10th International Conference on Information, Communications and Signal Processing
Singapore

Multicasting in Wireless Networks using Rateless Codes and Opportunistic Routing

G Barath Shankar   Gagandeep Singh   Sanjay K Bose
Department of EEE, IIT Guwahati, India

Wen-De Zhong
School of EEE, NTU, Singapore

Outline

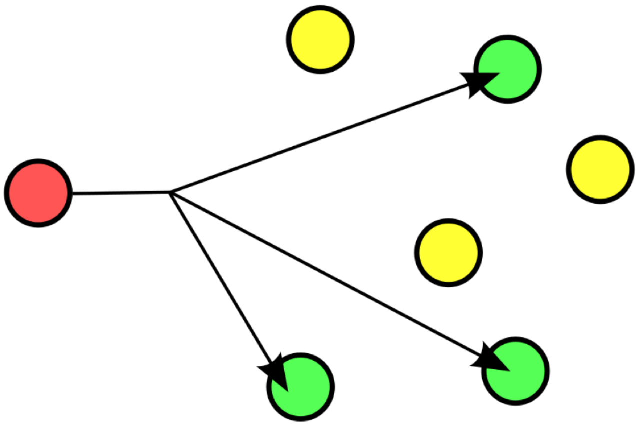
- Introduction
- Problem Discussion
- Rateless Codes
- Link Cost Formulation
- Opportunistic Routing
- Network Model
- Routing Algorithms
- Simulation Results
- Conclusions

Introduction

- Address the problem of reliable multicasting in wireless networks
- No backbone infrastructure network
- Each node can act as a forwarder
- Use of Rateless Codes for transmission
- Rateless codes offer specific advantages in multicasting
- Use of opportunistic routing instead of fixed routing

Problem : Reliable Multicasting

- Message from source node to be transmitted simultaneously to a group of destination nodes
- Other nodes may act as relays
- Message should reach each destination complete and uncorrupted
- Wireless channels are modeled as erasure channels
- A message on a link is either received completely correctly or is fully lost, i.e. erased

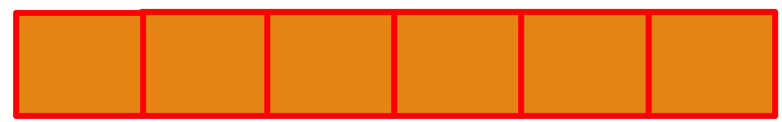


Traditional Multicasting : Feedback Problem

- Traditionally, reliable communication on erasure channels ensured by feedback and retransmission of lost messages
- In a multicast scenario, different receivers might have different missing messages
- This might lead to feedback implosion involving high feedback and retransmission costs
- Rateless codes provide an elegant solution to this problem

Rateless Codes

- Message is composed of *N* packets, each packet is treated as a symbol.
- Message is a sequence of the *N* symbols, called the source symbols.
- Potentially unlimited sequence of encoding symbols can be generated from linear combinations of the *N* source symbols.
- Source symbols can be recovered from any subset of *N* independent encoding symbols.
- Encoding symbols can be generated on the fly, as few or as many as required.
- Examples are LT codes, Raptor codes.

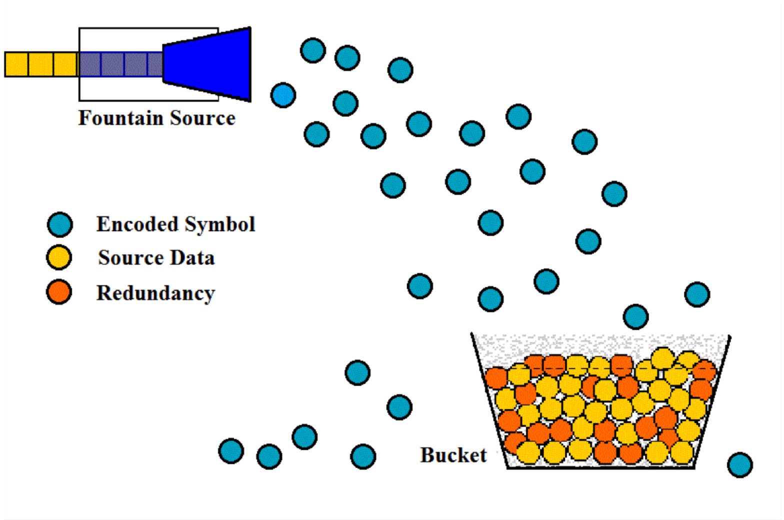


Packets/Source Symbols

Encoded Symbol

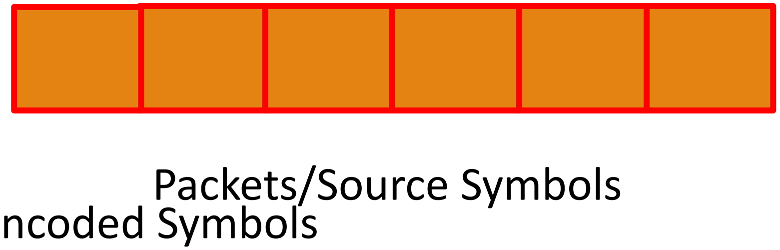
Rateless Transmission

- Sender is unaware of which packets have been received correctly by each receiver.
- Each receiver sends an acknowledgement packet ACK after it has successfully receives enough packets to decode the complete message.
- Thus, there is a substantial reduction in feedback.
- Hence, particularly useful for one to many transmissions.



