Submitted to the *Workshop on High Speed Local Networks* at *IEEE LCN* 2002 (correspondence author is Christensen)

A First Look at Wired Sensor Networks for Video Surveillance Systems

Vijay Chandramohan and Ken Christensen
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620
{vchandr2, christen}@csee.usf.edu

Abstract

Sensor networks are a major new area of research. Some sensor applications, such as video surveillance, will need to be tethered for reasons of bandwidth and power requirements. To support ad hoc, economical installation of video cameras there is a need for new shared-medium protocols. IEEE 1394b FireWire is investigated as a near-future candidate for a shared-medium wired sensor network (WSN). Simulation results show that FireWire can transport packetized video with low delay. In the future, WSN nodes will be capable of store-and-forward and of acting as routers and caches for arbitrary topologies. New routing protocols for attribute and location routing will be needed. We investigate a new hybrid routing scheme that uses distributed location servers to minimize broadcasting. Source routing is used as the packet forwarding mechanism. The distributed location servers contain knowledge of sensor locations and source routes between sensors and other nodes.

1. Introduction

The cost of computing and communications continues to drop. This makes possible new applications based on autonomous and low-cost sensors that interact with their environment to achieve a given sensing goal. Sensor networks, also called Embedded Networks (EmNets) [4], are envisioned to tie together embedded systems. EmNets can enable distributed processing and entirely new computational models. In many applications, sensor networks are wireless [1], [2], [5], [21]. For applications such as habitat monitoring in a jungle [1] or tracking of items in a warehouse [5], a wired network is not possible. However, for sensor applications with fixed locations and high bandwidth and power demands a wired sensor network (WSN) is needed. One such application of national importance is video surveillance. Open problems in video-based networks are described in [6].

Most existing video surveillance systems are built using a single coaxial cable per analog camera wired to a central location. Emerging video surveillance systems are based on digital cameras and use a single 100BaseT Ethernet unshielded twisted-pair cable per camera [17]. As the cost of cameras continues to drop, an underlying dedicated-medium Ethernet network will soon become the cost and performance bottleneck to further deployment of large-scale (e.g., thousands of cameras in one installation) video surveillance systems. Better technology also means that video cameras can contain built-in processing for target recognition. Existing, low-cost cameras are already capable of motion-detection [3]. Image processing localized in, or even distributed between, cameras is needed to reduce the video load to the human "eyeball" monitoring the cameras. To reduce the cost of the underlying network, shared-medium daisy-chained physical and MAC layer protocols need to be investigated. In this paper, we study IEEE 1394b FireWire [9] as a possible technology suitable for implementing large-scale WSNs in the near future. Image processing for reducing video information overload to humans is beyond the scope of this paper.

Sensor networks are different from existing networks in their focus on location and attribute of sensor nodes. In existing "data networks" we are primarily interested in finding information (e.g., a given document) and less so in where the information comes from. In a sensor network, information is meaningful only when coupled with the location of the sensor. In video surveillance there is a need to be able to query all cameras in a geographic area (e.g., "all cameras in the international terminal" or "all cameras within a 100 foot radius of the camera in the tool room") or by attribute (e.g., "all cameras with a person wearing a red shirt in view"). Future WSNs may evolve to store-and-forward nodes allowing for arbitrary network topologies. In such networks, routing is needed. Existing routing protocols do not consider geographic location or attributes. In this paper, we investigate a hybrid source routing and distance vector routing protocol that uses distributed route servers. The distributed route servers maintain route-attribute-location knowledge for routing in WSNs.

This work is funded by a Spring 2001 Florida Space Research and Education Grant.

The remainder of this paper is organized as follows. Section 2 reviews existing work in sensor networks and geographic routing and overviews source routing and IEEE 1394b FireWire. In Section 3 we describe applications and the evolution of video surveillance. Section 4 evaluates the performance of FireWire for packet-based video transmission. Section 5 describes a new *hybrid routing* scheme for future WSNs with store-and-forward nodes and arbitrary topologies. Section 6 presents a summary and discusses future work.

2. Background

In this section we review the current state of sensor network research, routing protocols for wireless networks, source routing, and FireWire.

2.1 The current state of sensor networks

Sensor networks encompass new applications, technologies, and protocols. Applications for wireless sensors nodes are in remote monitoring (even for planetary exploration of Mars [2]) and mobile uses. In a wireless sensor network, nodes can act as sensors, store-and-forward routers, and/or caches. Sensor networks are different from existing IP/Ethernet networks. Sensor networks are:

1) Autonomous in their operation and must automatically install and configure themselves, and detect/isolate failures.
2) Very large with hundreds to thousands of nodes in a single autonomous network.
3) Constrained by physical node size, cost, power, processor cycles, memory, and communications bandwidth.

Sensor networks are environment centric with nodes embedded within the environment in a possibly ad hoc fashion. Sensor data is only meaningful if coupled with geographic location. For fixed nodes, geographic location can be determined by geographic position systems (GPS) at time of installation. For mobile nodes, location can be determined by an integrated GPS or by presence in a particular wireless cell (where the wireless cell is at a known fixed location).

Sensor network research is dominated by a view of sensors as wireless nodes. There are many open problems in the wireless domain. Wireless nodes are not tethered to a power source and use internal batteries for power. In some applications, batteries can be recharged via solar cells. Energy conservation is a major issue for wireless sensors. Entirely new computing and communications paradigms are being investigated to reduce power consumption of sensor nodes [8]. Geographic routing is an open problem in sensor networks.

