An Adaptive Distance Vector Routing Algorithm for Mobile, Ad Hoc Networks

IEEE INFOCOM 2001
Rajendra V. Boppana
Computer Science Division
The Univ. of Texas at San Antonio
Satyadeva P Konduru
Nexsi Corp.

Speaker: Hicks Chen

Outline

- Introduction
- Current routing algorithms for MANETs
- Adaptive distance vector
- Performance analysis
- Conclusions

Introduction

- There are two category of routing algorithms
  - *Proactive* routing algorithms
    - *Distance Vector routing, RIP, DSDV*
  - On-demand routing algorithms
    - AODV, DSR
- A combination of proactive and on-demand routing algorithms
  - *Adaptive Distance Vector (ADV)*

Introduction

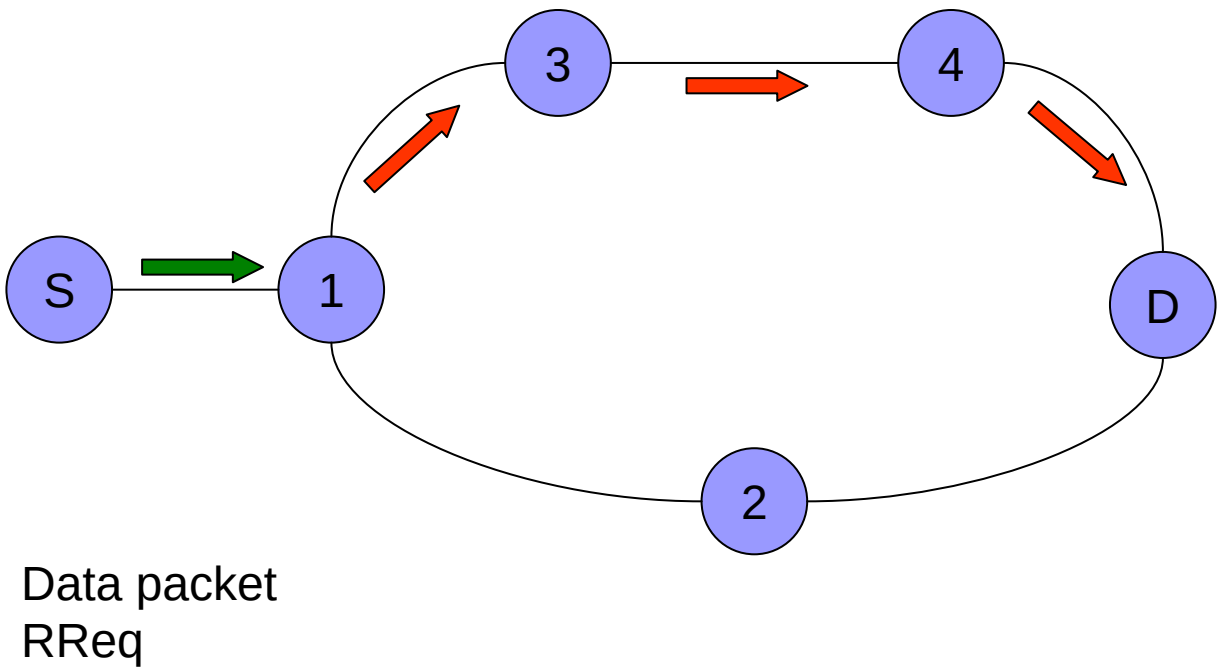
- This paper compare ADV with two on-demand protocols
  - Ad hoc On-demand Distance Vector (AODV)
  - Dynamic Source Routing (DSR)
- And a *Distance Vector* protocols
  - *Destination-Sequenced Distance Vector (DSDV)*

Current routing algorithms for MANETs

- *Distance Vector Routing*
  - The Routing Information Protocol (RIP) is an example of distance vector (DV) routing
    - Counting-to-infinity problem
  - *Destination-Sequenced Distance Vector (DSDV)*
    - DSDV solves counting-to-infinity problem in DV routing by attaching sequence numbers to routing entries.

