

# Utility-Based Flow Control for Sequential Imagery over Wireless Networks

Tomer Kishoni, Sara Callaway, and Mark Byers

**Abstract**— Wireless sensor networks provide a unique set of characteristics that make them suitable for building surveillance, including their small size and unobtrusiveness and their capabilities for rapid deployment, decentralized monitoring and control. Bandwidth utilization constrains the success of a wireless surveillance system, especially when the network serves to transmit video imagery. Creating an effective information management policy will help to obtain an allocation of bandwidth that is “best” from the end user’s perspective. This paper presents a decentralized pricing-based flow control method to allocate bandwidth across multiple competing cameras that are simultaneously attempting to make use of a shared, low bit-rate wireless channel. At present, this flow control model is being integrated into a building surveillance system in which off-the-shelf cameras detect intruders and then start streaming data according to user-specified preferences for quality, where various components of the flow control system monitor bandwidth utilization and adjust the quality settings of image streams to maximize end-user utility.

## I. INTRODUCTION

Wireless sensor networks have provided the ability to monitor, collect, and analyze large volumes of data in a variety of environments and applications. However, wireless networks are subject to limited resources, such as bandwidth. Overloading the network with information can cause network errors. As network usage increases, a need arises to develop an efficient information management policy to allocate bandwidth to competing applications. Information management is the decision making process about the use of network resources. Successful creation and implementation of an information management policy will provide benefits to all wireless sensor network applications by allowing the sensors to intelligently meet the needs of its users.

Flow control, or the adjustment of transmission rates for individual traffic sources in a network, forms a core foundation for an efficient information management policy.

Manuscript received April 9, 2007. This work was supported by the Systems Technology Integration Lab of the Department of Systems and Information Engineering at the University of Virginia, Charlottesville, VA. T. Kishoni is with the Systems and Information Engineering Department, University of Virginia, Charlottesville, VA 22904 USA (email: tk4k@alumni.virginia.edu). S. Callaway is with the Systems and Information Engineering Department, University of Virginia, Charlottesville, VA 22904 USA (email: sec9b@alumni.virginia.edu). M. Byers is with the Systems and Information Engineering Department, University of Virginia, Charlottesville, VA 22904 USA (email: mab6nu@alumni.virginia.edu).

Inefficient flow control can lead to sources greedily using bandwidth, overloading the system and causing network congestion. A congested network delays or loses information sent to the user, which decreases the overall effectiveness of the network. Conversely, an ideal flow control algorithm would allocate bandwidth so as to give higher priority (more bandwidth) to sources that the user deems important. This paper develops a pricing-based flow control method for a wireless surveillance system where various areas of a building vary in significance to the user.

The algorithms presented in this paper adapted to a specific building surveillance scenario. In this system, a user requires image streams to make accurate decisions regarding the action to take against intruders. Additionally, multiple streams limit the cognitive ability of the user, which increases the difficulty the user incurs in making an effective decision. Therefore, the user requires accurate information that has the necessary level of fidelity to make decisions.

## II. LITERATURE REVIEW

### *Price-Based Flow Control (Economic Model)*

The method of flow control used in this paper revolves around a pricing model that is similar to recent work in the networking literature on optimal flow control. For example, in [1], Kelly et al. examine the stability and fairness of two interacting rate control algorithms, decomposing the problem into constituent primal and dual parts solved independently by the network and all users, respectively. The network algorithm serves to set a price for bandwidth utilization based on network congestion, while users adjust transmit rates so as to maximize net utility (utility for bit rate offset by cost). The basis for fairness derives from an earlier paper [2], also due to Kelly, which provides two common examples of fairness: max-min fair and proportionally fair. The max-min fairness criterion requires that smaller flows receive their requested flow rates before larger flows and asserts that increasing the rate of flow for larger sources at the expense of the smaller flows is unfair. To consider flow rates proportionally fair, the rates must be feasible and for any other feasible rates, the aggregate of proportional changes is zero or negative [2].

In [3], Low and Lapsley expand on [1]-[2] by creating synchronous and asynchronous versions of both algorithms. A synchronous algorithm arises when updates to flow sources and network links occur at the same time interval. Their experimental results show that their prototype behaved as expected and that the algorithm tracks the theoretical bandwidth allocation optimum [3].

