Providing QoS to Real and Non-Real Time Traffic in IEEE 802.16 networks

Vinod Sharma
Joint work with my student Harish Shetiya

Dept. of Electrical Communication Engg.,
Indian Institute of Science,
Bangalore

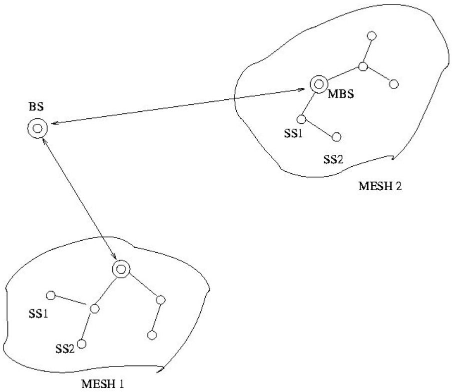
Overview

- An Overview of IEEE 802.16 Standards.
- Routing and Scheduling schemes for network throughput maximization.
- Scheduling schemes for providing the required QoS to individual UDP and TCP flows.

Overview of IEEE 802.16 standard

- Technology for wireless broadband access in the Metropolitan Area Network (MAN).
- Supports data rates upto 120 Mbps per Base Station (BS).
- Each BS can serve an area of radius upto 10 miles.
- Supports two modes of operation:
  - Point to Multipoint mode
  - Mesh mode
- Will cover PHY and MAC layer description in mesh mode only.

Mesh Mode



Physical Layer

- Flexibility in channel size (i.e. 3.5MHz, 5MHz, 20MHz etc.)
- Supports five different PHYs for both LOS and NLOS conditions.

Designation	Channel Access	Frequency band
WirelessMAN-SC	Single Carrier TDMA,DAMA/TDM	10-66 GHz
WirelessMAN-SCa	Single Carrier TDMA/TDM	$<$ 11GHz licensed
WirelessMAN-OFDM	OFDM TDMA/TDM	$<$ 11GHz licensed
WirelessMAN-OFDMA	OFDMA	$<$ 11GHz licensed
WirelessHUMAN	any of the above	$<$ 11GHz unlicensed

- Only WirelessMAN-OFDM and WirelessHUMAN PHYs with 256 point FFT OFDM modulation supported in Mesh mode.
- Adaptive Modulation and coding depending upon channel condition supported.

MAC Layer

- MAC layer divided into three sub-layers:
  - Service Specific Convergence Sub-layer
  - MAC Common Part Sub-layer
  - Privacy Sub-layer
- Only Time division Duplex(TDD) supported for Mesh mode.
- Frame consists of:
  - *Control subframe*, which is used for network creation and maintainence and also for co-ordinated scheduling of data transfer.
  - *Data subframe*, which consists of MAC PDUs transmitted by different SSs. MAC PDU consists of MAC header, Mesh subheader and optional data.

Scheduling

- TDMA used and resource allocation in terms of time slots.
- Three types of scheduling supported:
  - *Centralized Scheduling*, where the resource allocation is done by the MBS in a centralized manner.
  - *Co-ordinated Distributed Scheduling*, where resource allocation is done in a distributed manner using the control subframe.
  - *Unco-ordinated Distributed Scheduling*, where schedules are established by directed requests and grants between two nodes.

Problem Definition

- The standard does not provide any specific routing and scheduling algorithm.
- Hence we consider the problem of *Centralized routing and scheduling* for Mesh networks for two cases.
    - Maximization of network throughput.
    - Providing QoS to individual Flows

Problem Definition

- Here we consider scheduling of two types of traffic:
  - *Real Time* traffic with hard delay constraints, where the data that cannot be delivered to the destination at the end of the frame is dropped.
  - *Non Real Time* traffic without delay constraints, where no data is dropped at the end of the frame.

Routing and Scheduling for Real-Time traffic

Problem Setup:

- There are M SSs in a mesh.
- A frame has N slots.
- All data are either sent to MBS or from MBS.
- Data λ_i needs to be sent from SSi in the beginning of a frame and may change from frame to frame.
- Link rates $r(i,j)$ stay constant during a frame but change from frame to frame.
- MBS knows the topology and $r(i,j)$ and λ_i at the beginning of each frame.
- Any data not transmitted in a frame is discarded at the end of frame.

Problem: Find an optimal routing and scheduling algorithm that will *minimize* the amount of data dropped at the end of frame.

