Project Report

Project Name: Experimental Analysis of Streaming of Surveillance Video over Small-World Ad-hoc Wireless Networks

Author: Mesut Ali Ergin, Marco Gruteser
WINLAB, Rutgers University

Date: 03/20/2006

Presentation Overview

- Project Achievements Summary
- Problem Definitions and Background
  - Problem Definition
  - Methods Proposed
  - Simulations
- Implementation
- Tests and Results
- Conclusion

Project Achievements Summary (1)

- Methods for Available Bandwidth Estimation are proposed (4/1-9/1 and 6/1-9/1 periods)
  - Finding channel utilization via monitoring **channel busy time**
  - Finding channel utilization via monitoring **gaps between subsequent packets**
  - Finding channel utilization via **correlating current link rate with amount of legitimate traffic** seen per unit time on the channel

Project Achievements Summary (2)

- A method for bandwidth consumption calculation is proposed (4/1-9/1 and 6/1-9/1 periods)
  - ***Per source bandwidth consumption*** calculation through application TSPEC and link settings, i.e.
    - Application : average packet size, average packet rate
    - Link : overhead (packet headers) length, link rate, timings (ACK, DIFS,SIFS,RTS,CTS)
  - ***Total bandwidth consumption per flow*** calculation through CSN neighbor counts
  - ***Total bandwidth consumption per flow*** approximation through 2-hop transmission range neighbors.

Project Achievements Summary (2)

- Simulation of the proposed scheme on NS-2 (6/1-9/1 period)
  - 35-node network on a 1.5km X 1km terrain is tested with average connectivity degree of four
  - Packet delivery rate, average end-to-end delay and throughput performance are inspected thoroughly
- Implementation of a subset of the features on GNU/Linux (9/1-1/15 period)
  - Bandwidth reservation has been integrated into Lunar for Linux kernel 2.4.X
  - Available bandwidth estimation (the third method mentioned in the earlier slide) is implemented as a user space program to report to Lunar kernel module
  - Substantial amount of time invested to create a representative proof-of-concept demo on ORBIT testbed (64-node grid) with a control graphical user interface
  - Implementation is ported to work on the 400-node grid

Project Achievements Summary (3)

- Performance evaluation and validation tests were carried on ORBIT (9/1-3/31 period)
  - Performance of bandwidth estimation methods has been evaluated on different basic topologies
  - Perceptual video streaming quality tests have been conducted
  - Admission control integration has been validated for multi-hop topologies
- Methods and some of the results are being compiled for possible submission to a related academic conference (3/1-3/31 period)

Project Achievements Summary (4)

- Brief conclusions from the project
  - Implemented bandwidth estimation algorithm is found to be very accurate for the topologies without much non-decodable carrier sensing (CS) activity
  - A variant of bandwidth estimation algorithm (without listening the air all the time) can also be used with some degradation in accuracy
  - Admission control integrated routing showed instant gains in throughput and delay, even in small dense networks without many CS-domains
  - Packet probing approach showed the potential to be an augmentation method for the implemented bandwidth estimation technique

Presentation Overview

- Project Achievements
- Problem Definitions and Background
  - Problem Definitions
  - Methods Proposed
  - Simulations
- Implementation
- Tests and Results
- Conclusion

Problem Definitions

- Providing QoS for Wireless Ad Hoc Networks is challenging
  - Lack of centralized control (need to use distributed algorithms)
  - **Shared nature** of wireless channel makes resource allocation very complex
  - **Mobility of nodes** often break connections and/or reservations
  - Involves **cross layer** actions between MAC, Network and Application layers

- A possible-to-implement method is needed for practical 802.11 based wireless ad-hoc networks
  - **Available bandwidth** needs to be estimated
  - Routing should integrate resource allocation along the **entire path** of communication

Presentation Overview

- Project Achievements
- Problem Definitions and Background
  - Problem Definitions
  - Methods Proposed
  - Simulations
- Implementation
- Tests and Results
- Conclusion

Estimation of Unused Bandwidth (1)

- Most feasible way of implementation is via measuring channel utilization (ρ)

- Then unused (available) bandwidth is given as:

$$B_{unused} = (1 - \rho) \cdot B$$

and it is smoothened with a weighted average as:

$$B_{unused} = \alpha B_{unused} + (1 - \alpha)(1 - \rho) B$$

where $0 \leq \alpha \leq 1$

Estimation of Unused Bandwidth (2)

- Although quite feasible with marketed legacy 802.11 hardware designs, none of today's 802.11 vendors expose carrier sensing information (thus channel utilization) to users
- This feature might be requested from vendor, if using a customized 802.11 hardware already
- Otherwise, alternative solutions are needed to be used on legacy 802.11 hardware…

Estimation of Unused Bandwidth (3)

- By exploiting overhearing capabilities of wireless nodes, one can implement a channel usage estimator (up to a certain accuracy level)
- Via promiscuous listening, a node can extract
  - Number of legitimate packets seen on air
  - Length of the packets those packets
- Together with current channel rate information, a node can calculate remaining unused portion of the channel with these information

- '*Algorithm details*' and 'w*hat limits the accuracy*' are explained in the following slides…

Estimation algorithm detail

- The available bandwidth is calculated:

$$B_{avail} = B_{\max} - \left(\frac{8\sum_i (L_i + \mathrm{Pr}\,eamble)}{T} + \frac{i.CW_{\min}.uSlot.B_{\max}}{T}\right)$$

L_i -- length of the i-th packet

i -- the number of packets heard in the interval T

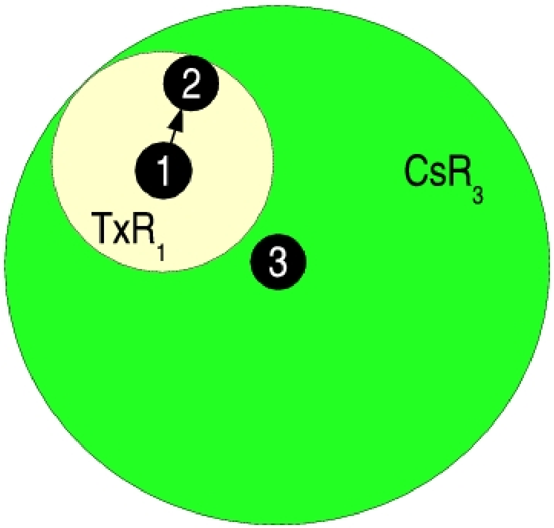
$CW_{\min}$ -- the minimum contention window

note:

we assume an ideal channel without much interference, thus P_e caused by collision can be assumed very low

Limitation of Estimation

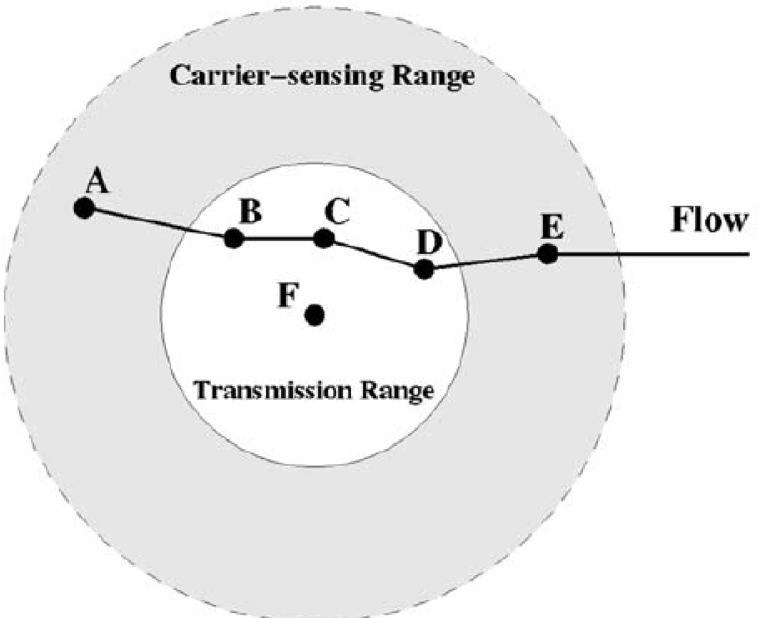
- Not only decodable packets consume available channel time
- Typically carrier sensing range of a legacy 802.11 hardware is much larger than its transmission range
- Thus, packets that are not decodable (but consume available bandwidth) can not be accounted for with the explained approach.