Encoded Symbols

- A simple rateless coding scheme was used:
  - For initial transmissions, encoded symbols are the same as source symbols.
  - Each of these N encoded symbols are transmitted once, where N is the number of packets in the message
  - The subsequent encoded symbols transmitted are random linear combinations of source symbols
  - Receiver sends ACK when it has correctly received enough packets to decode message completely



Link Cost

In our abstraction of rateless codes, reception of *N* linearly independent encoded symbols is sufficient to decode the original *N* source symbols

Let N^* be the total number of symbols transmitted by the sender for the *N* source symbols to be received

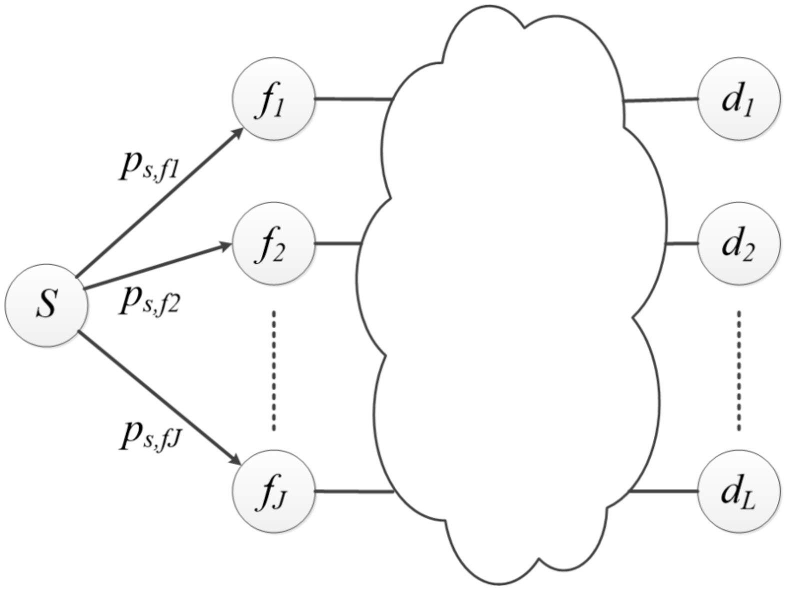
Link Cost is then defined as $$C_{link} = \frac{N^*}{N}$$

Opportunistic Routing

- Traditional routing selects a forwarder based on routing tables, prior to message transmission at each hop.
- In opportunistic routing, instead of selecting the next forwarder a-priori, the relay node is determined while the message is being transmitted.
- The relay or forwarder node is selected based on which downstream nodes have actually received the message.
- In general better message progress towards destination.
- In wireless transmission, multiple neighbouring nodes anyway receive transmissions simultaneously and hence no added usage of network capacity.

Network model

- Source node S
- A set $F = \{f_1, f_2, \dots f_J\}$ of neighbouring nodes.
- p_{S,f_j} is the symbol erasure probability of the link between S and node f_j.
- $D = \{d_1, d_2, \dots, d_L\}$ be the set of destinations.
- The cost C_{f_j,d_l} is defined as the cost along the minimum unicast path from f_j to d_l and is computed using standard *Djikstra* or *Bellman-Ford* algorithms.



Network model

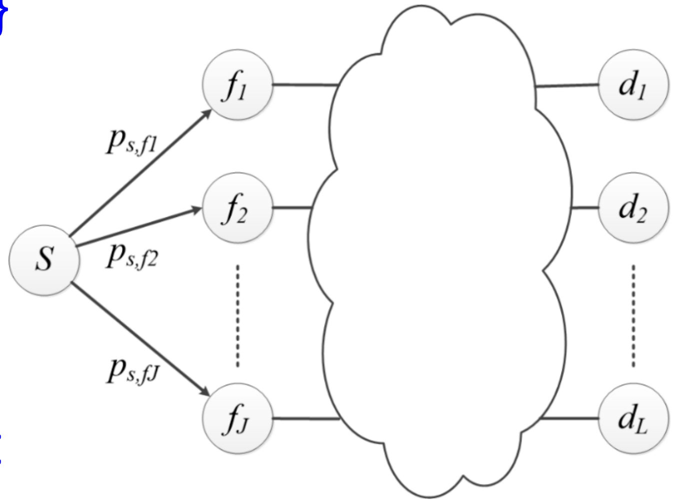
Source Node S

Destination Set $D = \{d_1, d_2, \ldots\ldots, d_L\}$

Forwarder Set $F = \{f_1, f_2, \ldots\ldots, f_J\}$

$p_{S,fj}$ = Symbol Erasure probability on link (S, f_j)

