Integrating DTN and MANET Routing

2006-11-09 DTNRG – IETF 67
(slides from ACM CHANTS workshop)

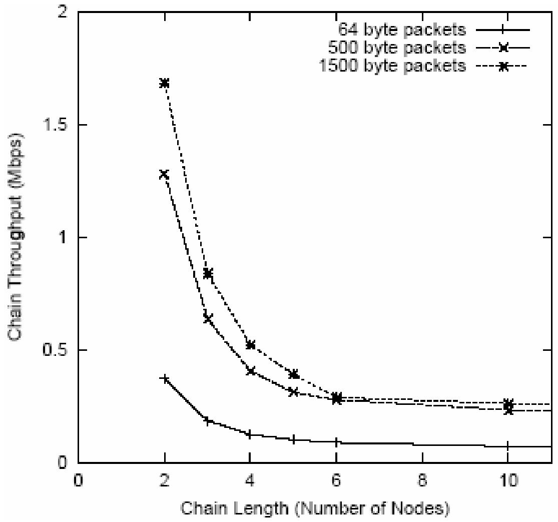
Jörg Ott jo@netlab.tkk.fi
Dirk Kutscher dku@tzi.org
Christoph Dwertmann nermal@tzi.org

Motivation: Ad-hoc Networking for Human Users

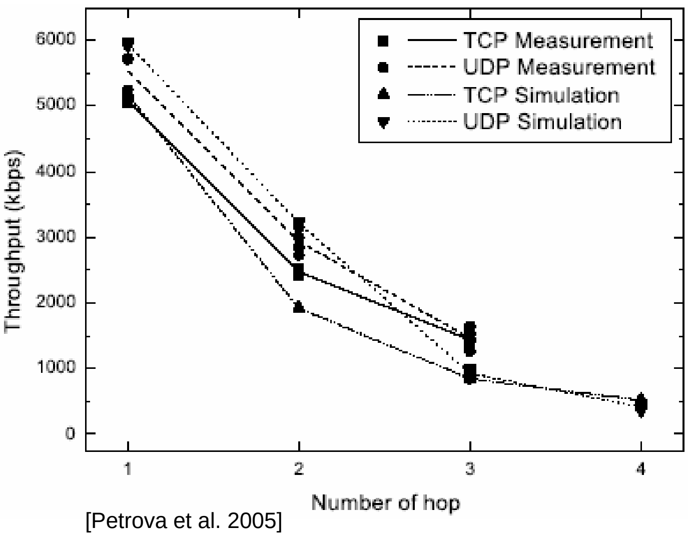
- Target field: interpersonal communication and Internet access
- Node density in environments without controlled deployment will often be sparse
  - Heterogeneity of consumer devices: different brands, versions, capabilities
  - Also: users may not want to cooperate
  - An end-to-end path may exist—but often will not
  - Calls for asynchronous communications: DTN
- User experience benefits from immediate end-to-end interactions
  - Users have grown expectations over time
  - Application protocols designed in this way anyway
  - Anything but end-to-end may only be second best
  - Yet better than not communicating at all

Motivation: Limitations in End-to-End Performance

- Communication performance degrades with increasing number of wireless hops in ad-hoc networks



[Li et al. 2001]



[Petrova et al. 2005]

- Path stability decreases with distance depending on mobility

Approach: Integrating DTN and AODV

- Let the application (and ultimately the user) decide
- Provide an interface to learn about the ad-hoc environment
  - Rather than completely abstracting from (i.e., hiding) it

- Combine AODV route search with DTN node location
  - Return available end-to-end route + path length
  - Return available DTN nodes in the vicinity + path lengths

- Application then chooses whether to use synchronous end-to-end or asynchronous hop-by-hop (DTN) communication
  - Example: Simply use DTN as fallback if end-to-end fails
  - Example: Path length, expected performance and stability vs. data volume

Approach: Integrating DTN and AODV

- Let the application (and ultimately the user) decide
- Provide an interface to learn about the ad-hoc environment
  - Ra
- Com
  - R
  - R
- Appl -end or as
  - Example: Simply use DTN as fallback if end-to-end fails
  - Example: Path length, expected performance and stability vs. data volume

<u>Applicability Statement:</u>

No, this does not work with existing applications

<u>But:</u>

Existing applications don’t work well with mobility anyway.

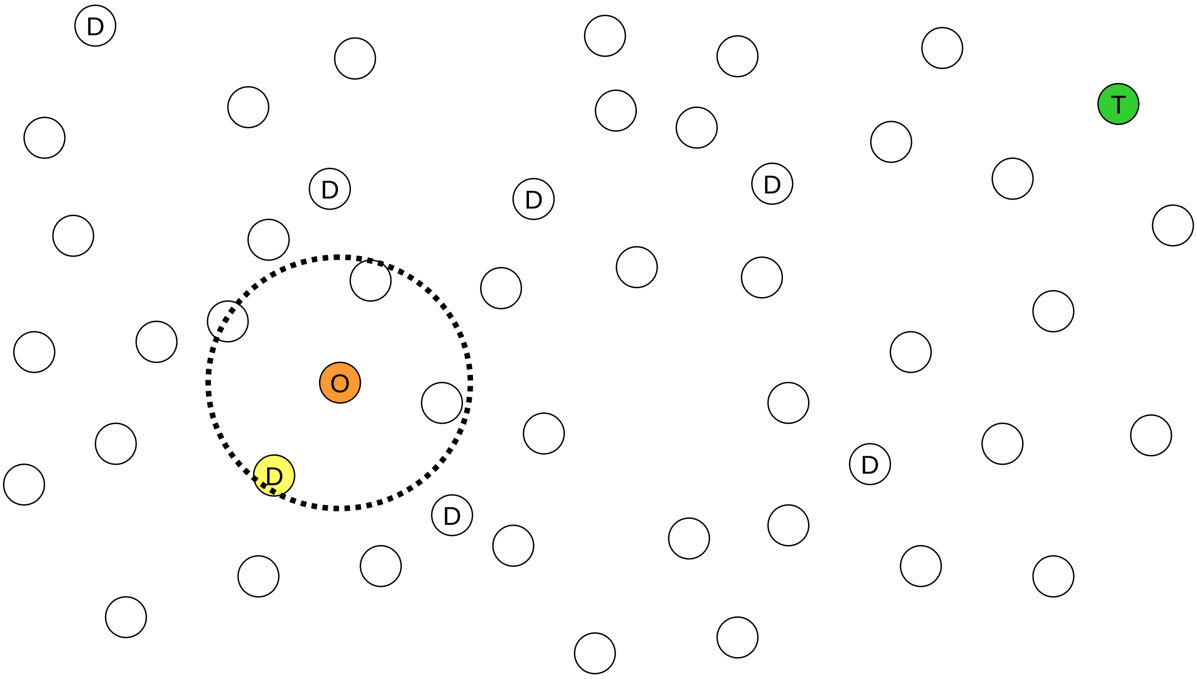
<u>Solutions:</u>

Short-term: use proxies as interim means for migration.

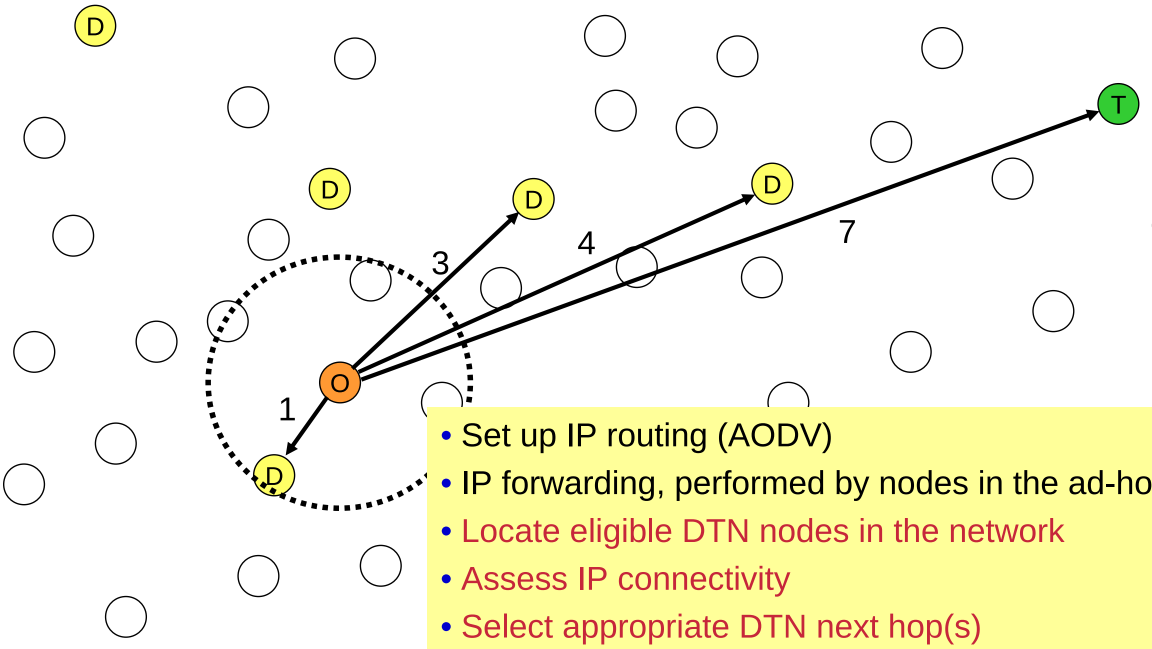
Long-term: design application protocols in a more suitable way.