Flow 1->2 consumes bandwidth of node 3, although node 3 can't tell this by counting the packets on air.

Prediction of bandwidth consumption (1)

- When making decision for admission control, bandwidth consumption due to parallel transmissions along the path must be considered
- E.g., flow 1 consumes five times its average rate at node C (or F)
- This consumption is dependent on the number of carrier sensing neighbors on the path of the flow.



Prediction of bandwidth consumption (2)

- In general, bandwidth consumption at a given node can be formulated as:

$$B_{consume} = (N_{cont} + 1) \times \frac{W}{1 - P_e}$$

N_{cont}: Number of CSNs on the path, excluding destination
W: Required bandwidth of the source node due to the intended flow
P_e: Average link packet error rate on the path

- In practice, it is very hard to calculate N_{cont} exactly. However, it can be approximated by:

$$\hat{N}_{cont} = \left\lceil \frac{\text{Carrier Sensing Range}}{\text{Transmission Range}} \right\rceil$$

- So, for 802.11, it is assumed to be $N_{cont} \simeq 2$

Admission Control

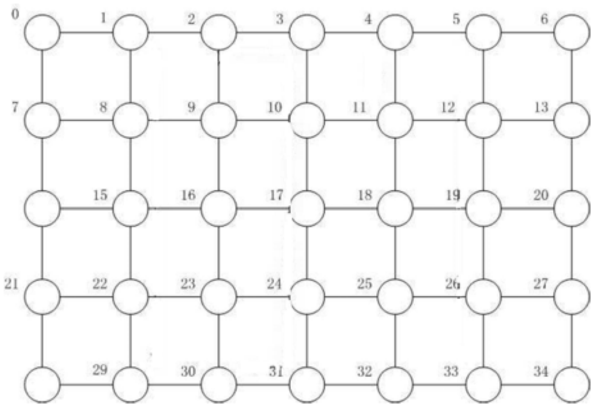
- For admission control, a node calculates the bandwidth consumption with the explained method, and checks whether its carrier sensing neighbors have enough bandwidth to support the flow.
- This mechanism has to be tightly coupled with the route establishment for the flow.
- Admission control design was selected to be integrated into LUNAR routing algorithm for wireless ad hoc networks.

Presentation Overview

- Project Achievements
- Problem Definitions and Background
  - Problem Definitions
  - Methods Proposed
  - Simulations
- Implementation
- Tests and Results
- Conclusion

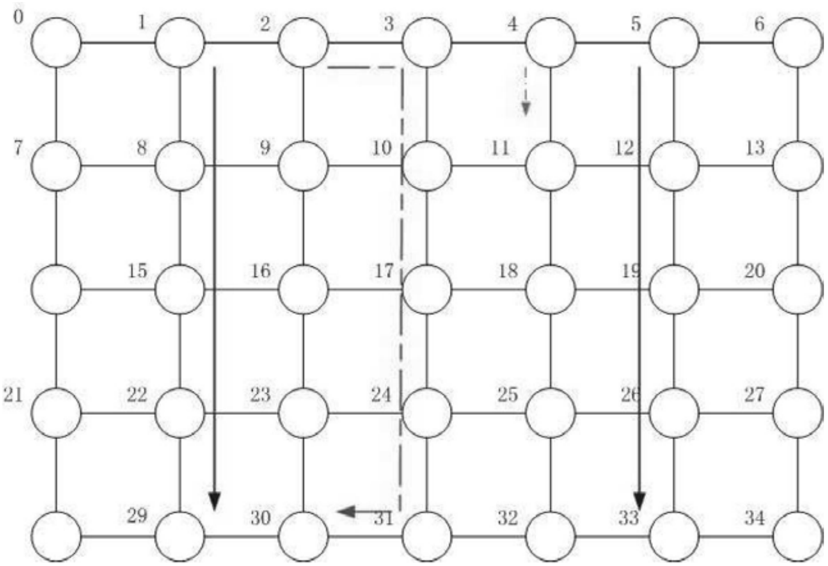
Simulation of the proposed methods (1)

- ns-2 simulator is used on a static network of a grid topology (5-by-7)
- Nodes have a 250m transmission range and 550m carrier sensing range
- Nodes are placed 250m apart on the grid
- IEEE 802.11 MAC @ 2Mbps is used
- Modified LUNAR is deployed on the network
- 230Kbits/s CBR-type flows are tested for correct operation of admission control



Simulation of the proposed methods (2)

- Two flows 1->29 and 5->33(solid arrows) are offered first. Both are admitted by using corresponding shortest paths.
- When the third flow 2->30 is offered, it is accepted to be on an alternative route, since the shortest path route has not enough available bandwidth due to the consumption of flow 1.
- Attempt of fourth flow 4->32 does not succeed. It is rejected due to lack of enough bandwidth within the whole network.

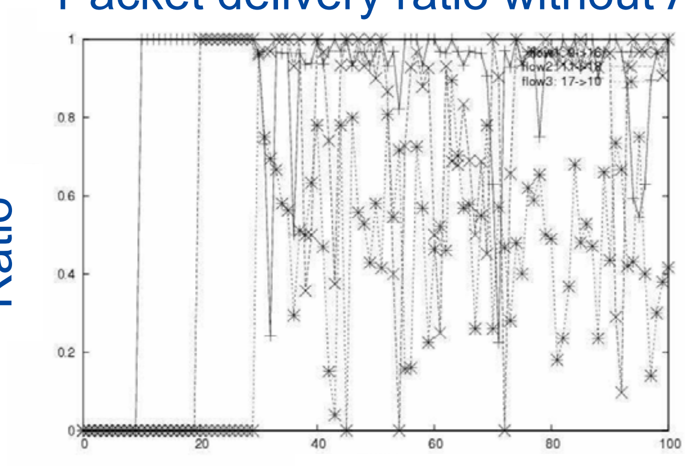


Simulation of the proposed methods (3)

- Another simulation experiment involves nine randomly chose senders, starting 110Kbps CBR streams.
- Each node starts 5-secs after the previous one
- Results are presented for the averages from runs with and without admission control

Packet delivery ratio without AC

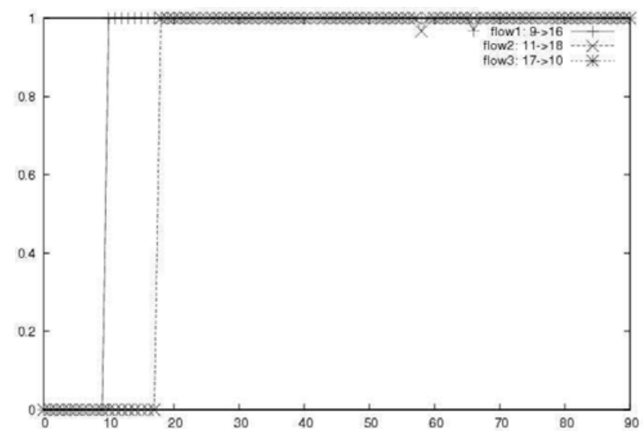
Ratio



Time (s)

Packet delivery ratio with AC

Ratio

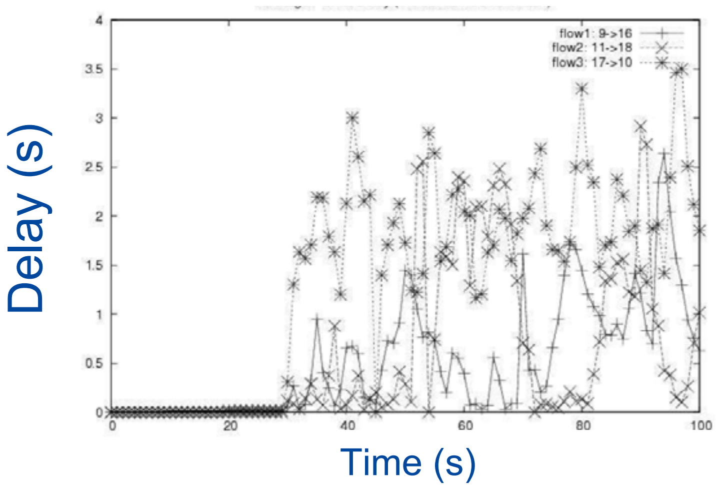


Time (s)