2.2 Routing in wireless networks

Routing in ad hoc wireless networks focuses on reactive routing schemes that consume less power than proactive routing protocols. Proactive routing protocols, such as distance vector and link state, require continuous transmission (e.g., of status packets) and maintain large routing tables in all routers. In reactive routing, a route is first established when needed. Dynamic Source Routing (DSR) [12] uses broadcast of discovery packet to a target node. Nodes snoop on source routes from observed packets and cache routes. The cached routes are used to reduce the scope of broadcast discovery packets (i.e., by having a nearby node reply with a route on behalf of a faraway node). DSR also limits the scope of broadcasts by having nodes not forward recently seen discovery packets (i.e., to prevent redundant copies of a discovery packets from traversing a link). DSR requires that all nodes be in promiscuous mode for purposes of snooping. This may place an undo burden on the node processor to filter-out uninteresting packets. Ad-hoc On-Demand Distance Vector (AODV) routing [20] uses soft state in nodes to build routes between nodes. Similar to DSR, a broadcast discovery packet is used.

Greedy Perimeter Stateless Routing (GPSR) [13] uses only knowledge of nearby nodes and simply forwards packets in the direction of a destination node. In this case, node coordinates must be known. GPSR has a means of routing around voids (areas in which a "backwards" hop must be made). Knowledge of geographic locations of nodes is needed. A Geographic Location Service (GLS) is proposed and evaluated in [15]. Individual nodes act as distributed location servers that can be queried to find a destination node. Prediction of future location is possible if the direction and velocity of a mobile node is known. Location-Aided Routing (LAR) [14] employs source routing, where the knowledge of the location and velocity of mobile nodes is used to minimize discovery packet flooding. Another approach to finding geographically specific data is directed diffusion [11]. In directed diffusion, attribute data moves and is cached in the general direction of requests reducing the distance that responses must travel.

In a WSN power is not as much of a constraint as it is in a wireless sensor network (where nodes are untethered and battery powered). However, processor cycles and bandwidth are constraints. In addition, in wired networks physical closeness does not (unlike in wireless networks) guarantee that communications is possible. The effects of these differences are investigated later in this paper.

2.3 Overview of source routing

For lightweight and reactive routing, source routing is a promising method. Source routing is a well-understood

mechanism for determining paths in a network. Source routing is used by IEEE 802.5 token ring for bridging. In source routing, a packet contains a routing indication field (RIF) with a hop-by-hop route in it. In token ring, each hop in the RIF is a ring and port id. The port id is the port to which this packet should be forwarded. The ring id identifies a unique ring number and prevents a source route with a loop from being defined. Source routing entails a route discovery process where a source node broadcasts a "discovery packet" to all rings when seeking a route to a target destination node. If the destination node receives the discovery packet, it returns a reply or "found" packet via broadcast to the source node. As the found packet is returned, each node adds its hop id. The original sending node receives the found packet(s) and can select and cache a source route for use by connections to the destination node.

Source routing simplifies the forwarding process in bridges. Only a very limited look-up is needed and no processor involvement is needed (e.g., to do learning or other table management). Source routing trades-off bridge for end-node complexity. A major issue with source routing is the explosion of discovery and/or found packets when broadcasting on parallel paths. Various means for reducing this broadcast explosion have been investigated:

- IEEE 802.5 uses a spanning tree algorithm to designate bridges for forwarding of source route discovery packets.
- DSR [12] uses a unique request id to discard redundant, recently seen discovery packets.

Later in this paper, we investigate using source routing in WSNs.

2.4 Overview of IEEE 1394b FireWire

IEEE 1394 FireWire is an extended serial bus technology first developed by Apple Corporation in 1987. FireWire supports both isochronous slotted and asynchronous packet transmissions. A FireWire is a shared-medium bus or tree with each node inserted in the repeat path. A FireWire cable contains two pairs of wires for full-duplex communication and a wire pair for power distribution. There exist three versions of the standard; IEEE 1394-1995 [10], IEEE 1394a, and IEEE 1394b [9]. The most recent is IEEE 1394b for which no products yet exist in mid-2002, but silicon is under development by major vendors. The original (IEEE 1394-1995) FireWire uses gaps to identify arbitration and fairness periods. The new IEEE 1394b employs overlapped arbitration request signaling with transmission. IEEE 1394b is capable of 100-meter reach between nodes, 63 nodes, and up to 1.6 Gbps data rate on fiber. Performance has been studied analytically in [18]. In [7], IP over FireWire was compared to IP over Gigabit Ethernet, and it was found that throughput was very similar.

In IEEE 1394b FireWire, arbitration decisions are made by a bus owner supervisor selector (BOSS) node. The BOSS is the last node to have transmitted a packet. The result of FireWire arbitration is round robin servicing of those nodes with packets queued. FireWire is the only existing high-speed technology that supports a shared-medium and has built-in power distribution. We believe that a shared-medium is necessary to support ad hoc installation of nodes and reduce cabling costs compared to dedicated medium technologies (such as switched Ethernet or ATM). Figure 1 shows an *N* node FireWire where each node is a video camera.