The cost $C_{fj,\, dl}$ is defined as the cost along the minimum unicast path from f_j to d_l and is computed using standard Djikstra or Bellman-Ford



Performance Metrics

Two metrics are used to assess the performance of the routing algorithms:

- Mean Number of Transmissions per packet (*MNT*)

  Mean number of packet transmissions required to send a message to all destinations divided by the number of packets per message.

- Mean Transmission Time (*MTT*)

  Time taken on an average for a message to reach all the destinations. In simulations, time taken to transmit one packet across a link is taken as the unit time.

Routing Algorithms

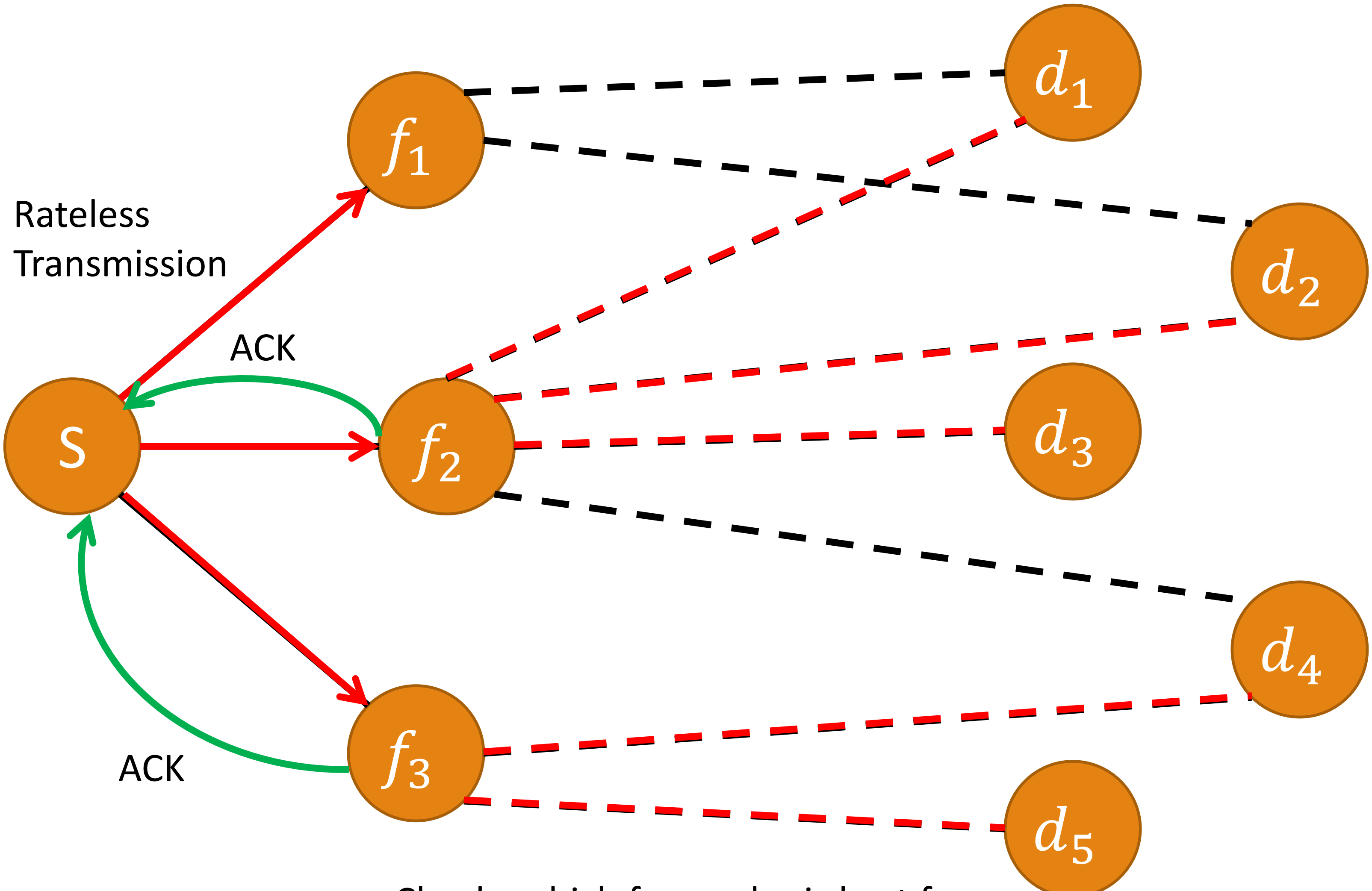
Three routing algorithms are presented next.