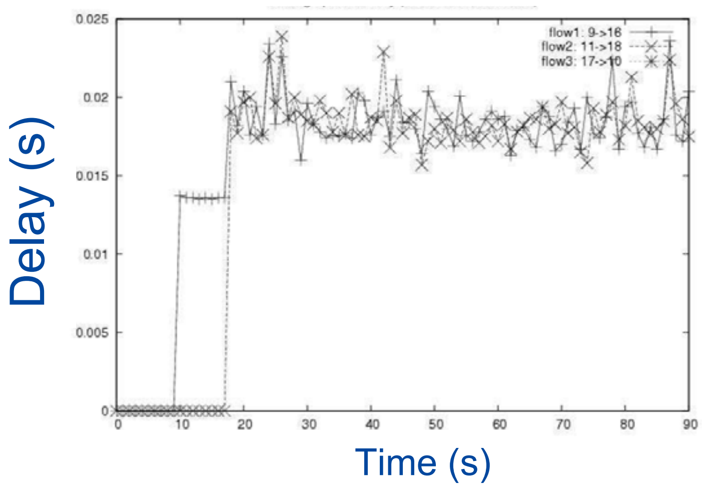
Simulation of the proposed methods (4)

- On average, five out of the nine flows are admitted
- With admission control, packet delivery ratios are stable and very close to one
- Admission control also reduces delay (almost two orders of magnitude) and delay variation

Average packet delay without AC



Average packet delay with AC



Presentation Overview

- Project Achievements
- Problem Definitions and Background
  - Problem Definitions
  - Methods Proposed
  - Simulations
- Implementation
- Tests and Results
- Conclusion

Implementation on ORBIT

- Proposed mechanisms are implemented and tested on ORBIT testbed using
  - Linux 2.4.X Kernel with Atheros 5212 based 802.11a/b/g wireless mini-PCI cards
  - LUNAR routing protocol ver.0.1.3
  - ORBIT nodeHandler ver.1.71
  - iperf traffic generator ver.2.0.2
  - vlc media player ver.0.8.4

- There are two major software components in the implementation: available bandwidth estimation module (`monitor` program), and modified LUNAR kernel module

`monitor` program

- `monitor` program is a user-space utility software that is estimating available bandwidth by using the method described earlier
- It uses /proc file system to report back to LUNAR kernel module
- Application traffic requirement (throughput in kbps), current bit rate (in kbps) and interface to exploit are provided from command-line to `monitor` program
- An example invocation follows:

```
ergin@external2:~$ ./monitor ath0raw 2000 150&
Reporting to /proc/pkt_rate ...
[1] 26528
ergin@external2:~$
```

monitor program structure

- Set up a raw socket:
```
socket(PF_PACKET, SOCK_RAW, htons(protocol));
```

- Open a `/proc` file to record the number of heard packets (bytes). `/proc` is used for communication between user space and kernel, since LUNAR is a kernel module

- Read data from the socket
```
ssize_t bytes = read(sock, Buffer, sizeof(Buffer));
```

- Differentiate between control and data packets
```
if (data packet)
    total_bytes += read_bytes
```
and write the total bytes to the file for LUNAR to read

LUNAR kernel module

- Lunar kernel modules `ksapf`, and `klunar` are modified to include bandwidth requirement information exchange during route establishment phase of the original Lunar protocol
- Operation of the original Lunar protocol is preserved during the implementation
- Loading related modules and configuring monitoring interface is done as follows:

```
insmod ./lunar_ac/lnx/ksapf.o debuglevel=10
insmod ./lunar_ac/lnx/klunar.o debuglevel=10 basedevname=ath0
sysctl -w dev.ath1.rawdev=1
ifconfig ath1raw up
sysctl -w dev.ath1.rxfilter=0x1ff
```

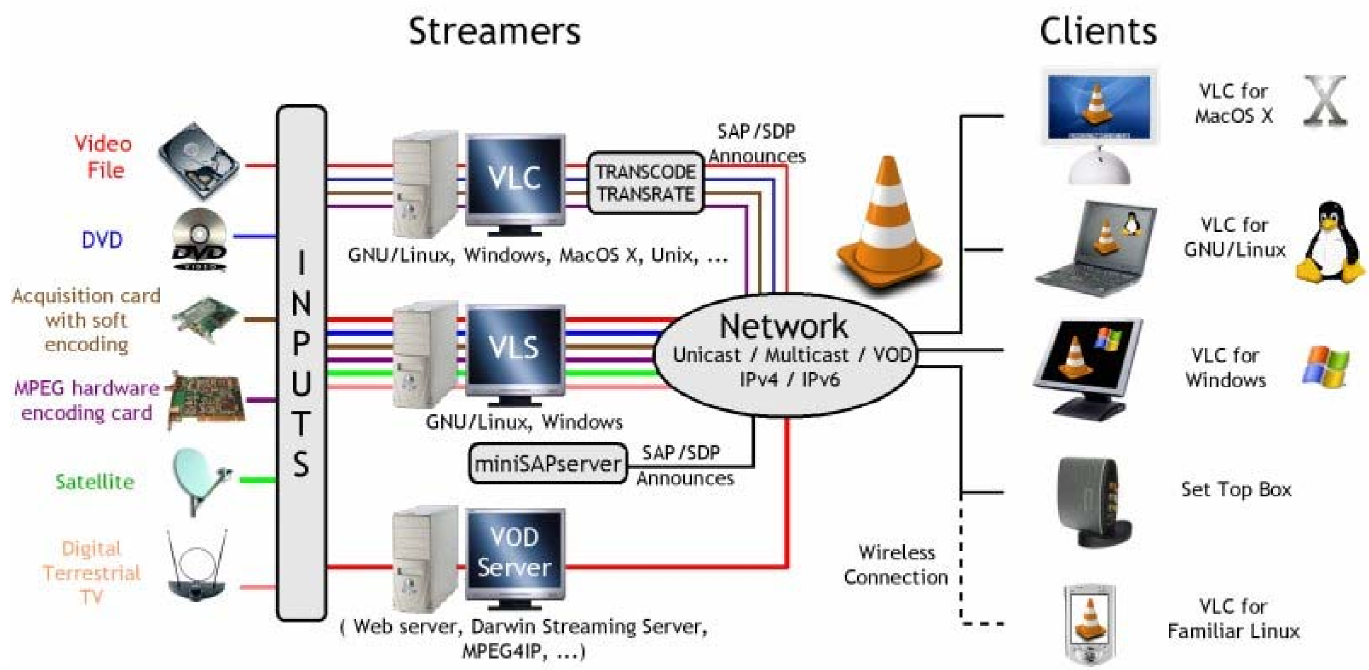
ORBIT experiment script

- Experimentation is conducted by an ORBIT nodeHandler script (in Ruby). An excerpt from the script follows:

```
require 'net/http'
require 'uri'

Experiment.name = "panasonic-test-4a"
Experiment.project = "orbit:panasonic"

# Define various elements used in the experiment
defProperty('rate', 300, 'Bits per second sent from sender')
defProperty('packSize', 256, 'Size of packets sent from sender')
defProperty('startExp', 0, 'Start experiment flag')

# Streaming Sources
defNodes('sender', [[1,2],[2,1],[3,2]]) {|node|
  # Dummy
  node.image = nil  # assume the right image to be on disk
  node.prototype("test:proto:sender", {
    'destinationHost' => '192.168.2.1',
    'packetSize' => Experiment.property("packetSize"),
    'rate' => Experiment.property("rate"),
    'protocol' => 'udp'
  })

node.net.w0.load = "ath_pci"
  node.net.w0.type = 'a'
  node.net.w0.essid = "%lunar%x"
  node.net.w0.mode = “ad-hoc"
  node.net.w0.rate = "6M"
  node.net.w0.ip = "%192.168.%x.%y"
  node.net.w0.bash = “./start-local-settings.sh"
}

...
# Streaming Destinations
...


#
# Now, start the application
#
whenAllInstalled() {|node|
  Experiment.props.packetSize = 1024
  Experiment.props.rate = 5000
  Experiment.props.startExp = 0

  wait 30000

  Experiment.done
}
```