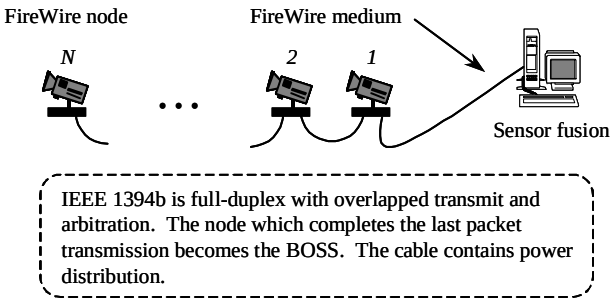

Figure 1. A simple FireWire video network

3. Networks for Video Surveillance

New applications continue to be found for video surveillance. These new applications will force an evolution in the underlying networks that support video surveillance systems.

3.1 New applications for video surveillance

Video surveillance is already used in public spaces for monitoring human behavior. This includes detection of theft and other criminal behaviors. We envision systems with thousands of cameras in airports and other transportation terminals. Cameras can be used to monitor crowd behavior and access to restricted areas (including aircraft and baggage storage areas). With facial recognition, monitoring extends to recognition of profiled individuals. Other novel applications include:

- Installation of cameras along airport runways to monitor for foreign objects. Such an installation could possibly have prevented the 2000 Concorde crash, which was caused by metal debris on the runway.
- Installation of cameras over all the seats in an airplane to monitor for suspicious activity by passengers. This application is already envisioned by Airbus [22].

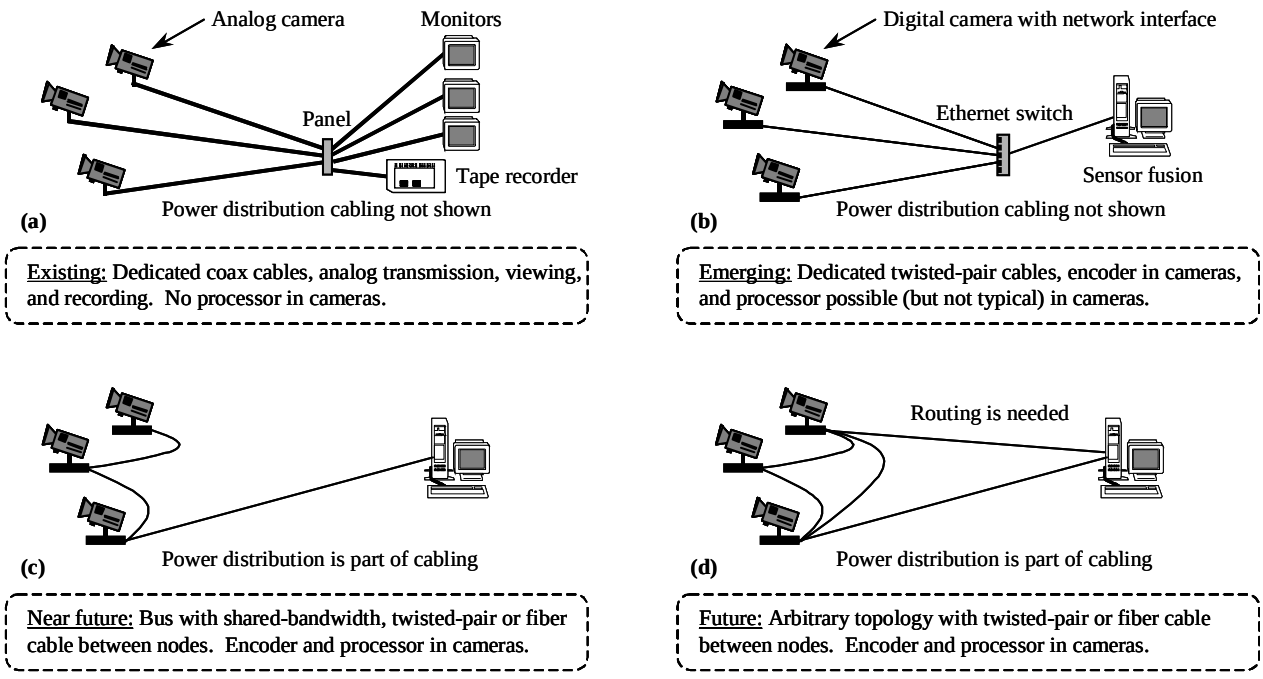

Figure 2. The four generations of video surveillance systems

With image processing included as part of an intelligent camera, many new applications can be envisioned. For example, monitoring for human presence (e.g., by detecting flesh tones) in dangerous areas within a factory becomes feasible. Common to all of these applications is the need to communicate video, still images, or event notifications between peer nodes and/or to one or more monitoring or sensor fusion points. At the monitoring points, some human or automated action is generated in response to a detected event.

3.2 Evolution of networks for video surveillance

We identify four generations of networks for video surveillance systems:

- Existing – based on analog cameras and dedicated coax cabling.
- Emerging – based on digital cameras and Ethernet or ATM with dedicated, unshielded twisted-pair cabling.
- Near future – based on intelligent digital cameras with processing capability and shared cabling in acyclic topologies.
- Future – based on intelligent digital cameras and arbitrary topologies with routing between nodes.

The near future and future generations are our predictions. Figure 2 shows the four generations as (a), (b), (c), and (d).