Special Application: DTN routers

Required Functions



Required Functions



- Set up IP routing (AODV)
- IP forwarding, performed by nodes in the ad-hoc network
- Locate eligible DTN nodes in the network
- Assess IP connectivity
- Select appropriate DTN next hop(s)
- General DTN routing
- DTN forwarding

AODV Extensions

- Additional attributes in RREQ and RREP packets
  - Indicate extension support
  - Report DTN node contact info
- Optional
  - Convey source, target EID
  - Include routing metrics
- Modified processing and timer handling rules
- Workable with plain AODV nodes

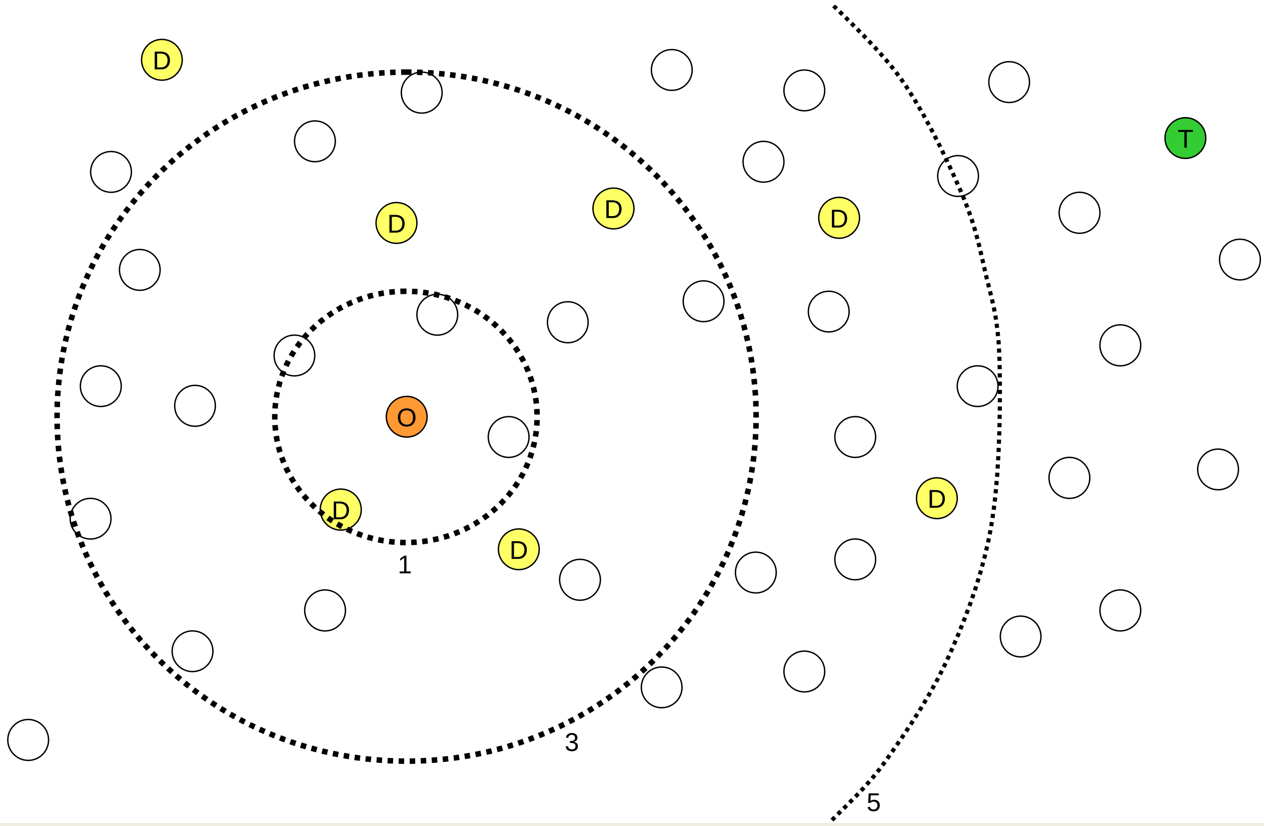
a) DTN router info + optional metric

<table>
<tr><th colspan="2">0 7</th><th colspan="2">8 15</th><th colspan="2">16 31</th></tr>
<tr><td colspan="2">Type = 29</td><td colspan="2">Length</td><td colspan="2">DTN router port</td></tr>
<tr><td colspan="6">DTN router IP address</td></tr>
<tr><td>T</td><td>Hop count</td><td colspan="2">Option EID</td><td>EID len</td><td></td></tr>
<tr><td colspan="6">DTN router EID</td></tr>
<tr><td colspan="2">Option Metric</td><td>D</td><td>Metric len</td><td colspan="2"></td></tr>
<tr><td colspan="6">DTN Routing Metric</td></tr>
</table>

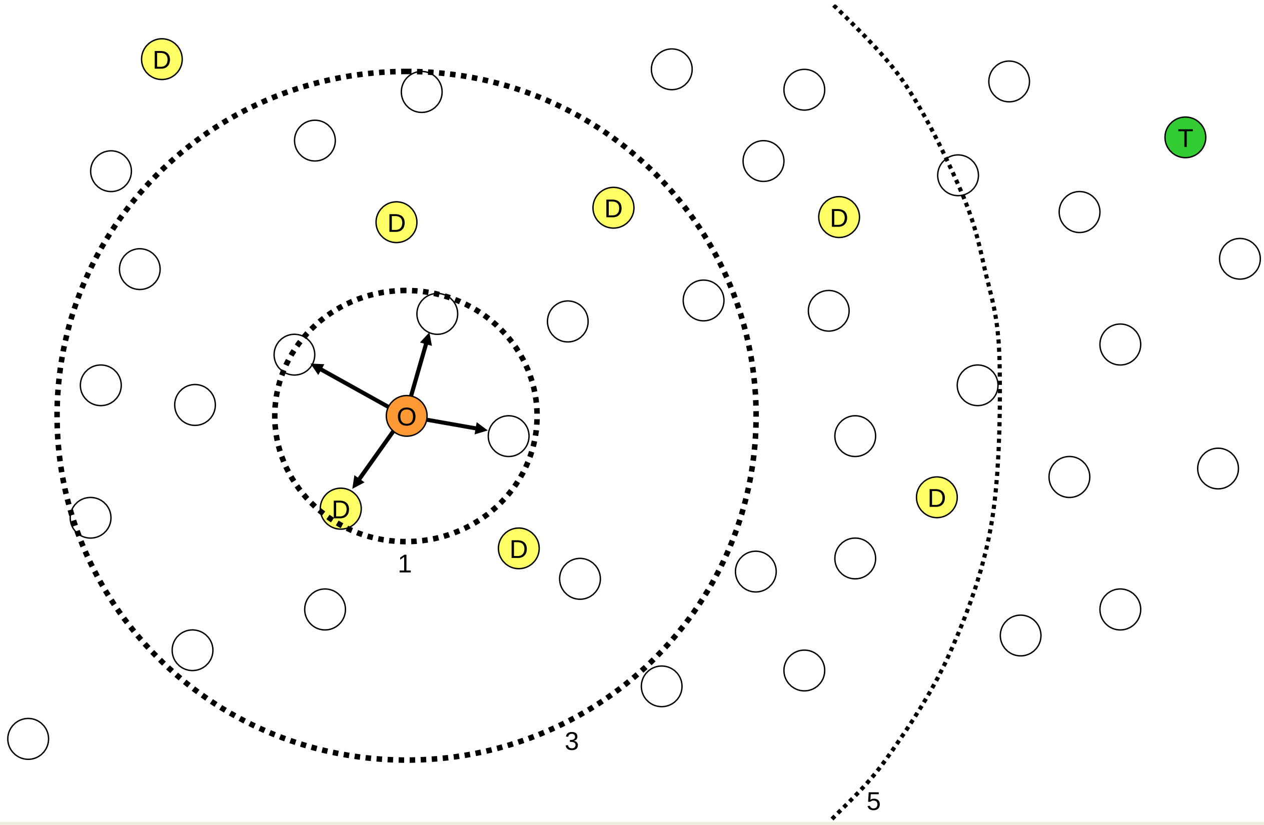
b) Optional DTN EID

<table>
<tr><th>0 7</th><th>8 15</th><th>16 31</th></tr>
<tr><td>Type = 30/31</td><td>Length</td><td></td></tr>
<tr><td colspan="3">DTN source or target EID</td></tr>
</table>