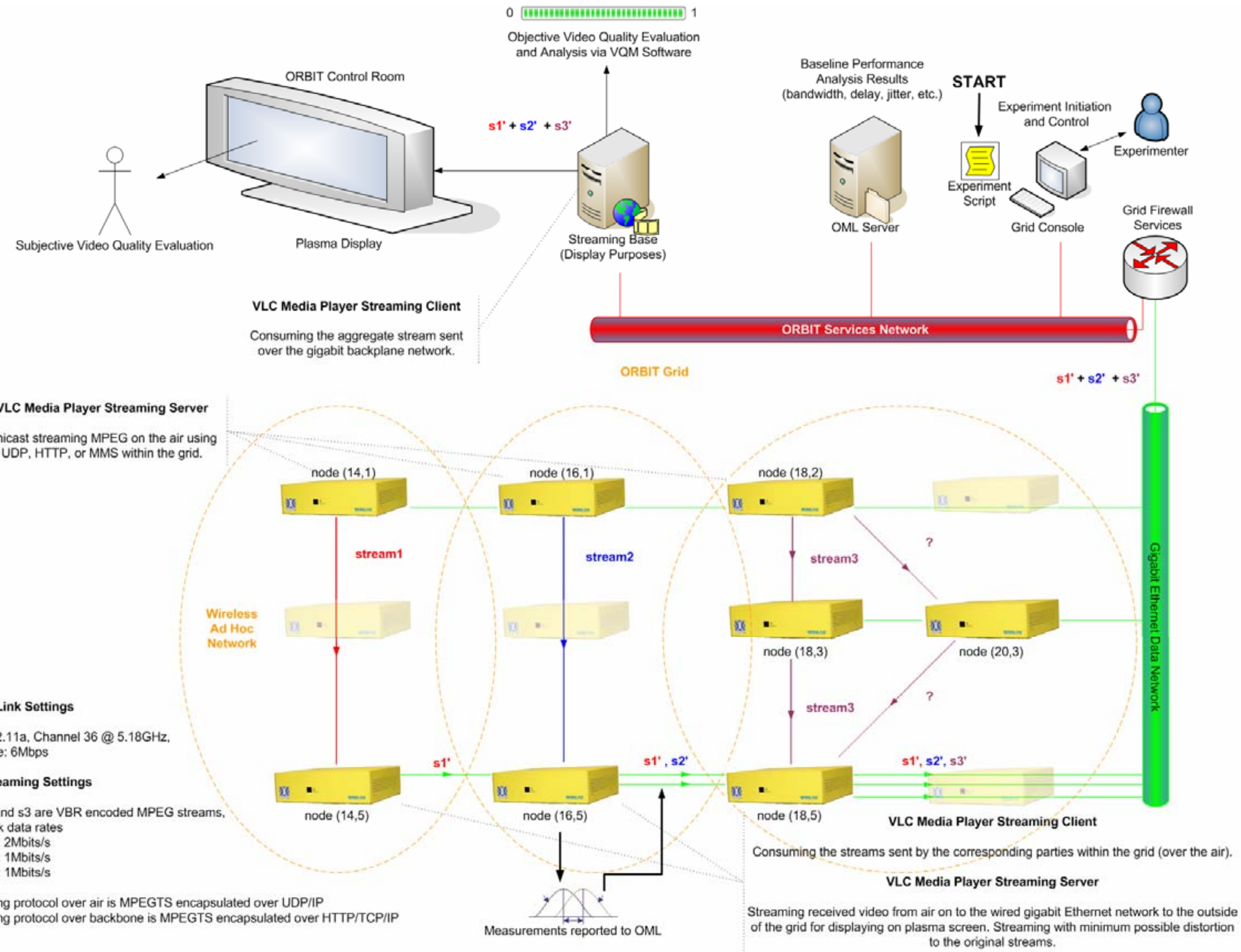
Video streaming software

- An open source cross platform media player software is used for streaming server and client needs : **VLC Media Player***
- VLC supports playback and streaming of MPEG-1, MPEG-2, MPEG-4, DivX, mp3, ogg formats.
- VLC has a capable built-in streaming server that can use many container formats and performs on-the-fly transcoding.



(*) Available for download from http://www.videolan.org

Elements of the ORBIT setup



Wireless Link Settings

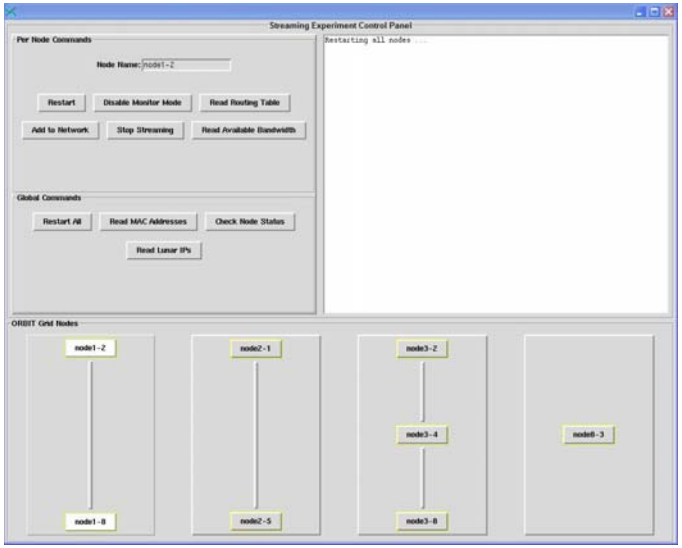
- IEEE 802.11a, Channel 36 @ 5.18GHz, Link Rate: 6Mbps

Video Streaming Settings

- s1, s2, and s3 are VBR encoded MPEG streams, with peak data rates
  - s1 : 2Mbits/s
  - s2 : 1Mbits/s
  - s3 : 1Mbits/s
- Streaming protocol over air is MPEGTS encapsulated over UDP/IP
- Streaming protocol over backbone is MPEGTS encapsulated over HTTP/TCP/IP

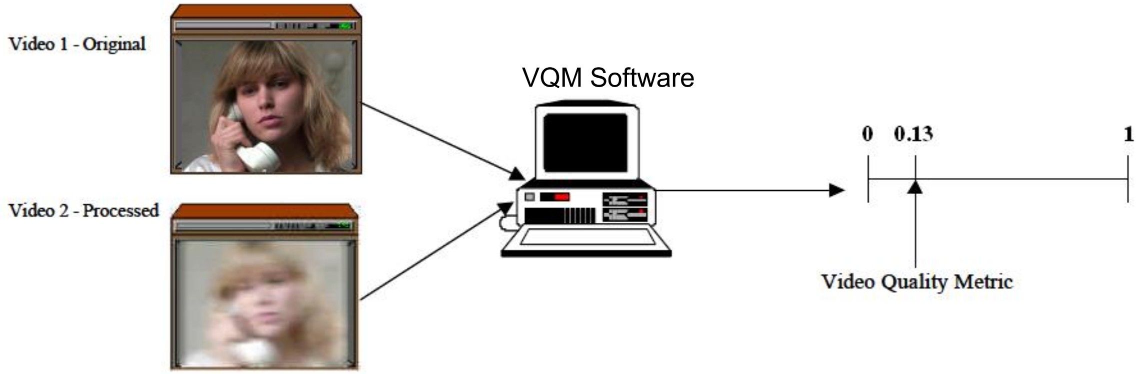
GUI for controlling the experiments

- A representative video streaming demonstration has been developed with the ORBIT setup
- Actual video streaming performance with and without admission control is presented to audience
- To increase understandability of the demonstration, a GUI to control the experiment has also been developed.
- The GUI can be used to issue, global and per-node commands to the grid.
- Feedback from the running algorithm, bandwidth estimation and routing tables can interactively be obtained with the help of the GUI.