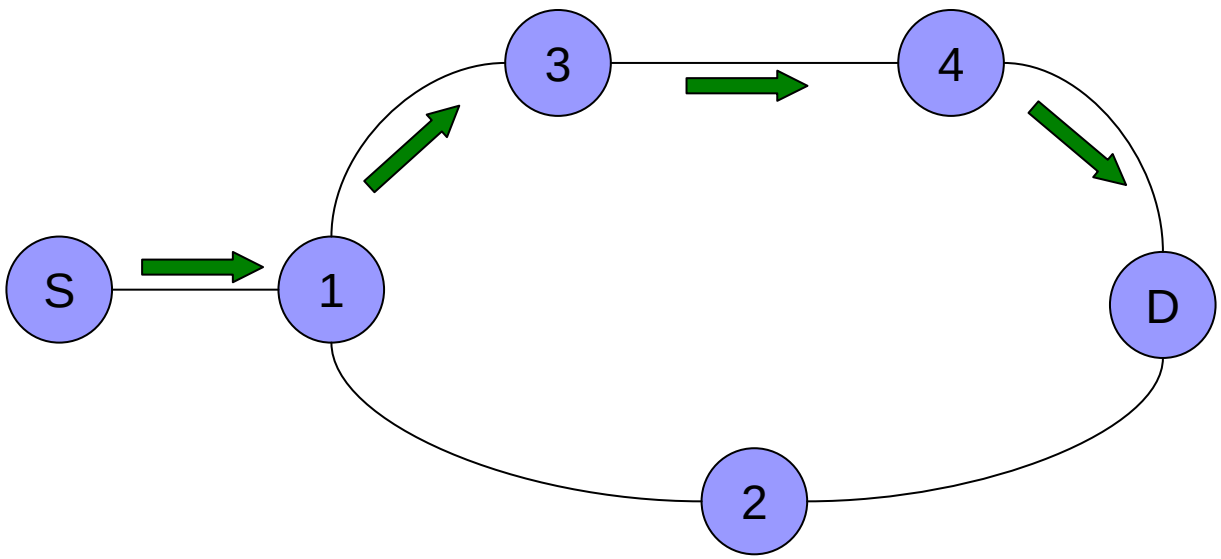
Current routing algorithms for MANETs

- *On-demand Routing Algorithms*
  - *Ad hoc On-demand Distance Vector –AODV*



Current routing algorithms for MANETs

- Dynamic Source Routing - DSR



Adaptive distance vector

- The Adaptive Distance Vector (ADV) starts with *DSDV*
- Using the following methods to reduce the routing overhead
  - *Varying the number of active routes maintained*
  - *Varying the frequency of routing updates*

Adaptive distance vector

- *Varying the number of active routes maintained*
  - ADV advertises and maintains routes for active receivers only
  - A node is an active receiver if it is the receiver of any currently active connection.

Adaptive distance vector

- At the beginning of a new connection, the source floods with an *init-connection* control packet
- If destination node is not an active receiver already, broadcasting with a receiver alert packet.
- When a connection is to be closed, the source broadcasts network-wide an *end-connection* control packet
- If the destination node has no additional active connections, then it broadcasts a non-receiver alert control packet

Adaptive distance vector

- *Varying the frequency of routing updates*
  - A node should trigger an update under three conditions
    - if it has some buffered data packets due to lack of routes
    - if one or more of its neighbors make a request for fresh routes
    - it is a forwarding node that intends to advertise any fresh valid/invalid route to the destination so as to keep the route fresh.

Adaptive distance vector

- *Then check trigger meter*
  - If > TRGMETER_FULL, send full update
  - Else if > trigger threshold*, send partial update*
- If the number of neighbor changes exceeds a preset number, then the node categorizes the network as HIGH_SPEED or else as LOW_SPEED network.

Adaptive distance vector

- *Sending routing updates*
  - With every routing update entry, a node sends an expected response value of ZERO, LOW, MEDIUM or HIGH
  - The expected response values are determined using the following rules

Adaptive distance vector

- An expected response of HIGH is given when there are packets waiting for this route in the node buffers regardless of the speed of the network.
- In a HIGH SPEED network, an expected response of MEDIUM is given if this node is a *forwarding node* for the destination given in the routing entry.
- In a LOW SPEED network, an expected response of LOW is given if this node is a *forwarding node* for any of its neighbors to the routing entry's destination.
- If none of the above criteria apply then the expected response is set to ZERO.