Kelly and Low provide a solid framework for flow control using a pricing scheme. Their thorough setup of a utility model provides the motivation to following a similar pricing scheme. However, the specific application discussed in this paper concerns that of transmitting streaming images to the user in a decision critical scenario. This directly translates to a different set of objectives as compared to Kelly and Low. Whereas in their work they sought to efficiently allocate transmission rates while minimizing the use of bandwidth, this model requires maximizing bandwidth in order to provide as much information as possible to the decision-making user. This presents an interesting problem that requires further discussion and analysis as done in the remainder of this paper.

### III. GENERAL UTILITY MODEL

In this section we discuss the algorithm used to allocate bandwidth to cameras in the network. This model presents a pricing model similar to that of Kelly and Low in [1]-[3], with two new features: multiple users and dynamic utility.

#### A. Utility Model

This section introduces the optimization problem and its structure. As in [1]-[3], we can consider a network that contains a unidirectional link with a capacity  $c$ . Connected to the network are  $K = (1, \dots, K)$  users. The network includes a set  $S = \{1, \dots, S\}$  of cameras that have the following characteristics:

- A subset of users  $L_s = (k \subseteq K)$  that receive streams from camera  $s$ .
- The utility function  $U_s^k : \mathfrak{R}_+ \rightarrow \mathfrak{R}$ , which we assume to be strictly concave and increasing. Specifically,

$$U_s(x_s, t) = w_s(t) \cdot u(x_s) \quad (1)$$

$$w_s(t) = r_s \cdot d(t) \quad (2)$$

$$r_s = \frac{\sum_{k \in L_s} a_k \cdot i_k}{\sum_{k \in L_s} i_k} \quad (3)$$

where:

- $x_s$ : transmission rate of camera  $s$
- $u(x_s)$ : strictly concave and increasing utility function based on the transmission rate
- $r_s$ : aggregate static priority of a camera stream weighted by users  $L_s$
- $a_k$ : user  $k$ 's static priority
- $i_k$ : user  $k$ 's weighting level ( $\mathfrak{R}_+$ )
- $d(t)$ : dynamic priority strictly decreasing in time

- Minimum and maximum transmission rates,  $m_s \geq 0$  and  $M_s < \infty$ , respectively.
- The camera randomly activates for a random sequence of time.

Additionally,  $R = (s \subseteq S)$  represents the subset of cameras that are active. This leads to the objective of the

problem, choosing transmission rates  $x = (x_s, s \in S)$  so as to

$$\max_{x_s} \sum U_s(x_s, t) \quad (4)$$

$$\text{s.t.} \quad \sum_{s \in R} x_s = c \quad (5)$$

According to [3], a unique maximizer of aggregate utility exists because the objective function is strictly concave and the feasible set of solutions is compact. Solving the optimization problem in a centralized fashion requires coordination among all active cameras and is impractical in real networks, leading to our proposal of a decentralized pricing-based flow control algorithm.

#### B. Pricing Algorithm for Flow Control

In this subsection, we present two cases for flow control prices. The first case introduces a single user scenario that excludes the dynamic priority and builds off the recursive model found in [1]-[3]. The second case provides a general model where quick responses are required due to the inclusion of the dynamic priority.

##### 1) Single User with Static Priority and Logarithmic Utility Function

In this case, a single user provides a static priority level to the cameras. Additionally, all these cameras exhibit a logarithmic curve, which reduces (1) to

$$U_s(x_s) = a \cdot \log(x_s) \quad (6)$$

This mimics the utility function by Kelly in [2] and a similar recursive method (7) solves the optimization problem.

$$p(t+1) = p(t) + \gamma \left( \sum_{s \in R} x_s - c \right) \quad (7)$$

The cameras allocate bandwidth based on a greedy approach by maximizing their benefit (or utility minus costs). (We will discuss the allocation scheme more in depth in the general case.)

$$x_s(t) = \frac{a_s}{p(t)} \quad (8)$$

In fact, since the optimization problem requires full network utilization at  $c$ , this specialized case simplifies to

$$x_s = \frac{a_s}{\sum_{s \in R} a_s} \quad (9)$$

Additionally, the allocation in (9) provides proportional fairness and derives all the attributes described by Kelly in [2].

