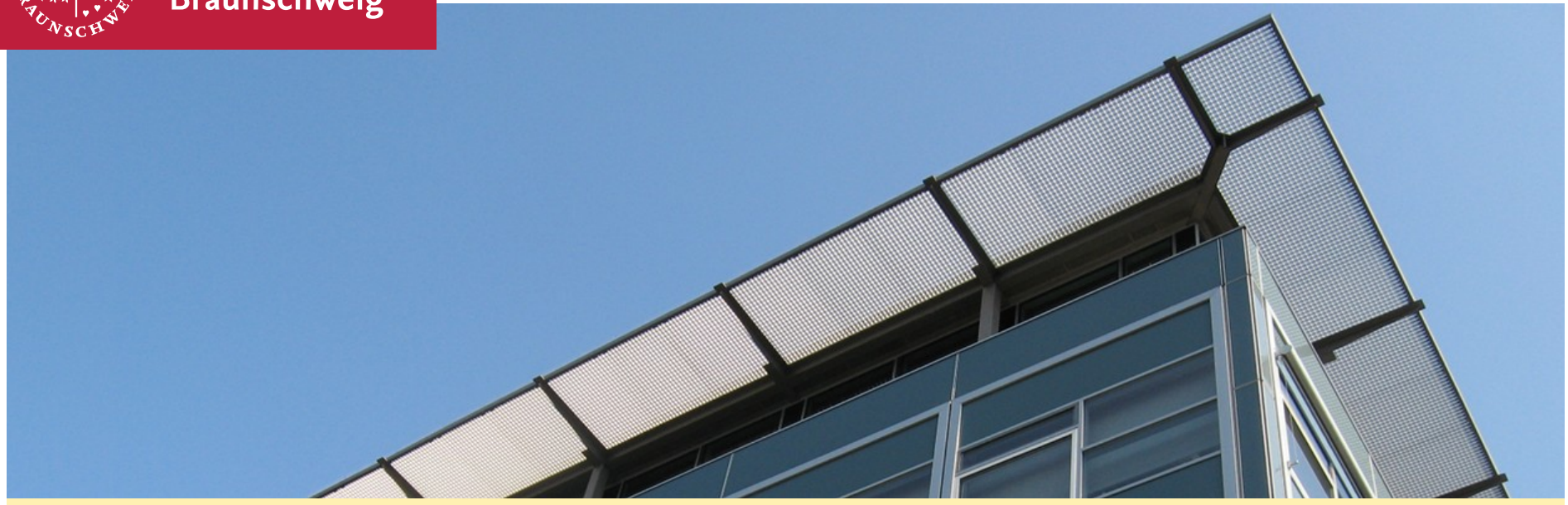
Technische Universität Braunschweig

Institute of Operating Systems and Computer Networks



NASDI - Naming and Service Discovery for DTNs in Internet Backbones

Sebastian Schildt, Wolf-Bastian Pöttner, Oliver Ohneiser, Lars Wolf,
Mobilware 2012, Berlin

DTN Primer

Delay & Disruption Tolerant Networking

- No continuous end-to-end connectivity in the network
- Interplanetary Networks (NASA)
- Mobile Sensor Networks
- Pocket Switched Networks (Mobile Phones)
- VANETs

A standardization effort for DTNs is underway in form of the Bundle Protocol, specified in RFC5050

Bundle Protocol: DTN-Internet Convergence

- Standard Bundle Protocol convergence layers (TCP-CL, UDP-CL) use internet standard protocols as transport
- DTNs are used in fringe networks, which have localized and intermittent Internet connectivity
  - Intermittent 3G connection on a mobile device
  - Internet enabled sink node in sensing apps

Bundle Protocol: DTN-Internet Convergence

- Standard Bundle Protocol convergence layers (TCP-CL, UDP-CL) use internet standard protocols as transport
- DTNs are used in fringe networks, which have localized and intermittent Internet connectivity
  - Intermittent 3G connection on a mobile device
  - Internet enabled sink node in sensing apps
- **Bad idea:** Use an application proxy at the border of the network to interface with “The Internet”