Routing

- Joint routing-scheduling optimal algorithm can be obtained via finite horizon Dynamic Programming (DP).
- However complexity of DP algorithm is exponential in number of nodes. The algorithm needs to run in *real-time* at the MBS. Thus not practical.
- Thus, we propose to decouple the routing and scheduling problems and obtain good suboptimal solutions.

Routing

For node i, the time required to transmit the entire data to the MBS is

$$T_i = \frac{\lambda_i}{r_{ip_1}} + \frac{\lambda_i}{r_{p_1 p_2}} + \ldots + \frac{\lambda_i}{r_{p_{h_i}0}} = \lambda_i \cdot \left(\frac{1}{r_{ip_1}} + \frac{1}{r_{p_1 p_2}} + \ldots + \frac{1}{r_{p_{h_i}0}} \right)$$

where the data is routed through nodes $\{p_1, p_2, \ldots, p_{h_i}\}$.

- If term in bracket minimized, the rate of data delivery to MBS maximized.
- Hence, for each node use *shortest path* routing with the link weights $\frac{1}{r_{ij}}$ to obtain a *tree* network.

We will use this routing in the rest of the talk. This leads to a tree network with MBS as the root node.

The Uplink Scheduling Problem

Problem setup:
$W_k(i)$ – data in the i^{th} node after the k^{th} slot.
Then,

$$W_{k+1}(i) = (W_k(i) + \sum_{j=1, j\neq i}^{M} 1_k(j,i)\cdot\tilde{r}_k(j,i) - \sum_{j=0, j\neq i}^{M} 1_k(i,j)\cdot\tilde{r}_k(i,j))^+$$

where,

$$\tilde{r}_k(i,j) = min(W_k(i), r_{ij}),$$

$$1_k(i,j) = \begin{cases} 1 & \text{if the } k^{th} \text{ slot is given to link(i,j)} \\ 0 & \text{otherwise} \end{cases}$$

and

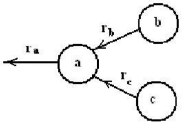
$$W_0(i) = \lambda_i$$

Problem: Maximize $(W_N(0))$ or minimize $(\sum_{i=1}^{M} W_N(i))$.

Scheduling Schemes

- These scheduling schemes applicable on the *tree network* obtained after the routing is fixed.
- Finite horizon Dynamic Programming(DP) over horizon N can be used.
- DP has a complexity of $O(M \cdot N \cdot \lambda^M)$, $\lambda = \max\{ \lambda_1, \ldots, \lambda_M\}$, which becomes prohibitive even for moderate sized networks.
- We can exploit the tree structure of the network and reduce the complexity of the algorithm.

Tree Dynamic Programming



- Consider the basic structure of the tree.
- $D_k(i)$ – maximum data that can be extracted from node i in k slots
- The optimal allocation of k slots to nodes b and c given by

$$n_b^k = argmax_{0 \leq n \leq k}\{D_n(b) + D_{k-n}(c)\},$$

$$n_c^k = k - n_b^k.$$

- The maximum data that can be extracted from the combination of nodes b and c is given by

$$D_k(b,c) = D_{n_b^k}(b) + D_{n_c^k}(c).$$

- Construct a table consisting of the following entries: $\{\ k,\ D_k(b,c),\ Sched_k(b,c)\}$, where $Sched_k(b,c) = (n_b^k, n_c^k)$ for k=1...N
- The optimal allocation of k slots to node a and jointly to nodes b and c is given by:

$$n_a^k = argmax_{0 \leq n \leq k}\{min(n \cdot r_a, \lambda_a + D_{k-n}(b,c))\}$$

$$n_{bc}^k = Sched_{k-n_a^k}(b,c).$$

- Nodes b and c should be scheduled before node a.

- Construct another table with the following entries: $\{\ k,\ D_k(a,b,c),\ Sched_k(a,b,c)\}$, where $Sched_k(a,b,c) = (n^k_{bc}, n^k_a)$. for k=1...N
- If a is a leaf node then the optimal allocation is:

$$n^k_a = argmax_{0 \leq n \leq k}\{min(n \cdot r_a, \lambda_a)\}.$$

  which depends only on parameters of a.
- Hence we can initiate the algorithm at the leaves and work towards the MBS.
- Complexity of the tree DP algorithm is $O(M \cdot N^2)$.

