QoS-aware, Adaptive Throughput-Enhancement for CSMA-based Mobile Ad Hoc Networks

Rainer Storn

Rohde & Schwarz GmbH & Co. KG, 81671 Munich Germany
Email: rainer.storn@rsd.rohde-schwarz.com
Overview

1. Motivation for IP-capable Mobile Ad-Hoc Networks (MANETs)

2. Identification of problems affecting throughput
   1. Multihopping
   2. MAC-efficiency
   3. Overhead

3. QoS-aware schemes to mitigate MAC influences
   1. Concatenation
   2. Piggybacking
Overview

1. Motivation for IP-capable Mobile Ad-Hoc Networks (MANETs)

2. Identification of problems affecting throughput
   1. Multihopping
   2. MAC-efficiency
   3. Overhead

3. QoS-aware schemes to mitigate MAC influences
   1. Concatenation
   2. Piggybacking
Networks in the military environment

Today's legacy networks:
- Organized in circuits
- Single-hop connections
- Voice and data in separate networks

Upcoming SDR-networks:
- Part of a larger heterogeneous network. Waveforms need to be optimized for the internet protocol (IP)
- Mobile Ad-Hoc Networking due to rapid deployability and absence of a single point of failure
- Range increase through multi-hop capability (IP)
- Need to handle different applications of differing importance (e.g. network control, voice, data) → QoS mechanisms needed
- For the case of heterogeneous traffic and varying load CSMA/CA MAC protocol is very popular (e.g. WiFi IEEE 802.11)
Overview

1. **Motivation for IP-capable Mobile Ad-Hoc Networks (MANETs)**

2. **Identification of problems affecting throughput**
   1. Multihopping
   2. MAC-efficiency
   3. Overhead

3. **QoS-aware schemes to mitigate MAC influences**
   1. Concatenation
   2. Piggybacking
Problems encountered with Multihopping, CSMA, and IP

a) **Multihopping reduces throughput $R$. With $N$ nodes and ideal MAC-Layer we get** 1):

$$\text{Throughput } R \sim \frac{1}{\sqrt{N \cdot \log(N)}}$$

b) **MAC-Layer has limited efficiency**
   a) CSMA $\rightarrow$ Collisions (TDMA $\rightarrow$ waste of bandwidth at low network utilization)

c) **Bandwidth is wasted due to overhead**
   a) UDP, TCP, IP ... header
   b) MAC preamble and header

---

Data Rate Definitions

Payload Rate $R_p$:

\[ R_{p,max} = R_{a,max} \cdot \frac{M_{payload}}{M_{payload} + M_{overhead}} \]

Aggregate Rate $R_a$:

\[ R_{a,max} = R_{n,max} \cdot K_{MAC} \]

Net Data Rate $R_n$:

\[ R_{n,max} = R_{b,max} \cdot R_c = 72 \frac{Mbit}{s} \cdot \frac{3}{4} = 54 \frac{Mbit}{s} \]

Gross Data Rate $R_b$:

\[ R_{b,max} = \frac{N \cdot b_N}{T_S} = 48 \cdot 6bit \cdot \frac{4 \mu s}{s} = 72 \frac{Mbit}{s} \]

Example: IEEE 802.11a, 64-QAM modulation
Data Rate Loss over the OSI Layers (example IEEE 802.11a)

Net rate = 54Mb/s
M_{overhead} = 48 Byte

Payload Size in Bytes

Rate Ratio

- payload rate / aggregate rate
- aggregate rate / net rate

Highest loss of data rate caused by MAC layer
Data Rate Loss over the OSI Layers (example IEEE 802.11a)

Net rate = 54Mb/s
M_{overhead} = 48 Byte

Payload Size in Bytes

Rate Ratio

payload rate / aggregate rate
aggregate rate / net rate

Highest loss of data rate caused by MAC layer
Overview

1. Motivation for IP-capable Mobile Ad-Hoc Networks (MANETs)

2. Identification of problems affecting throughput
   1. Multihopping
   2. MAC-efficiency
   3. Overhead

3. QoS-aware schemes to mitigate MAC influences
   1. Concatenation
   2. Piggybacking
DiffServ: Mapping of Services to Tx Priorities (Example)

Mapping of Service Classes to Tx Priorities

- Priority-Based Queuing
  - Prio 1 Queue
  - Prio 2 Queue
  - Prio 3 Queue
  - Prio 4 Queue

Mapping onto IPSec

- DSCP Converter
- DSCP Mapper
- Classifier & Marker

IPv6 Routing (black side)

Network Layer (red side)

DS Field

DSCP = DiffServ Codepoint
ECN = Explicit Congestion Notification

Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Source Address
Destination Address

Audio Packet
FTP Packet
Email Packet
Video Packet
Adaptive Concatenation within Priority Queues

Hop a  Hop b  Hop c

next-hop-Tx subqueues:

5  4  3  1  2

Tx-hop-token-queue:

a  c  b

pseudo timestamp 

"1" says that this packet arrived first

next-hop-subqueue 

"b" will be serviced first

Prio x Queue
Adaptive Concatenation within Priority Queues

- Tx-Queues will be emptied as far as possible
- Concatenation increases payload and hence increases throughput
- Latency increase is adaptive and stays as small as possible
- Complex queue processing
- Busy period is increased
- Packet loss destroys more information

next-hop-Tx subqueues:

