MASC Prefix Allocation Algorithm

Deborah Estrin, Ramesh Govindan,
Mark Handley, Dave Thaler,
Pavlin Radoslavov

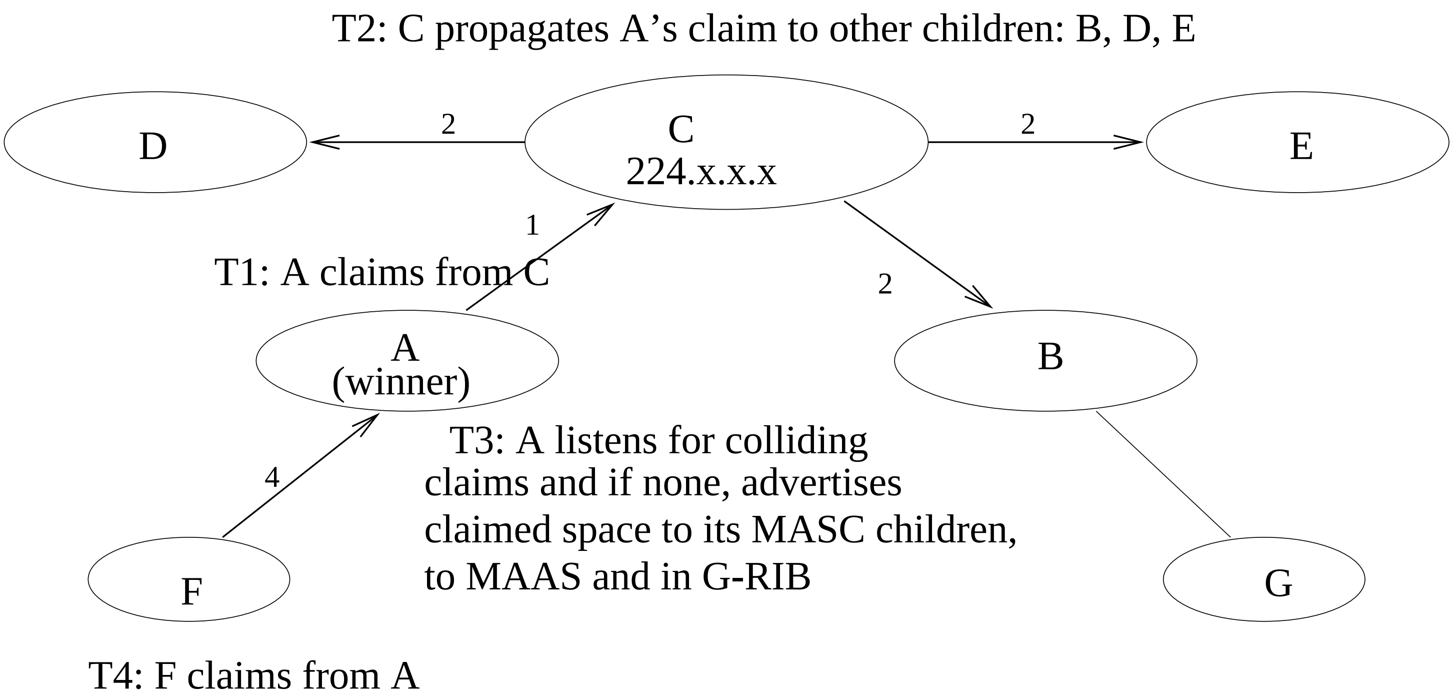
Overview

1. MASC Overview
2. Prefix Expansion Algorithm
3. Basic Scheme Issues
4. Algorithm Description
5. Simulation Results
6. Conclusions

MASC Overview

- "Multicast-Address-Set Claim" (MASC):
  - Associates group ranges (prefixes) with AS's
  - Injects local associations into G-RIB (to be used by BGMP)
- Goals:
  - state aggregation (few prefixes per domain)
  - efficient use of address space
  - minimize flux in G-RIB
  - avoid third-party dependency (e.g. no registry)

MASC Overview (cont.)