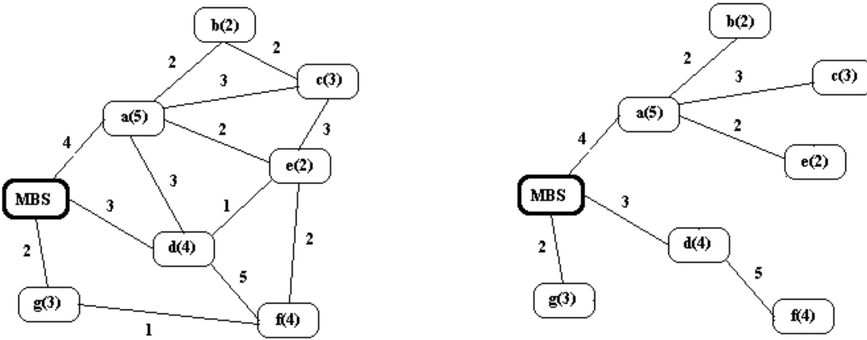
Downlink Scheduling

Problem: Maximize the amount of data delivered to the MSSs in N slots

- The downlink scheduling problem is converted into an uplink scheduling problem considering the downlink data requirement of node i i.e. λ_i, to be the uplink data requirement.
- Tree DP is now applied to obtain the schedule $\{u_1, u_2, \ldots, u_N\}$.
- The downlink schedule $\{d_1, d_2 \ldots, d_N\}$ is obtained by reversing the uplink schedule i.e. $d_i = u_{N-i+1}$.

Example

Consider the following mesh network



The Schedule

- Uplink Schedule
  - The schedule: c, e, a, a, a, f, d, d, d, g.
  - The total amount of data delivered to the MBS = 20.
  - The amount of data dropped at the end of 10 slots = 3.
- Downlink Schedule

Slot No.	1	2	3	4	5	6	7	8	9	10
transmitter	MBS	MBS	MBS	MBS	d	MBS	MBS	MBS	a	a
receiver	g	d	d	d	f	a	a	a	e	c
data of	$g(2)$	$d(3)$	$d(1)f(2)$	$f(2)$	$f(4)$	$a(4)$	$a(1)c(3)$	$e(2)$	$e(2)$	$c(3)$

Routing and Scheduling for Streaming traffic

Problem Setup:

- Streaming audio/video applications.
- Traffic carried by UDP.
- But here, there is no hard delay bound – No packets are dropped at the end of the frame.

Problem: To find a routing and scheduling scheme that maximizes the long term average total throughput to the destination (MBS in uplink and SSs in downlink).

Routing

Notation:

- λ_i = external arrival rate at SS_i
- $W_k(i)$ = queue length at SS_i at the end of k^{th} frame
- $A_k(i)$ = new external arrivals in k^{th} frame
- $X_k(i)$ = arrivals from immediate children to SS_i in k^{th} frame
- r_{ik} = rate of output link of SS_i in k^{th} frame.

Assumptions:

- $\{ A_k(i), k \geq 0 \}$ is iid, independent of the j^{th} stream.
- $\{ r_{ik}, k \geq 0 \}$ is iid, independent of the other stream.

Routing(Contd...)

- If n_i = No. of slots assigned to output link of SS_i in k^{th} frame

$$W_{k+1}(i) = (W_k(i) + A_k(i) + X_k(i) - n_i \cdot r_{ik})^+$$

- For Stability need

$$n_i > \frac{\lambda_i + \sum_{j=1}^{m_i} \lambda_{a_{(i,j)}}}{\mathrm{E}[r_i]} \text{ for all } i = 1 \ldots M,$$

where $\{\ a_{(i,1)}, a_{(i,2)}, \ldots a_{(i,m_i)}\}$ are the nodes whose data passes through node i.

- Since $\sum_{i=1}^{M} n_i = N$, we obtain

$$\sum_{i=1}^{M} \left(\lambda_i \cdot \sum_{j=1}^{h_i} \frac{1}{E[r_{p_{(i,j)}}]} \right) < N$$

where $\{p_{(i,1)}, \ldots p_{(i,h_i)}\}$ are the nodes through which the data of node i is routed.