## 2) General Model of Pricing

The ideas from [1]-[3] and the specialized case do not depend on a dynamically changing variable. In the case of a dynamically changing utility function, the system requires quick responses, something not guaranteed in a recursive model. The general case we present here provides a method to allocate bandwidth in a dynamic environment.

In order for the network operator to supply bandwidth properly, the link must monitor the aggregate utilization of bandwidth and calculate a price per unit of bandwidth consumed. We attribute to the link a disutility associated with bandwidth consumption, where the form of the disutility function reflects the effect of congestion at full network utilization. Similar to a traffic jam where an additional car brings flow to a halt, an additional bit at full network utilization may cause loss of information. In this paper, we assume that, an exponential price curve accurately reflects the network operator's increasing disutility associated congestion. Specifically, the link determines the price using a lookup method (10), as opposed to the numerical method in (7).

$$P(b) = e^b \quad (10)$$

where:

$$b = \sum_{s \in R} x_s$$

Therefore, the two steps necessary for the link to supply bandwidth involve (i) monitoring the network utilization and (ii) calculating the price based on the disutility at that utilization.

Acting as a consumer, each active camera seeks to consume the network's bandwidth, and it does so on behalf of the end user associated with the camera. Using its default behavior, each camera will attempt to maximize its private net benefit (12) according to the utility function set by its associated user. The point of maximization indicates the optimal bit rate for the camera. Each camera uses a similar logarithmic utility function differentiated by priority settings as seen in (1). The priority setting allows the user to give bandwidth preference to certain cameras.

$$C_s(x_s) = p \cdot x_s \quad (11)$$

$$B_s(x_s) = U_s(x_s, t_s) - C_s(x_s) \quad (12)$$

$$x_s = \arg \max(B_s(x_s)) \quad (13)$$

where:

C: cost      B: benefit      p: price

Changes in bandwidth consumption propagate back to the link (network operator) which then calculates a new price reflective of the new level of utilization. This supplier-consumer relationship iterates and produces a damped oscillation around an equilibrium price. Since the packets sent by the cameras contain time sensitive information, a method to damp further the oscillation may be needed, as in the following price averaging scheme:

$$p(t) = \frac{p(t-1) + p_c(t)}{2} \quad (14)$$

where:

$p_c$ : current link calculated price

A drawback of the price-setting formula of (10) is that the prices and end user utilities are miscalibrated, resulting in equilibrium bandwidth allocations that fail to consume all available bandwidth. To fully maximize aggregate utility (as in (4)), an additional step is required to ensure that network resources are fully utilized. The mechanism we adopt in this paper is to send a proportional multiplier (as computed by the network operator according to (15) below) to all cameras when the bandwidth prices have reached relative equilibrium (prices vary by less than 5%).

$$m = \frac{c}{b} \quad (15)$$

When the equilibrium bandwidth allocation would otherwise settle on a network operating at less than full capacity, the normalization factor serves to increase the bandwidth allocation for each camera, while maintaining the proportional bandwidth allocation between the cameras. This model describes the general method to reach equilibrium in allocation. However, it remains to be seen whether the allocations solve the optimization problem in (5). We plan to further investigate this model to test for optimality, which will provide the focus of future papers.

### C. Simulation Proof of Concept

In this subsection, we present simulation results that provide a proof of concept for our distributed pricing algorithm above. Rather than presenting a complete numerical evaluation of our approach, we illustrate the effect of the pricing scheme in a simple scenario. (We plan to document a more complete numerical evaluation of the scheme in a future publication.) The simulation results below were computed in Matlab using a synchronous implementation of the price/bandwidth updates of (10), (13), (14), and (15).

To set the stage for the numerical results, consider single wireless link with 11 Mbps total capacity, of which 8 Mbps are to be allocated (so that the "target bandwidth" is 8 Mbps). We assume that there are five cameras, which are either active or asleep in each of 10 "rounds", where each round is 20 units of time in length. Each price/bandwidth update consumes one unit of time, so that there are up to 20 price updates per round. We assume that all cameras have the same utility function, with different priority settings ( $a = 1, 2, \text{ or } 3$ ) and a dynamic priority decreasing each subsequent iteration. Figs 1 and 2 show the effect of the simulation with and without the price averaging scheme.