Prefix Expansion Algorithm

- Reverse Bit Expansion Algorithm (courtesy of Dave Thaler; inspired by Kampai)

3 bits space:	xxx
A:	000
B:	100
A:	00x
B:	10x
C:	010
C:	01x

- Why not Kampai (non-contiguous masks)?
  - non-contiguous masks are maybe problematic

Basic Scheme Issues

- For two-level hierarchy the overall utilization can be as low as 25% (50% per level)
- The Reverse Bit Expansion algorithm does not work well in a hierarchy:

Parent : 00xxx	Parent double: 0xxxx
Child A: 000xx	but neither A or B can
Child B: 001xx	expand

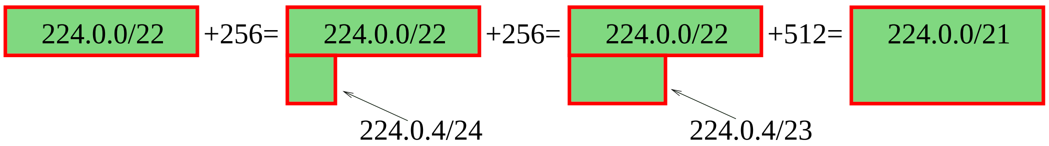
- Prefixes cannot be shrunk if utilization becomes very low

The Adaptation Mechanism

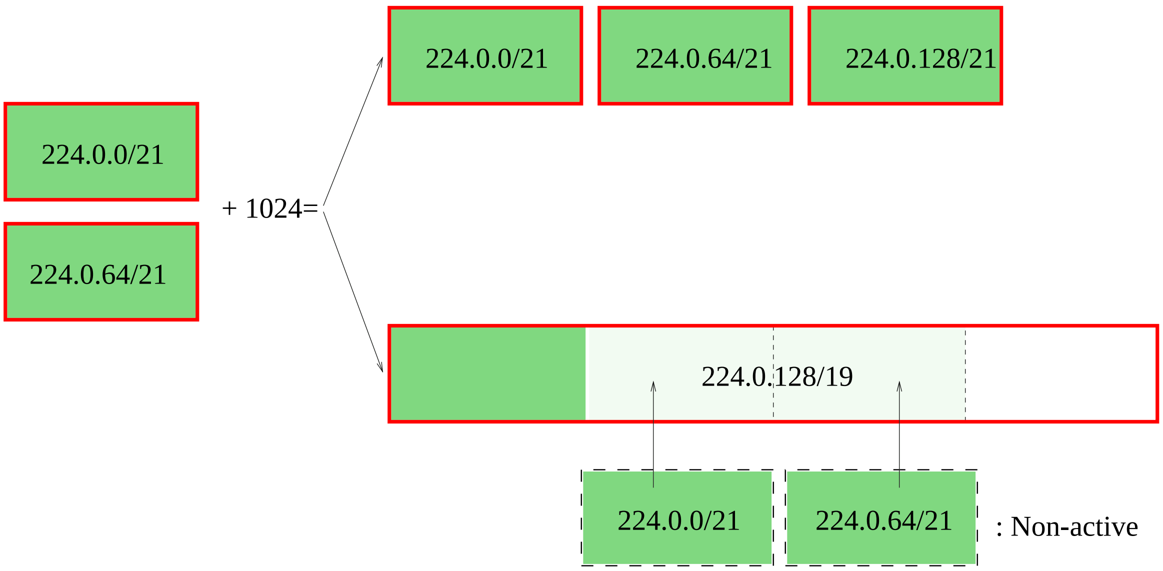
- Two prefixes per domain
- No need for 100% utilization
- If a prefix cannot expand, get a bigger one, large enough to accommodate current usage and stop reusing the old prefix(es)
- If the utilization goes down, get a smaller prefix and stop reusing the old larger prefix(es)
- Optimized for stability: only if the utilization is low for relatively long time, then trigger the adaptation