For the existing generation, the coax cables for all cameras are brought into a common control room in which the analog video signals are monitored by a human and/or recorded. In the control room human operators observe a large wall of monitors, each with rotating views from multiple cameras. The operators may have the ability to lock a monitor to a specific camera and physically control the orientation and/or zoom-in of a camera. By controlling the orientation and zoom-in of a camera, a suspicious target can be manually followed. This existing generation is limited by the cost of the coaxial cabling (which can exceed camera costs) and the number of video streams that can be monitored by a human. Image processing of the video streams would be difficult given that only centralized processing is possible.

The emerging generation uses low-cost digital cameras and incorporates an Ethernet [17] (or an ATM network [19]) connection in the camera unit. Video is transmitted from the camera to a central point as MPEG-2 using IP. By using Ethernet, lower cost unshielded twisted-pair cabling (e.g., UTP-5 for 100BaseT) can be used. With the video stream already in digital and packet format, transmission over an existing IP network and/or recording on a PC hard disk are easily accomplished. With a continued decrease in the cost of cameras, but no similar decrease in the cost of copper or labor to install cabling, a bottleneck will soon be hit.

We envision that the future generations of video surveillance systems will require shared-medium and direct-wired networks to reduce cabling costs and allow for ad hoc installation of cameras. In the near future, we envision acyclic networks with intermediate clustering points. Tree branches can be extended with a new camera, or a camera inserted into a branch. In the longer-term future, we envision each node in a WSN to be store-and-forward and include routing and caching capabilities. With such nodes, arbitrary network topologies become possible with redundant links for reliability and added bandwidth. Very significantly, these future WSNs can enable new models of distributed image processing.

For high-resolution video and localized processing in each node, power cannot be delivered for very long by a battery. Thus, wiring is needed for power distribution. Existing video surveillance systems have two cabling systems, one for power and another for communications. Power distribution can be combined with communications. IEEE 802.3af standardizes power distribution on an Ethernet link to allow for nodes without separate power wiring. FireWire includes power distribution as part of the standard cable bundle. WSNs will combine power distribution with communications. We expect that the power requirements for sensor nodes will motivate the need for WSNs.

4. Performance of Packet Video on FireWire

The throughput performance of IEEE 1394b FireWire is largely dominated by the repeat and propagation delays as node count and network span increases. This per hop delay is an overhead for each packet transmitted. Define L as packet length in bits, R as link rate, and T as the overhead delay per packet. Then the service time for a packet transmission is $x = L/R + T$. The service rate, μ, is the reciprocal of x. The link utilization is $\rho = \lambda x$ for an arrival rate of λ packets per second. Offered packet load is equal to $\lambda L/R$ and overhead load is equal to λT. Thus, the relative overhead scales-up with the arrival rate if T is a fixed value (i.e., not relative to R). The overhead delay is a function of the mean distance between nodes, D, the number of nodes, N, the repeat path delay T_{repeat} per node, and the propagation delay T_{prop} per meter of medium. For a FireWire of N nodes, all with packets queued for transmission, the service time for a node i ($i = 0,1,\ldots,N-1$) is,

$$x_i = \frac{L}{R} + \left[i\left(DT_{prop} + T_{repeat}\right)\right], \quad (1)$$

and the mean service time for all nodes is,

$$\bar{x} = \frac{L}{R} + \left[\frac{(N-1)}{2}\left(DT_{prop} + T_{repeat}\right)\right]. \quad (2)$$

Delay performance is the queuing delay at a node and is a function of the packet arrival rate and the distribution of packet interarrival and services time.

4.2 Simulation model and experiments

Using simulation we evaluate the queuing delay of a IEEE 1394b FireWire for MPEG-2 video streams. A discrete-event queueing simulation model of IEEE 1394b FireWire was built using the CSIM18 function library and is freely available from the authors of this paper. The model includes T_{prop} and T_{repeat} delays, standard FireWire packet start and end overhead (100 and 260 nanoseconds each, respectively) and response time (244 nanoseconds) for the BOSS. Packetized video transmission is done using IEEE 1394b asynchronous streaming packets.

4.2.1 Traffic models for simulation experiments

We used two traffic models to evaluate performance. The first traffic model was based on MPEG-2 frame length traces from the 1996 Olympic games [16]. Each trace was for 40 minutes of a sporting events and a total of 20 traces were available. The MPEG-2 frame traces were converted into packet sizes with 52 bytes of overhead (representing LAN, IP, and TCP headers) per packet. Fragmentation of MPEG-2 frames into Ethernet packets was assumed to occur in zero time. Figure 3 shows a rate snapshot of an athletics event. The video rate is 25 frames per second with a mean data rate of about 5 Mbps The peaks occurring periodically every 0.6 seconds represent MPEG-2 I frames. For the simulation evaluation, frame traces from 20 different Olympic events were used (one trace per simulated camera node). For the 20 MPEG-2 sources, the mean packet length was 1459.67 bytes and the total offered packet load was 101.48 Mbps. The frames from multiple sources were not synchronized.

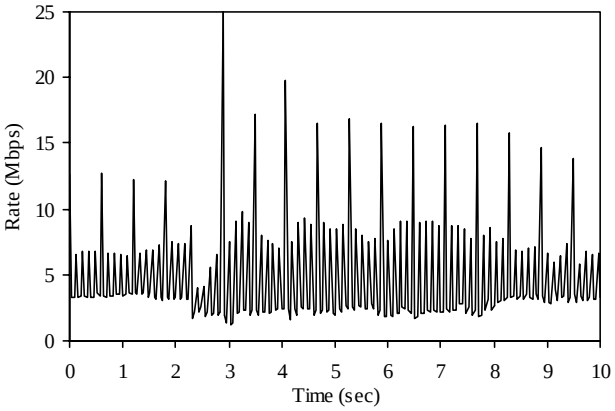

Figure 3. Rate plot of an MPEG-2 video clip

The second traffic model was Poisson arrivals of fixed length packets. The packet length used was the mean packet length of the MPEG-2 video sources. The second traffic model was synthetically generated with no limit on the number of nodes.