These are motivated with some specific goals in mind

- Greedy Forwarder Select-Routing Algorithm (GFS-RA)
- Constrained Forwarder Set-Routing Algorithm (CFS-RA)
- Minimum Forwarder Set-Routing Algorithm (MFS-RA)

Greedy Forwarder Select-Routing Algorithm (*GFS-RA*)

- A simple greedy algorithm where at each hop, the source maintains a list of those of its forwarder nodes that have sent an ACK.
- Upon receiving an ACK, the source adds the node to the list of ACKed nodes and marks all the destinations that it can reach.
- The transmission from source continues as long as enough nodes have sent an ACK so that all destinations have been marked.
- Once all destinations are marked, then for each destination, the source selects the node from the list of ACKed nodes which has the lowest cost of reaching the destination, as the forwarder for that destination.
- This algorithm is used as a benchmark to assess the performance of the other algorithms.



Checks which forwarder is best for which destination

Constrained Forwarder Set-Routing Algorithm (*CFS-RA*)

- The motivation is to provide good MTT performance.

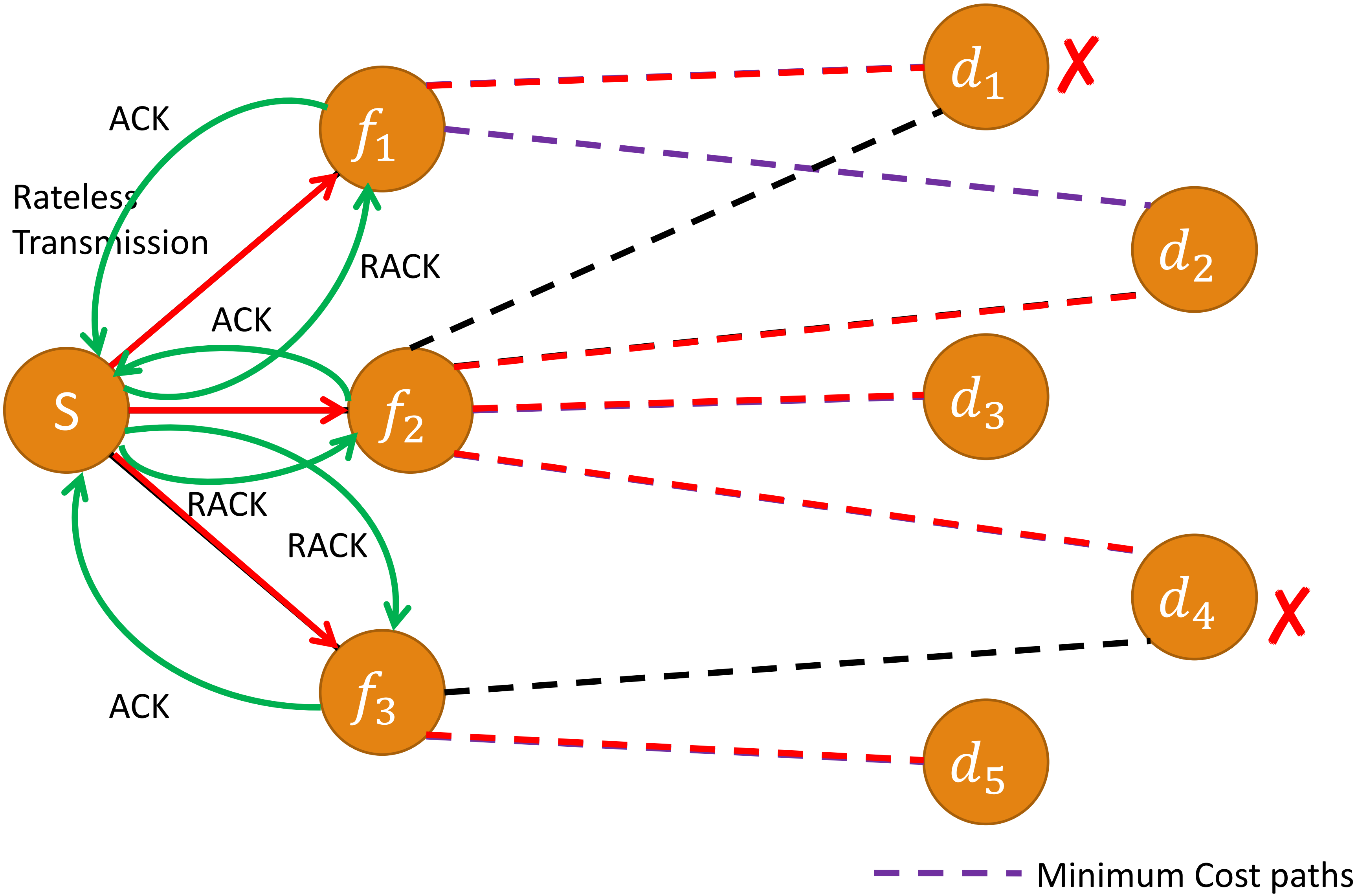
- The aim is to pass along the message to forwarders as soon as possible while also ensuring the transmission path does not veer too much away from the shortest path.

