

# Experiences with RPL: IPv6 Routing Protocol for Low power and Lossy Networks

T. Clausen, J. Yi, A. Colin de Verdiere  
LIX, Ecole Polytechnique

U. Herberg  
Fujitsu Laboratories of America



# Background

From where do these experiences come?

## 1. jRPL - Implemented, started in draft-....-rpl-07

- Continued updates, compliant to RFC6550
- Simulations, lab deployments

## 2. ContikiRPL - deployments, tests

## 3. New impl. on specific, constrained, platform

- Deployments



# Goals and non-Goals

Why are we doing this?

- Generalize, share our experiences
- Solicit feedback, experiences from community
  - Since draft-.....-00, much such received - thank you!
- Contribute to understand
  - Where RPL is applicable
  - Where RPL is not applicable
  - Directions where further work is suggested



# Discussed in I-D

## **draft-clausen-11n-rpl-experiences**

- DODAG Root Requirements
- RPL Traffic Flows
- Fragmentation
- Downward and Point-to-Point Routes
- Address Aggregation and Summarization
- Links Assumed Bi-Directional
- Neighbor Unreachability Detection For Unidirectional Links
- Implementability and Complexity
- Underspecification
- Convergence
- Loops



# Discussed in Presentation

## `draft-clausen-11n-rpl-experiences`

- ✓ DODAG Root Requirements
- RPL Traffic Flows
- ✓ Fragmentation
- Downward and Point-to-Point Routes
- Address Aggregation and Summarization
- Links Assumed Bi-Directional
- ✓ Neighbor Unreachability Detection For Unidirectional Links
- Implementability and Complexity
- Underspecification
- ✓ Convergence
- ✓ Loops



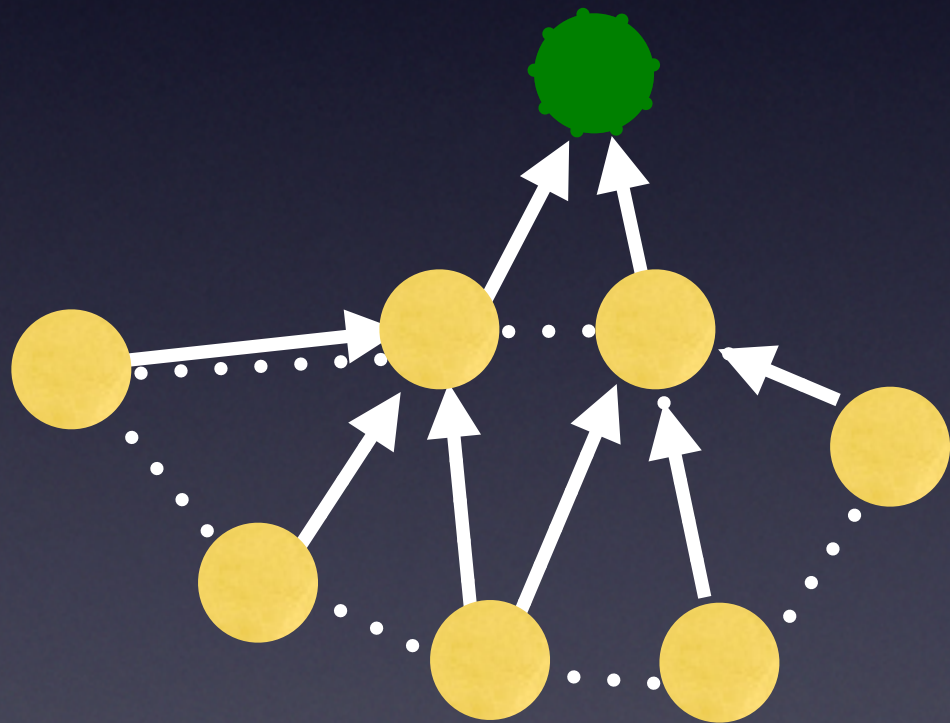
# DODAG Root Requirements (I)

- “Evaluating the Performance of RPL and 6LoWPAN in TinyOS” (Culler et. al, 2011)
- *Storing mode: 10KB of RAM, about 30 RPL Routers in the LLN*
- *We found that to be true, also*