4.2.2 The simulated configuration

A bus topology of *N* nodes, each node an independent traffic source, was modeled. Each node was assumed to have an infinite buffer for packets being sent on the link. All packets were destined to the head end, which acts as the sensor fusion node. The response variable was queuing delay (mean and 99%). Control variables were:

- Offered packet load on the link
- Distance between nodes
- Number of nodes
- Traffic model (MPEG-2 or Poisson)

T_{repeat} is 144 nanoseconds per node (worst case for FireWire [9]). T_{prop} is assumed to be 5 nanoseconds per meter.

4.2.3 Simulation experiments

We defined three experiments to evaluate the performance of IEEE 1394b FireWire with MPEG-2 and Poisson traffic models. The experiments were:

Load experiment: The offered packet load was increased from 10% to 95%. The number of nodes was fixed at 20 and the distance between nodes was 10 meters.

Node distance experiment: The distance between each node was increased from 10 meters to 100 meters. The number of nodes was fixed at 20 and the offered packet load was fixed at 70% and 90%.

Node count experiment: The number of nodes was increased from 10 to 100. The offered packet load was fixed at 70% and 90% and the distance between nodes was fixed at 10 meters. For this experiment, only the Poisson traffic model was used.

To achieve a target offered packet load, we varied the link rate (*R*) as 101.48 Mbps divided by the target offered load for the first two experiments. For the node count experiment we varied the arrival rate (λ) and maintained a constant link rate of 400 Mbps. The per packet overhead remained the same for all offered packet loads.

4.3 Results from the simulation experiments

Figures 4 and 5 show the load experiment results for MPEG-2 and Poisson traffic sources, respectively. IEEE 1394b queuing delay increases with load, but remains below a mean of 10 milliseconds and 99% of 50 milliseconds even for 90% offered load. This 50-milliseconds 99% delay is well within the tolerance of human response time. The Poisson source results in delays about one magnitude less than the burstier video source. Figures 6 and 7 show the node distance experiment results. It can be seen that with IEEE 1394b, queuing delay is not sensitive to distance. This is not the case for IEEE 1394-1995 or IEEE 1394a where scaling-up to 20 nodes and 100 meter between nodes was not possible at offered load above 70% (results not shown here). Again, Poisson source queueing delay is significantly less than queueing delay for an MPEG-2 source. For node count scaling, mean and 99% delay increase by slightly less than one magnitude for 90% offered load as the number of nodes increases by one magnitude (from 10 nodes to 100 nodes). For lower offered load, the increase in delay is much less.

These results show that IEEE 1394b performs and scales very well for transporting MPEG-2 video traffic. This demonstrates the potential for IEEE 1394b to be a near-future WSN for video surveillance applications.

