Route-Over Forwarding Techniques in a 6LoWPAN

Andreas Weigel, Martin Ringwelski, Volker Turau, Andreas Timm-Giel

MONAMI
September 24\textsuperscript{th}, 2013

Institutes of Telematics and of Communication Networks
TUHH
IPv6 over low power WPAN (6LoWPAN)

Vision: Internet of Things

Every device should have its own IP address and should be directly accessible through the Internet.

- IPv6 supports approximately $3.4 \cdot 10^{38}$ addresses, but:
  - 802.15.4 supports frames up to 127 byte
  - IPv6 requires a MTU of at least 1280 byte!

- Solution for using IPv6 on 802.15.4 is 6LoWPAN:
  - Intermediate layer for header compression,
  - Packet fragmentation and
  - Mesh routing (Mesh under) ability

The base specification document is RFC 4944
IPv6 over low power WPAN (6LoWPAN)

Vision: Internet of Things

Every device should have its own IP address and should be directly accessible through the Internet.

- IPv6 supports approximately $3.4 \cdot 10^{38}$ addresses, but:
  - 802.15.4 supports frames up to 127 byte
  - IPv6 requires a MTU of at least 1280 byte!

- Solution for using IPv6 on 802.15.4 is 6LoWPAN:
  - Intermediate layer for header compression,
  - Packet fragmentation and
  - Mesh routing (Mesh under) ability
  - Route-over

The base specification document is RFC 4944
Motivation

- Fragmentation can cause trouble!
  One lost fragment results in a lost datagram

- Big packets needed by:
  - Smart Metering
  - Firmware Updates
  - ...
  - If it is possible, people will use it
Motivation

- Fragmentation can cause trouble!
  One lost fragment results in a lost datagram

- Big packets needed by:
  - Smart Metering
  - Firmware Updates
  - ...
  - If it is possible, people will use it

- Compare different forwarding techniques
- Introduce enhancements
Route-Over Forwarding Strategies

Assembly:
- On each hop: Wait for every fragment
- Reassemble datagram and send to IP Layer
- IP Layer sends datagram back to 6LoWPAN
- 6LoWPAN recreates fragments

Direct:
- On each hop: Look into the first fragment
- If not for this node lookup route
- Directly send to next Hop
- Safe routing information for next fragment
Problems

Assembly:
- Needs a big buffer
  A node needs a buffer for every incoming datagram
- Does not allow pipelining

Direct:
- Can lead to heavy losses
  A node tries to forward a frame while the next is being send
Enhanced Modes

Direct-RR:
- sending rate of the queue is restricted
- Inter frame delay (between 15 and 21 ms)

Direct-ARR:
- Similar to Direct-RR
- Adaptive delay (EWMA filter on last delay)

Retry Control:
- Progress-based Retry Control (PRC)
- Later Fragments of a datagram get increased number of maximum retries
CometOS

- A Component-based, extensible, tiny Operating System for wireless sensor networks
- Developed at the Institute of Telematics (TUHH)
- Code written in C++
- One implementation for OMNeT++ and hardware
- Own implementation of the 6LoWPAN stack
Topologies

Chain-Network

Star-Network

LongY-Network

RealSim

Edges represent static routes, the dark gray node is the sink.
Methodology

Settings

- Static Routing
- All implementations use same amount of RAM
- Perfect Links in the Chain- and Star-Network
- UDP Packets
- Sending Rate for each Node: $37.5 \frac{\text{Byte}}{s}$
- Simulation:
  - payload [Byte] = 50, 100, 200, 400, 800, 1200
  - 2000 Packets, 5 runs
- Testbed:
  - 48 000 Bytes in
  - payload [Byte] = 100, 400, 1200
Chain-Network - PRR

![Graph showing PRR vs Payload for different methods: Assembly, Direct, Direct-RR, and Direct-ARR.](image-url)
Chain-Network - Latency

Per hop latency and PRR for 1200 Byte Payload
Star-Network PRR

![Graph showing Star-Network PRR](image-url)
LongY-Network - PRR

![Graph showing PRR vs Payload size for Assembly, Direct, Direct-RR, and Direct-ARR]
LongY-Network - Latency

Assembly Mode

Direct Mode

Direct-RR Mode

Direct-ARR Mode

Per hop latency and PRR for 1200 Byte Payload
RealSim-Network - PRR

The graph shows the Packet Reception Rate (PRR) in percentage as a function of payload size in bytes, comparing different forwarding techniques:

- Assembly
- Direct
- Direct-RR
- Direct-ARR

The x-axis represents the payload size in bytes, ranging from 0 to 1,200. The y-axis shows the PRR percentage, ranging from 0% to 100%. The graph illustrates how the PRR decreases as the payload size increases for each technique.
Testbed - PRR

![Graph showing PRR (%) versus Payload (Byte)]

- **Assembly**
- **Direct**
- **Direct-RR**
- **Direct-ARR**
RealSim-Network - Latency

**Assembly Mode**

- Latency [ms] vs Hops
- Per hop latency and PRR for 1200 Byte Payload

**Direct Mode**

- Latency [ms] vs Hops
- Per hop latency and PRR for 1200 Byte Payload

**Direct-RR Mode**

- Latency [ms] vs Hops
- Per hop latency and PRR for 1200 Byte Payload

**Direct-ARR Mode**

- Latency [ms] vs Hops
- Per hop latency and PRR for 1200 Byte Payload
Conclusion

- 6LoWPAN enables 802.15.4 nodes to use IPv6
- Different forwarding strategies for fragments
- Significant difference between Direct and Assembly Mode
- Rate Restriction provides better PRR
- Direct-ARR scales best, but may increase latency for small hop distances
- Retry Control has very limited impact
Outlook

- Compare selective retry control with flat increased retries
- Implement a Hop-based Retry Control
- Implement a fragment recovery mechanism
- Evaluate different settings of the MAC configuration
Route-Over Forwarding Techniques in a 6LoWPAN

Andreas Weigel, Martin Ringwelski, Volker Turau, Andreas Timm-Giel

Martin Ringwelski
Research Assistant

Phone  +49 / (0)40 428 78 3387
e-Mail  martin.ringwelski@tuhh.de

http://www.ti5.tu-harburg.de/staff/ringwelski
Appendix
Packet Fragmentation

- Header with 4 for first and 5 byte for following fragments
- Allowed fragmented datagram size of up to 2048 byte
- Header inherits size and tag of the IP datagram
- Position of the fragment in the datagram is in header
  \[\Rightarrow\] Fragments do not need to arrive in order, but one lost fragment results in a lost datagram