Adaptive distance vector

- Processing received updates
- the trigger meter is incremented by TRGMETER_HIGH, TRGMETER_MED or TRGMETER_LOW for an expected response of HIGH, MEDIUM or LOW respectively.

Adaptive distance vector

- To mitigate the effect of the periodic transmission of the updates, we have proposed some adaptive criteria that trigger routing updates based only on network load and mobility conditions.
- The overhead is reduced by varying the size and the frequency of the routing updates dynamically.
- We have shown using simulations that ADV outperforms on-demand protocols like AODV and DSR in many instances.

Performance analysis

- *ns-2* simulator with the CMU extensions
- CMU extensions include detailed implementations of IEEE 802.11 wireless LAN and ad hoc routing protocols DSDV, AODV, and DSR

Performance analysis

- *Mobility models*
  1. 50 nodes randomly placed on a 1000m x 1000m field
  2. 100 nodes placed on a 2200m x 600m rectangular field
  - The nodes randomly choose a direction and speed and travel for a certain distance

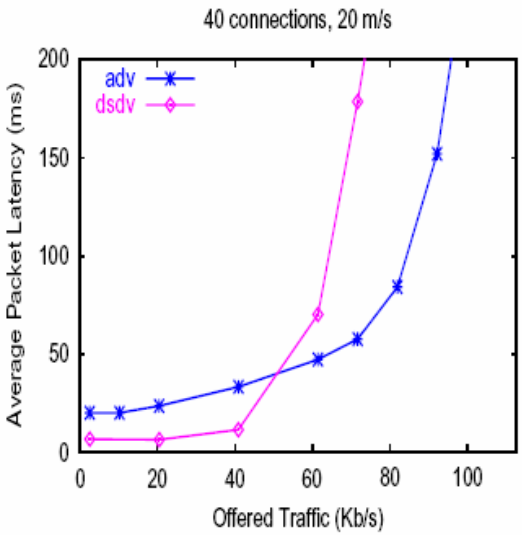
Performance analysis

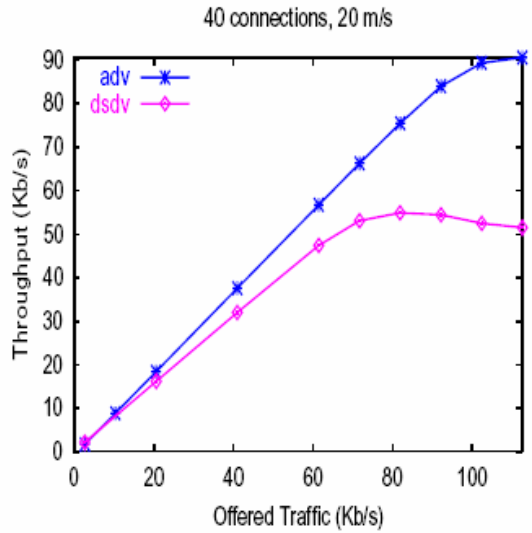
- *Traffic load*
  - CBR of 40 connections
- Performance metrics
  - average data packet latency
  - network throughput
  - percentage of data packets delivered
  - overheads of routing algorithms
    - IP layer
    - MAC layer