Perceived video quality measurements

- Apart from classical baseline performance metrics, such as throughput, end-to-end delay, packet delivery ratio etc., perceived quality of the resulting video streaming is very important
- A procedure to evaluate this perceived quality has been devised for ORBIT setup
- Freely available **VQM*** (Video Quality Metric) software is used
- A VQM score is assigned to each video clip reflecting average user's perception of quality.
- VQM Score has a scale of 0 (best) – 1(worst)
- Software also provides PSNR, block distortion rate etc. measurements.
- Implements video analysis algorithms as specified in ANSI T1.801.03-2003 and ITU Recommendations J.144R



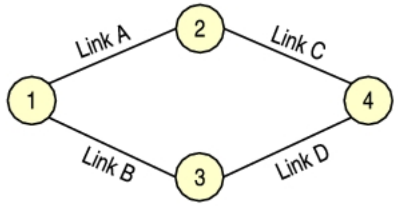
(*) Available for download from http://www.its.bldrdoc.gov/n3/video/vqmsoftware.htm

Presentation Overview

- Project Achievements
- Problem Definitions and Background
  - Problem Definitions
  - Methods Proposed
  - Simulations
- Implementation
- Tests and Results
- Conclusion

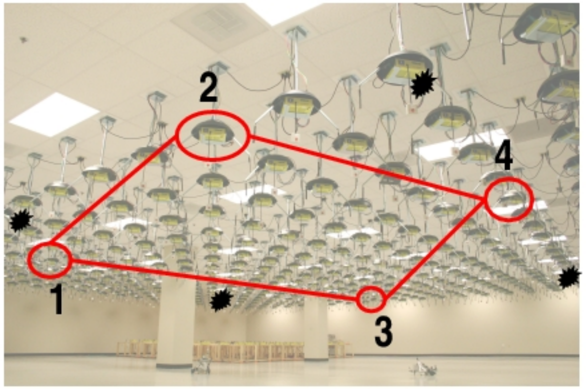
Test topology on ORBIT

- For the performance evaluation tests the illustrated topology has frequently been used.
- Characteristics of the links of the topology have been experimented well to understand the underlying structure.
- Multi-hops are generated by using noise injection capabilities of the ORBIT grid via four emitting antennas @ -1dBm.



Attribute Summary for The Experiments

Attribute	Value
Radio Nodes	1GHz VIA C3 Processor, 512MB RAM, 20GB HDD
Wireless Interfaces	2 X Atheros AR5212 based mini-PCI 802.11a/b/g
PHY/LLC/MAC Used	IEEE 802.11b @ Channel 6
PHY Link Speed	1 Mbps (fixed)
Wireless output power	18 dBm
OS Used	Linux 2.4.26
Wireless Card Driver	MadWifi r.1417 [17]



Signal strength measurements from setup

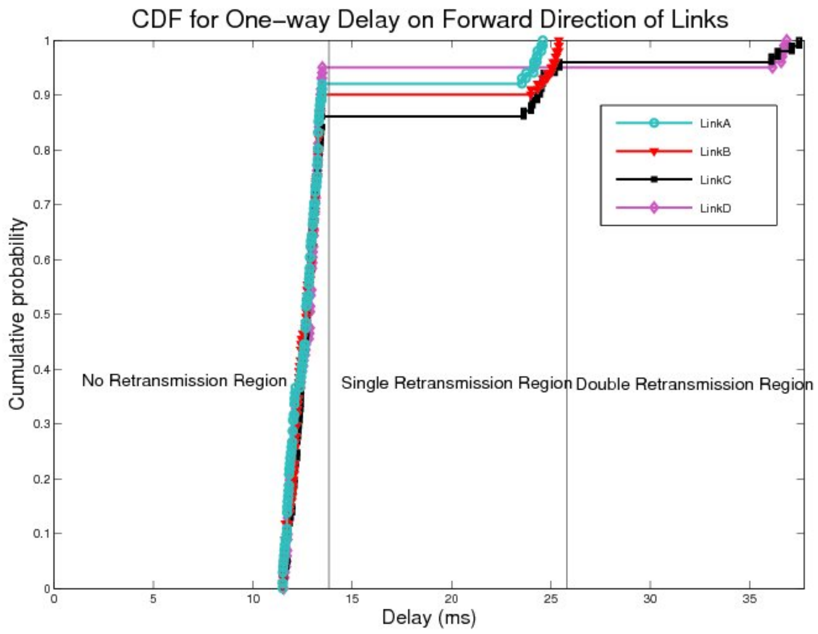
- For link test measurements, each link is individually excited with successive raw-IP packets of size 1408-bytes for a duration of 120sec
- MAC layer retransmissions are enabled
- Timing information is recorded on receiver side by means of the 64-bit μs resolution wireless card hardware timer
- Average of received signal powers (with their std. deviations) and link quality indicators (as reported by the wireless cards) during this experiment is provided below:

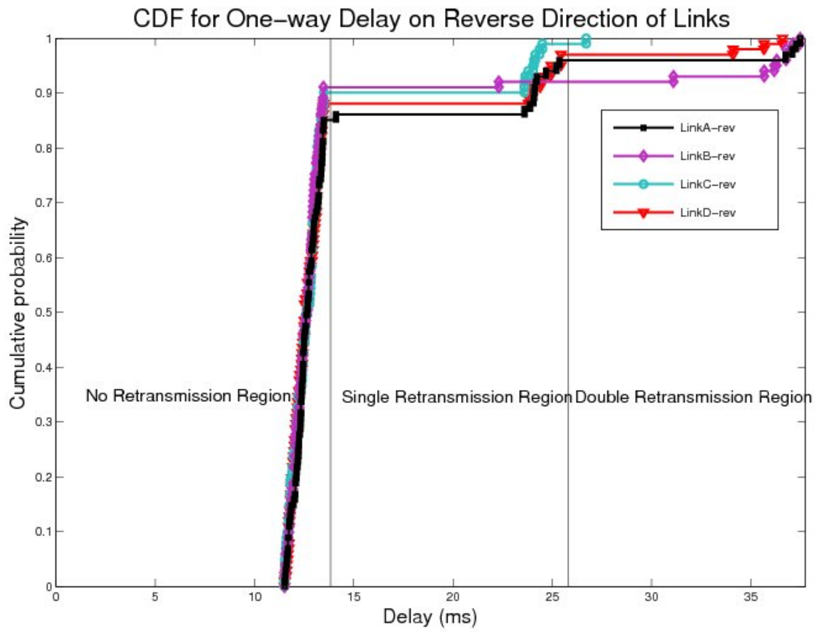
Nodes	1	2	3	4
1	X	-81dBm ±0.3 14/94	-83dBm ±0.2 12/94	X
2	-79dBm ±0.2 16/94	X	X	-87dBm ±0.2 8/94
3	-88dBm ±0.3 7/94	X	X	-82dBm ±0.3 13/94
4	X	-81dBm ±0.1 14/94	-82dBm ±0.2 13/94	X

Delay distribution on the links of setup

Panasonic
Systems Solutions
Development Center
USA

- Following results show delay distribution among the links of the setup for the same experiment
- Particular packet size used in the experiment implies appx. 12ms one-way delay for each packet. Hence, delay distribution graphs can divided into regions of retransmission as follows