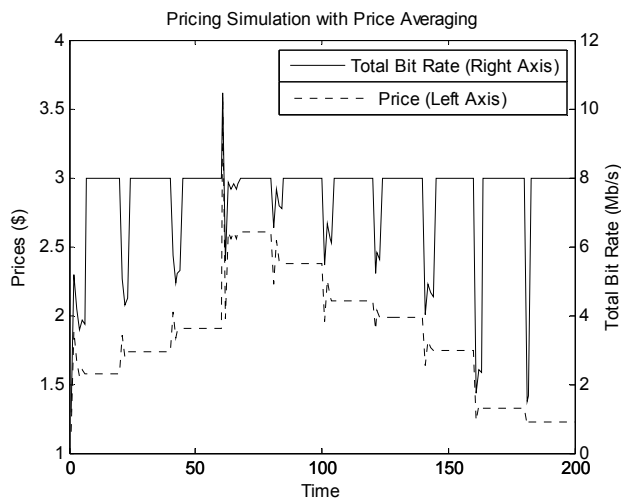


Fig. 1: Pricing Simulation with Price Averaging.  
Active cameras indexed by round = (1, 2, 3, 5, 4, 3, 2, 1, 0, 1)

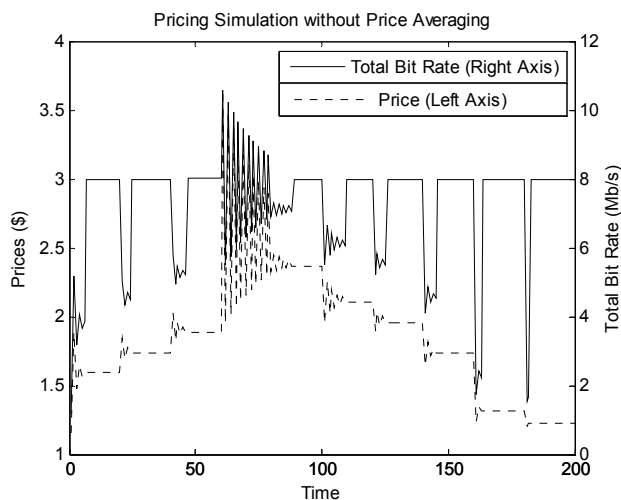


Fig. 2: Pricing Simulation without Price Averaging.  
Active cameras indexed by round = (1, 2, 3, 5, 4, 3, 2, 1, 0, 1)

As seen in Fig. 1 and 2, the effect of the price averaging is significant. This simulated results show that user will have a relatively constant stream of fidelity using this scheme.

#### IV. IMPLEMENTATION OF THE FLOW CONTROL ALGORITHM

The research of this paper was conducted in the context of a larger project on the development of system that uses wireless cameras to provide facility video surveillance. An image-processing application from [4] provided the method to stream a sequence of encoded images from cameras. The encoding scheme uses a change detection algorithm to stream only the changes relative to a background image, which then reconstructs itself on the interface using the same background image. Other components of the larger project are the development of a web-based interface for setting up the surveillance network and viewing motion detections [5],

and a simulated environment to simulate the affect of monitoring multiple buildings [6]. The information management policy, implemented on the nodes of the network (server and camera), is used to control information flow by adjusting the quality controls of the images streams. This section serves to describe both the hardware and software aspects of our pricing-based flow control algorithm within the overall system.

#### A. Implementation Architecture and Strategy

##### 1) Hardware

A laptop with the Linux laptop serves as a central server. This server provides all instances of the web-based interface access to available image streams so that they can be seen by end users. (While the web interface can be made available to any number of users, it actually encodes only one user's set of preferences for video from individual cameras.) The same laptop also serves to implement the "link" functionality of our flow control algorithm. Specifically, the laptop monitors the bandwidth utilization in the network and computes the prevailing price-per-unit-bandwidth according to (10).

Cameras (Logitech Quickcam Pro 4000 [7]) are deployed throughout the building for surveillance and are attached to Stargate computers that run the change detection and image streaming application of [4]. (Stargates are small wireless devices which communicate using 802.11 as their wireless protocol [8].) The Stargates also implement the bandwidth update functionality of our flow control algorithm, receiving prices periodically from the laptop and adjusting image streaming parameters.