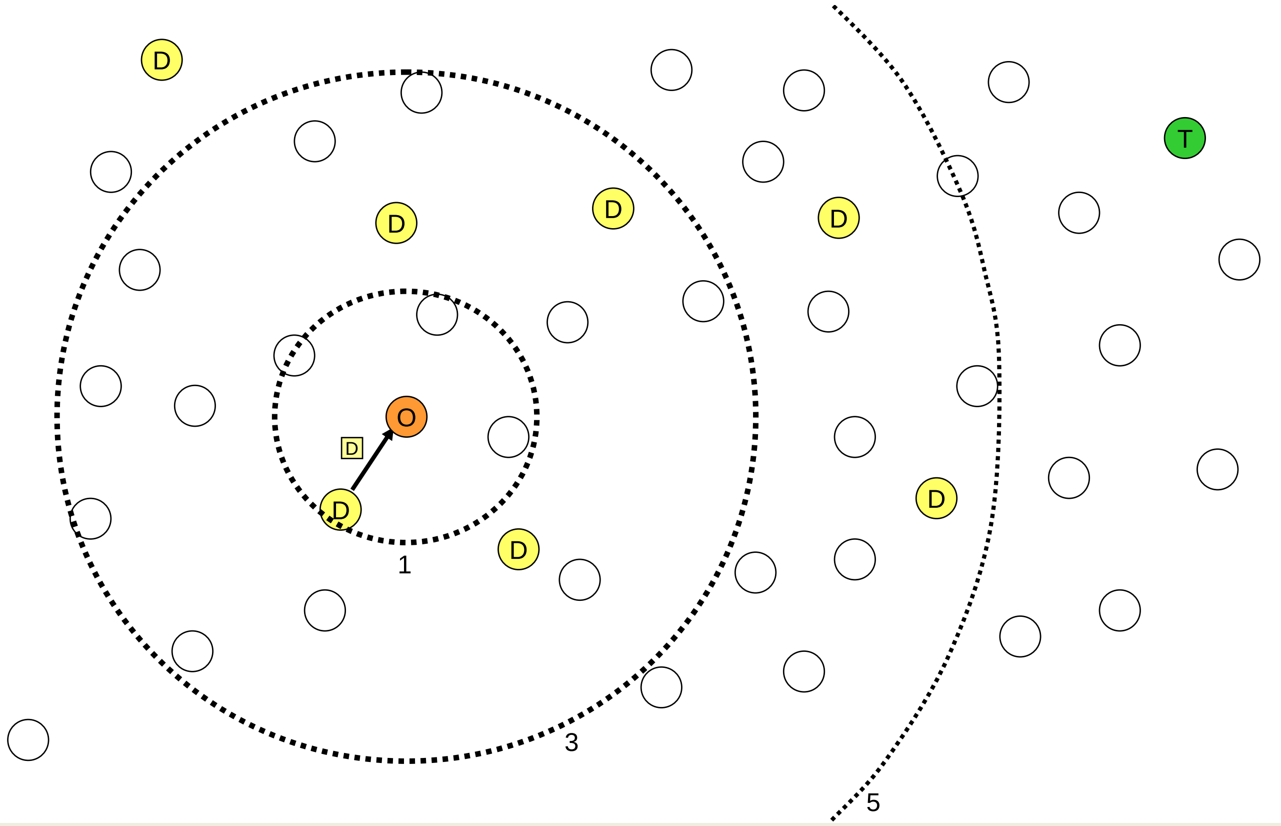
Extended AODV Operation



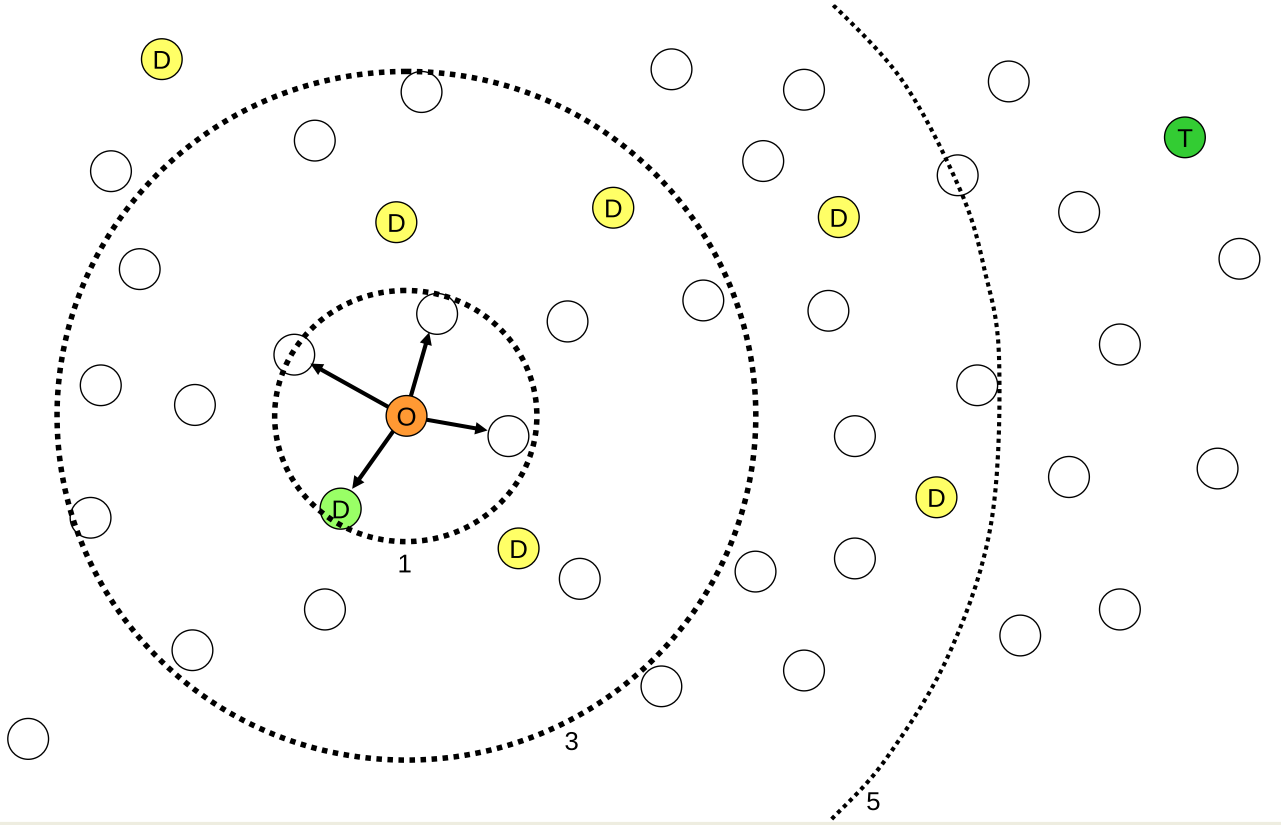
Extended AODV Operation (TTL 1)