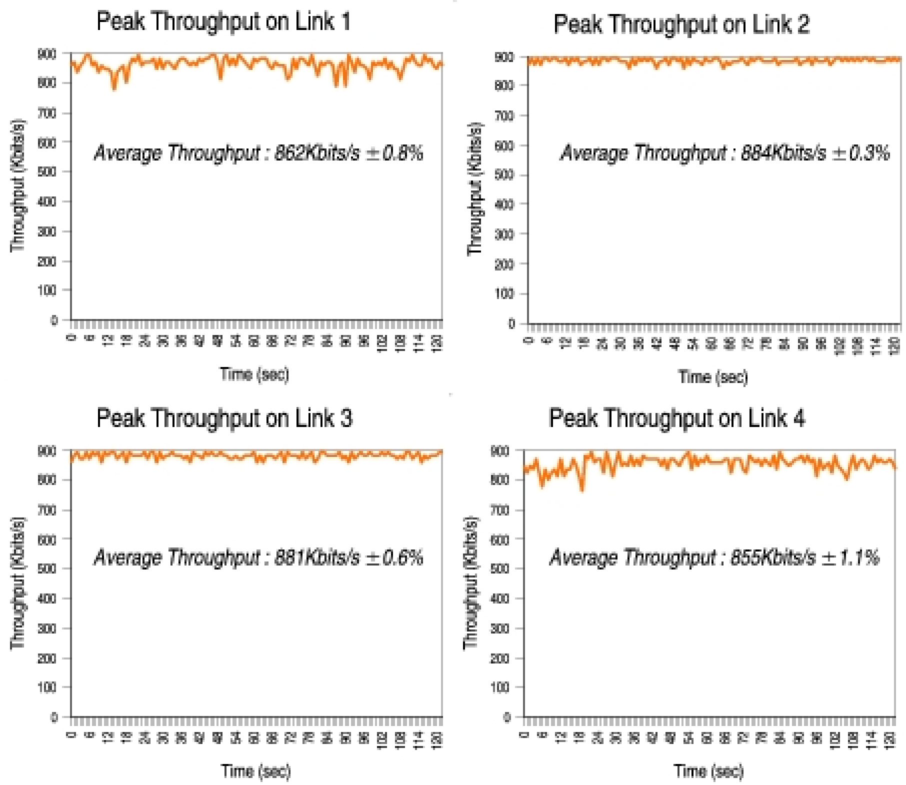


Observations from link experiments

- From these measurements, we concluded that none of the links of the setup had a significant disadvantage so far as the received signal strength is concerned
- From delay distribution experiment, we observe that 90% of the time on average, links of the setup deliver frames on the first transmission attempt. Also, it is verified that no more than two retransmissions are necessary to deliver a frame successfully
- For the topology under test, 100% frame loss is observed between nodes 2-and-3, and nodes 1-and-4
- With the help of retransmissions, rest of the links had 0% frame loss during link characterization experiments
- Hence, topology created via noise has been verified

Throughput limits per link

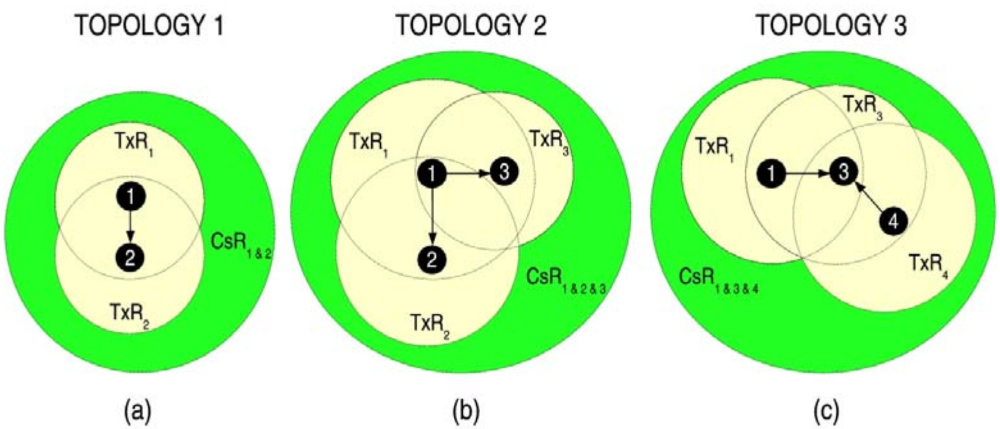
- To provide a baseline performance for available bandwidth estimation comparisons, layer-3 maximum achievable throughput tests have been done
- Raw IP packets of 1408-byte are sent as fast as possible (saturated links) one link at a time for a duration of 120 sec.
- Average results are very close to 890Kbits/s, max. achievable theoretical throughput on 802.11 at 1Mbits/s.
- Slight deviations are observed due to occasional retransmissions



monitor program performance (1)

Panasonic
Systems Solutions
Development Center
USA

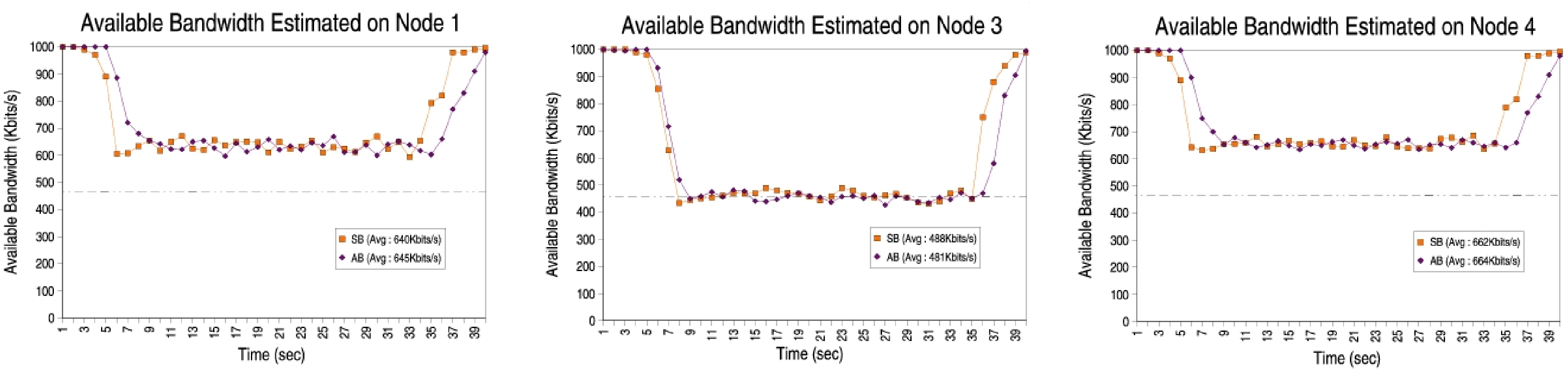
- Accuracy of the available bandwidth algorithms are tested on three basic representative sub-topologies



- Two variants of the discussed algorithm are tested
  - Simple Bookkeeping (SB): Estimator node uses only the packets that are destined for itself (no promiscuous mode due to the need for energy saving)
  - Advanced Bookkeeping (AB): Estimator node uses all decodable packets on the air (promiscuous mode allowed)
- Flows are started on 5th second of the experiments and stopped on 35th second of the experiments.

monitor program performance (2)

- Results from the third topology are shown below. Available bandwidth estimation from the viewpoint of the three nodes involved in this topology are illustrated. Calculated available bandwidth is plotted with a straight dashed line for each of the nodes.