Examples

(a) Incremental increase

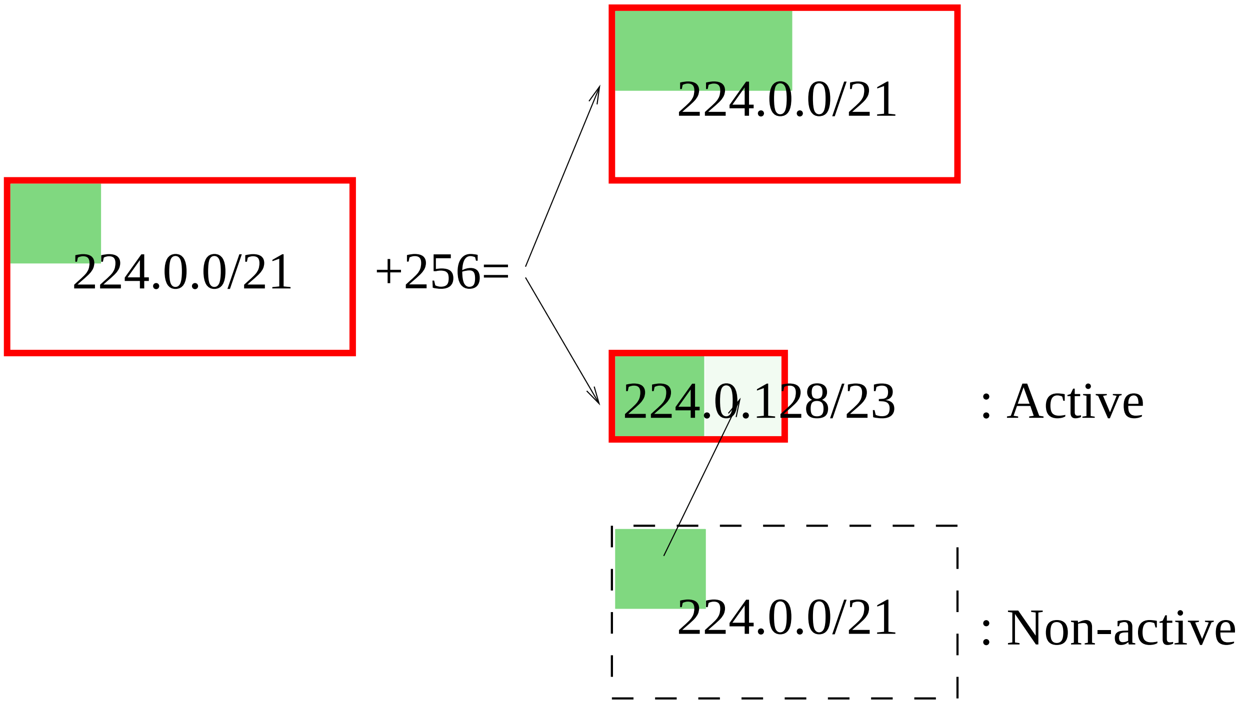


(b) Adaptive increase



Examples(cont.)