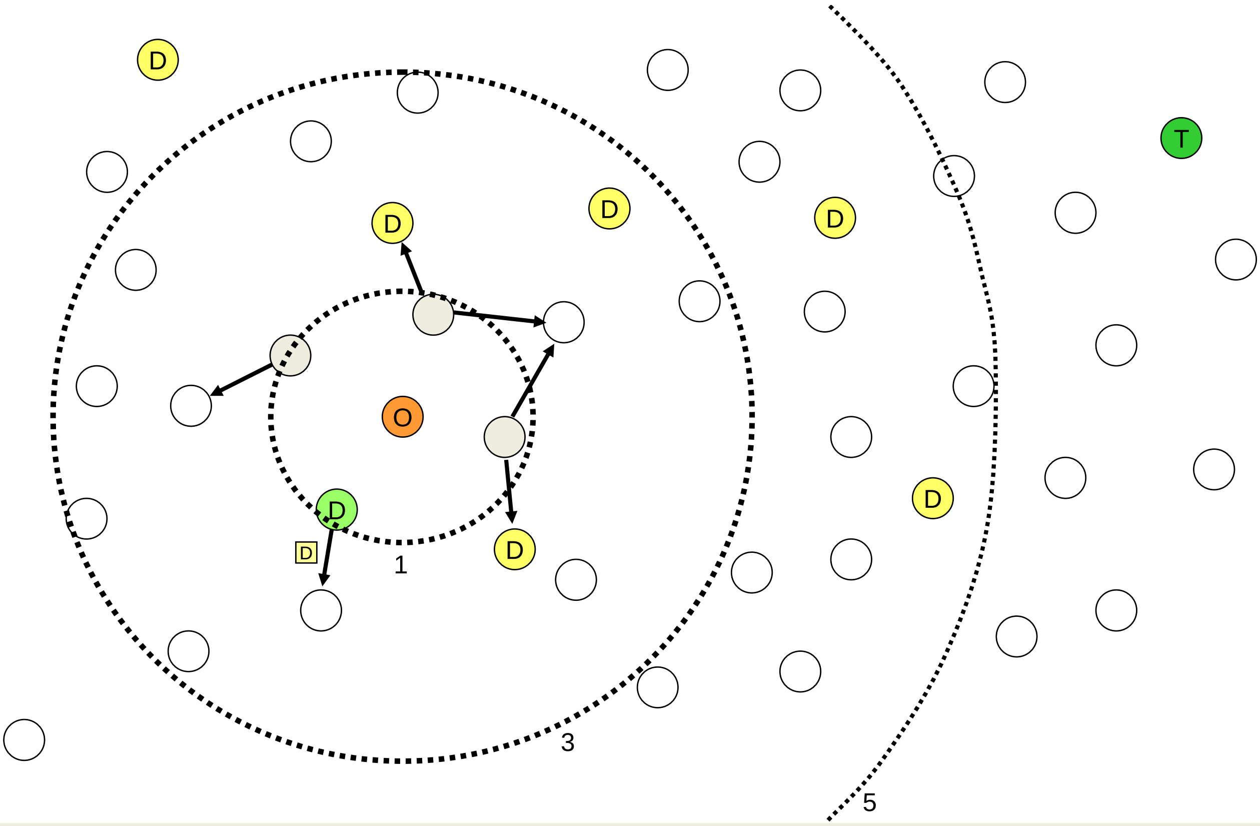
Extended AODV Operation (TTL 1)



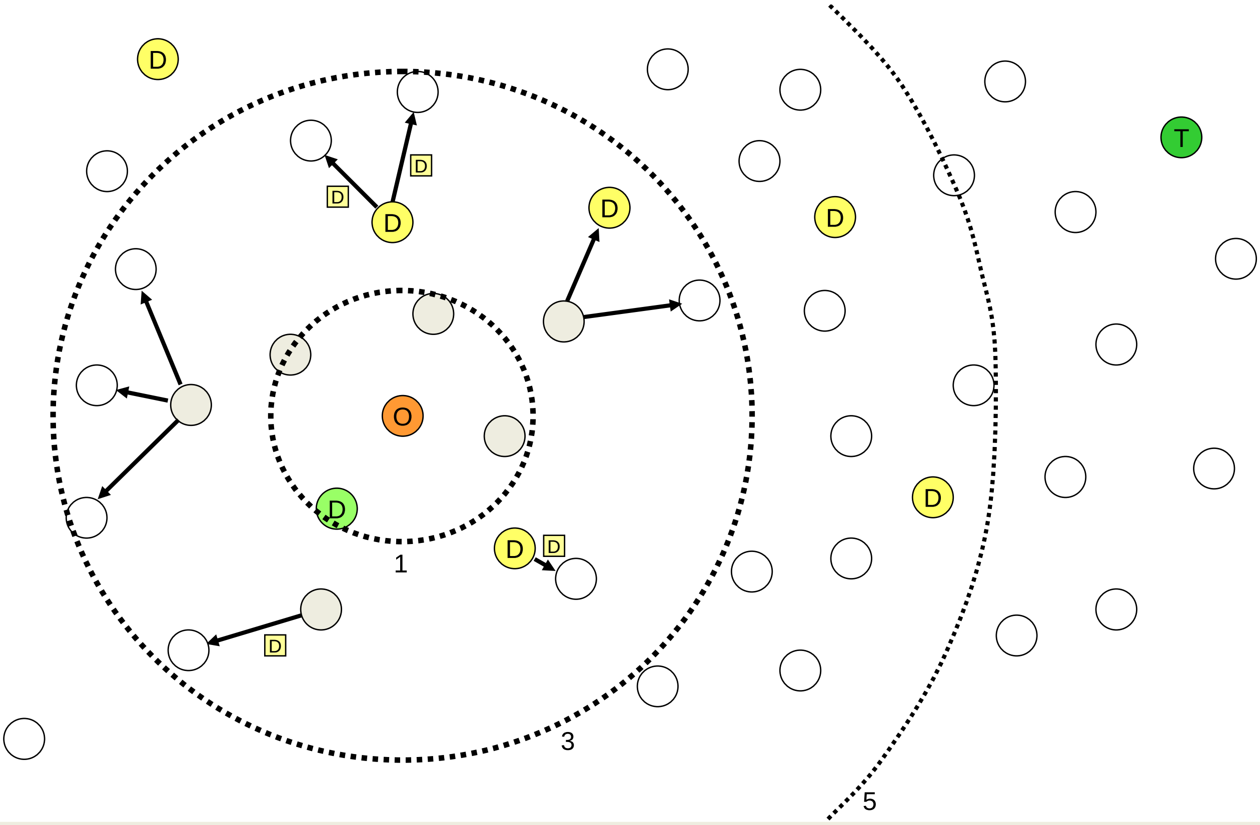
Extended AODV Operation (TTL 3)



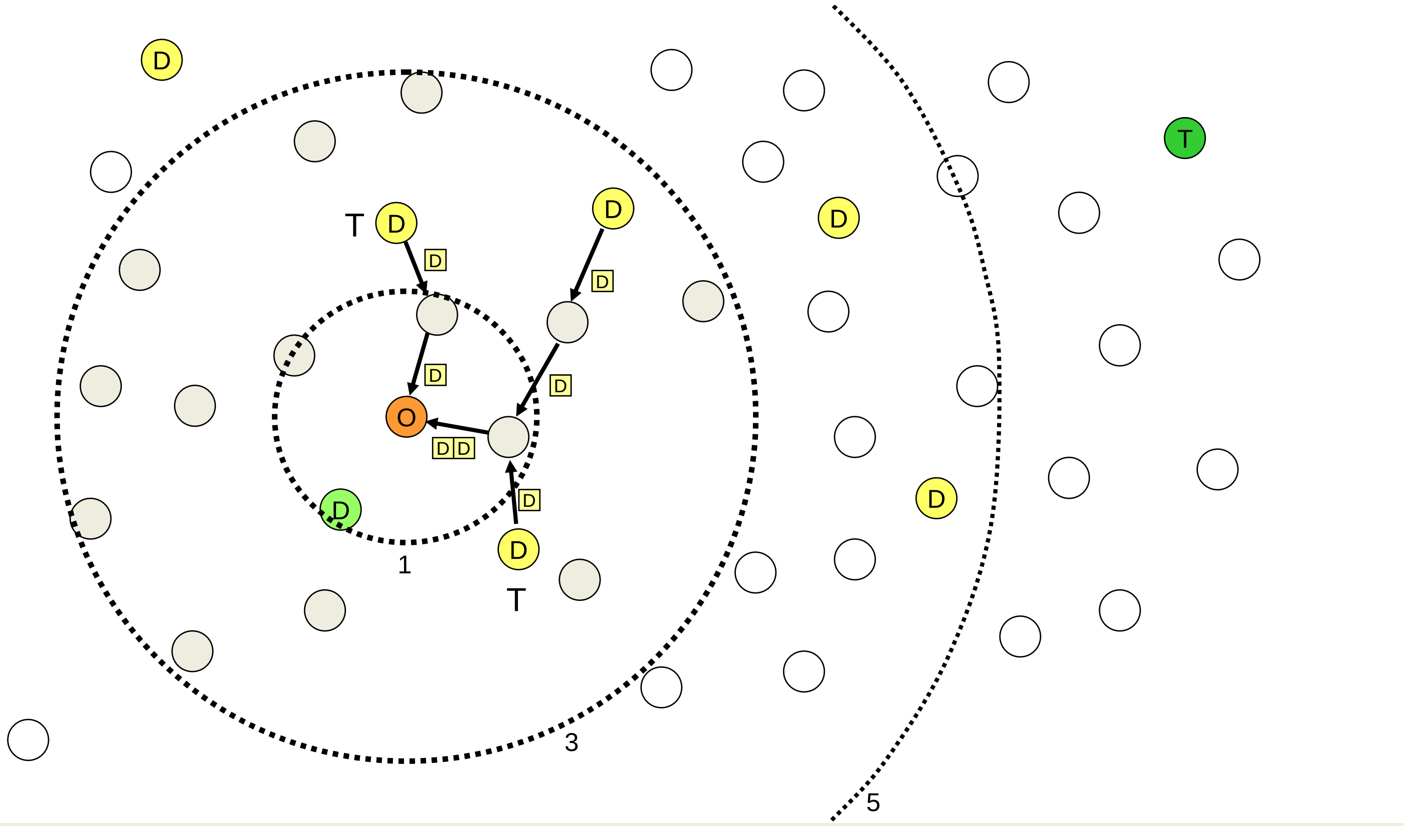
Extended AODV Operation (TTL 3)