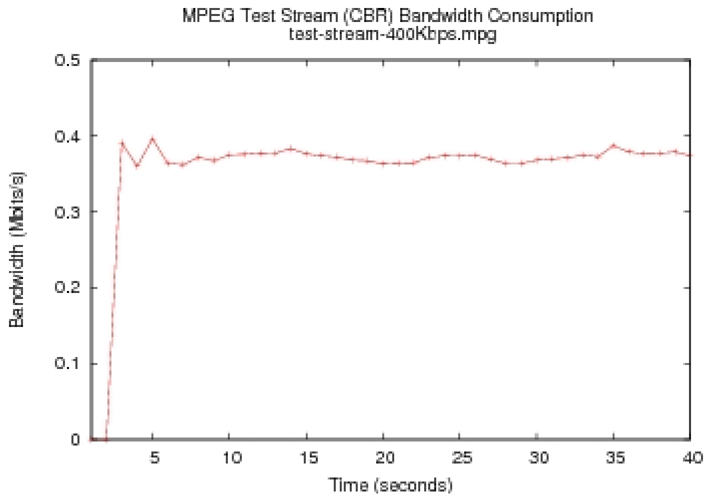


(c) Topology 3

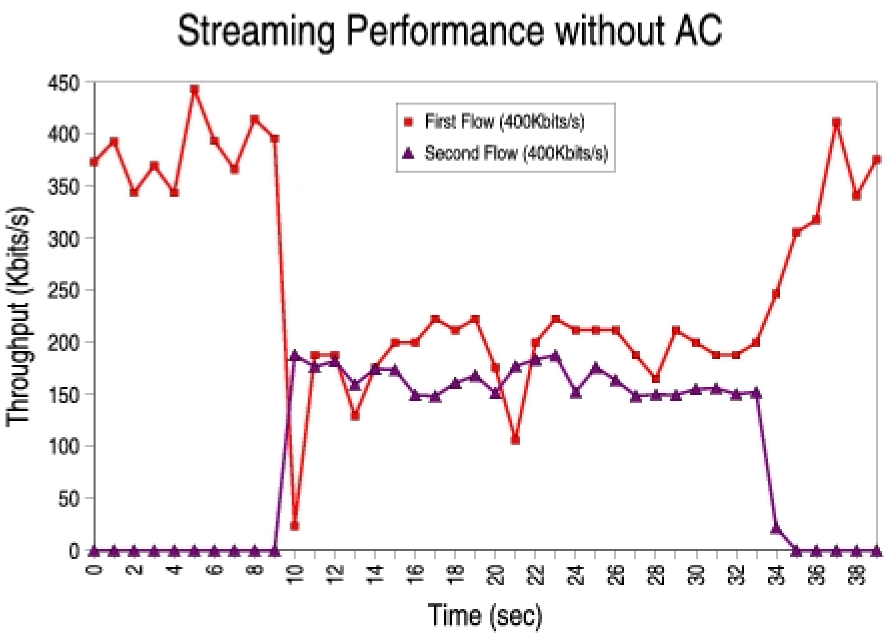
- SB and AB provide accurate estimation of available bandwidth on node 3, since all packets that consume the bandwidth are decodable on this node.
- Node 1 and 4 can not decode each others packets, thus they overestimate the available bandwidth.

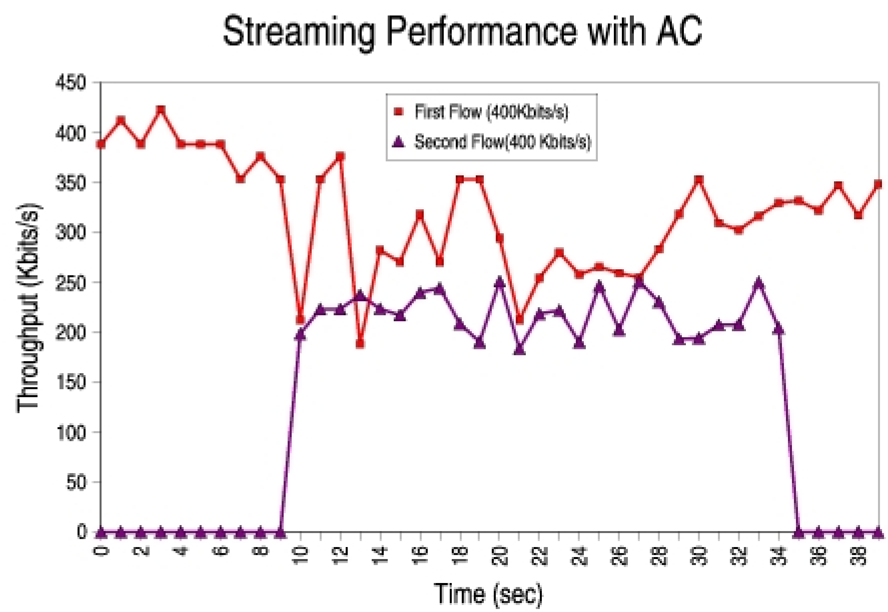
Admission control throughput/delay (1)

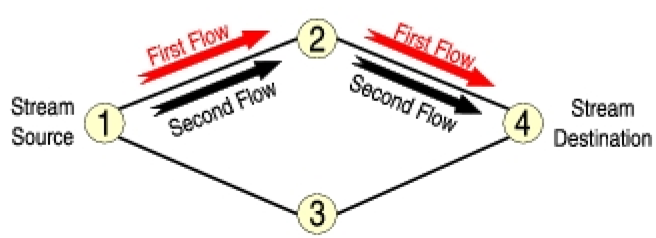
- To test admission control performance, a 400Kbits/s CBR video streaming experiment has been done
- Two parallel streams are offered to network between nodes 1 and 4.
- During 40-sec experiment, second stream is started on 9^{th} sec, and stopped on 33^{rd} second.
- Throughput and one-way delay for the streaming session has been recorded.
- Encoded bit-rate variation plot for the MPEG-2 video clip used in the experiment is given here

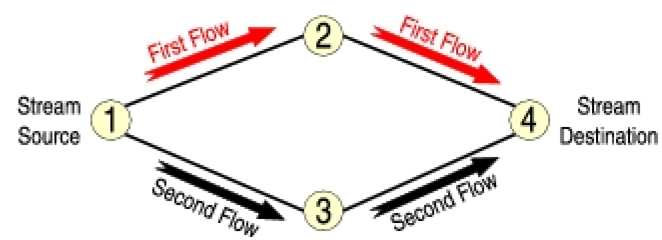


Admission control throughput/delay (2)





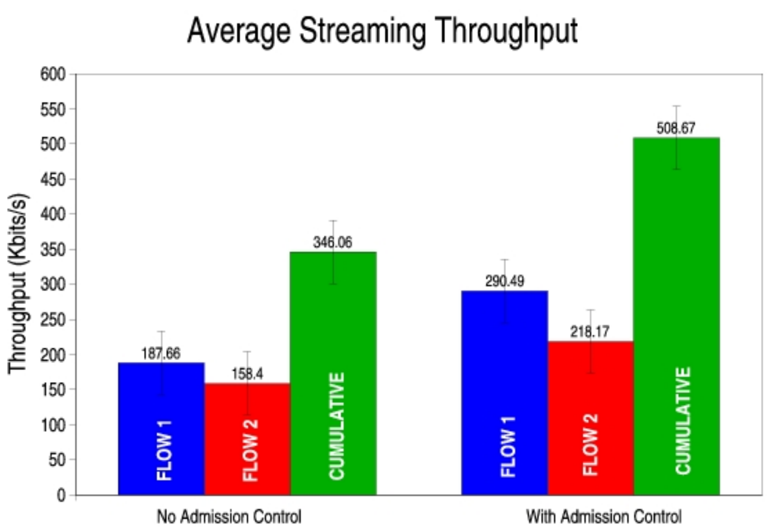


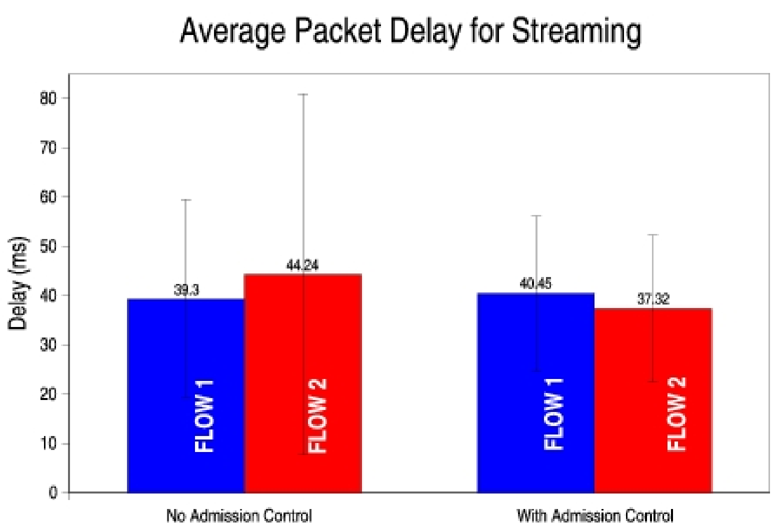


Admission control throughput/delay (3)