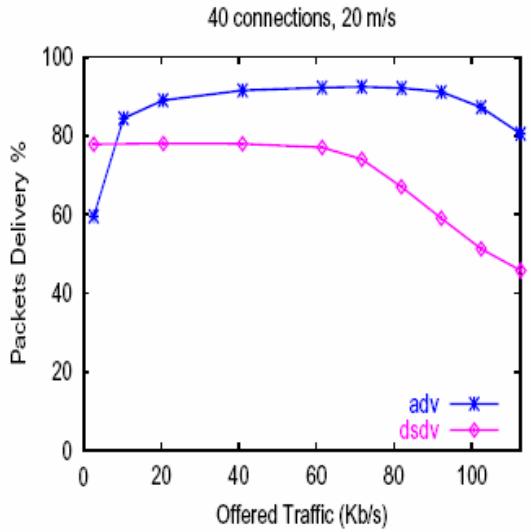
Performance analysis

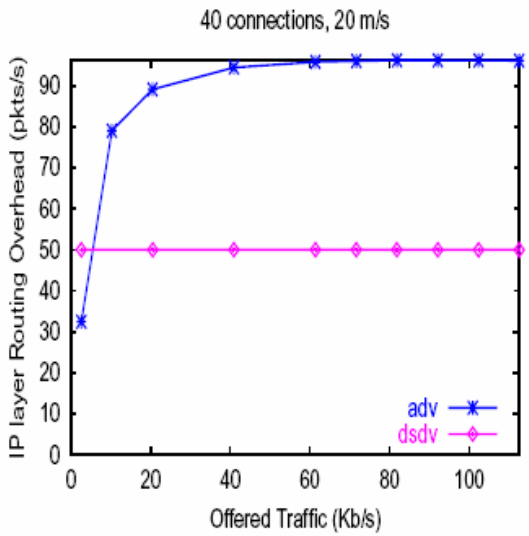
- *ADV vs. DSDV*
- *ADV vs. On-demand algorithms*
  - Steady-state behavior
  - Transient state behavior
    - 10 new connections are initiated every 60 seconds

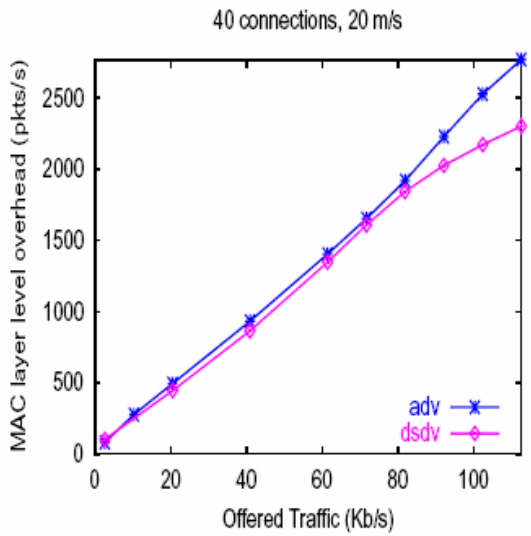
ADV vs. DSDV


40 connections, 20 m/s
Average Packet Latency (ms)
Offered Traffic (Kb/s)
adv
dsdv