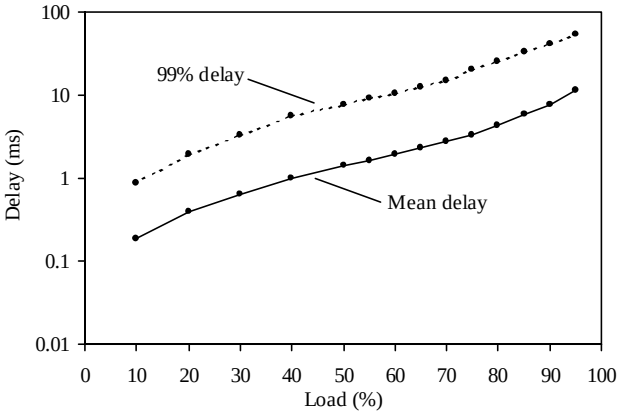

Figure 4. Load results for MPEG-2 source

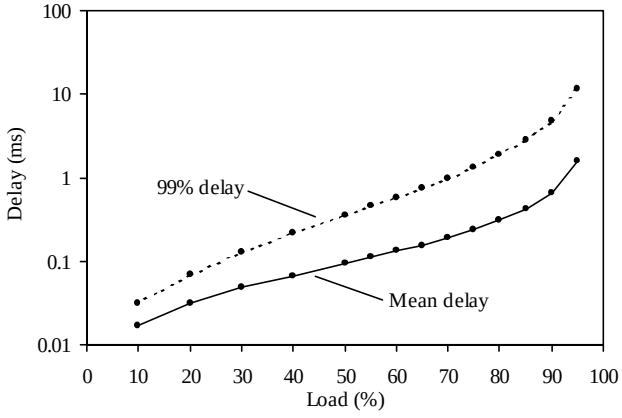

Figure 5. Load results for Poisson source

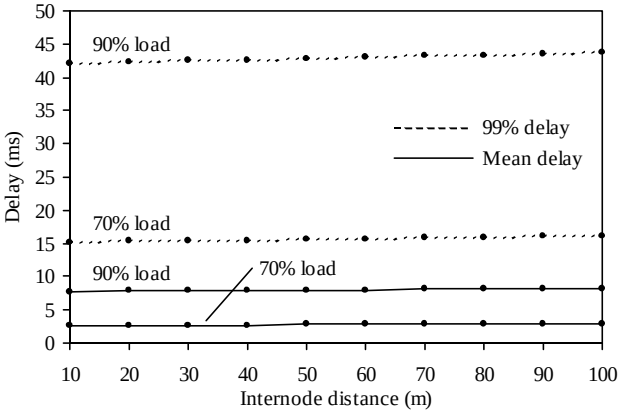

Figure 6. Node distance results for MPEG-2 source

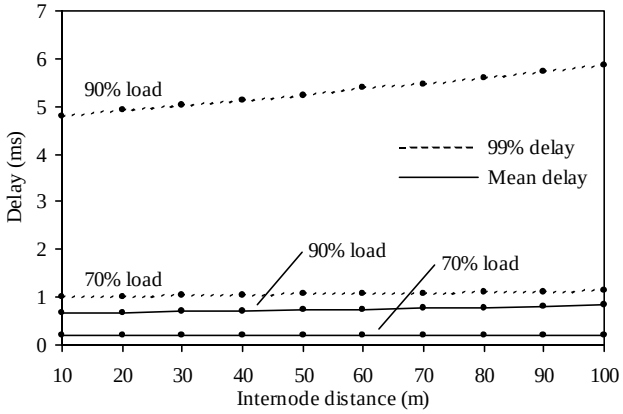

Figure 7. Node distance results for Poisson source

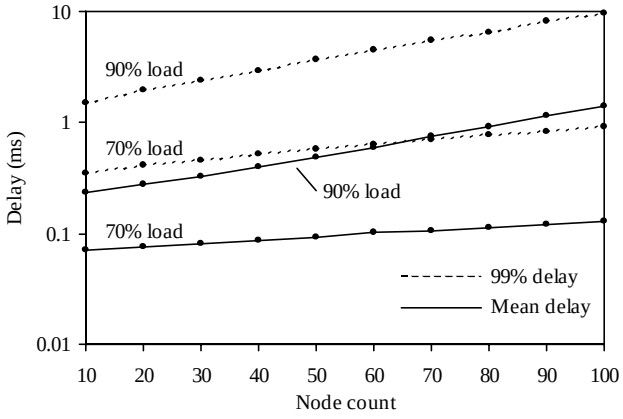

Figure 8. Node count results (Poisson source)

5. Evolving to a Store-and-Forward WSN

FireWire is a shared-medium protocol where each node is a part of the data path (i.e., the repeat path). FireWire does not allow for loops and thus supports only acyclic bus and/or tree topologies. Acyclic topologies cannot load balance traffic on multiple links and do create a disconnected network when a link failure occurs. The next step will be to WSNs that support store-and-forwarding. Simple, full-duplex Ethernet can be used between nodes. With store-and-forward capability, each node can act as a sensor, router, and/or cache. Thus, future WSNs can have arbitrary topologies for improved performance and robustness. Routing is needed to find a path between nodes in the network.

5.1 Routing in a WSN

In a WSN geographic routing cannot be based on greedy directional approaches used in wireless networks. In a wireless network, a geographically nearby node is also a neighbor node for communications. In a WSN, a geographically near node is not necessarily an immediate one-hop neighbor and a distant node may be only one hop away. This difference between wireless and wired networks makes the geographic routing problem in WSNs difficult. Directional geographic routing also assumes that the longitude/latitude between two nodes is different. For nodes very close together (i.e., on opposite sides of a wall), it may not be possible to differentiate their location by a GPS. Additionally, locations based on longitude/lattitude may be less useful than human identified locations such as “airplane seat 37E” or “door facing camera in tool room in terminal A”. Using string identifiers for geographic location allows the use of a geographic database with relational fields.

A WSN node has limited processor resources. Processor resources are needed for localized image processing and/or other localized sensor fusion. Link bandwidth is also a limited resource. Memory and power are less limited. Existing link-state and distance-vector routing protocols are processor and bandwidth intensive. It is also not necessary for every node to contain routing information for all other nodes. We propose to use a hybrid link state and source routing approach with distributed route servers. This approach minimizes node processor load, reduces routing-related broadcast traffic, and is a solution to geographical routing. We call our scheme *hybrid routing* since it is a fusion of two routing methods (source routing and distance vector) and is a mixture of centralized and distributed.

Each node in a WSN has a unique address (e.g., an Ethernet and/or IP address) and an associated geographic location string. Figure 10 shows a WSN with 16 sensor nodes, two route servers, and a single sensor fusion node.

The route servers are manually installed in ad hoc locations distributed throughout the network. The route servers all have a common, well known multicast address. Each route server contains a database of <node_address, node_location, IP_address> records and a network map built from received link state updates and shared information from other route servers. If the route servers do not exist, or if they all fail, nodes can use broadcast discovery to locate target nodes and determine source routes to them. If an individual route server fails, nodes associated with this server can automatically discover another route server and begin using it (or degenerate to using broadcast). In Figure 10 it is shown how nodes associate with the closest (by hop count) route server. In the case of multiple route servers the same distance, one is chosen at random (this is the case for nodes 3 and 10 in the sample WSN of Figure 10).