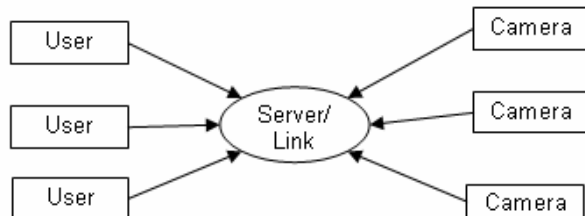


Fig. 3. System Architecture

##### 2) Software

As shown in Fig. 3, all communication will go through the central server, which resides on the Linux laptop discussed previously. When cameras detect motion, they send the resulting images to the server creating input traffic on the server. When users connect to the server to view image streams, output traffic is created.

The laptop also supports the "link" functionality of our pricing-based flow control algorithm, and central to this measurement capability for assessing bandwidth utilization. At present we are using a Linux utility known as "iptables" [9] to monitor the bandwidth utilization in the network. Iptables is generally used to create firewalls on a machine [9]. In this case, it is used to create a firewall that does not actually block any network traffic. Instead, rules about packets being sent through the network card are used to monitor the network usage [9]. These rules count the traffic in three categories: input, output, and forward. This traffic

is monitored for a period of time. This period is an input that fits the needs of the network being monitored. A more volatile network will require a longer interval to smooth out the observations. The total monitored bandwidth is divided by the number of seconds in the time interval to determine the average bits/s during the period. After each measurement, the counters are reset to zero and the period restarted. The link functionality implemented on the laptop uses this measured bandwidth utilization to compute prices (per unit of bandwidth), which are then sent to the flow control algorithms running on the Stargate computers (to which the cameras are attached).

The bandwidth updating functionality of the flow control algorithm resides on the Stargates, where link prices and utility functions are reconciled and new video encoding parameters are computed. When encoding parameters change, they are written to a configuration file for the encoding software, which also resides on the Stargates, and the system send a message to the encoder telling it to read the new configuration file.

### B. Component Details

Our pricing-based flow control algorithm required separate routines for the link and camera, implemented in Java using standard libraries. Java’s portability and object-oriented structure allow for straightforward algorithm implementation.

#### 1) Link Routine

We refer to the price-setting mechanism that resides on the server as the “link routine” or simply “the link”. Implementing the link required developing a multithreaded program. Fig. 4 presents the functional relationship between the threads and the various Java methods we developed.

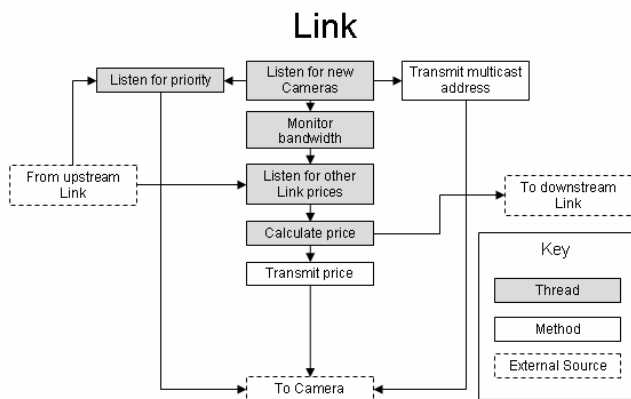


Fig. 4: Link Routine

The functions marked in the grey box represent concurrent threads. To transmit the prices from the link the cameras, a multicast UDP packet is sent. Additionally, the link listens for the user set priority and relays the level to all cameras in its network. Although not discussed in this paper, the link routine is actually implemented in a way that allows it to be used in the case of a multilink environment.

#### 2) Camera Routine

We refer to the bandwidth updating mechanism of our flow control algorithm, which resides on the Stargates, as the “camera routine” or simply “the camera”. Similar to the link routine, the camera routine implements a multithreaded application as seen in Fig. 5.

