, with scientific research and homeland security, devices deployed at either a hazardous location or a distant planet may collect audio and video recordings from their environment for subsequent retrieval and processing. With entertainment, communities of Home-to-Home Online [8, 21, 11] (H2O) devices may collaborate to provide video-on-demand service to each household.

These applications might be deployed using a variety of stand-alone and networked environment. The focus of this paper is on those deployments that must support both environments. We detail a science of design for applications that store and deliver continuous media. This science is essential to development of complex systems such as Memex [4] and MyLifeBits [9]. Given the current technological trends, it is not far fetched to envision a personal Memex system that fits on a key chain. This device might display its data either on an individual's glasses or a near-by monitor. It would communicate with other nearby devices using its wireless card. These devices would collaborate to provide each user with access to a large volume of data. In addition to operating when in contact with other devices, this device must operate in a stand-alone mode when its user moves to a location that is not network reachable by other devices. Thus, mobility is one factor that impacts data services provided by a device. By predicting mobility, a device may perform preventive operations to increase its Quality of Service (QoS). (Section 2 details alternative metrics to quantify QoS.) To illustrate, consider a user who employs his or her device to record data from an on-going conversation with a colleague. If this device detects movement towards an area sparsely populated with other devices, then it might start to free its storage by pushing its recently recorded data to other devices for permanent storage. Once in a remote area and operating in a stand alone mode, if this device detects shortage of space, it can delete this pushed data in order to continue recording additional data.

In addition to storing data, applications may desire to display data. Continuous media consist of a sequence of quanta, either audio samples or video frames, that convey meaning when presented at a pre-specified rate [10, 18]. Once the display is initiated, if the data is delivered below this rate (without special precautions) then the display might suffer from frequent disruptions and delays, termed hiccups. Depending on their display time, audio and video clips are typically large in size when compared with traditional media such as text. For example, a 90 minute DVD quality video is typically quoted as 2.5 Gigabytes in size.

Applications of continuous media can be broadly categorized into those that manipulate either static, dynamic, or both content types. Static content pertains to delayed mode of communication consisting of pre-recorded data archived for future retrieval and display, e.g., display of a recordings from a recent meeting or seminar. Dynamic content corresponds to immediate mode of communication and generated in almost real-time for display at a client, e.g., scientists communicating and collaborating across different laboratories. The boundary between these two types of content is defined by the delay from when data is produced to the time it is displayed. As this delay increases, dynamic content becomes static. An example is when the collaborating scientists record their real-time communication for future display and analysis.

We believe there is a science of design for continuous media applications. This science strives to meet an application's requirements given the physical constraints of an environment. It consists of three steps, see Figure 1. Step 1 produces a multi-dimensional framework to describe the requirements of applications sharing data. Step 2 requires a system designer to taxonomize available algorithms and their tradeoffs. Step 3 maps Steps 1 and 2 to produce an application-centric solution constrained by the environment. When considering performance objectives of an application, this science must consider the characteristics of the target hardware platform. Step 3 might perform the mapping between Steps 1 and 2 in either an off-line or an on-line manner. When applied in an off-line manner, Step 3 computes parameters of algorithms that constitute a system for deployment purposes. This is appropriate when the characteristics of the target hardware platform is fixed. This is typically not the case with an ad-hoc network of collaborating devices. In particular, the bandwidth of a wireless network connection between two nodes may vary as a function of physical changes to the environment. In this case, an on-line application of Step 3 is more appropriate because it constantly computes and adjusts the parameter settings of algorithms to meet the application's requirements.

This science is important because continuous media is wide spread and impacts design and implementation of many diverse applications with different requirements. By identifying this scientific process and its design principles, one builds a foundation for systematic creation of application-centric, software intensive systems that store and retrieve continuous media. Its systematic property empowers system designers to understand the peculiar oddities of their design decisions (law of large numbers) in advance and requires them to examine those that are not tolerable and must be addressed. The principles guide the designers toward parameterized (general) algorithms that strike a compromise between conflicting factors that are application dependent. This increases robustness of software where the same algorithm supports diverse applications. The only difference is the parameter settings of an algorithm.

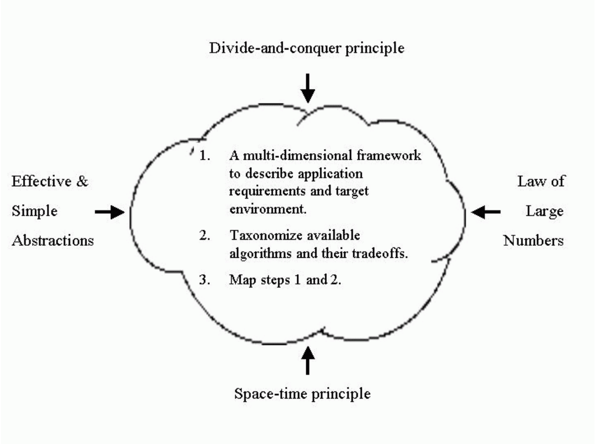
Divide-and-conquer principle
Effective & Simple Abstractions
Law of Large Numbers
1. A multi-dimensional framework to describe application requirements and target environment.
2. Taxonomize available algorithms and their tradeoffs.
3. Map steps 1 and 2.
Space-time principle

Figure 1. Science of physical data design and four principles that guide this process.