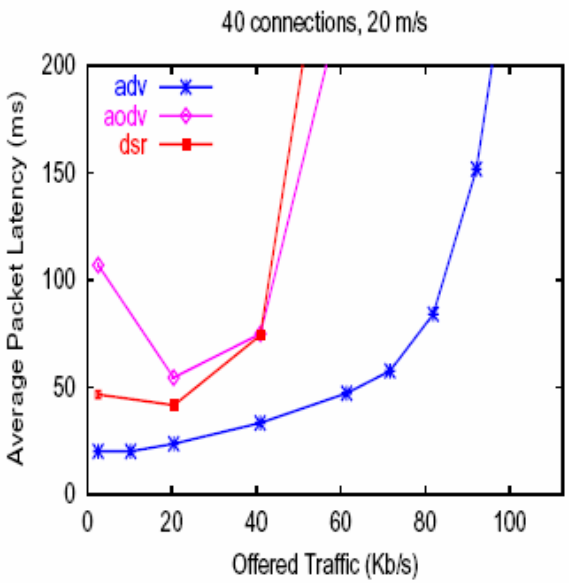

40 connections, 20 m/s
Throughput (Kb/s)
Offered Traffic (Kb/s)
adv
dsdv

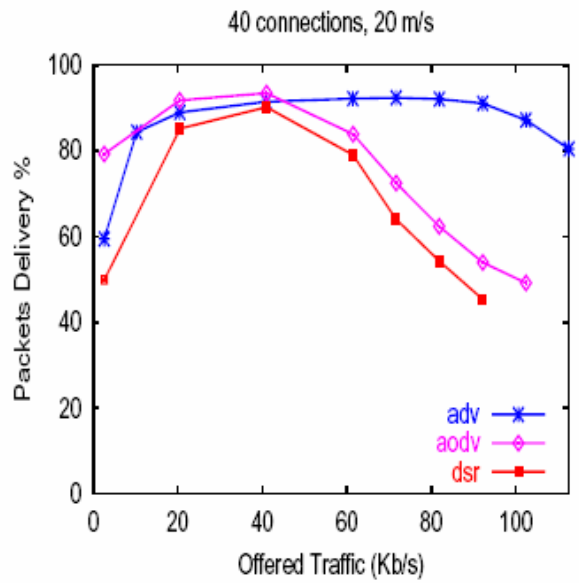

40 connections, 20 m/s
Packets Delivery %
Offered Traffic (Kb/s)
adv
dsdv

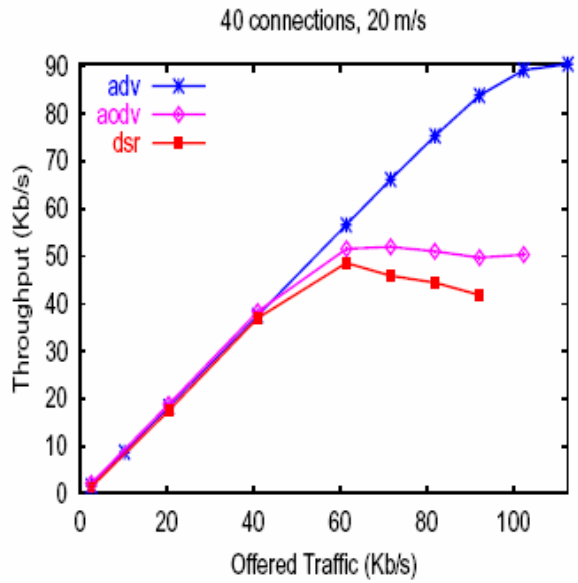

40 connections, 20 m/s
IP layer Routing Overhead (pkts/s)
Offered Traffic (Kb/s)
adv
dsdv

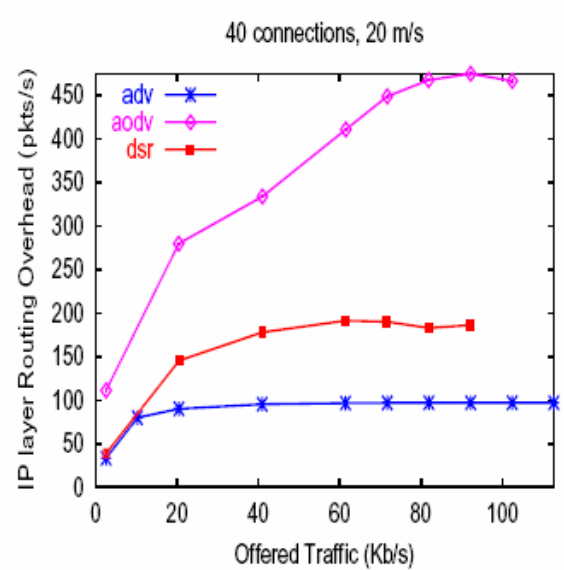

40 connections, 20 m/s
MAC layer level overhead (pkts/s)
Offered Traffic (Kb/s)
adv
dsdv