(c) Adaptive decrease

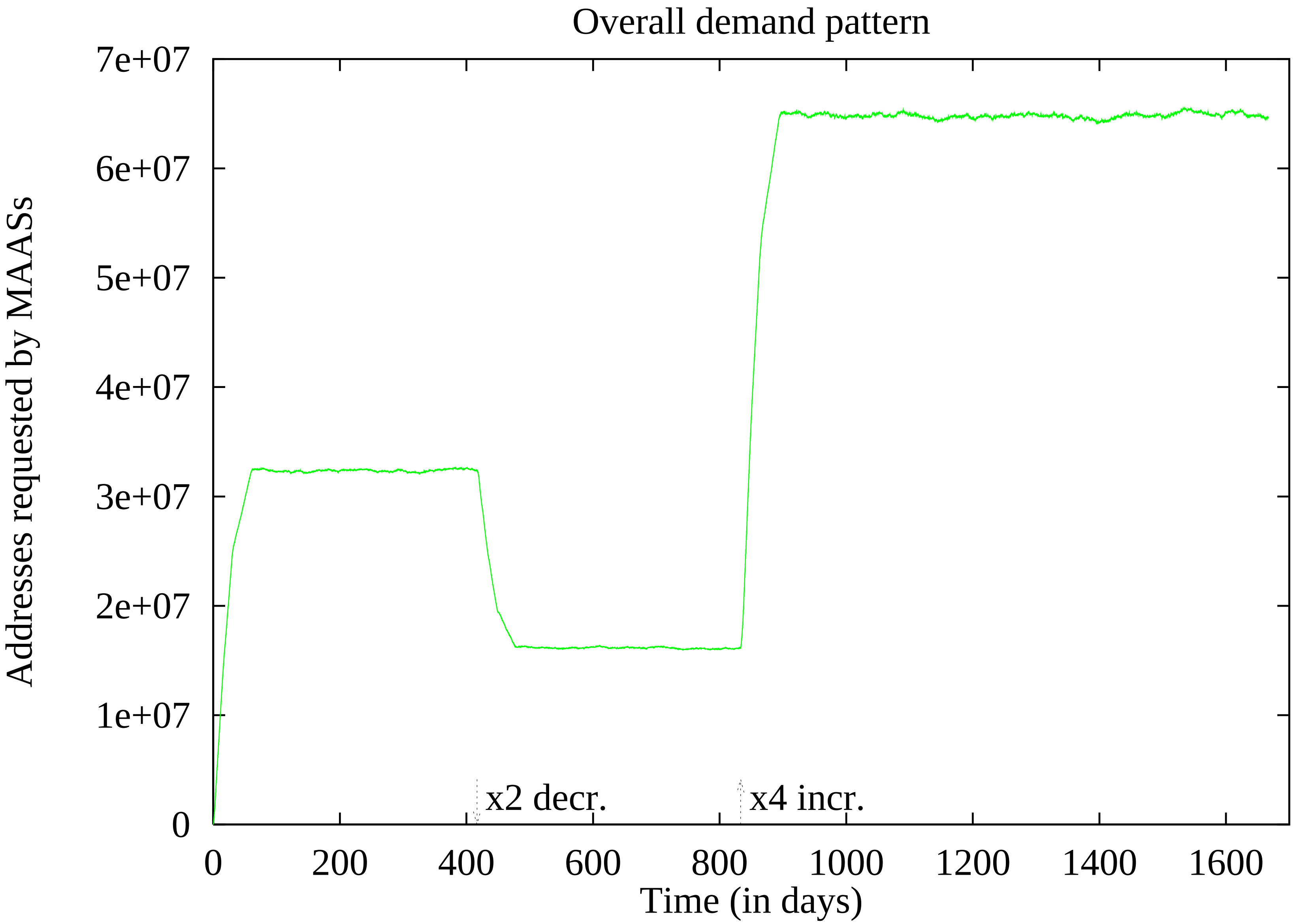


Simulation Results

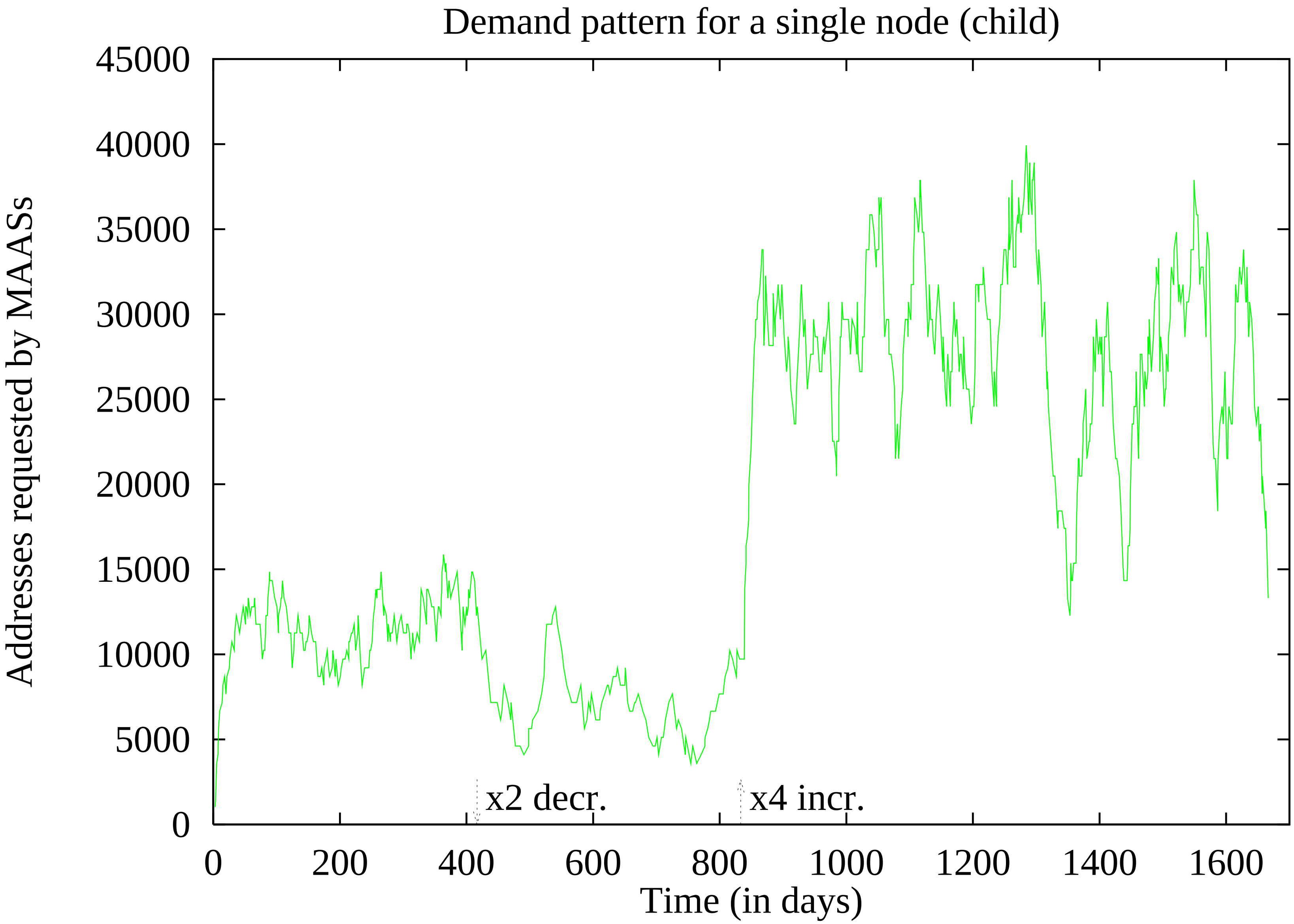
- Two-level hierarchy (50 parents, 50x50 children)
- MAASs request size: 256-1024 addresses
- Inter-request time: 1 hour - 8 days
- Lifetime: 15 days - 60 days
- Measurements:
  - G-RIB size
  - G-RIB flux
  - Space utilization