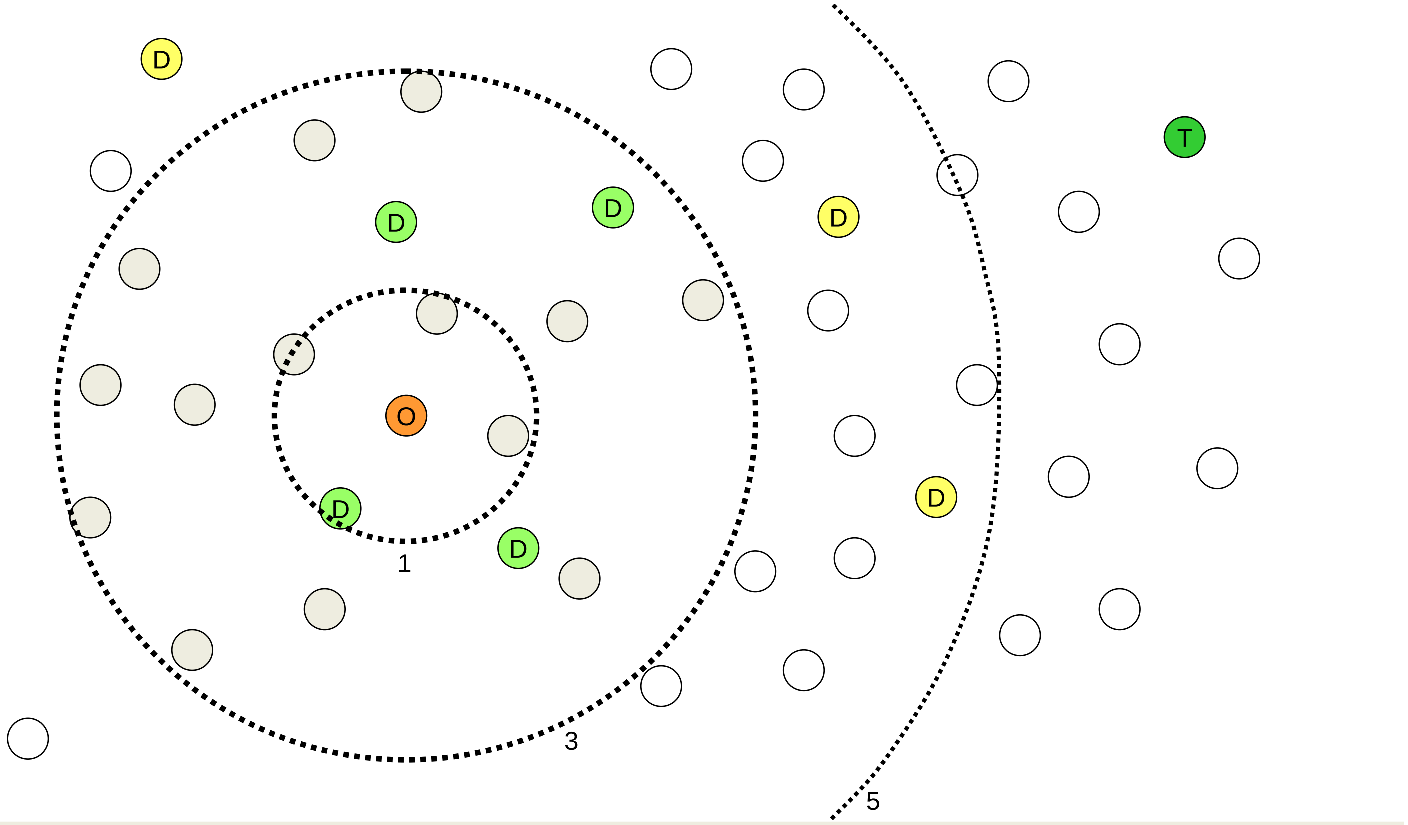
Extended AODV Operation (TTL 3)



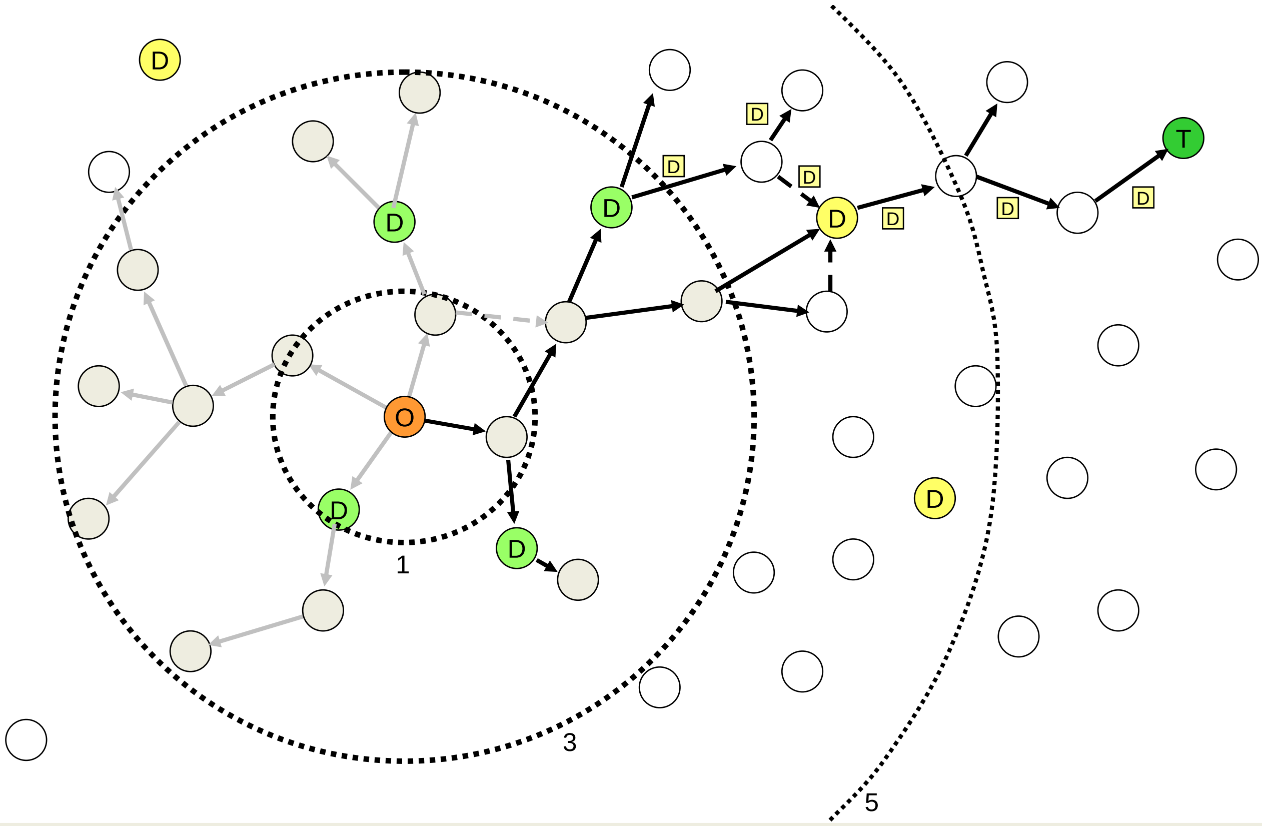
Extended AODV Operation (TTL 3)