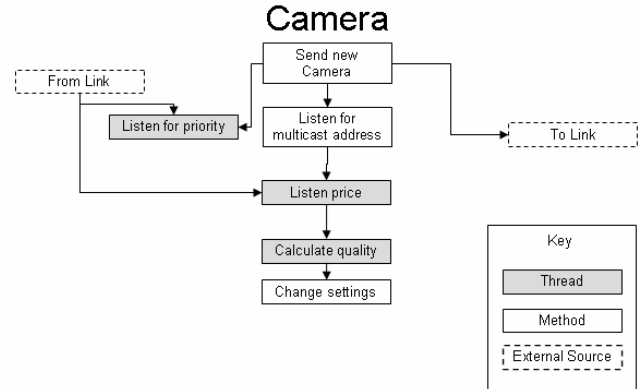


Fig. 5: Camera Routine

#### a) Video Encoding Routine

The output of the camera routing is a new set of encode parameters for the encoder developed in [4], with the goal being to set encoding parameters to achieve the optimal bandwidth allocation computed by our flow control algorithm. The relevant encoding parameters include (i) frames per second and (ii) change detection threshold. The ‘frames per second’ parameter greatly influences a camera’s bandwidth and serves as a coarse bandwidth adjustment mechanism. The ‘change detection threshold’, which describes the sensitivity the program to changes in the image, allows for more precise control of the bandwidth utilization. The combination of these two parameters offers us a method to match the calculated transmission rates.

#### b) Camera/Link Initialization

To initialize the system, each camera must notify the link that it has entered its network. This is the very first step done by every camera. Upon registration of each camera, the link assigns and transmits a multicast address to the camera. After completing these initial steps, each instance of the camera routine proceeds as required by the flow control algorithm: listening for prices and adjusting its image-processing settings.

### C. Status

At the time of this writing, the separate components of the larger facility video surveillance project are being combined into a single working system. A demonstration of the final system is scheduled to occur on April 16, 2007 at the Nuclear Reactor Building at the University of Virginia.

## V. CONCLUSIONS

The model presented in this paper provides a method to efficiently allocate bandwidth to cameras with varying levels

of importance to the user. The pricing model uses a multi-round process to allocate bandwidth to the cameras. In each round a link monitors current bandwidth utilization and calculates a new price-per-unit-bandwidth. The cameras receive this price and select the bandwidth that maximizes its own benefit (net utility). Achieving this new bandwidth allocation requires adjustment to two video encoding parameters, ‘frames-per-second’ and ‘change detection threshold’, as specified by the video encoding format of [4]. As encoding parameters are adjusted, the link routine continues to monitor the bandwidth utilization by means of the Linux utility, iptables. This allows the link to calculate a new price for the next round, and the process restarts. The algorithm produces, in a decentralized fashion, an equilibrium bandwidth allocation for the camera streams maximizing aggregate utility. Investigating the optimality of this equilibrium provides the focus of future research.

An effective information management policy amplifies the utility gained by system users. The techniques of this paper also apply to other applications, such as home network bandwidth management. The importance of creating an effective utility-maximizing flow control algorithms increases with the addition of more and more Internet-connected consumer products. For example, users may download movie trailers to their television, play online video games, or watch streaming media. An effective information management policy will efficiently allocate bandwidth to all the users according to priorities (utilities) set by the network operator.

#### REFERENCES

- [1] F. Kelly, A. Maulloo, & D. Tan, Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(5), 237-252, 1998.
- [2] F. Kelly, Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1), 33-37 1997.
- [3] S. Low, & D. Lapsely, Optimization flow control. I. Basic algorithm and convergence. *Networking, IEEE/ACM Transactions on*, 7(12), 861-874, 1999.
- [4] M. Farrell, Personal Communication, 2007.
- [5] F. Thompson, A. Vaswani, B. Walthall, M. DeVore, Computer vision information systems for wireless surveillance, 2007.
- [6] D. Chao, K. Datt, J. Rekhi, G. Learmonth, Human-in-the-loop simulation testbed for wireless sensor networking for infrastructure surveillance, 2007.
- [7] Logitech Homepage. [www.logitech.com](http://www.logitech.com)
- [8] Crossbow: Wireless Sensor Technology: Homepage. [www.xbow.com](http://www.xbow.com)
- [9] G. Beekmans, “Bandwidth monitoring with iptables.” Linux.com. <http://www.linux.com/article.pl?sid=05/12/15/177232>