Conclusions

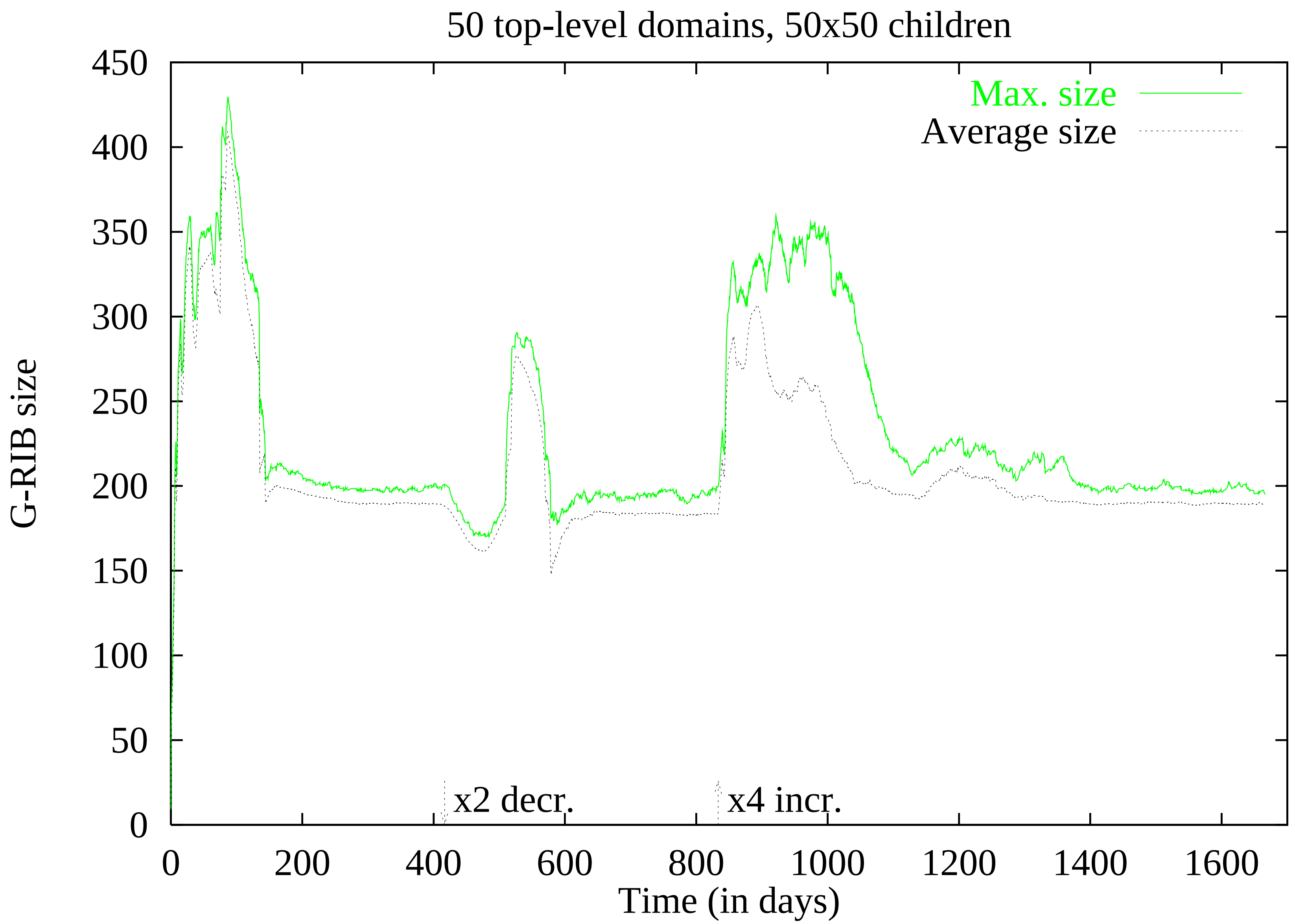
- Adaptive prefix allocation algorithm
- No more than two prefixes per domain (in relatively steady state)
- 50% overall address space utilization
- Good prefix reusability