2 Overview

The following sections are organized around the three steps of our proposed science of design and a general purpose software architecture for a wireless network of ad-hoc devices, see Figure 2. Section 3 details the functionality of system (first two layers). This section focuses on hiccup-free display and details the requirements of an application along three dimensions: Quality of Service, efficiency, and availability of data.

Section 4 elaborates on the underlying algorithms and software components that strive to support the specified functionalities and requirements. This discussion corresponds to the second step of our science of design. It identifies the following three principles: space-time, divide-and-conquer, and law of large numbers. These principles are not intended to be an exhaustive list. Instead, they are a subset of those that we intend to develop in the next few years.

Finally, Section 5 focuses on the third step, mapping between parameters of different algorithms and the application requirements. It describes the importance of abstractions that enable a system designer to analyze how well a certain design decision impacts an application's requirements. These abstractions might be based on analytical and simulation models. They can be used to estimate the capacity of a specific deployment. Moreover, they can be applied at run-time (in an on-line manner) to adjust the parameter setting of algorithms in response to environmental changes in order to meet the requirements of target applications.

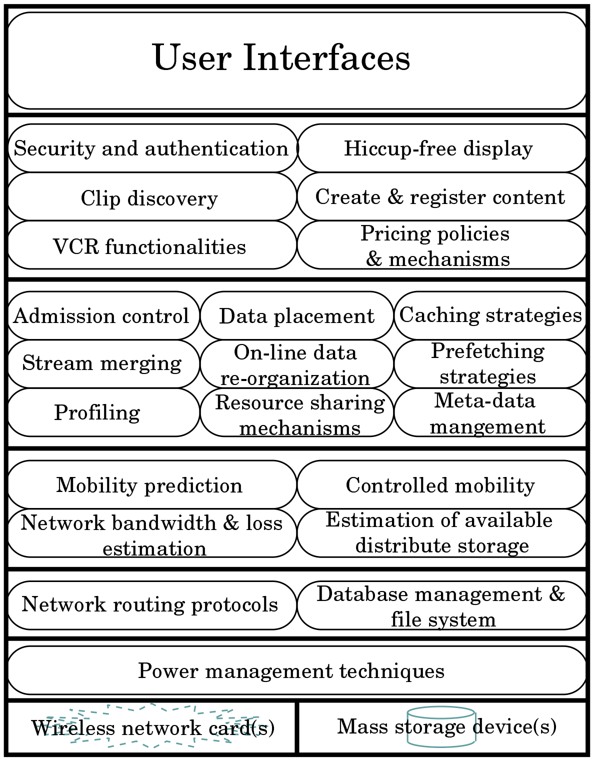
User Interfaces
Security and authentication
Hiccup-free display
Clip discovery
Create & register content
VCR functionalities
Pricing policies & mechanisms
Admission control
Data placement
Caching strategies
Stream merging
On-line data re-organization
Prefetching strategies
Profiling
Resource sharing mechanisms
Meta-data mangement
Mobility prediction
Controlled mobility
Network bandwidth & loss estimation
Estimation of available distribute storage
Network routing protocols
Database management & file system
Power management techniques
Wireless network card(s)
Mass storage device(s)

Figure 2. Components of a software architecture.

3 Application requirements and functionalities

The requirements of applications are dictated by their desired functionalities. As detailed in Section 1, these functionalities can be categorized into real-time versus delayed mode of communication, display versus recording of continuous media. One may find a mix of these functionalities in an application. In the following, we identify the key functionalities required from a system. Subsequently, we focus on hiccup-free display and the requirements of an application.

Effective user interfaces enable an individual to utilize devices to their full potential. These interfaces must be simple and intuitive. A keyboard to enable users to input their user-id and password is most likely the wrong interface to either facilitate privacy of content or identify a user. Novel approaches are required to enable an individual to query and discover static contents and candidate sources of dynamic content, select and preview content, publish new content for either private use or sharing with other users, etc.

Security of content and privacy of users is an important challenge faced by applications that manipulate continuous media. These challenges include diverse topics such as preventing improper profiling and their use, and preventing un-authorized display of private content. In the following, we elaborate only on these two topics. With the first, an application may gather profiles of how each device is used in order to predict future pattern of use in support of preventive operations. At the same time, profiles might be abused by advertisers for target advertising. All target advertising is not bad. For example, a user may elect to participate in target advertising in return for a service, e.g., video e-mail. Another example is a household that desires a profile describing what percentage of their time is spent viewing adult content and during what time of day. They may desire this profile in order to discontinue programming not suitable for children. While this appeals as a service, this same profiling might be perceived as a violation of privacy when it is gathered without a household's permission.