# DODAG Root Requirements (2)

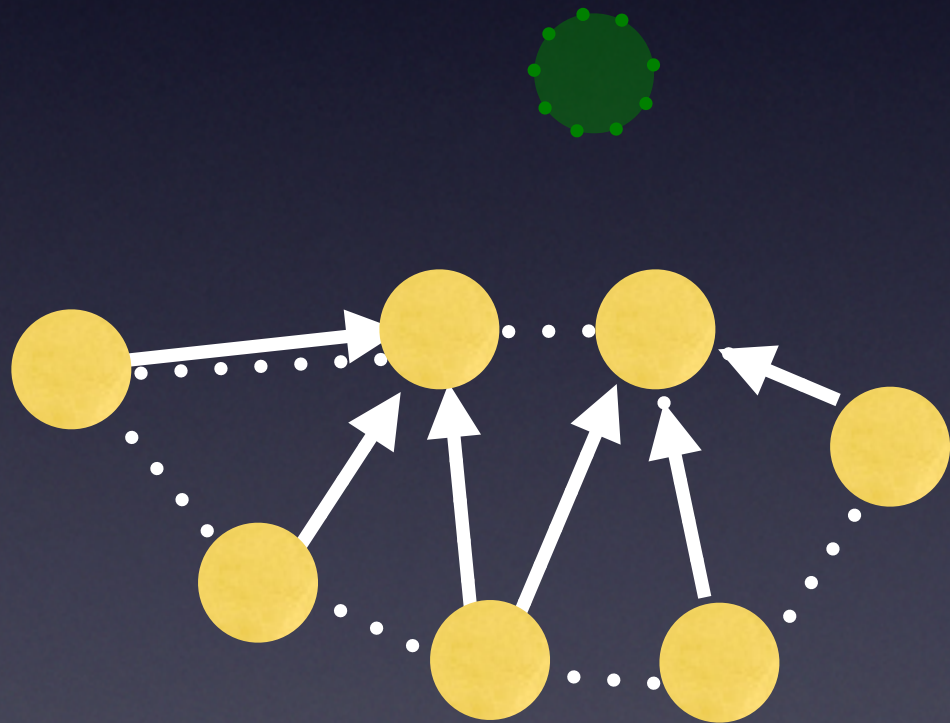
- Non-storing mode: state in DODAG Root
- But ....





# DODAG Root Requirements (2)

- Non-storing mode: state in DODAG Root
- But ....





# DODAG Root Requirements (2)

- Non-storing mode: state in DODAG Root
- But ....





# DODAG Root Requirements (2)

- Non-storing mode: state in DODAG Root
- But ....





# DODAG Root Requirements (2)

- Non-storing mode: state in DODAG Root
- But ....





# DODAG Root Requirements (2)

- Non-storing mode: state in DODAG Root
- But ....





# DODAG Root Requirements (2)

- Non-storing mode: state in DODAG Root
- But ....

Become Root for “Floating DODAG”  
Must be provisioned with sufficient resources

## Observations:

....etc...

- Disconnections from DODAG Root: not rare
- If using “Floating DODAG” for internal connectivity
  - “Floating Root” must be provisioned as DODAG Root
  - Root (floating or not): much more state than storing mode



# DODAG Root Requirements (2)

- Non-storing mode: state in DODAG Root
- But ....

## Observations:

- Disconnections from DODAG Root: not rare
- If using “Floating DODAG” for internal connectivity
  - “Floating Root” must be provisioned as DODAG Root
  - Root (floating or not): much more state than non-storing mode



# DODAG Root Requirements (2)

- Non-storing mode: state in DODAG Root
- But ....

## Observations:

- Disconnections from DODAG Root: not rare
- If using “Floating DODAG” for internal connectivity
  - “Floating Root” must be provisioned as DODAG Root
  - Root (floating or not): much more state than non-storing mode
- Unless very planned, structured deployment
  - All LLN nodes must be provisioned as DODAG Root



# Fragmentation (I)

- IEEE 802.15.4 + 6LoWPAN leaves 79 octets
- Example (fairly standard) RPL DIO:
  - ICMP (4) + DIO base (24) + metrics (8) + Conf. Obj (16) + prefix (32) = 84 octets
- Fragmentation of RPL control messages



# Fragmentation (2)

- Data-Traffic, non-storing mode
  - Source routing header (at least, 8 octets)
  - Variable available payload for data
    - P2P: src does not know the MTU when DODAG Root adds source-routing header
    - Src may chose inefficient data payload size