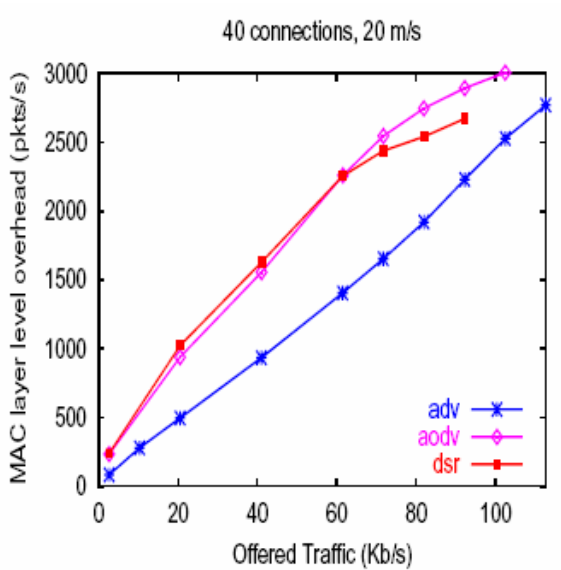
Steady-state behavior



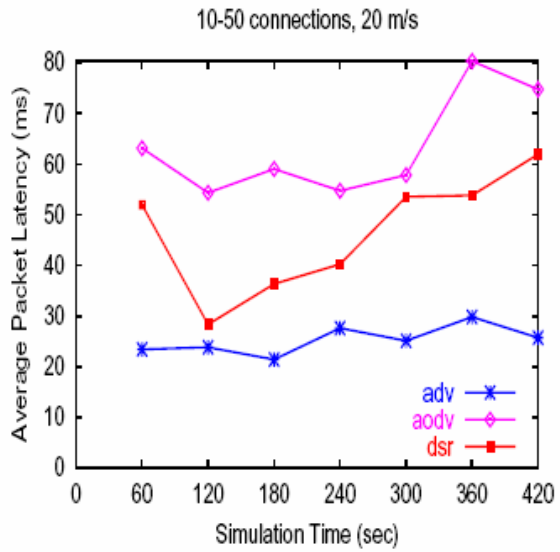


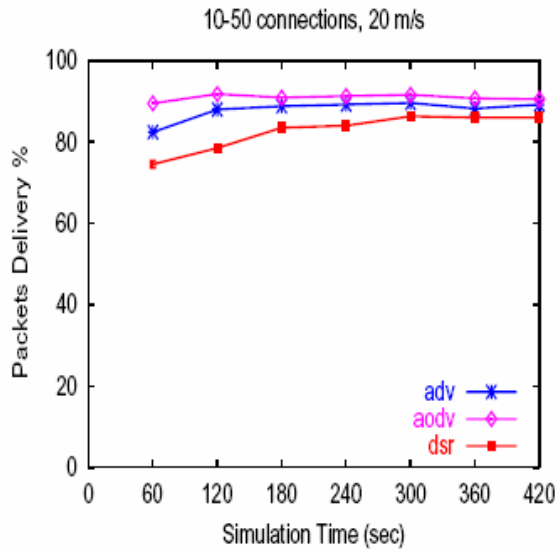


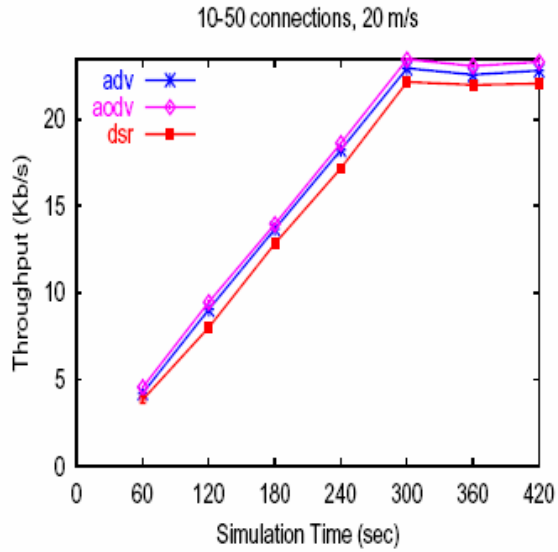