With the second, users may employ devices to store and retrieve a variety of personal content such as video emails and diaries. "**Create and register content**" component of Figure 2 enables users to upload new content. While a user may want to display this content remotely, viewing of this private content must be limited to those authorized to view this data. The deployed policies and mechanisms to address this issue must be flexible enough to empower a household member to (1) view content using a display located anywhere with ability to connect to a device, and (2) grant and revoke permission to others who may want to display a clip. The system may encrypt content to provide both privacy of content and implement a business model that is realized using a variety of **pricing policies and mechanisms** (another component of Figure 2). Finally, the system may track which users display content marked as private and provide this information to the owner of the content. Such a profile would enable the content-owner to detect violation of privacy. These violations can be prevented by either authoring new policies or adjusting existing policies.

A **discovery process** computes available content for display to a device. This functionality might be two fold. First, one may ask whether a specific content or source of content is available. For example, with a personal Memex system, a user may query whether a friend is network reachable for real-time communication. Second, the application may ask for all available content. In our example, a user may request all those instructional clips that describe Einstein's theory of relativity. In both cases, the discovery module should display those content that satisfy both their specified privacy and security policies, and QoS requirements of the target application. This module might employ either a detective or a predictive approach. A detective approach waits for a user to request identity of available content before computing the available data. A predictive approach may maintain a list of available content up to date. This might be performed in a variety of ways ranging from periodically issuing detective queries to employing statistical approaches that employ mobility to refine the list of available titles.

In [15], we investigate availability of clips in the context of vehicle entertainment systems. This study assumes vehicles are equipped with C2P2 [17] devices. Each device might be configured with hundreds of gigabytes of storage and collaborate with other devices to provide data services to passengers of a vehicle, e.g., video-on-demand. We assume a two-tier architecture which leverages the low-rate cellular infrastructure as a control plane for the high-rate data plane consisting of the C2P2 ad-hoc network. A cellular base station maintains the identify of titles available on each C2P2 device. When a C2P2 device enters the coverage area of a base station, the base station determines the list of titles available on this C2P2 device by contacting the base station that handed it this vehicle. Next, if the change in its list of available titles is greater than some threshold, it broadcasts the change to all vehicles in its coverage area. PAVAN [15] is a policy to estimate which titles are available to a C2P2 device within the next δ time units. PAVAN is invoked by each C2P2 device independent of others. It employs a Markov mobility model for its prediction. Results presented in [15] shows that the quality of PAVAN's predictions is dependent on degree of replication for titles, and relative display time of a clip relative to trip duration of a C2P2 device.

The discovery process impacts availability of data. If a continuous media clip cannot be discovered then it is available to those knowledgeable individuals who are aware of its existence, and know how to retrieve it.

Hiccup-free display of a clip requires data to be delivered to a client in a manner that prevents it from starving for data once it has initiated the display of a clip. With digital audio, hiccups result in random noises that sound like pops and cracklings. With digital video, the display freezes on a frame that might be partially distorted. Typically, a client overlaps display of a clip with its retrieval from a remote server to minimize the amount of delay from when the user requests the clip to the onset of its display, termed startup latency. This form of data delivery is termed streaming.

An alternative to streaming is for the client to download the entire clip prior to initiating its display. In a wireless network with a limited amount of bandwidth, this may result in long delays. To illustrate, consider a 90 minute clip with a bandwidth requirement of 4 Mbps. If the available bandwidth between the client and a server is 4 Mbps then the client must wait for 90 minutes before initiating the display. With streaming, the client initiates the display of the clip once the first few blocks of the clip arrive. This is because the network bandwidth is sufficient for data production to match data consumption. This discussion motivates the need for an admission control module that may block a new request by another client in order to prevent an active display from observing hiccups.

VCR functionalities are available in consumer VCR and DVD players, and many professional video editing equipments. These functionalities are almost mandatory for digital video. They enable users to pause, fast-forward, and fast-rewind a display.

An application's requirements for a given functionality can be quantified along different dimensions. To illustrate, the requirements of a hiccup-free display can be quantified along the following three dimensions: (i) Quality of Service (QoS), (ii) efficiency, and (iii) availability of data. We describe each in turn. Quality of Service (QoS) is a qualitative requirement, characterizing the perception of end users. It can be quantified using different metrics. One is startup latency. A second might be the frequency and duration of hiccups. A third is the resolution of streamed clip. While the displayed data might be hiccup-free, it might have been encoded with a significant loss. In particular, scalable compression techniques [28, 7, 20, 25] control the quality of streamed data based on the available amount of network bandwidth.