- This is done by constraining the set of neighbouring nodes that can potentially act as forwarders for each destination.

Continued in next slide

Constrained Forwarder Set-Routing Algorithm (*CFS-RA*)

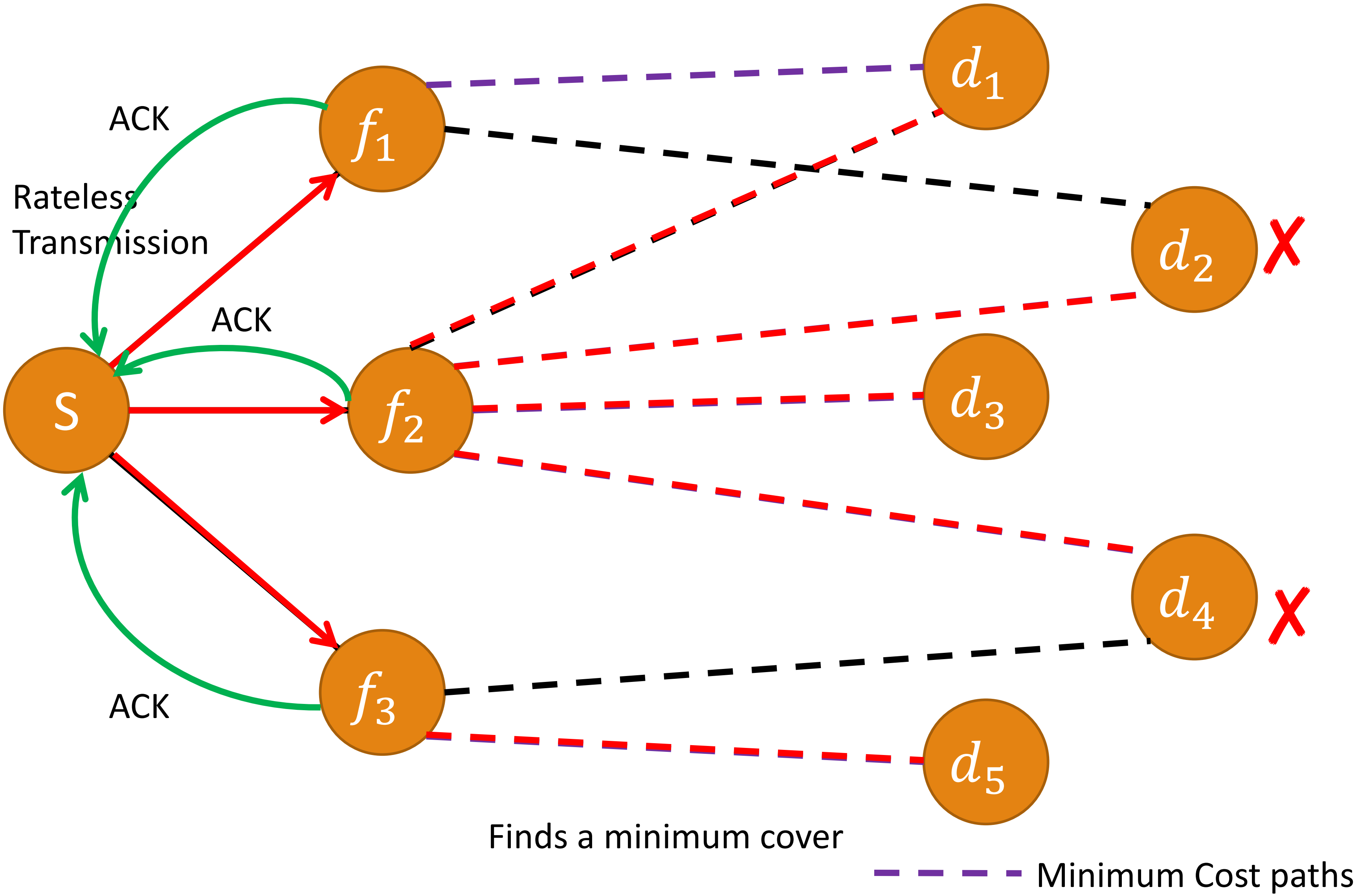
- A neighbour can act as a forwarder to a destination only if the cost from it to the destination is less than (1+α) times than the minimum of such costs from all the neighbours. This is defined as the Threshold Cost. (The value α is a parameter to be chosen appropriately.)
- Upon receiving ACK from a node, the source directs it to act as a forwarder to all destination nodes for which it can be a forwarder and start transmission immediately
- Transmission form source continues until forwarders are identified for all the destination nodes