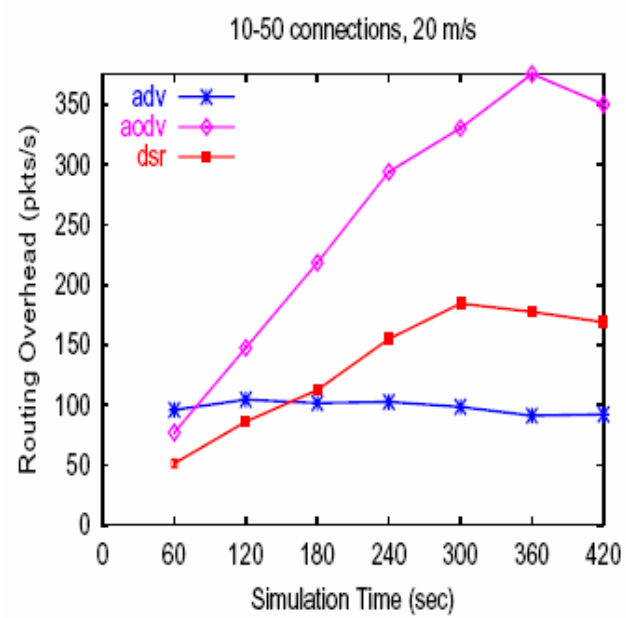


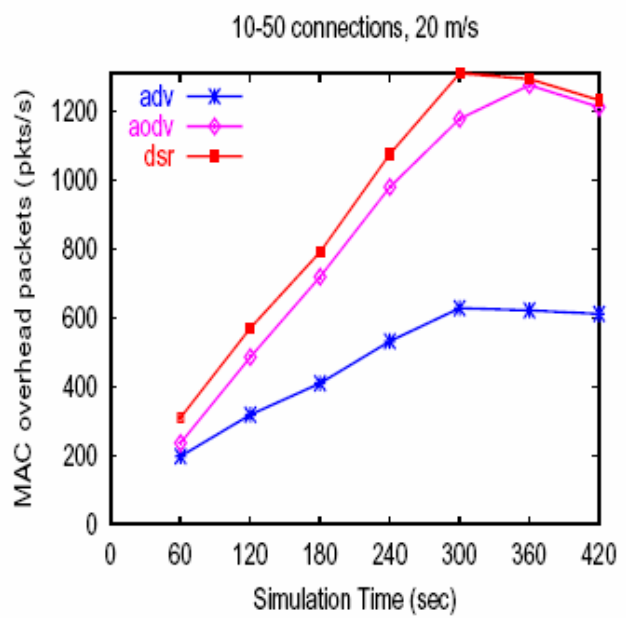
Transient state behavior



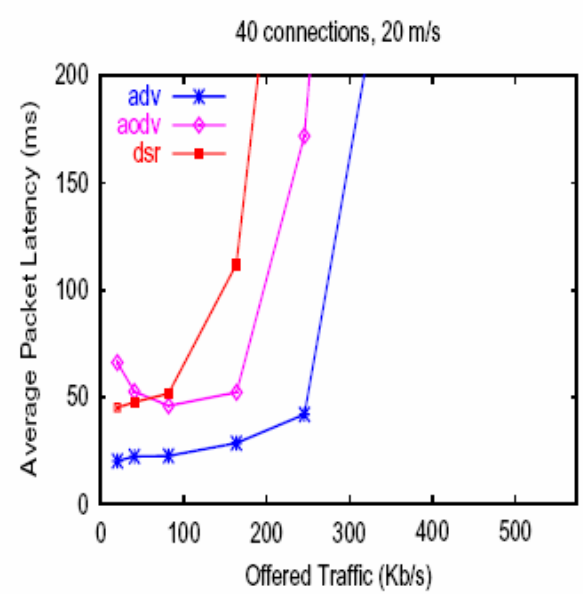


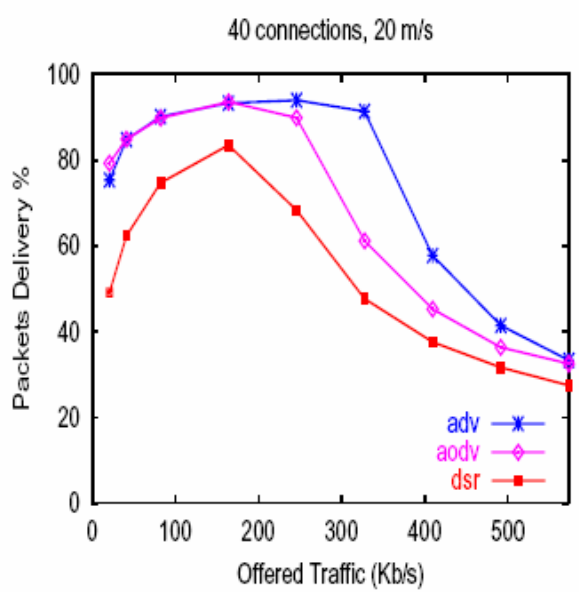