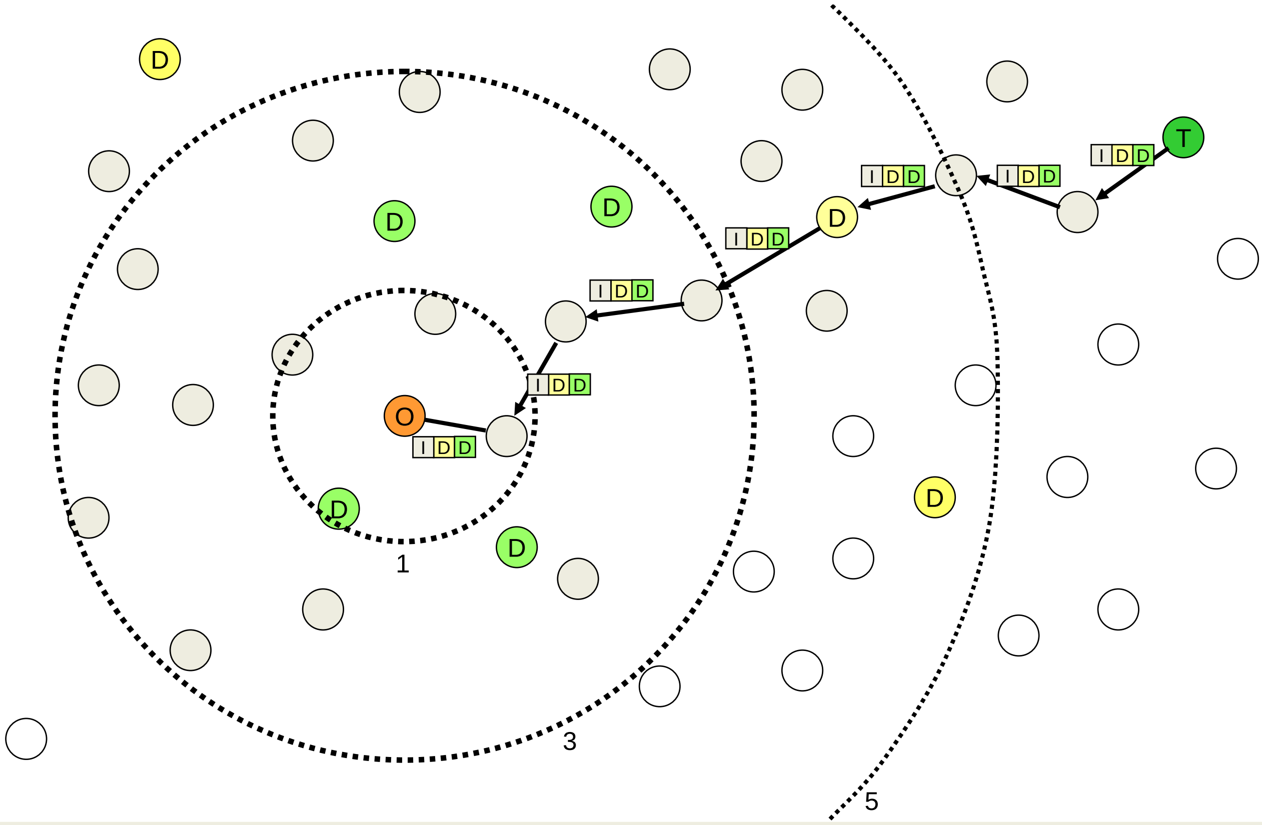
Extended AODV Operation (TTL 3)



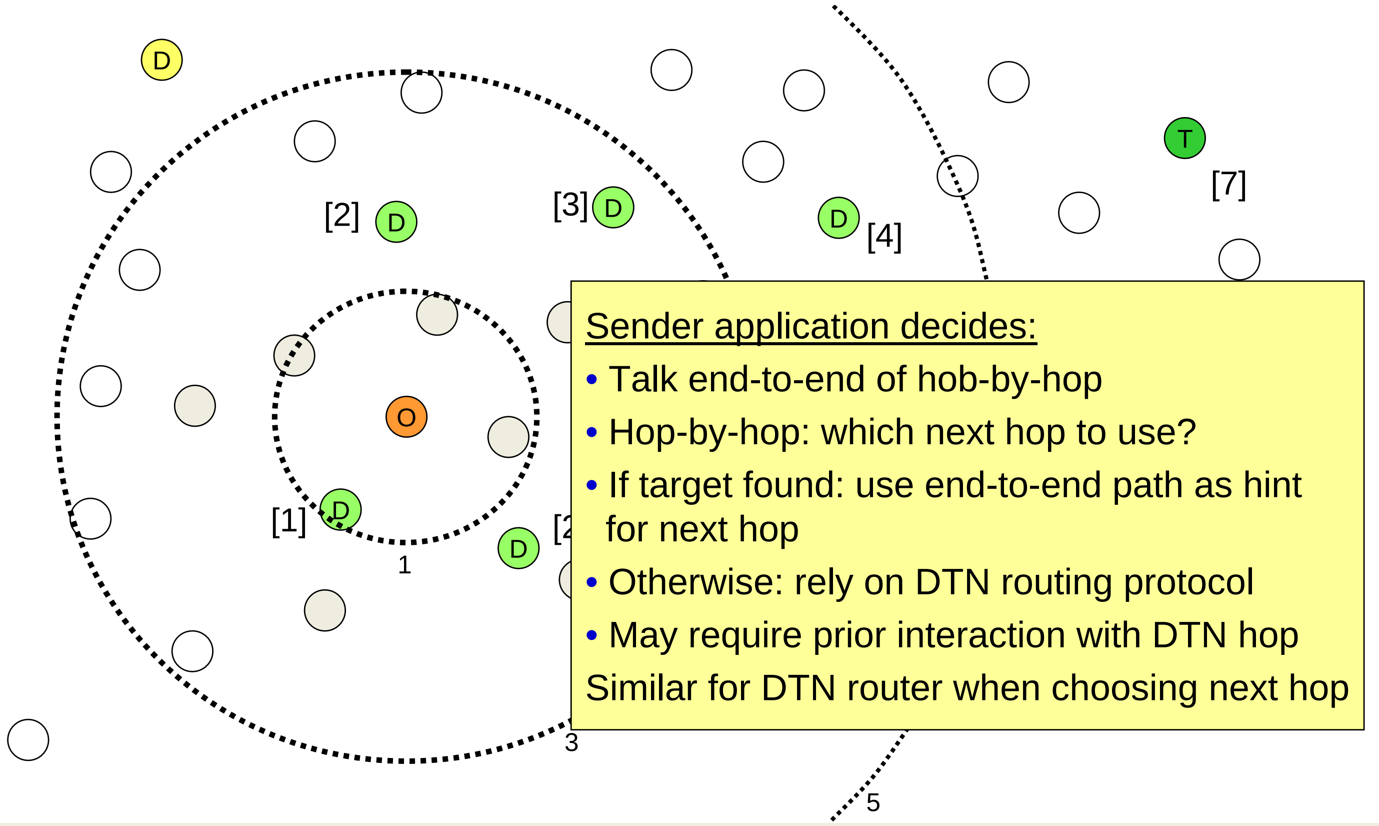
Extended AODV Operation (TTL 7)



Extended AODV Operation (TTL 7)



Extended AODV Operation



<u>Sender application decides:</u>

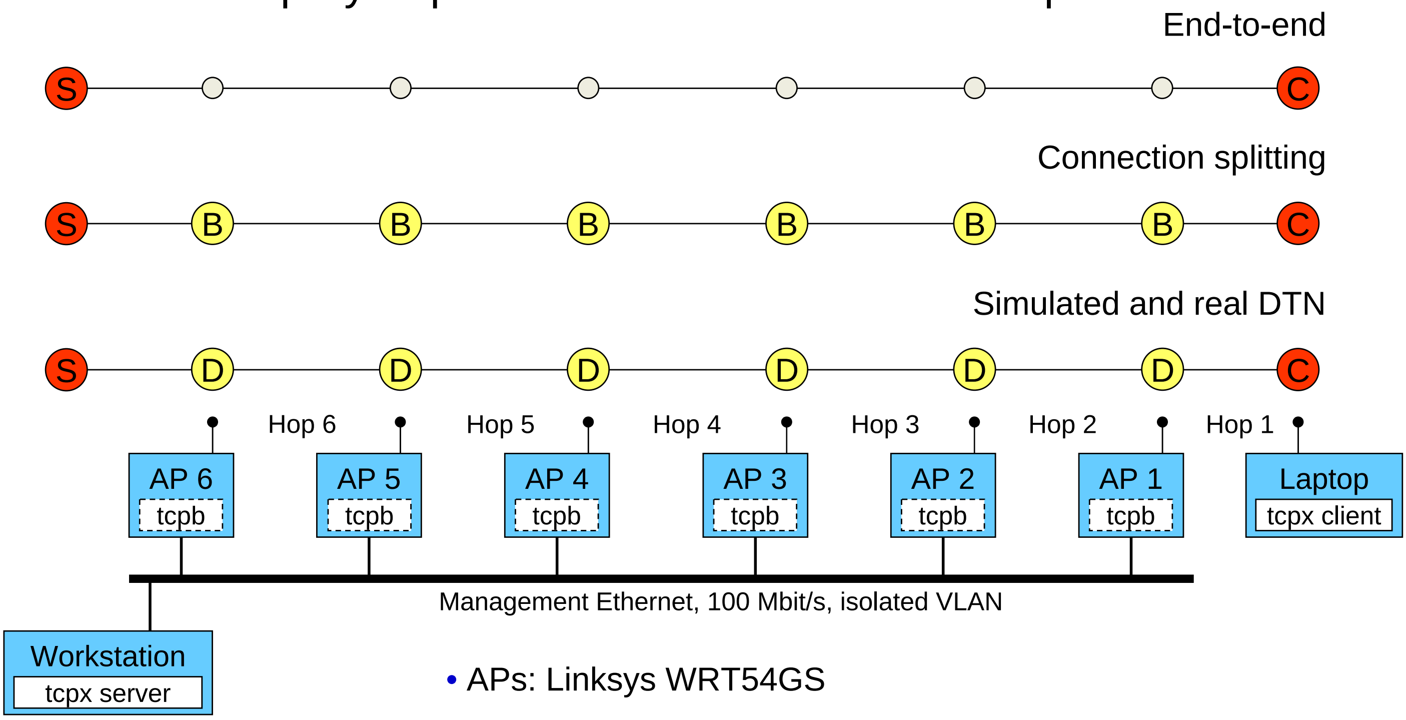
- Talk end-to-end of hob-by-hop
- Hop-by-hop: which next hop to use?
- If target found: use end-to-end path as hint for next hop
- Otherwise: rely on DTN routing protocol
- May require prior interaction with DTN hop