Efficiency describes how intelligently resources are utilized when streaming content. A key metric with static content is how many devices may display their referenced content simultaneously. With a network of $\mathcal{N}$ candidate clients, ideally all $\mathcal{N}$ clients should be able to display clips. With dynamic content, a key metric is the number of devices that can participate in real-time communication simultaneously. Typically, either the available network bandwidth, the available storage, or their combination is a bottleneck resource. These resources must be managed intelligently to satisfy requirements of an application.

Availability quantifies what fraction of accessible content or producers of content are available to a client at a given instance in time. It is defined by the system's ability to discover and retrieve data. The system must be able to deliver discovered content to the requesting device in order for that content to be advertised as available. As detailed below, streaming of data between a producing and a consuming device requires reservation of bandwidth. With ad-hoc networks of wireless devices, this may exhaust certain paths in the network, isolating a candidate client device from those devices that contain relevant clips. These clips remains unavailable until the reserved paths are released. Another important factor is the characteristics of the wireless network. Both the bandwidth and loss rate of an 802.11 connection between two devices is dependent on its deployed environment and might vary from one minute to the next due to factors such as exposed node [27, 26, 3]. Finally, mobility of devices poses both challenges and opportunities. It is a challenge when it renders relevant content un-deliverable or leads to degradation in quality of service. It is an opportunity when it provides relevant content to a potential consumer directly instead of a multi hop transmission.

4 Design principles

Components of Figure 2 consist of parameterized algorithms whose settings is a tradeoff between QoS, efficiency, and availability metrics. The purpose of this section is to highlight three design principles for these components: space-time, divide-and-conquer, and law of large numbers. This discussion introduces each component and its functionality. The presented principles are not intended to be exhaustive. We have dropped a few intentionally because they are well known, e.g., separation of policies from mechanisms [22].

4.1 Space-time principle

Space-time principle states time is not completely separate from and independent of space. Instead, space and time are combined with one another to form space-time [19]. This early 1900 insight by Einstein (theory of relativity) and Poincare enabled distance to be measured precisely where the meter is defined to be the distance traveled by light in 0.00000000333564092 seconds, as measured by a cesium clock. The space-time principle paved the way to many important discoveries of natural phenomena such as black-holes.

When designing algorithms for hiccup-free display, storage space (either disk or memory) must be considered as an integral component of time (latency, bandwidth, cost of bandwidth). In particular, data can be prefetched in combination with streaming to minimize startup latency. The prefetched data occupies storage space until display is initiated. Prefetching is particularly appropriate when the available network bandwidth (denoted B_{Path}) between a data producing and a consuming device is below the clip's required bandwidth for a hiccup-free display (denoted $B_{Display}$). Assuming S_i denotes the size of the referenced clip i, the amount of prefetch data $S_{P,i}$ is [12]: $S_{P,i} = S_i - \lfloor \frac{B_{Path}}{B_{Display}} \times S_i \rfloor$. The time required to stage $S_{P,i}$ bytes of data at the displaying client dictates the incurred startup latency.

To illustrate prefetching, consider delivery of a 27 minute video clip with a display bandwidth requirement of 4 Mbps from a server to a client across a network connection that offers only 3 Mbps. The client must buffer 6.75 minutes of this clip prior to initiating its display in order to prevent hiccups. To prevent a user from waiting 6.75 minutes, one may pre-stage the first 6.75 minutes of the clip onto each candidate client, see [12] for a formal proof and relevant formulas. There are multiple ways of describing this space-time example. One may state "This design uses 6.75 minutes worth of storage (space) to reduce delay observed by a user to zero seconds (time)". Another interpretation might state "6.75 minutes of storage (space) is used to support immediate hiccup-free display of a clip delivered at a rate below its $B_{Display}$ (bandwidth)". In essence, storage space is an integral component of time and bandwidth.

The pre-stagging of data in anticipation of its future use is termed data placement, see Figure 2. A smart data placement technique enhances both system efficiency and QoS metrics such as startup latency. A data placement technique must address three key issues: 1) What is the granularity of data (blocks versus clip) for placement across devices? 2) How many replicas of a data item should be constructed in the system? and 3) How should the data items and their replicas be placed across devices? We explore the answer to question 1 in Section 4.2. Solutions to the remaining two questions are provided below.