Routing Schemes

Two schemes used:

- *Statistical Shortest Path* routing where the routing is fixed over all the frames for each node along the path that minimizes $\sum_{j=1}^{h_i} \frac{1}{E[r_{p_j}]}$.
  From last slide, this *maximizes stability region* in the class of fixed routings.
- *Instantaneous Shortest Path* routing, where the routing is changed every time the channel condition is updated, along the path that minimizes $\sum_{j=1}^{h_i} \frac{1}{r_{p_j}}$.

Uplink Scheduling

- *Problem*: To minimize the mean system queue length i.e.

$$min\{\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}x_k\}$$

  where, x_k is the total queue length in the system at the end of the k^{th} scheduling frame.
- We can use Markov Decision Theory with partial information but it will be very complex because frame size is quite large (in no. of slots).
- Hence, we develop four *suboptimal* scheduling policies.

Minimum Data Scheme

- At the end of each frame(N slots), the sum of the queue lengths is minimized i.e. the schedule for the k^{th} scheduling frame is such that

$$\sum_{i=1}^{M} W_{k,N}(i)$$

  is minimized.
- The tree dynamic programming scheme is applied in every scheduling frame.

Minimum Work Scheme

- At the end of each frame, the amount of work left in the network is minimized i.e. for the k^{th} frame,

$$\text{minimize} \sum_{i=1}^{M} \left\{ \left(\sum_{j=1}^{h_i} \frac{1}{r_k^{p_(i,j)}}\right) \cdot W_{k,N}(i) \right\}$$

  where, $\{\ p_{(i,1)} \ldots\ p_{(i,h_i)}\ \}$ are the nodes through which the data of node i is routed

- A slightly modified version of the *Tree DP* algorithm is used.

Ordering Scheme

- Based on *Klimov's* scheduling policy.
- Order nodes based on the rate of data delivery to the MBS.
    - For each node i calculate ,

$$a_i = \frac{1}{\sum_{i=1}^{h_i} \frac{1}{r_k^{p(i,j)}}}$$

    - Order nodes such that if $a_i > a_j$, the node i is higher than node j.
- Assign each slot n to the first node in the ordering which *fully* utilizes the slot.
- If all nodes *underutilize* the slots, then assign the slot to the first node in the ordering that has non zero data to transmit.

Maximum Transmission Scheme

- A very simple scheme.
- Each slot is assigned to that node which can transmit the maximum data to its next hop node.
- Slot n is assigned to node j if

$$j = \text{argmax}_{i=1\ldots M}\{\min(W_{k,n}(i), r_k^i)\}$$

where $W_{k,n}(i)$ is the queue length of the i^{th} node in the n^{th} slot.

Fixed Allocation Scheme

- A fixed number of slots allotted to each node in each frame.
- n_{ij}, the number of slots allotted to node i at node j calculated from the following optimization.

$$min \sum_{i=1}^{M} (\lambda_i - n_{ii} \cdot E[r_i])^2$$

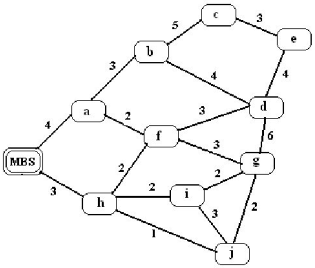
Subject to

$$\sum_{i=1}^{M} \sum_{j=1}^{M} n_{ij} \leq N.$$

Adaptive Fixed Allocation Scheme

- Slot assignment done as in Fixed Allocation Scheme.
- Channel i marked bad if $r_i < R_{th}(i)$.
- Slot assignment to bad channels deferred and the slots missed by a node is credited to it.
- If total credits $c_i > C_i$ then no slot deferment done.
- Given requirements $\{\lambda_1 \ldots \lambda_M\}$ and channel statistics, use Stochastic Approximation to calculate optimum value of $(n_i, R_{th}(i), C_i)\ \forall i = 1 \ldots M$

Simulations

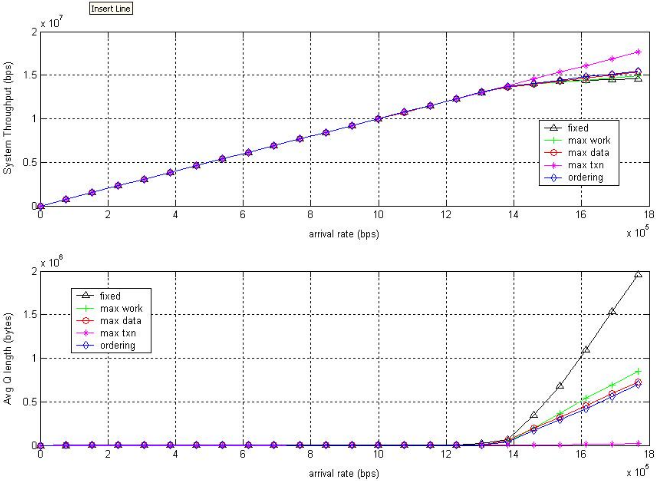


- Channel gain is Rayleigh distributed.
- Link parameter is the mean data rate per slot expressed in terms of burst profile.