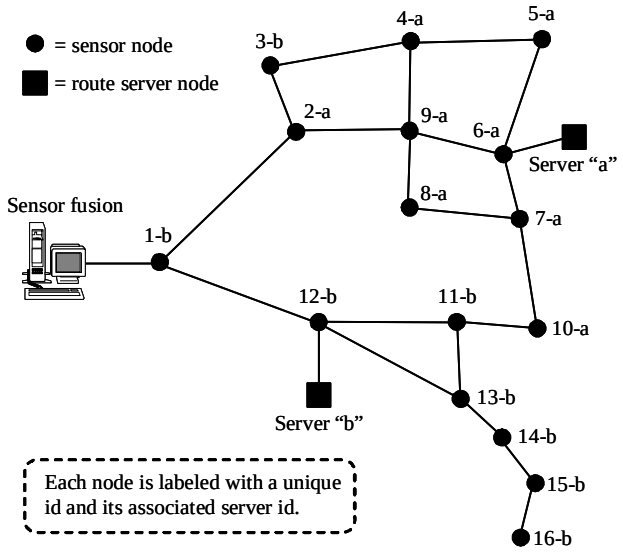

Figure 10. A WSN showing distributed route servers

5.1 Source routing mechanisms in a WSN

We propose to use source routing as the packet forwarding mechanism between nodes in a WSN. The distributed route servers return source routes to nodes in response to location queries (this is described in Sections 5.2 and 5.3). The source route is a series of port numbers for sensor nodes. For an Ethernet packet, we envision appending a source route RIF as part of the preamble. The RIF consists of hops, a sequence number, and a TTL field. The sequence number and frame CRC are used to identify redundant discovery packets. The TTL field is used to prevent infinite looping. The TTL is initially set to the maximum span of the network. Packets leaving a WSN would have their RIF stripped off.

5.2 Hybrid routing - the distributed route server

Multiple route servers can exist in a WSN. The purpose of these route servers is to:

1) Contain associations of node address and location.
2) Possibly contain associations of node address to IP address (i.e., to act as an ARP server).
3) Contain a network map used to determine best routes between two nodes.

Route servers run two internal processes (daemons):

- route_server - receives and handles discovery, link state update, and route query packets.
- route_server_update - maintains and shares network map information with other route servers in the WSN.

Figure 11 shows these two processes.

```
PROCESS route_server
BEGIN
  IF (receive a packet to server) THEN
    IF (discovery packet) THEN
      reply with a "found server" to node
    IF (link state packet) THEN
      update network map
    IF (route query packet) THEN
      determine route from network map
      reply with a route to node
END

PROCESS route_server_update
BEGIN
  PERIODICALLY DO
    share map information w/ route servers
    update based on shared information
END
```

Figure 11. Route server processes for hybrid routing

5.3 Hybrid routing - the sensor node

A sensor node contains sensing, caching, and routing capabilities. The sensing capability can both receive and send packets. Each sensor node acts as a source routing switch with link state updates sent to a route server node (if it exists). The sensor node runs three processes:

- find_server - searches for a route server.
- link_state - if a route server is known to exist, sends link state updates to the route server.
- handle_packet - receives packets and forwards them by source routing or broadcast.

Figure 12 shows these three processes. The find_server process finds the nearest route server. The TTL variable in the discovery packet limits the scope of the packet to TTL hops from the sending node. If multiple replies are received, indicating multiple route servers at the same

hop count distance, one server is randomly chosen. The node caches the source route to the server to direct link state updates and route queries to the route server. The `link_state` process sends directed link state updates to the route server when major link status changes occur. The handle_packet process forwards only "new" broadcast packets out all ports except the port on which the packet was received. Forwarding by source routing can be implemented directly in hardware and need not burden the node processor. A signature (e.g., CRC32 or MD5 hash) is made for each broadcast packet received and is stored. This signature is used to identify packets recently seen which do not be broadcast in multiple copies. Signatures age-out of storage. A similar method is used in DSR [12] to prevent duplicate broadcasts in wireless networks that use source routing.

```
PROCESS find_server
BEGIN
  PERIODICALLY DO
    server_exists = FALSE
    FOR i = 1 to maximum span
     broadcast discovery packet with TTL = i
     wait for a found reply from a server
     IF (receive a reply from a server) THEN
       server_exists = TRUE
       cache route to server
       send link status to server
END

PROCESS link_state
BEGIN
  WHILE (server_exists is TRUE) DO
    IF (link status changes) THEN
      send link status to server
END

PROCESS handle_packet
BEGIN
  IF (receive a packet) THEN
    IF (packet addressed to this node) THEN
      copy packet to application buffers
    IF (routed packet) THEN
      forward packet by source route
    IF (broadcast packet) THEN
      compute packet signature
      IF (signature has been seen before) THEN
        discard the packet
      ELSE
        store the signature
        forward the packet to all ports
END
```

Figure 12. Sensor node processes for hybrid routing

A sensor node caches routes to other nodes by address or location. If the cache does not contain a route for a packet queued for transmission and if a route server exists, then the route server is queried. If there is no route server, routes are found via a broadcast discovery packet. The function in Figure 13 is executed when a packet is to be transmitted to another node. The argument dest_addr is the destination node address or location identifier. The function returns a source route. This function can be extended to return multiple source routes for a query that is satisfied by multiple nodes.

```
FUNCTION find_route(dest_addr)
BEGIN
  IF (route to dest_addr is in cache) THEN
    return(source route)
  ELSE IF (server_exists is TRUE) THEN
    send route query to known server
  ELSE
    broadcast query to all nodes
  wait for a reply
  get source route from reply and put in cache
  return(source route)
END
```

Figure 13. Sensor node function for hybrid routing

5.4 Future work in WSN routing

Additional work needs to be done in two areas; evaluating scalability and support of QoS.