In [1], we explored alternative techniques to compute the number of replicas when the granularity of data placement is a clip. Assuming the frequency of access to a clip i is known (f_i), this study identified a family of techniques that replicates a clip i proportional to $(f_i \times B_{Display})^{\alpha}$. This study focused on two possible values for α: 1.0 and 0.5. It shows that $\alpha = 0.5$ (square-root) results in a higher efficiency when compared with $\alpha = 1.0$. It defines efficiency as the number of devices that can display either the same clip or different clips simultaneously [1]. This study analyzes other alternative such as computing number of replicas proportional to $B_{Display}$, $B_{Display} \times (f_i)^{0.5}$, clip size ($S_i$), display time ($d_i$), $(S_i \times f_i)^{\alpha}$ and $(d_i \times f_i)^{\alpha}$ where α

is either 0.5 or 1. As a comparison yard stick, we employ a random policy that decides the number of replicas using a random number generator and ignores clip properties such as frequency of access and $B_{Display}$. Obtained results show efficiency is maximized with $(f_i \times B_{Display})^{\alpha=0.5}$. Moreover, the other alternatives are at times worse than random.

In [14], we extended our study of replication criterion to a mobile environment. In this study, we focused on a technique to estimate availability latency for a clip as a function of its number of replicas. Availability latency is defined as the delay observed from when a moving device requests a clip to the time it encounters a device containing the referenced clip. This study showed that a random policy for deciding the number of replicas is as good as any for a certain threshold defined as a function of the number of moving devices, storage per device, and movement pattern of devices. A technique that determines the number of replicas as a function of f_i^{α} with α=0.5 is typically superior to α=1.

Techniques for on-line data re-organization adjust the original placement of data in response to changes in frequency of access to data over time. It might be performed in either an eager or a lazy manner. With lazy, it stores the data it might be routing for display to another device. With eager, it initiates retrieval of data from other devices to either occupy its free space or replace its existing data. Both decision making processes require mechanisms to estimate benefits and costs of the current and a target future placement.

The space-time principle offer the following advantages. First, it states a system may satisfy the requirements of an application by utilizing its storage (Step 1). Second, it dictates flexible algorithms that can effectively trade space for time (Step 2). Third, it requires an understanding of this space-time tradeoff in order to map algorithms to applications effectively (Step 3). This is further elaborated in Section 5.

4.2 Divide and conquer

This principle breaks an instance of a problem into many smaller instances. It solves each sub-instance independently and combines their solutions to yield a solution for the original problem. With continuous media, a designer may divide hiccup-free display of a long clip as hiccup-free display of many short sub-clips. To illustrate, consider display of a clip X with display time of 27 minutes and a tolerable startup latency of δ minutes. The system may divide this into display of 27 one minute long video sub-clips: X_0, X_1, X_2, ..., X_{26}. Each sub-clip X_i tolerates a startup latency of $\delta+i$ minutes. Given a system with 9 devices, these sub-clips can be dispersed across all devices. In general, the system may represent a clip as a sequence of fix-sized blocks (instead of sub-clips). The number of blocks that constitute the clip is a function of both its size and block size. The system may replicate the first few blocks (say X_0 and X_1) more aggressively because they are required more urgently [16]. In [16], we describe a decentralized data placement technique to compute the number of replicas for blocks of a clip where the blocks are replicated based on their tolerable delay, i.e., sum of δ and when the block is required for display. This technique is termed partial-clip replication.

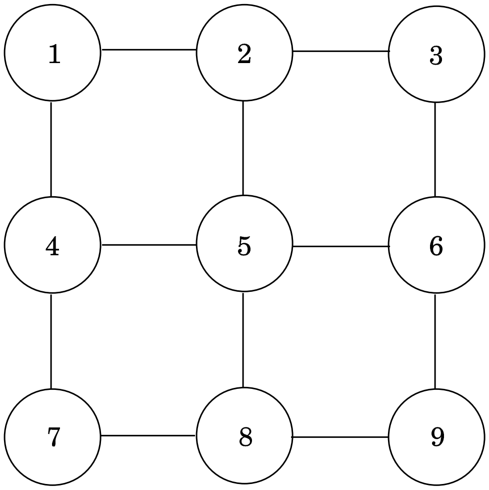

Figure 3. A 2-dimensional grid consisting of 9 nodes.

There is a host of tradeoffs associated with a divide-and-conquer approach when considering the requirements of an application. First, by placing clips at the granularity of blocks, a device may contain no single clip in its entirety. Instead, it may contain blocks of many different clips. If this device becomes isolated and separate from other devices then it supports no display. This renders all clips unavailable. To address this limitation, one may assume the presence of 802.16 [6] as the backbone for an ad-hoc network. As detailed in Section 4.3, when a device becomes network unreachable, it may utilize its 802.16 bandwidth to offer clips with a higher startup latency[1].