ACK
Rateless Transmission
RACK
ACK
S
f_1
f_2
f_3
RACK
RACK
ACK
d_1
d_2
d_3
d_4
d_5
Minimum Cost paths

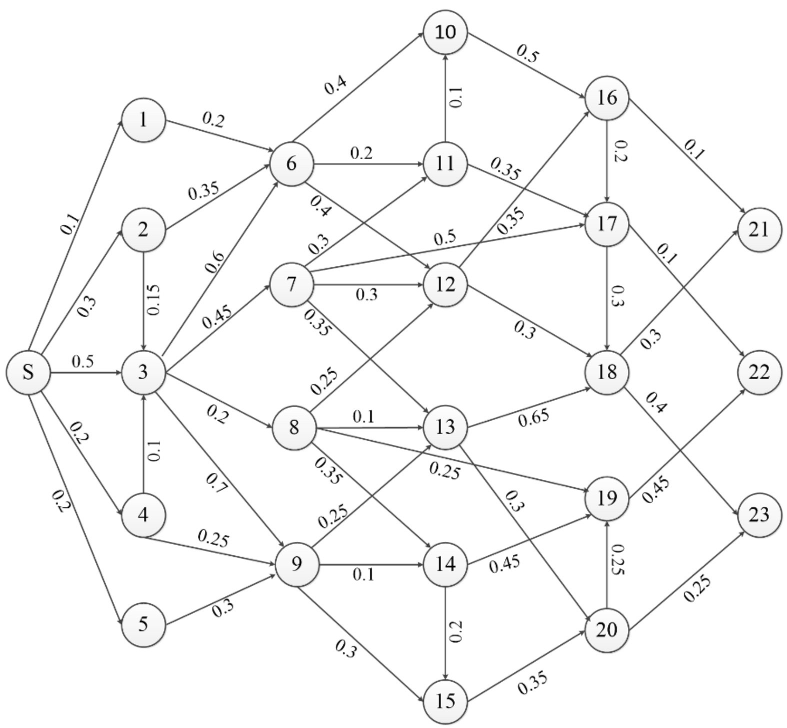
Minimum Forwarder Set-Routing Algorithm (*MFS-RA*)

- Emphasis on improving *MNT* performance. Tries to reduce the branching of the multicast tree at each hop.
- Potential forwarder set for each destination is constrained using parameter α as in *CFS-RA*.
- *S* transmits until enough neighbours have received the message to be able to reach all destinations.
- Then first selects the node which can reach maximum number of destinations as the forwarder to all those destinations.
- Repeats the process so as to get a minimum covering set of forwarders for all destinations.

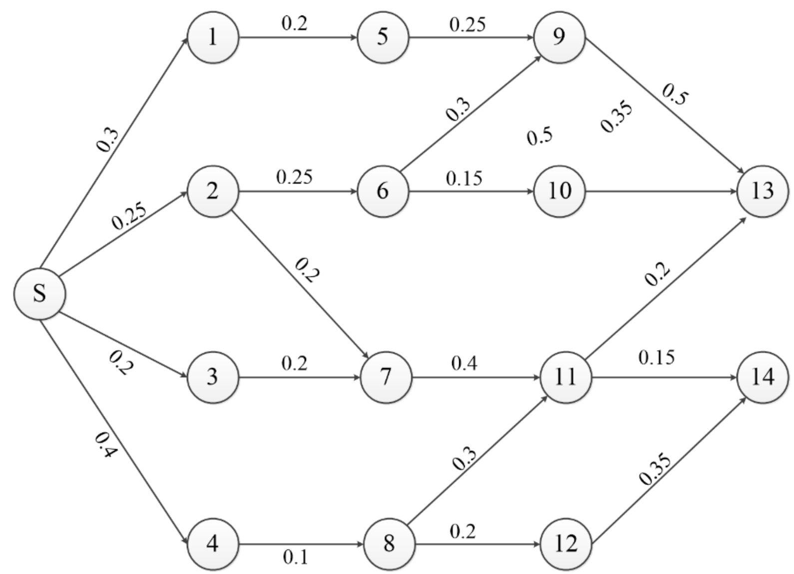

ACK
Rateless
Transmission
ACK
ACK
S
f_1
f_2
f_3
d_1
d_2
d_3
d_4
d_5
X
X
Finds a minimum cover
Minimum Cost paths

Simulation Results

The following two networks were used for carrying out simulations.