# Fragmentation (3)

- Data-Traffic, non-storing mode
  - Source outside LLN
    - IPv6-over-IPv6 tunnel
    - Border gateway adds RPL option to outer IPv6 packet

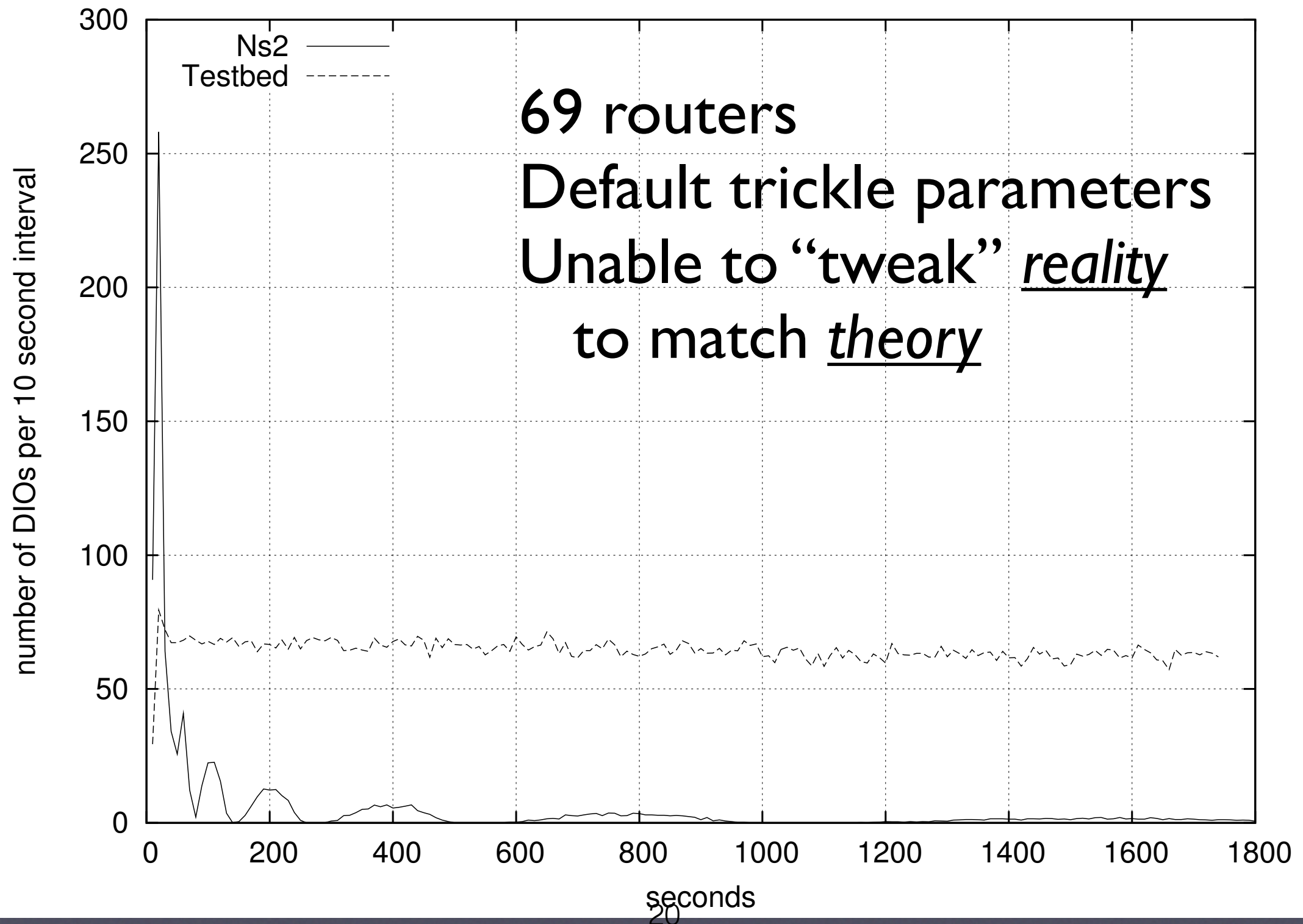


# Trickle timer

- Hard to set trickle parameters properly
  - A (redundancy) constant of 3-5 has been found adequate in deployments... (draft-gnawali-roll-rpl-recommendations-03)
  - AMI deployments SHOULD set DIORedundancyConstant to a value of at least 10. (draft-ietf-roll-applicability-ami-05)
- Very sensitive setting - hard to get just right
- Hard (impossible) to identify universal setting