Second, assuming connectivity among devices, a divide-and-conquer utilizes network bandwidth more efficiently to support a higher number of simultaneous displays when compared with a full-clip replication strategy. In [13], we demonstrate this for a variety of configurations consisting of a different number of devices and storage capacities. Here, we illustrate this observation for a simple environment consisting of nine nodes. Assume these nodes are organized in

[1] This is because the bandwidth offered by 802.16 is expected to be in the order of 1 Mbps. This is lower than the standard 4 Mbps bandwidth quoted for DVD quality video.

a two-dimensional grid structure with 3 nodes on each axis, see Figure 3. In this figure, an arc shows two devices that are in the same radio range. Assume each device has sufficient storage capacity to hold a 27 minute long video clip and the database consists of nine such clips. Thus, only one replica of each clip can be accommodated in the system. A data placement technique at the granularity of a clip would construct one copy of each clip and assign it to each node. Partial-replication would assign 3 minutes worth of each clip to each node (assuming a homogeneous collection of devices with identical network bandwidth among devices).

Now, compare clip and partial-clip replication techniques. With clip replication, in the best case, a request is issued by the client containing the clip, requiring no (zero) network bandwidth to transfer data. In the worst case, the client and server are diagonally opposing nodes, e.g., nodes 1 and 9 of Figure 3. In this case, the 27 minute clip must be transmitted across four network links to arrive at its destination, consuming 108 minutes of network resources.

With partial-clip replication, in the best case, the client is the center node (node 5). Assuming a uniform distribution of the 27 minute clip across nine devices, this client contains 3 minutes of the clip locally and must retrieve the remaining 24 minutes from the other eight devices; 3 minutes from each device. Four devices will deliver their fraction in one hop. The remaining four devices require two hops. In sum, this consumes 36 minutes of network resources. In the worst case, a node at one corner (say node 1) references the clip. Assuming a uniform distribution of data across devices, two devices require one hop to deliver their data fragments, three devices require two hops, two devices require three hops, and one device requires four hops. This consumes 54 minutes of network resources. This is one half of the worst case scenario with full clip replication. If one compares the average of best and worst case scenario (which may not be representative of the true average case) of clip and partial-clip replication techniques, partial-clip replication requires 45 minutes of network resources when compared with 54 minutes required by clip replication. We refer the reader to [13] for details of data placement and a comparison of partial-clip and full-clip replication techniques.

4.3 Law of large numbers

This principle has several different worded definitions with the same implication. The wording most relevant to our proposed science of design is as follows. Many odd coincidences are likely to happen with large populations. This law explains diverse coincidences such as a woman winning the New Jersey lottery twice, and approximately sixteen million people sharing the same birthday. It also implies design of a system must consider odd coincidences. A system designer may not be able to identify all possible coincidences in advance. Thus, profiling mechanisms (see Figure 2) are needed to gather sufficient information from the environment and users' pattern of access to discover as many coincidences as possible. Coincidences might be either opportunistic or adversarial. We describe each in turn.

An opportunistic coincidence helps with the requirements of an application. One example is a device that references a clip which it is actively routing to another device. Another is a scenario where many devices in a geographical area reference the same clip at approximately the same time. With stream merging (a component of Figure 2), the system employs the space-time principle to use storage of several client devices to transmit only one stream to satisfy many different displays at the same time.

An adversarial coincidence might be an isolated device that encounters other devices rarely and for short periods of time. In the worst case, this device is completely isolated. Another example scenario is a device that is connected to other devices for half a day and isolated for the other half. This device issues requests only when it is isolated. In these scenarios, the system must be robust enough to utilize full-clip replication. Moreover, a system designer might employ either a preventive or a detective approach to refresh the content of each device. With a preventive approach, a device might be equipped with an alternative form of communication mode. One example is a low-bandwidth cellular communication mode. An emerging alternative is 802.16 (WiMax) which offers a higher bandwidth for a single connection with coverage areas similar to a cellular base station [6]. Another alternative is datacasting, a one-way channel for delivering large blocks of data using existing Television broadcasting infrastructure[2]. A limitation of datacasting is its inability to retrieve content recorded on a device for storage on remote servers. This is important for those applications that enable clients to upload their content.

With a detective approach, the system detects the occurrence of an adversarial coincidence and resolves it. In our example scenario, an environment may consist of special devices whose mobility pattern can be controlled. The system may manipulate mobility of these devices in a manner to prevent an autonomous device from becoming isolated (or starve for data when displaying a clip). The controlled mobility component of Figure 2 is designed in support of this functionality.

[2] Products such as MovieBeam receiver are advertised to come with 100 movies already stored inside. This device uses datacasting to replace 10 movies with new ones every week.

5 Mapping between the first two steps

The primary reason for mapping between the first two steps of Figure 1 is to satisfy the requirements of an application. This mapping must consist of effective abstractions describing the following: First, the requirements of an application. Second, the characteristics of the target environment. Third, the parameter settings of an algorithm and their impact on other algorithms and the overall requirements of an application. The space of possible mappings might be extremely large, requiring the use of heuristics.

At times, an environmental parameter might be unspecified and one may utilize the mapping process as a part of an optimization to compute the missing parameter value. In essence, this becomes a configuration planner for the system and is typically invoked in an off-line manner. An example is when deciding the amount of storage for a C2P2 device given an expected respository size.