Simulation Network I



Simulation Network II

Importance of Parameter α

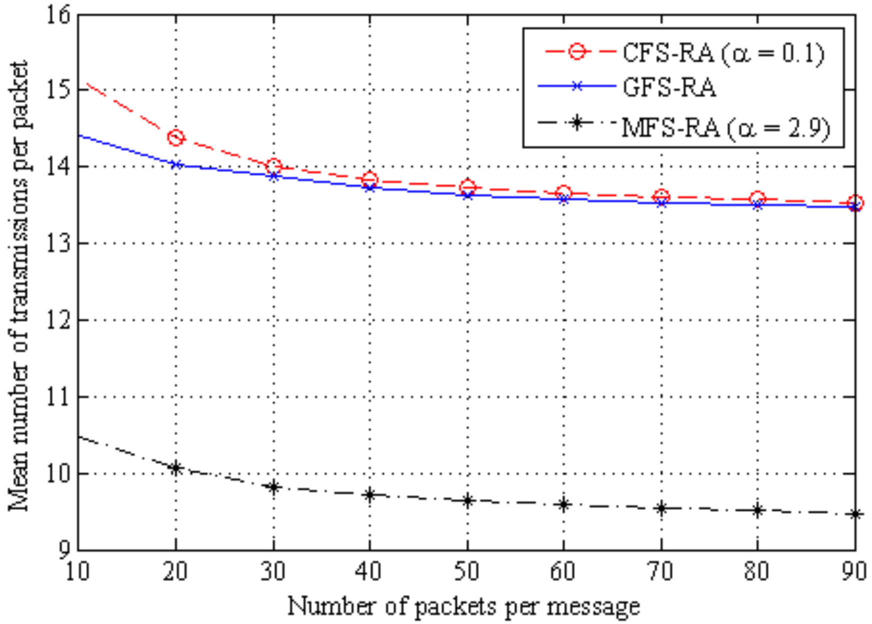
Destination Set	CFS-RA					MFS-RA				
	α	MTT	MNT	$\alpha = \infty$ MTT	$\alpha = \infty$ MNT	α	MTT	MNT	$\alpha = \infty$ MTT	$\alpha = \infty$ MNT
1,14,21	0.1	283.3	9.77	330.6	10.98	2.9	337.0	8.42	338.3	8.45
16,18,20	0.2	230.7	10.58	296.1	14.98	0.1	230.4	10.54	307.1	12.50
16,22,23	0.2	293.4	11.41	428.4	13.44	0.1	332.9	10.20	456.9	11.42
8,15,17,21	0.1	283.2	13.82	329.8	14.22	2.0	336.3	9.70	337.4	9.74
3,7,10,19,23	0.2	292.3	12.32	428.8	17.78	0.5	396.3	11.64	379.0	12.1

CFS-RA and MFS-RA performance for Network I. Results shown are for $\alpha=\infty$ (all forwarders acceptable!) and for the optimum value of α – i.e. α minimizing MTT for CFS-RA and α minimizing MNT for MFS-RA

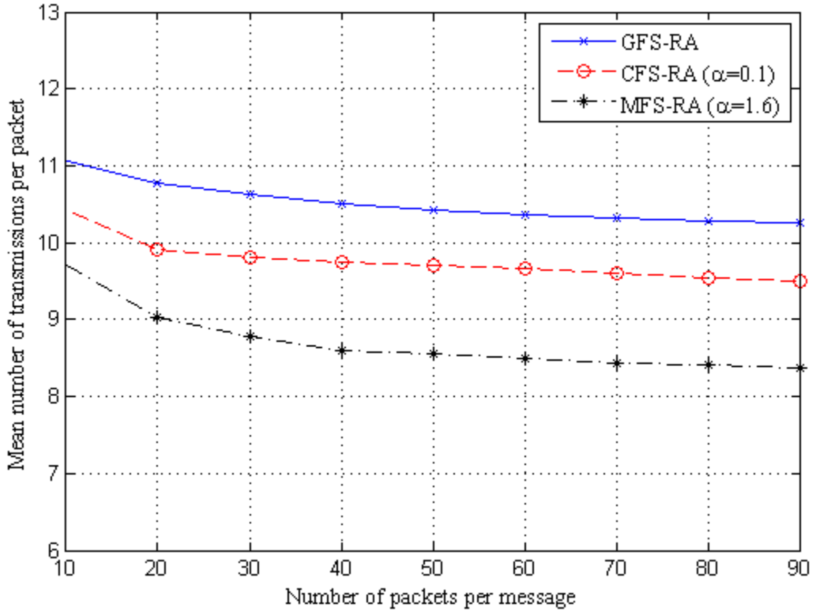
Appropriate Choice of α

- Optimal choice of α depends on the network, the source and the destination set.
- However, for *CFS-RA,* a small value of α (around 0.2) generally gives good performance.
- For *MFS-RA* , α can be low or high
- For subsequent simulations on network I, α=0.1 for *CFS-RA* and α=2.9 for *MFS-RA* were chosen while for Network II, α=0.1 for *CFS-RA* and α=1.6 for *MFS-RA* were chosen.