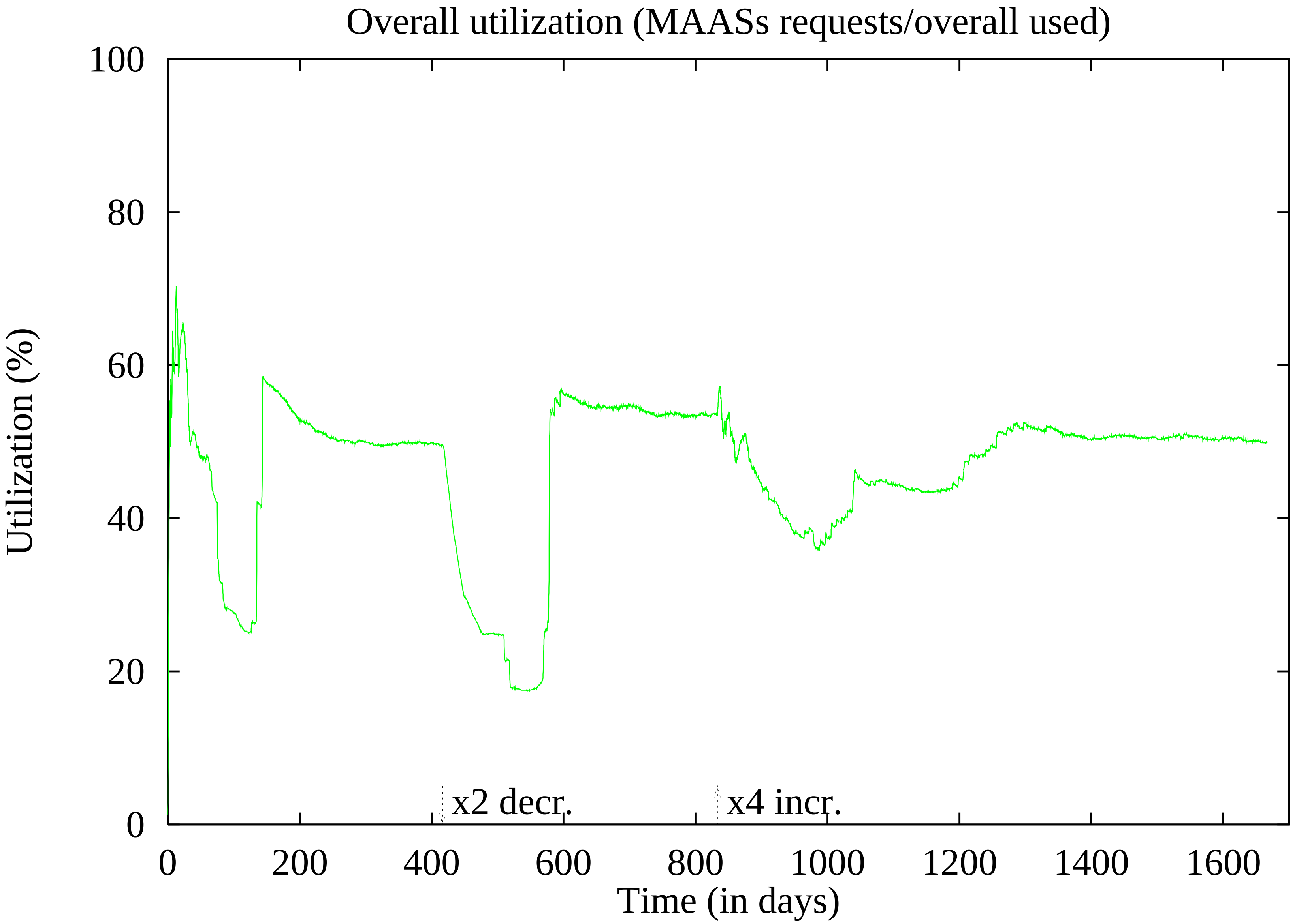


Overall demand pattern
Addresses requested by MAASs
7e+07
6e+07
5e+07
4e+07
3e+07
2e+07
1e+07
0
x2 decr.
x4 incr.
0
200
400
600
800
1000
1200
1400
1600
Time (in days)