Once the environmental parameters are fixed, the mapping process is applied in either an off-line or an on-line manner. It is applied in an on-line manner when the environment is dynamic and its characteristics change over time (which is typically the case when considering the bandwidth and loss rate of 802.11 wireless network connection). This enables a device to adjust the parameter values of an algorithm in response to environmental changes. Note that a device may anticipate changes and adjust parameter values in a preventive manner.

To illustrate the dynamics of an environment and its characteristics, in [3], we investigated the feasibility of 802.11a to realize a community of Home-to-Home (H2O) devices. H2O devices collaborate as peers to stream audio and video clips to a household. We conducted many experiments that considered: distance between participating devices, number of intermediate H2O devices used to route a stream from a producing H2O device to a consuming H2O device, and simultaneous number of active streams in the same radio range. Obtained results make a convincing case for the feasibility of H2O devices using 802.11a. At the same time, they show significant variation in bandwidth and loss rate attributed to the physical characteristics of an environment. (This observation is also reported by studies such as [2, 23, 21].) As an example, in some of our experiments where the participating devices were separated by 192 feet, we would observe a bandwidth of 22 Mbps. The same experiment in a slightly different location would yield a bandwidth of 2 Mbps. At times, the nodes would even fail to detect one another. A mapping technique may adjust placement of data dynamically to respond to environmental variations. It may reject system-wide publication of new clips because an application's requirement cannot be satisfied when the content size exceeds a threshold.

Abstractions that describe the behavior of an algorithm's parameter settings might be either analytical models, simulations, or a combination of these two. Typically, simulations approximate reality more accurately. However, they may require substantial amount of time to compute the behavior of the algorithm, rendering them as too slow for on-line decision making. To illustrate, in [1], we developed analytical models to estimate the efficiency of (number of simultaneous displays supported by) a H2O framework as a function of the number of replicas for different clips in the system. The analytical models could not capture network bandwidth that remains idle due to fragmentation of the ad-hoc network. The network becomes fragmented because of link reservations made by the admission control component of the system. To elaborate, the admission control component may exhaust the bandwidth of certain paths, forcing some links to remain idle while other requests wait. A simulation model can captures this reality, providing better accurracy when compared with the analytical model. One must consider the tradeoff between simulation and analytical models carefully when designing a mapper that is invoked at run-time.

6 Conclusions and future research directions

The proposed science of design and its principles are a preliminary effort to specify a structure for applications that deliver continuous media using a wireless network of mobile devices. We envision this effort to develop a tool-kit of plug-and-play components that are employed depending on the requirements of an application. One such a tool-kit is shown in Figure 2. Determining the parameter settings of different techniques in this tool-kit is application dependent and dictated by its requirements.

Our discussions considered two possible deployments for a network consisting of $\mathcal{N}$ devices: pure Ad-Hoc (AdHoc) and Mixed Ad-Hoc (MAdHoc). The former is realized when $\mathcal{N}$ devices form an ad-hoc network and communicate with one another. A MAdHoc deployment complements an AdHoc network with additional infrastructure to provide it with an alternative mode of communication. For example, a cellular base station (or an emerging WiMax, IEEE 802.16, base station [6]) enables a device to communicate with other devices when it is disconnected from the ad-hoc network. Such infrastructure may provide a device with connectivity to remote servers deployed on a wired infrastructure either directly or via multiple hops.

Policies that control availability of data across the network have been implicit to our discussion. Both static and dynamic data might be designated as either No-Loss or Lossy. With the first, a device, say D_p, may contribute data to the network and require the network to not loose this data as long as D_p is a participant. The second variation allows the network to forget a published clip or data source. This frees up both bandwidth and storage for other frequently accessed data sources. The wide spread availability of clips depends on an application's requirements and its parameter settings. An application may publish clips and require them to be available only to its ad-hoc network even though the environment is MAdHoc and remote servers are available. This informs devices to hide this data from the wired infrastructure. This flexibility enables a system designer to implement a host of security and business policies.

We are extending this research in several ways. First, we intend to formalize other principles that are appropriate for our proposed science of design. Second, we intend to demonstrate the feasibility of both the proposed science of design and its principles in different applications, e.g., C2P2 [17] and H2O [11].

7 Acknowledgments

The material presented in this paper have benefited grately from neumerous brain storming sessions with Bhaskar Krishnamachari, Tooraj Helmi, Shyam Kapadia, Samip Bararia, and Ashkan Aazami. This research was supported in part by NSF grant IIS-0307908, National Library of Medicine LM07061-01, and an unrestricted cash gift from Microsoft Research.

References

[1] A. Aazami, S. Ghandeharizadeh, and T. Helmi. An Optimal Continuous Media Replication Strategy for Ad-hoc Networks of Wireless Devices. In *Tenth International Workshop on Multimedia Information Systems (MIS)*, August 2004.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000.