# Convergence





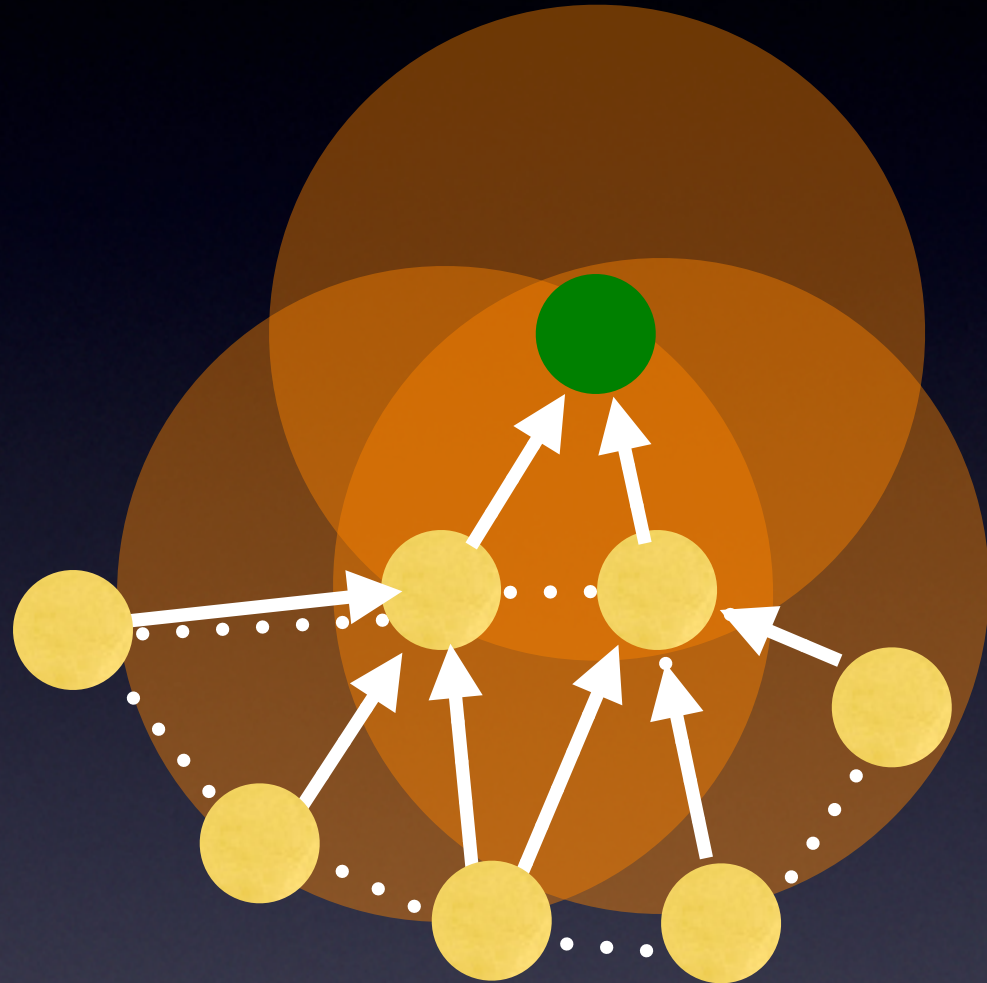
# Loops

- No loop freedom guarantee
- Loop detection during data traffic transmission
  - Longer delay
  - Buffer for the data traffic
- Loops exist - frequently
  - In test (69 routers, previous slide):
    - Snapshot every 10 sec
    - At least one loop in 74.14% of snapshots



# DODAG Construction

## Unidirectional Links



- Connectivity  $\uparrow$  based on messages received  $\downarrow$

Uni-directional  
links common?

*Oh yes...*



# NUD for Unidirectional Links

- Upward routes were build assuming link to be bidirectional
- NUD invoked only when unicast traffic fails
- Might not be able to react to connection loss (if no other parents)
- NUD from upper layer: problem not necessarily stems from preferred parent being unreachable.