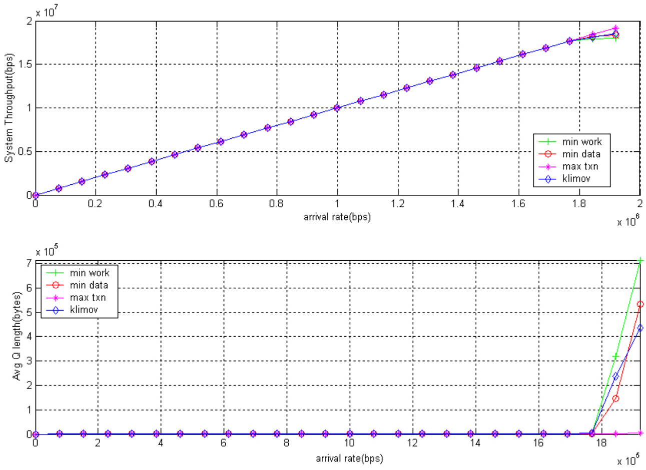
Parameters Used

Bandwidth	20 MHz
Number of Subcarriers	256
Frame Duration	10ms
No. of OFDM symbols / frame	844
No. of OFDM symbols / minislot	4
Total No. of minislots / frame	211
No. of minislots / frame for uplink Centr. Sched	50
No. of minislots / frame for downlink Centr. Sched	50

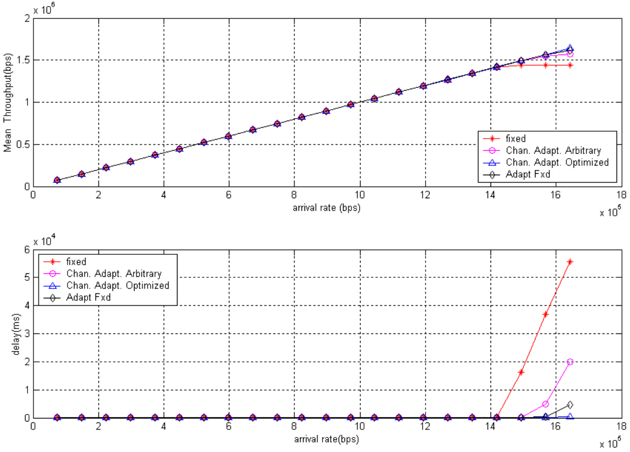
Uplink Scheduling with Statistical SP Routing



Uplink Scheduling with Instantaneous SP Routing



Comparison of Fixed Scheduling Schemes



Observation

- Under low traffic conditions, all schemes perform identically.
- Maximum transmission scheme performs better than other schemes - has larger stability region and hence lower queue lengths under heavy traffic.
- Overall, statistical routing with maximum transmission scheme has the best performance and is also computaionally the simplest. Furthermore, its stabilty region and queue lengths (under heavy traffic) are much better than a static routing and scheduling policy.

QoS for individual flows: UDP traffic

- CBR traffic requirements: Throughput $= \lambda_i$ Bytes/frame, Delay $< D_i$, Data Drop Probability $< \Delta_i$.
- Use Fixed Allocation Schemes to provide per flow QoS.
- Delay guarantee provided by dropping the unsent data at the end of the scheduling frame.
- Calculate the number of slots required to satisfy drop probability requirement.

QoS for UDP-CBR Traffic (Contd...)

- Let data of node i be routed through node $\{p_{(i,1)}, \ldots, p_{(i,h_i)}\}$.
- Find $\{\delta_{p_{(i,1)}}, \ldots, \delta_{p_{(i,h_i)}}\}$ such that $\prod_{j=1}^{h_i}(1-\delta_{p_{(i,j)}}) > (1-\delta_i)$.
- Find $n_{ip_{(i,j)}}$ such that $\mathrm{P}(n_{ip_{(i,j)}} \cdot r_{p_{(i,j)}} < \lambda_i) < \delta_{p_{(i,j)}}$.