Transmission hop-token-queue:

"1" says that this packet arrived first

next-hop-subqueue "c" will be serviced next
Modulation-aware Hop Token Queue

- Hop a (QPSK)
- Hop b (QPSK)
- Hop c (64-QAM)

next-hop-Tx subqueues:
- Hop a
- Hop b
- Hop c

Tx-hop-token-queue:
- Prio x Queue
- "1" says that this packet arrived first
- next-hop-subqueues "b" and "a" will be serviced together
**Modulation-aware Hop Token Queue**

- **Hop a** (QPSK) - next-hop-Tx subqueues
- **Hop b** (QPSK)
- **Hop c** (64-QAM)

**Tx-hop-token-queue:**
- **pseudo timestamp** "1" says that this packet arrived first
- **next-hop-subqueues** "c" comes next

**Prio x Queue**

- **Tx-Queues will be emptied as far as possible**
- **Concatenation increases payload and hence increases throughput**
- **Latency increase is adaptive and stays as small as possible**
- Complex queue processing
- Busy period is increased
- Packet loss destroys more information
- Modulation has to be monitored and hop-token-queue content updated continuously
Further structure of next-hop-Tx-Subqueue

Data-type-subqueues:
- Voice 1
- Network Control
- Other

Payloads:
- Payload 1
- Payload 2
- Payload 3 (overhead)
- Payload 4
- Payload 5
- Payload 6

Pseudo-timestamp "4" says that this packet arrived fourth

MAC frame structure for aggregated data:

<table>
<thead>
<tr>
<th>Next Type</th>
<th>Traffic Class</th>
<th># packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Voice (Voice 1)</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>Network Control</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>Other (other)</td>
<td>1</td>
</tr>
</tbody>
</table>

Common Voice Overhead
- Payload 1
- Payload 4
- Payload 5

Common Network Control Overhead
- Payload 2
- Payload 6

Individual Overhead for payload 3
- Payload 3

MAC Forward Error Correction
Piggybacking of DATA on MAC-layer

Prio X Queue of node B may be selected for transmission in two ways:

- By scheduler
- By incoming ACK

+ No address comparisons
+ ACK is sent immediately & independently of data destination address
+ ACKs are never sent without data at the forwarding node
+ QoS not violated

- Busy period is increased
- Modulation B→C and B→A has to be taken into account

Since ACK 1 is sent within busy period, DATA 1 can be sent along with it.

Tying DATA 1 to ACK 1 is QoS neutral, because DATA 1 has won the contention phase

If only one forwarding hop is used ACK 1' terminates the piggybacking.
Combination of Piggybacking and Concatenation

Prio X Queue of node B may be selected for transmission in two ways:

- By scheduler
- By incoming ACK

+ No address comparisons
+ ACK is sent immediately & independently of data destination address
+ ACKs are never sent without data at the forwarding node
+ QoS not violated
- Busy period is increased
- Modulation B→A and B→C has to be taken into account

Not only DATA 1 is sent along with ACK 1 but also (potentially) the content of one or more next-hop-subqueues (depending on modulation scheme)

Works only if B can send to A and C with the same modulation scheme
Benefits of concatenation and piggybacking

- Due to adaptivity no concatenation and hence no increased latency for lightly loaded networks (important for realtime traffic like voice)

- High load $\rightarrow$ high probability of collisions $\rightarrow$ high probability of concatenation
  - 2 packets concatenated $\rightarrow$ throughput increases 2 times
  - 4 packets concatenated $\rightarrow$ throughput increases 3 times
  - Piggybacking renders a further throughput increase of up to 50% \(^1\)

- Schemes look promising, but realistic \(^2\) simulations needed that consider:
  - Error correction scheme and packet loss due to collisions
  - Traffic mix, traffic load, packet arrival process
  - Network type, size and node movement
  - MANET protocol

<table>
<thead>
<tr>
<th>Payload packet size</th>
<th>Payload rate (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>1.7</td>
</tr>
<tr>
<td>64</td>
<td>3.5</td>
</tr>
<tr>
<td>128</td>
<td>5.2</td>
</tr>
<tr>
<td>192</td>
<td>6.9</td>
</tr>
<tr>
<td>256</td>
<td>11.0</td>
</tr>
<tr>
<td>512</td>
<td>13.6</td>
</tr>
<tr>
<td>1024</td>
<td>22.1</td>
</tr>
</tbody>
</table>

Example: net rate 54 Mb/s, IEEE 802.11a, 64-QAM

---


Conclusion

- The main throughput-reducing effects in IP-based MANETs have been summarized
  - Multihopping
  - Low MAC-efficiency, especially for small packets (e.g. voice)
  - Protocol-Overhead, especially due to IP and potentially IPSec

- Concentration on MAC for CSMA/CA since most of the incurred data rate loss happens in the MAC layer

- Several schemes to counteract data rate loss
  - Adaptive concatenation (next hop aware, modulation aware, application aware)
  - Piggybacking

- Schemes look promising, but realistic simulation needed to find out net benefit
Thank Your For Your Attention
IPv6- and UDP-Overhead

Overhead: 48 Byte
IPv6- and UDP-Overhead if IPSec is used

Overhead: 100Byte