Similar for DTN router when choosing next hop

Performance Observations

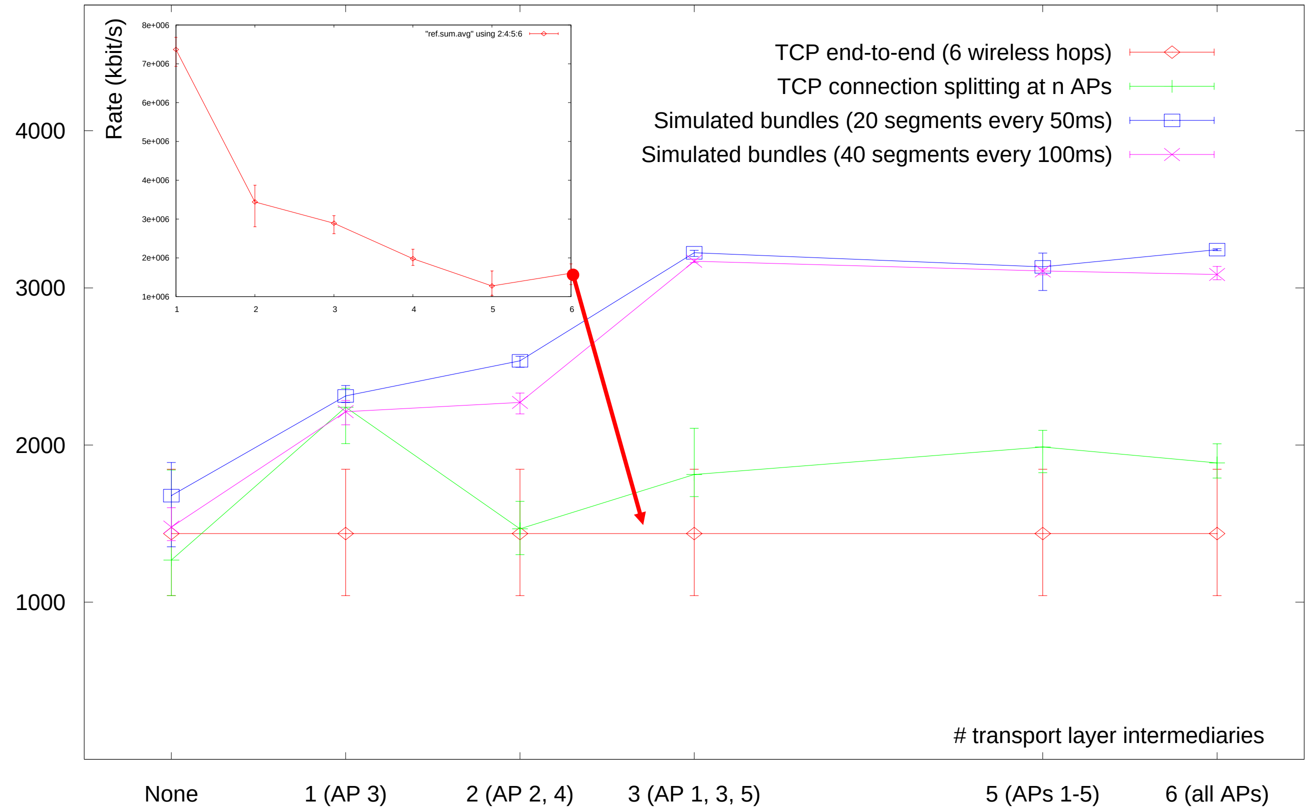
Measurements (1)

- Part 1: Hop-by-hop vs. end-to-end in a static setup

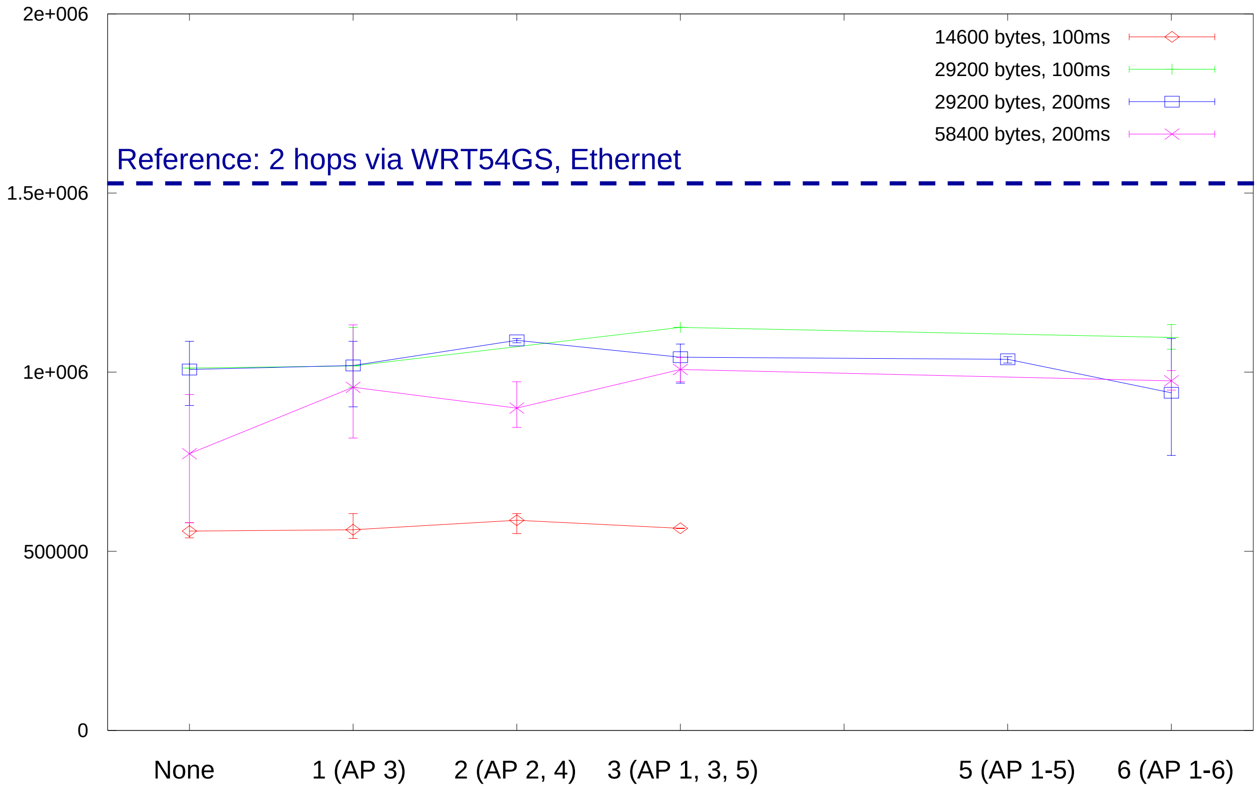


- APs: Linksys WRT54GS

Results (1.1)