QoS for UDP-VBR Traffic

- K State Markov Source. Rate in state j is R_j.
- Transition Probability Matrix assumed to be known.
- QoS Requirement: Delay $< D_i$, Data Drop Probability $< \Delta_i$.
- Use Fixed Allocation Schemes to provide per flow QoS.
- Delay guarantee provided by dropping the unsent data at the end of the scheduling frame.

QoS for UDP-VBR Traffic (Contd...)

- Can use Marginal Buffering Analysis.
- Let $(1 - \delta_i(1))(1 - \delta_i(2)) \geq (1 - \Delta_i)$.
- $B_k(i,j)$ is the amount of data arriving at node i for flow j in frame k.
- If there are J flows at node i, then find n_i such that

$$P(J \cdot c_i > n_i \cdot r_i) \leq \delta_i(2)$$

and

$$P(\sum_{j=1}^{J} B_k(i,j) > J \cdot c_i) \leq e^{-J \cdot I(c_i)} \leq \delta_i(1)$$

where, $I(c_i) = sup_\theta(\theta \cdot c_i - log(E(e^{\theta \cdot B_i})))$.

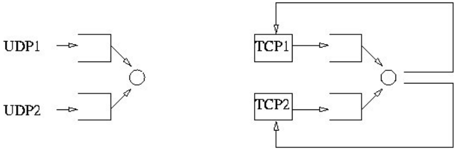
Scheduling Algorithms for TCP Traffic

We assume

- A fixed number of number of TCP connections each sending an infinitely long file (Persistent TCP connections).
- Connection i has minimum mean throughput requirement of λ_i pkts/sec.

Due to *window flow control* and infinitely long files TCP flows behave fundamentally different from UDP flows.

Comparison with UDP



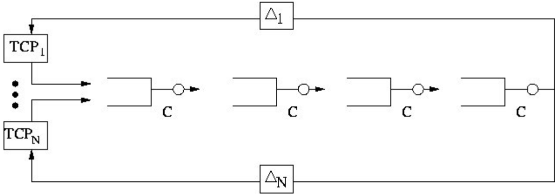
- r_1,r_2 fixed, $r_1 > r_2$
- UDP1 generates traffic at rate $0.3r_1$ and UDP2 at rate $0.6r_2$
- Algorithms mentioned earlier (except fixed and adaptive fixed), will serve UDP1 till it has packets and then UDP2. Both connections will get their packets through.

TCP Scheduling

- Replace UDP connections with TCP connections TCP1 and TCP2.
- Then all packets will transmit *only* TCP1 packets and TCP2 will starve.
- A TCP connection passing through multiple hops will get less throughput except in case of fixed and adaptive fixed algorithms.

Thus we will use only fixed or adaptive fixed algorithms for TCP connections.

TCP Scheduling



- Different TCP connections sharing the same link
- TCP i has window size W_i, packet size s_i
- Link bandwidth is c bits/sec

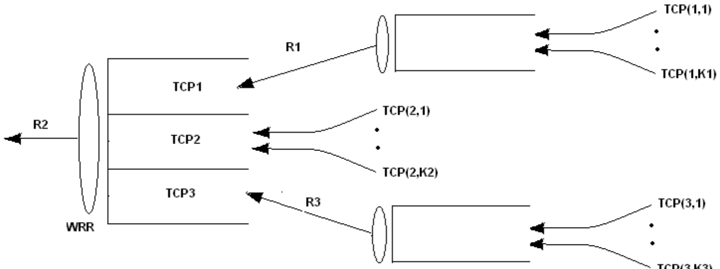
TCP Scheduling

Then Throughput obtained by TCP_i is

$$\frac{W_i \cdot c}{\sum_{j=1}^{N}(W_j - \lambda_j \cdot \Delta_j) \cdot s_j + 3 \cdot s_i + \Delta_i \cdot c} \text{ packets/sec.} \qquad (1)$$

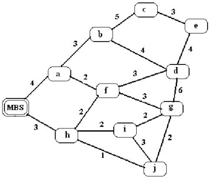
- Thus throughput of a TCP connection depends on the TCP parameters of all TCP connections sharing the link.
- Throughput obtained may not satisfy the QoS requirement.
- By dropping packets of TCP_i (say via RED control) its window size W_i can be decreased to $E[W_i]$
- c needed at a node equals the total throughput requirement of all TCP connections passing through the node.
- To avoid problems of bandwidth hogging by TCP with few hop routing, use WRR to guarantee BW to connections originating at a particular node.