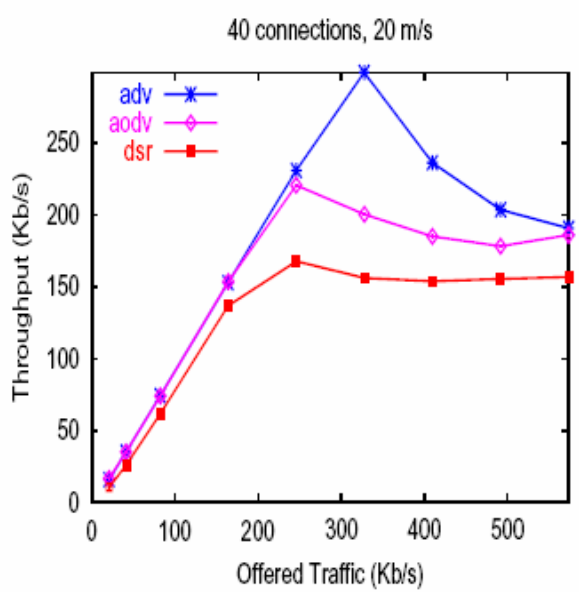


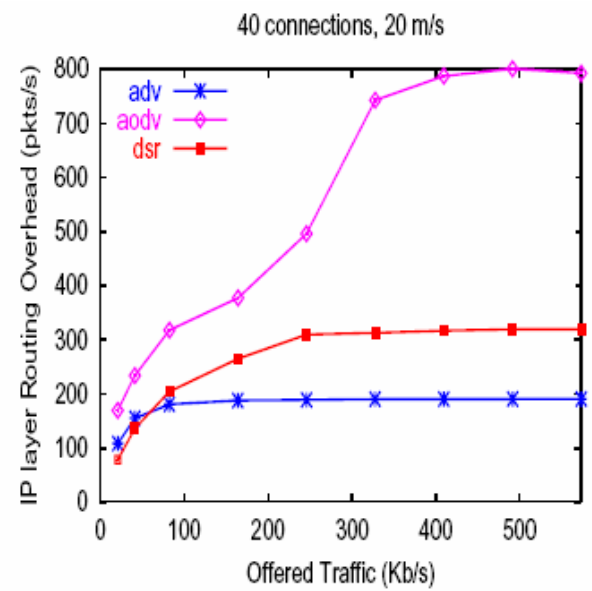


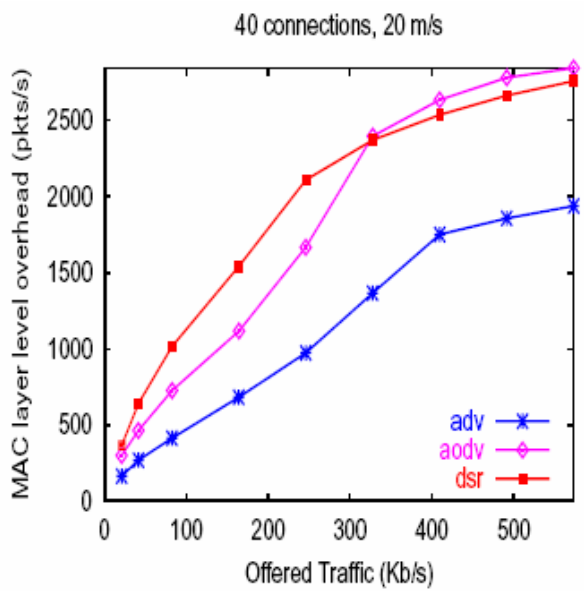
Steady-state behavior with 100 node











Conclusions