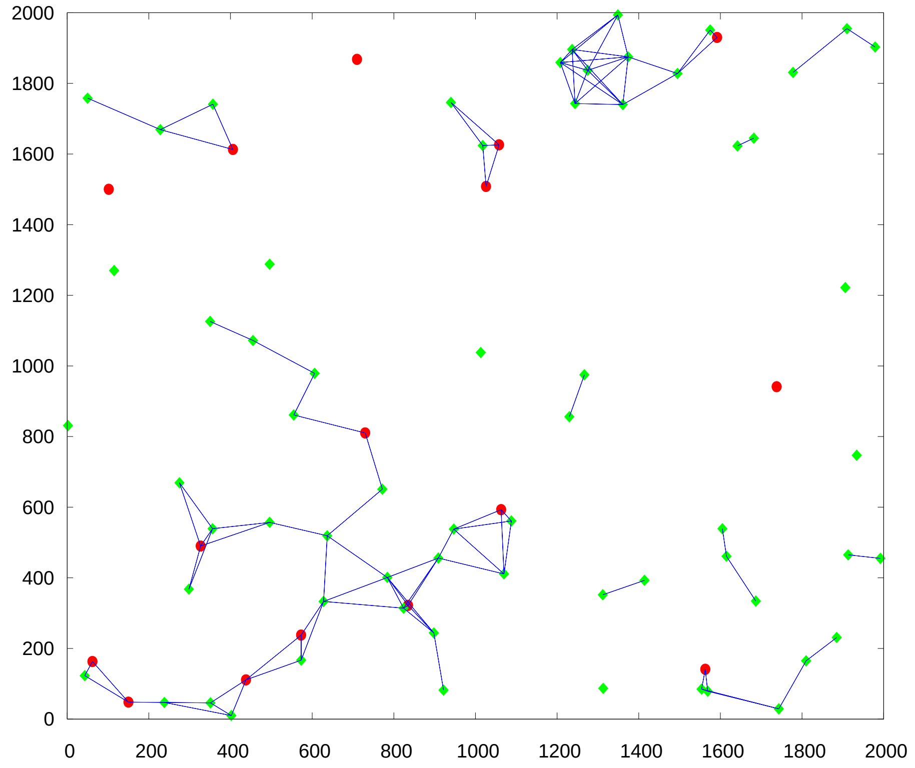
Results (1.2): dtnd



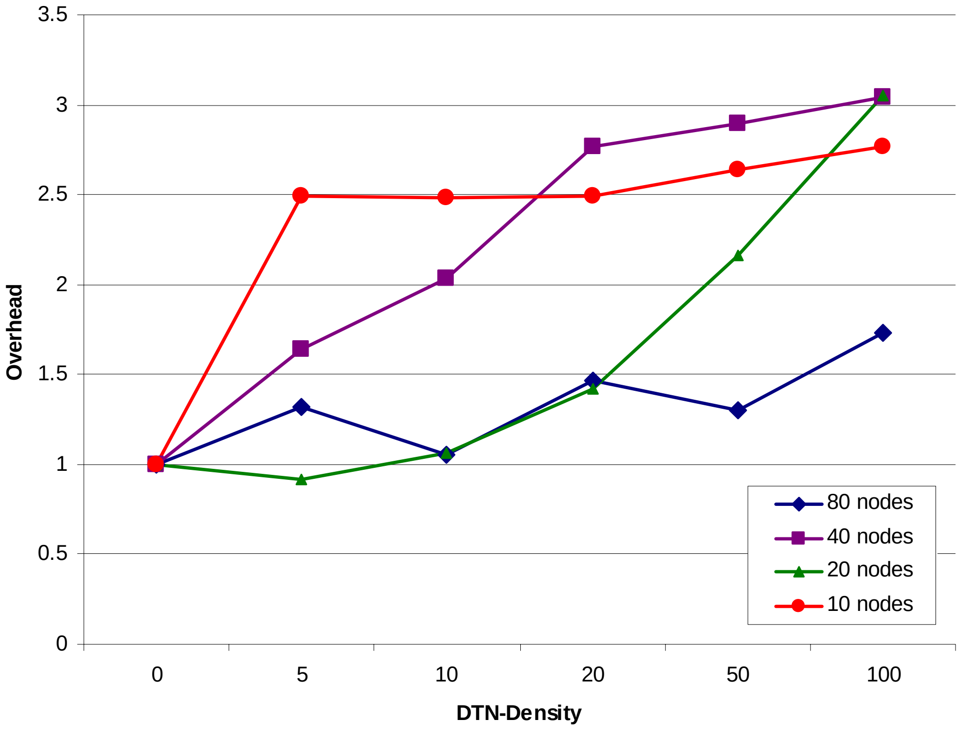
Measurements (2)

- Emulation environment
  - Dedicated emulation machine: 2 x Intel dual core @ 3.7 GHz (16 GB RAM)
  - Xen 3.0 (unstable) for up to 100 virtual machines (32 MB memory each)
  - Debian 2.6.16.13
  - AODV UU from Uppsala University + AODV extensions implemented
  - ns-2 + nsemulation patch to simulate motion + wireless connectivity
  - tcpdump
- AODV Extension Overhead
  - Repeated ping from one node to another
  - 2000m x 2000m area, 200m wireless communication range, 0…5 m/s
  - 20, 40, 80 nodes
  - 0, 5, 10, 20, 50, 100% of which support DTN
  - Compare against plain AODV overhead

Measurements (2)

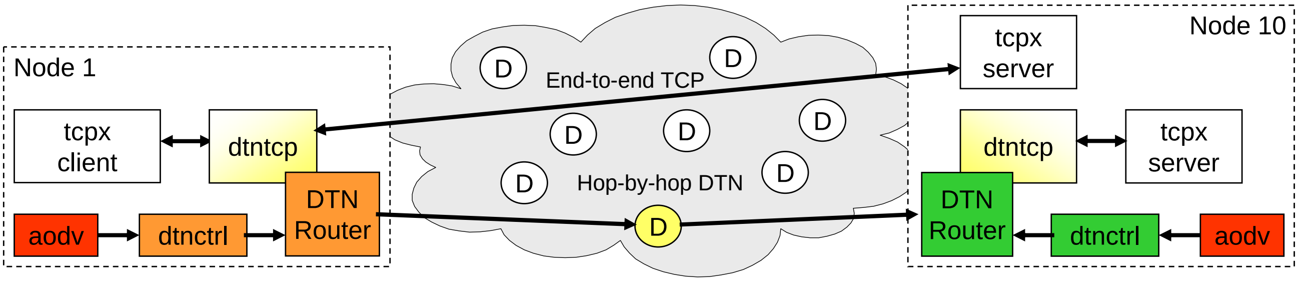


Results (2)



Measurement (3)

- dtnd from the DTN reference implementation (Intel, UC Berkeley)
- tcpx + dtntcp
  - Send via plain TCP end-to-end whenever a path exists upon transfer initiation
  - Otherwise fall back to DTN
- Initial tests with chain of 10 nodes and on/off links
- Proof of concept emulations
  - 2000m x 600m area, 10 nodes: random walk, 0…20m/s, 29200 bytes every 150s
  - Hack: n > m routing
  - Transmission from node 1 to node 10 in regular intervals
  - DTN augmented TCP-based delivery well in most cases with only 5min lifetimes



Discussion: API and Applications

- Current approach: implicit signaling (e.g., TCP connect())
  - Little chance to communicate EIDs
  - Approach limited to locating DTN routers
  - Cumbersome retrieval of “search results”
    - Correlation may be tricky
    - How long to wait? (TCP timeout? …?)
    - Protocol-dependent feedback does not work with all protocols
- Better: dedicated API for explicit control
  - Trigger route search independently
  - Allow providing and retrieving parameters as needed
  - Current thoughts: interfacing through dtnd vs. bypassing dtnd
- Applications need to be adapted for mobility anyway

Discussion: Routing Integration

- Simple case: just find DTN routers
  - Implicit service location
- Advanced case: learn about DTN route metrics
  - Data volume + processing overhead vs. improved routing decisions
- How much subsequent handshaking is needed?
  - Mutual authentication (which may decide about accepting a bundle)
  - Possible other reasons for not accepting a bundle
  - How much information can and should we convey *before*?
    - Taking into account that AODV is expensive
  - How will DTN routing protocols work in ad-hoc environments?
- How to embed routing hints in DTN bundles?
  - (and processing/queuing hints for the local DTN router)

Conclusion and Next Steps

- AODV extensions enables route search to locate DTN routers
  - Re-using a reactive routing protocol as opposed to separate service location
  - Insights into requirements for underlying routing protocols
- Allows applications to make e2e vs. hop-by-hop tradeoffs
- Allows DTN routers to discover peers beyond next hop
- Allows DTN routers to forward packets instead of bundles
- DTN-based communication may outperform end-to-end TCP

Next steps:

- Interaction with emerging DTN routing protocols
- Performance tradeoffs in different MANET environments
- API, applications and policies
- Internet Draft describing all the details