Mean Number of Transmissions per Packet (*MNT*) Performance

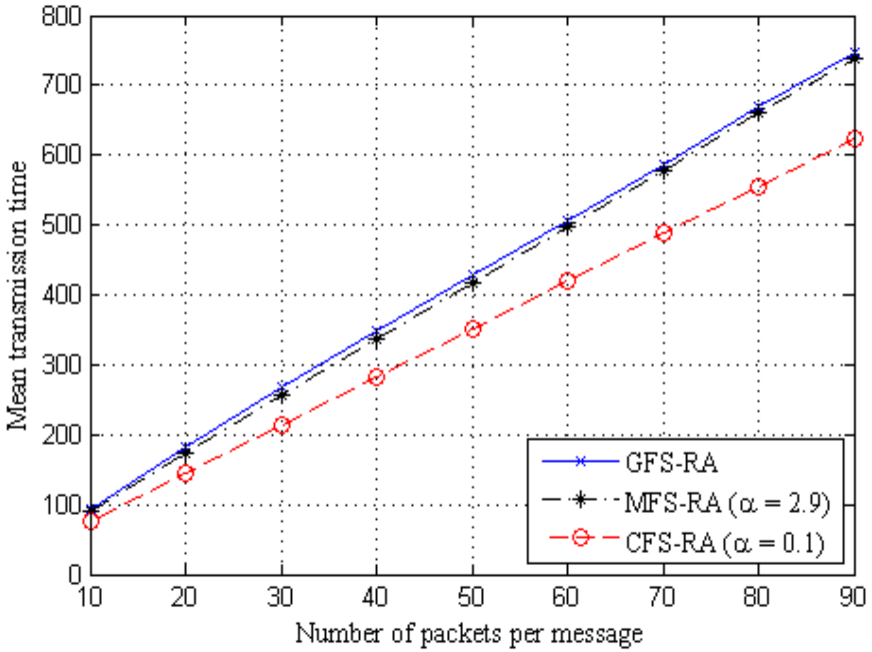


MNT performance for Network I

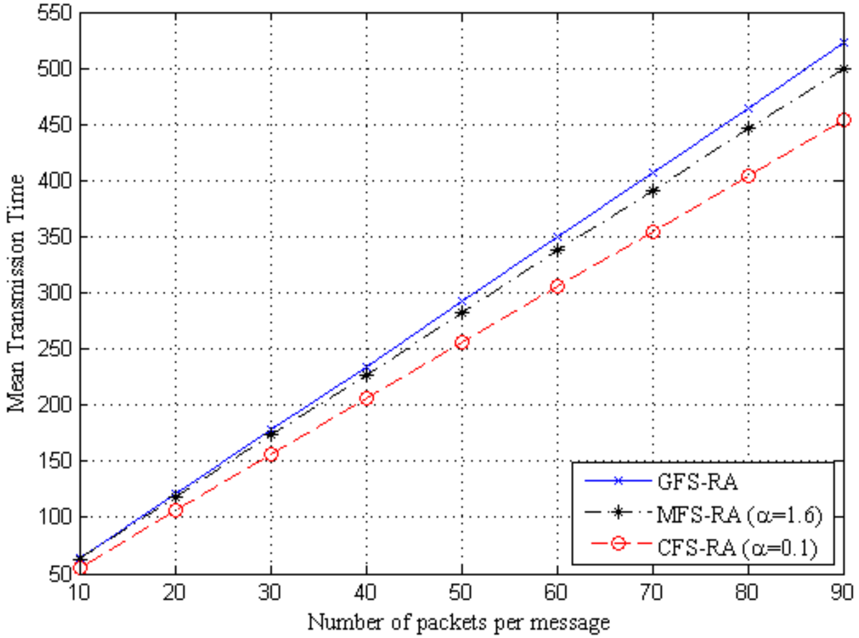


MNT performance for Network II

Mean Transmission Time (*MTT*) Performance



MTT performance for Network I



MTT performance for Network II

Conclusions

- Rateless coding and opportunistic routing together provide an effective framework for multicasting in wireless networks.
- For applications requiring low latency, *CFS-RA* is the recommended algorithm. However, it suffers from a relatively higher *MNT.*
- *MFS-RA* , by its superior *MNT* performance, is recommended for networks with limited resources like bandwidth and power.

Thank You!