TCP Scheduling (Contd. . .)



Simulation Results

Sample Network A sample network



- Channel gain is Rayleigh distributed.
- Link parameter is the mean data rate per slot expressed in terms of burst profile.
- 12 TCP flows per node
- 4 classes of traffic with throughput requirement 50kbps,100kbps,150kbps and 200kbps.

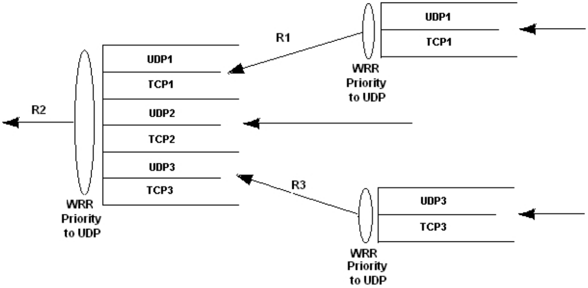
Simulation Results

Percent Error	Number of Flows
$<$ -25	0
-20 to -10	0
-10 to 0	31
0 to 10	54
10 to 20	35
20 to 50	0
$>$ 50	0

Joint Scheduling of TCP and UDP Traffic

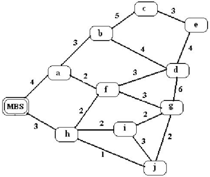
- TCP requirement: Data Rate $\geq \lambda_i^T$ Bytes/frame.
- UDP requirements: Data Rate $= \lambda_i^U$ Bytes/frame, Delay $< D_i$, Data Drop Probability $< \Delta_i$.
- Provide priority to UDP traffic over TCP traffic.
- Let n_i^U number of slots required to satisfy UDP requirements.
- Let $n_i^T = \frac{\lambda_i^T + \lambda_i^U}{E[r_i]}$.
- Then actual number of slots allotted to node i is $n_i = max(n_i^U, n_i^T)$.

Joint Scheduling of TCP and UDP Traffic (Contd. . .)



Simulation Results

Sample Network A sample network



- 12 TCP flows per node with 3 classes of traffic with throughput requirement 40kbps,80kbps and 120kbps.
- 3 CBR flows per node with requiremets: Data rate=64kbps, Delay $\leq$ 30ms and Drop Prob $\leq$ 2%
- 3 VBR flows per node characterized by 4 state Markov Chain with rates { 20, 40, 80, 120} kbps, Delay $\leq$ 30ms and Drop Prob $\leq$ 2%.

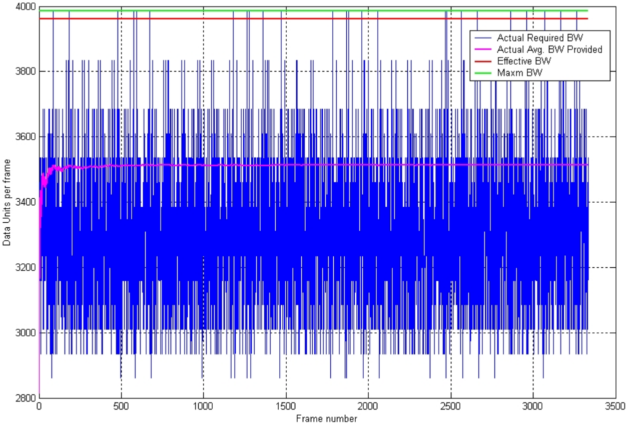
Simulation Results

- TCP Flows:

Percent Error	Number of Flows
$<$ -25	0
-20 to -10	2
-10 to 0	31
0 to 10	54
10 to 20	19
20 to 50	14
$>$ 50	0

- VBR UDP Flows:
  Maximum Average Network Delay=9.48 ms
  Maximum Drop Probability=0.0002%
- CBR UDP Flows:
  Maximum Average Network Delay=9.16 ms
  Maximum Drop Probability=0.0002%

Simulation Results



Conclusion

- Considered routing and scheduling problem for multihop WiMax mesh network.
- Developed efficient algorithms for uplink and downlink for network throughput maximization.
- Developed scheduling algorithms providing QoS to individual UDP and TCP flows.