5.4.1 Scalability and performance of hybrid routing

The scalability of the hybrid routing scheme needs to be investigated. The distributed route servers maintain a global network view but are only accessed by a subset of nodes. In particular, the following need to be studied:

1) Where and how should route servers be located?
2) What mechanisms should be used for sharing information between route servers?
3) How should partial or stale network map information within a route server be dealt with?

The structure of geographic information within a route server needs to be studied. A route server can be a relational database with keys based on geographic identifiers (text strings and GPS coordinates). Nodes can be queried by location or attribute. For example, queries could be made by a sensor fusion node for:

- "Views from all cameras in all tool rooms" if an intrusion in an unspecified tool room is suspected.
- "Views from all cameras within 500 feet of the ticket counter camera in terminal A" if searching for an individual recently reported at this ticket counter.

5.4.2 Support for QoS in hybrid routing

Some nodes may require paths with guaranteed bandwidth and bounded delay. The route servers offer a mechanism to have knowledge of bandwidth commitments on paths and to make reservations. The broadcast source routing discovery can also serve as a means of bandwidth probing and reservation where nodes with insufficient link bandwidth can drop the discovery packet. These are areas for future work.

6. Summary and Future Work

Building large-scale video surveillance systems is of national importance. With camera and other sensor costs continuing to drop, we believe that shared-medium networks are needed. We demonstrated via simulation that IEEE 1394b FireWire can transport packetized video streams with very good performance Even 99% queueing delays are under 100 milliseconds for over 90% offered load. For a 50% offered load, the 99% queueing delay is less than 10 milliseconds.

In future wired sensor networks (WSN), each node may act as a sensor, router, and cache. We proposed the use of distributed route servers to solve the geographic routing problem. Namely, how can a node access sensor data by location including human specified location and/or GPS coordinates. We propose that source routing be used as a packet forwarding mechanism. The route servers receive link state updates from sensor nodes and respond to queries with source routes. We call our new scheme *hybrid routing*. Additional work is needed to evaluate the performance of hybrid routing.

References

[1] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat Monitoring: Application Driver for Wireless Communications Technology," *Proceedings of the First ACM SIGCOMM Workshop on Data Communications in Latin America*, 2001.

[2] K. Delin and S. Jackson, "Sensor Web for In Situ Exploration of Gaseous Biosignatures," *Proceedings of the IEEE Aerospace Conference*, pp. 465-472, 2000.

[3] Detect and Photograph Intruders With a Portable, Motion-Sensing Camera!, SMARTHOME, Inc., 2002. URL: *http://www.smarthome.com/764801.html*.

[4] *Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. Committee on Networked Systems of Embedded Computers. National Academy Press, Washington, DC, 2001.

[5] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proceedings of Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 263-270, 1999.

[6] W. Feng, J. Wadpole, W Feng, and C. Pu, "Moving Towards Massively Scalable Video-Based Sensor Networks," *Large Scale Networking Workshop*, 2001.

[7] "FireWire versus Gigabit Ethernet: Dare to Compare," 2002. URL: *http://www.unibrain.com/products/ieee-1394/fw_vs_gbit.htm*.

[8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proceedings of the 33rd Annual Hawaii International Conference on Systems Sciences*, 2000.

[9] IEEE P1394b, Draft Standard for a High Performance Serial Bus (High Speed Supplement), (*http://www.zayante.com/p1394b/drafts/p1394b1-33.pdf*), Draft 1.3.3, November 16, 2001.

[10] IEEE Std 1394-1995, Standard for a High Performance Serial Bus, 1995.

[11] C. Intanagonwiwat, R. Govidian, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, pp. 56-67, 2000.

[12] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, Kluwer Academic Publishers, 1996.

[13] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, pp. 243-254, 2000.

[14] Y-B. Ko and N. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *Wireless Networks*, Vol. 6, No. 4, pp. 307-321, 2000.

[15] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, "A Scalable Location Server for Geographic Ad Hoc Routing," *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, pp. 120-130, 2000.

[16] A. Mukherjee and A. Adas, "An MPEG2 Traffic Library and its Properties," (*http://www.knoltex.com/aboutKnoltex/people/amarnath/papers/mpeg2Library.htm*), 1998.

[17] Network Camera, DVR and Video Servers, Axis Communications, Inc., 2002. URL: *http://www.axis.com/products/camera_servers/index.htm.*

[18] T. Norimatsu, H. Takai, and H. Gail, "Performance Analysis of the IEEE 1394 Serial Bus," *IEICE Transactions on Communications*, Volume E84-B, Issue 11, pp. 2979-2987, 2001.

[19] "People-Mover Project Brings 21st Century Surveillance System to Dallas Airport," Telindus, 2002. URL: *http://www.cellstack.com/news_info/case_dallas_airprt.pdf*.

[20] C. Perkins and E. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, 1999.

[21] H. Qi, S. Iyengar, and K. Chakrabarty, "Distributed Sensor Networks – A Review of Recent Research," *Journal of the Franklin Institute*, Vol. 338, pp. 655-668, 2001.

[22] T. Radford (science editor), "In-Flight Cameras to Curb Hijack Fears," *The Guardian*, May 9, 2002.