[3] S. Bararia, S. Ghandeharizadeh, and S. Kapadia. Evaluation of 802.11a for Streaming Data in Ad-hoc Networks. In *Fourth Workshop on Applications and Services in Wireless Networks*, August 2004.

[4] V. Bush. As We May Think. *The Atlantic Monthly*, 176(1):101–108, July 1945.

[5] S. Cherry. Across the Great Divide. *IEEE Spectrum*, 41(1):36–43, January 2004.

[6] S. M. Cherry. WiMax and Wi-Fi: Separate and Unequal. *IEEE Spectrum*, 41(3):16–16, March 2004.

[7] C. Chrysafis and A. Ortega. Efficient Context-Based Entropy Coding Lossy Wavelet Image Compression. In *Designs, Codes and Cryptography*, pages 241–250, 1997.

[8] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *MobiCom*, Sept 2004.

[9] J. Gemmell, B. Gordon, R. Lueder, S. Drucker, and C. Wong. MyLifeBits: Fullfilling the Memex Vision. In *ACM Multimedia*, December 2002.

[10] J. Gemmell, H. M. Vin, D. D. Kandlur, P. V. Rangan, and L. A. Rowe. Multimedia Storage Servers: A Tutorial. *IEEE Computer*, 28(5):40–49, 1995.

[11] S. Ghandeharizadeh. H2O Clouds: Issues, Challenges and Solutions. In *Fourth IEEE Pacific-Rim Conference on Multimedia*, December 2003.

[12] S. Ghandeharizadeh, A. Dashti, and C. Shahabi. Pipelining Mechanism to Minimize the Latency Time in Hierarchical Multimedia Storage Managers. *Computer Communications*, 18(3):38–45, March 1995.

[13] S. Ghandeharizadeh and T. Helmi. An Evaluation of Halo Data Placement Strategy for Continuous Media in Multi-hop Wireless Networks. In *Submitted for publication*, 2004.

[14] S. Ghandeharizadeh, S. Kapadia, and B. Krishnamachari. An Evaluation of Alternative Data Replication Strategies for Moving Devices. In *Submitted for publication*, 2004.

[15] S. Ghandeharizadeh, S. Kapadia, and B. Krishnamachari. PAVAN: A Policy Framework for Content Availability in Vehicular Ad-hoc Networks. In *First ACM Workshop on Vehicular Ad Hoc Networks (VANET)*, Oct 2004.

[16] S. Ghandeharizadeh, B. Krishnamachari, and S. Song. Placement of Continuous Media in Wireless Peer-to-Peer Networks. *IEEE Transactions on Multimedia*, April 2004.

[17] S. Ghandeharizadeh and B. Krishnamechari. C2P2: A Peer-to-Peer Network of On-Demand Automobile Information Services. In *First International Workshop on Grid and Peer-to-Peer Computing Impacts on Large Scale Heterogeneous Distributed Database Systems (GLOBE'04)*, August 2004.

[18] S. Ghandeharizadeh and R. Muntz. Design and Implementation of Scalable Continuous Media Servers. *Parallel Computing*, 24(1):91–122, May 1998.

[19] S. Hawking. *A Brief History of Time From the Big Bang to Black Holes*. Bantam, 1988.

[20] K. Illgner and F. Mueller. Hierarchical Coding of Motion Vector Fields. In *IEEE International Confernece on Image Processing*, October 1995.

[21] R. Karrer, A. Sabharwal, and E. Knightly. Enabling Large-scale Wireless Broadband: The Case for TAPs. In *HotNets*, 2003.

[22] R. Levin, E. Cohen, W. Corwin, F. Pollack, and W. Wulf. Policy/Mechanism Separation in Hydra. In *ACM Symposium on Operating System Principles*, pages 132–140, 1975.

[23] D. Maltz, J. Broch, and D. Johnson. Experiences designing and building a multi-hop wireless ad hoc network testbed, 1999.

[24] A. Pentland, R. Fletcher, and A. Hasson. DakNet: Rethinking Connectivity in Developing Nations. *IEEE Computer*, 37(1):78–83, January 2004.

[25] R. Rejaie and A. Ortega. PALS: Peer to Peer Adaptive Layered Streaming. In *Proceedings of NOSSDAV*, June 2003.

[26] S. Sharma. Analysis of 802.11b MAC: A QOS, Fairness, and Performance.

[27] D. Shukla, L. Chandran-Wadia, and S. Iyer. Mitigating the Exposed Node Problem in IEEE 802.11 Ad Hoc Networks. *IEEE International Conference on Computer and Communication Networks (ICCN)*, Oct 2003.

[28] D. Taubman and A. Zakhor. Multirate 3-D Subband Coding of Video. *IEEE Transactions on Image Processing*, 3(5):572–588, September 1994.