Panasonic
Systems Solutions
Development Center
USA

- It is observed that routing algorithm forces utilization of the same route (due to route cache) for two streams if no admission control is deployed
- Introducing admission control yields an increased cumulative throughput and reduced delay variation even though all nodes were in the same carrier sensing range (possible room for improvement in congestion scenarios)





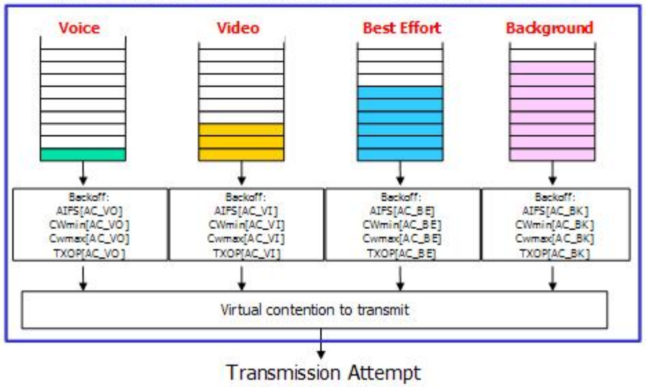
Packet Pair Probing

- Counting decodable packets approach for available bandwidth estimation has been shown to have limitations for some topologies
- An alternative approach to augment this implementation might be packet pair probing technique
- Classically, packet pair probing has been used on wired networks (esp. Internet) to detect congestion and bottlenecks on certain paths under inspection
- In the most basic case, two probe packets are sent back to back from the probe source towards the destination. Dispersion between the probe packets at the destination is highly correlated with the queuing and processing delay on the path
- Similar idea can be used to take effects of non-decodable activity on the wireless channel into account when making channel busy rate predictions

Packet Pair Probing Implementation

- A proof of concept implementation has been done on ORBIT by making use of the first topology of bandwidth estimation tests.
- Two 1300-byte raw IP probe packets are sent back-to-back. Measurements are taken from the receiver. Same sender-receiver pair is also excited with a known amount CBR-type background traffic.

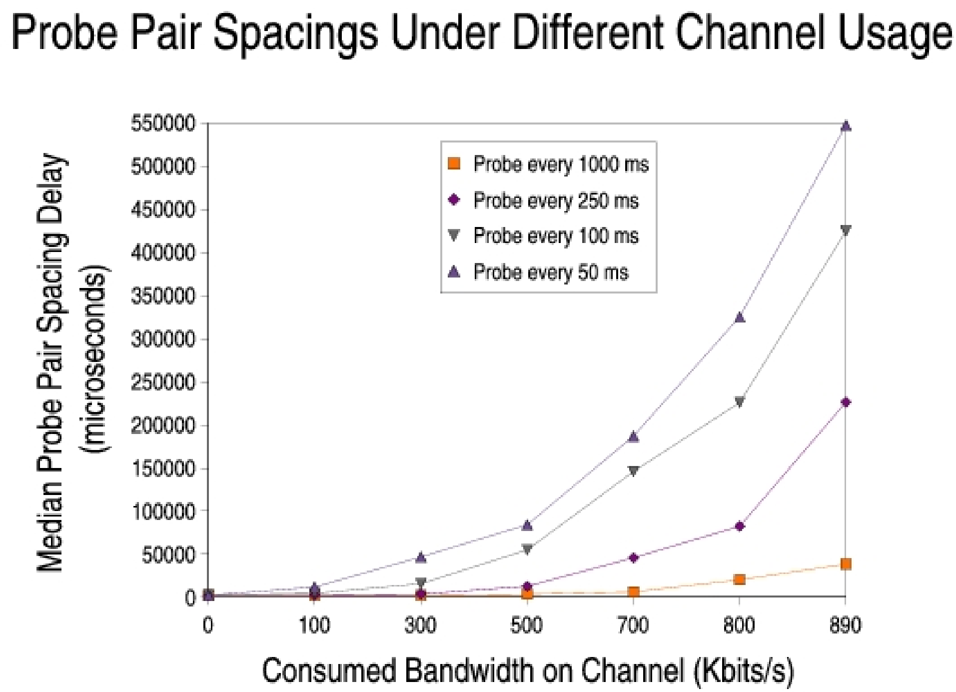
Queue Structure of a WMM Enabled Generic Source Station



- First packet is sent out from WME_VO queue of Atheros HW (High Priority)
- Second packet is sent out from WME_BK queue of Atheros HW (Lowest Priority)
- Due to the implemented operation of the priority mechanism of 802.11e, two probe packets are expected to sandwich regular traffic in between.
- Probe spacing measurement at destination should reflect amount of traffic that falls in between the two pairs.
- Two approaches possible in implementing packet probing: Sender Side Measurement/ Receiver Side Measurement
- Two critical parameters have most influence on performance: Initial probing gap, probe repeating frequency.

Packet Pair Probing Results

- From the preliminary tests, initial probing gap and probing frequency values have been found to be very important for accurate estimations.
- Depending on the desired accuracy level, this approach might require inserting a large number of probes to obtain statistically valid results.
- Close-to-congestion operation can easily be detected, thus packet pair probing method might provide an emergency warning to augment the previous estimation approach
- This is especially beneficial if significant underestimation of channel use frequently occurs in typical topologies of interest



Presentation Overview

- Project Achievements
- Problem Definitions and Background
  - Problem Definitions
  - Methods Proposed
  - Simulations
- Implementation
- Tests and Results
- Conclusion

Conclusions (1)

- In this project, practical methods for **available bandwidth estimation** and **admission control integrated routing for QoS** are proposed.
- The proposed methods are first studied and evaluated by means of **ns-2 simulations**.
- Then, **real implementations** of the methods are completed and important **validations** are made by designing experiments and demonstrations with these implementations.
- Throughout the ORBIT experimentation phases, many **overlooked** practical problems for wireless ad-hoc network QoS have been brought to attention.
- Important **motivating results** have been gathered to do more interesting research on different issues of this challenging topic.

Conclusions (2)

- In this project, practical methods for **available bandwidth estimation** and **admission control integrated routing for QoS** are proposed.
- Implemented bandwidth estimation algorithm (AB) is found to be **very accurate** for the topologies without much non-decodable carrier sensing (CS) activity
- If **energy consumption** constraints are strict for the target platform, a variant of bandwidth estimation algorithm (SB) can also be used with some degradation in accuracy
- Admission control integrated routing showed **instant gains** in throughput and delay, even in small dense networks without many CS-domains. Potential gains in very large ad-hoc network deployments would boost a **significant performance improvement**.
- Packet probing approach showed the potential to be an **augmentation method** for the implemented bandwidth estimation technique. This method can be used to prevent overall system going into the congestion state, no matter the decodable packet approach reports.

References

[ORBIT]

• D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, “Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols”, Proceedings of IEEE WCNC 2005, pp.1664-1669, March 2005.

• S. K. Kaul, M. Gruteser, and I. Seskar, “Creating Wireless Multi-hop Topologies on Space-Constrained Indoor Testbeds Through Noise Injection”, To appear in Proceedings of IEEE Tridentcom 2006, Spain, March 2006.

• J. Lei, R. Yates, L. Greenstein, and H. Liu, “Wireless Link SNR Mapping Onto An Indoor Testbed”, Proceedings of IEEE Tridentcom 2005, pp.130-135, Italy, February 2005.

• M. Ott, I. Seskar, R. Siraccusa, and M. Singh, “ORBIT Testbed Software Architecture: Supporting Experiments as a Service”, Proceedings of IEEE Tridentcom 2005, pp.136-145, Italy, February 2005.

[LUNAR]

C. Tschudin and R. Gold, "LUNAR: Lightweight Underlay Network Ad-Hoc Routing," tech. rep., University of Basel, Switzerland, January 2002.

[ns-2]

The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/, March 2006.

[MADWiFi]

MADWIFI: Multiband Atheros Driver for WiFi, http://madwifi.org, February 2006.

END