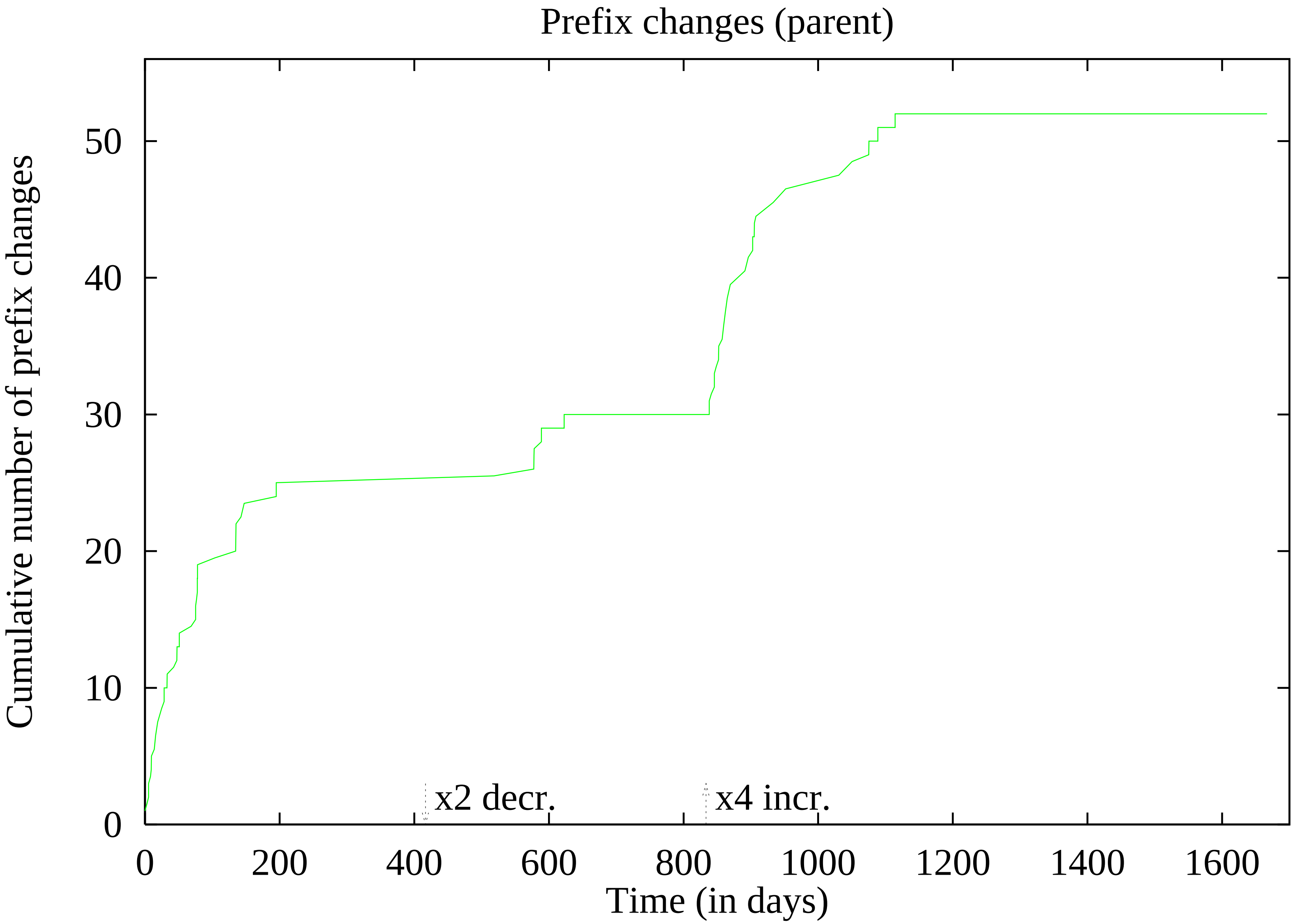


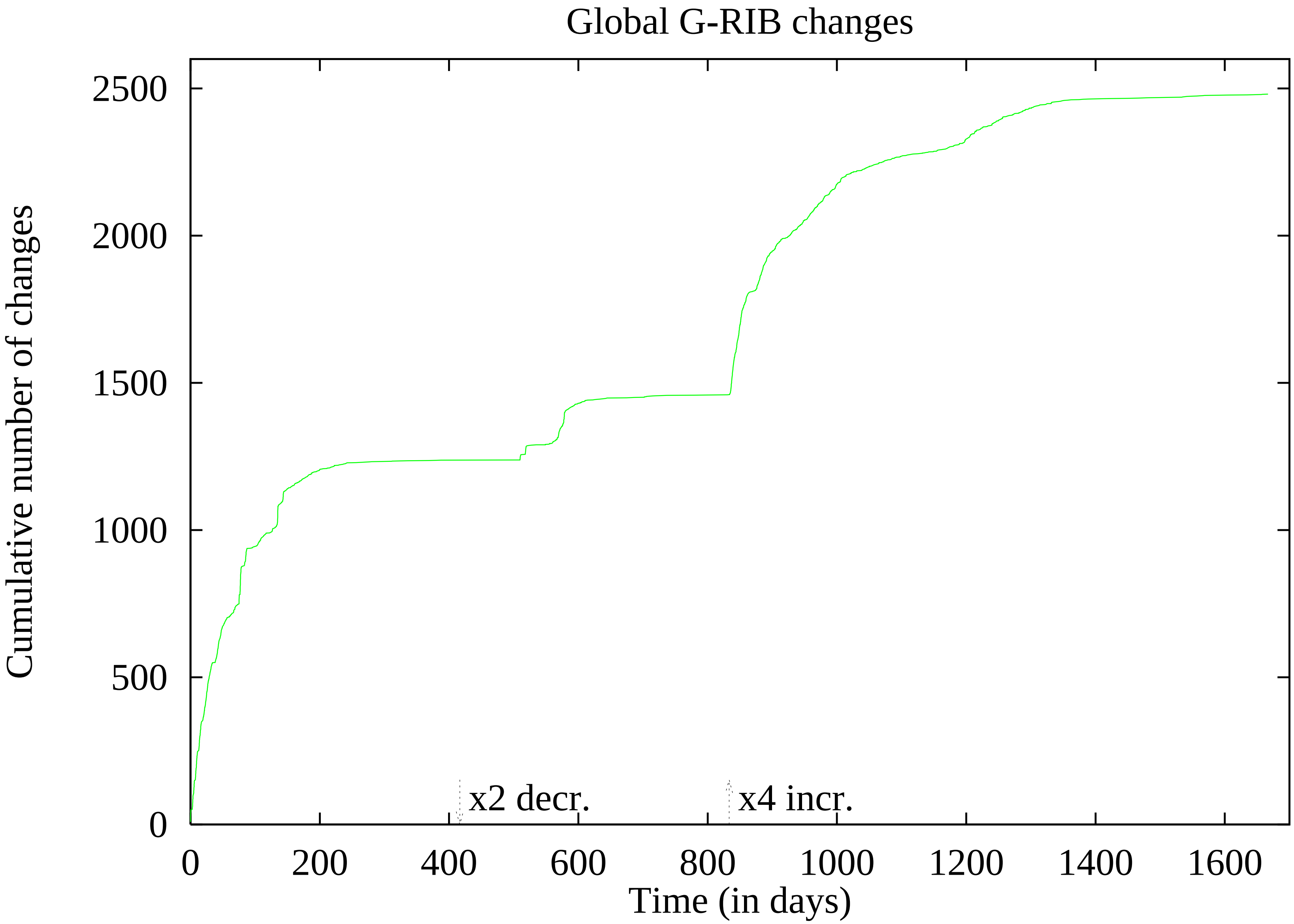
Demand pattern for a single node (child)
Addresses requested by MAASs
Time (in days)
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
0
200
400
600
800
1000
1200
1400
1600
x2 decr.
x4 incr.


50 top-level domains, 50x50 children
Max. size
Average size
G-RIB size
Time (in days)
x2 decr.
x4 incr.
0
50
100
150
200
250
300
350
400
450
0
200
400
600
800
1000
1200
1400
1600



Overall utilization (MAASs requests/overall used)
Utilization (%)
100
80
60
40
20
0
x2 decr.
x4 incr.
0
200
400
600
800
1000
1200
1400
1600
Time (in days)


Prefix changes (parent)
Cumulative number of prefix changes
0
10
20
30
40
50
0
200
400
600
800
1000
1200
1400
1600
Time (in days)
x2 decr.
x4 incr.


Global G-RIB changes
Cumulative number of changes
0
500
1000
1500
2000
2500
0
200
400
600
800
1000
1200
1400
1600
Time (in days)
x2 decr.
x4 incr.