Bundle Protocol: DTN-Internet Convergence

- Standard Bundle Protocol convergence layers (TCP-CL, UDP-CL) use internet standard protocols as transport
- DTNs are used in fringe networks, which have localized and intermittent Internet connectivity
  - Intermittent 3G connection on a mobile device
  - Internet enabled sink node in sensing apps
- **Bad idea:** Use an application proxy at the border of the network to interface with “The Internet”
- **Good idea:** Use Bundle Protocol end-to-end

Bundle Protocol State of the Art

Routing

- Epidemic, PRoPHET, Direct-Contact, Static, ...
- Decide what to transfer when meeting a *neighbor*

Bundle Protocol State of the Art

Routing

- Epidemic, PRoPHET, Direct-Contact, Static, ...
- Decide what to transfer when meeting a *neighbor*

Neighbor (Discovery)

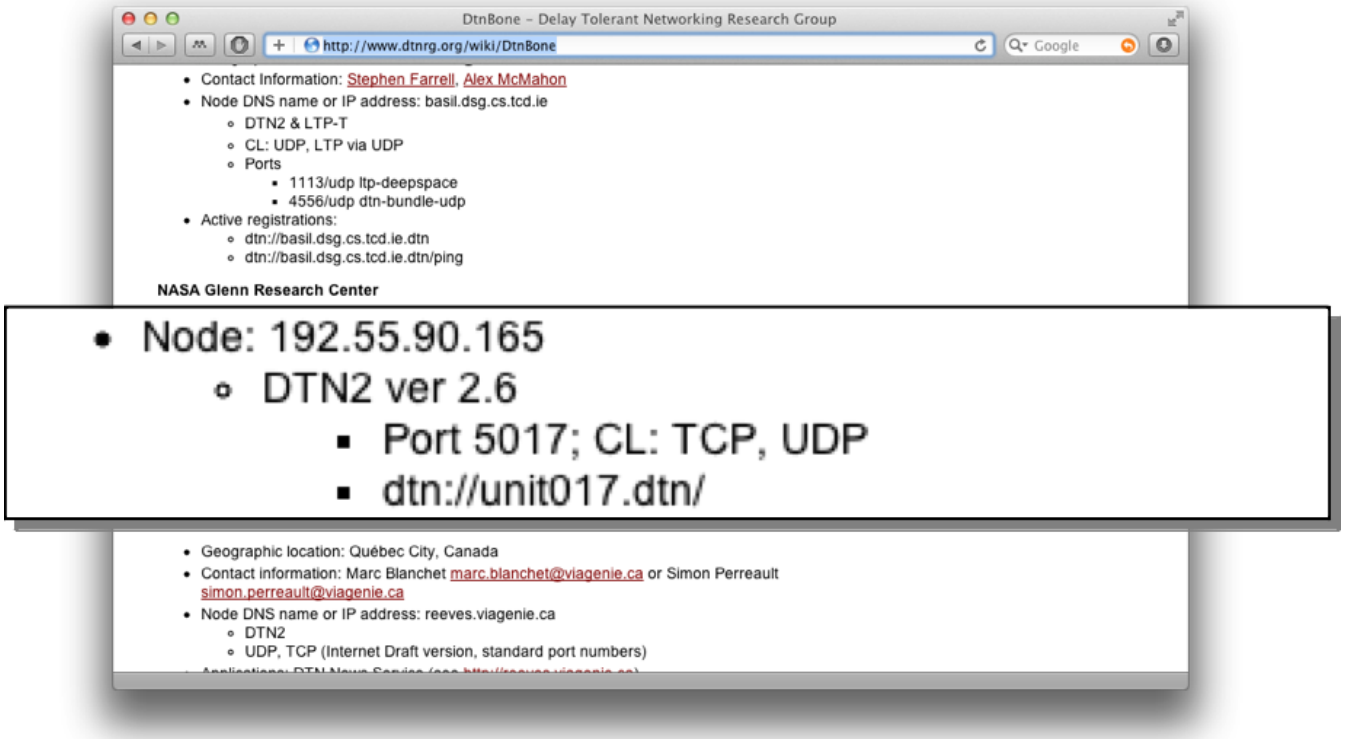
- Configure static contacts
- IP Neighbor Discovery (IETF Draft)
- Proprietary DTN2 neighbor discovery

The Result: DTNBone “WNS”

“ *dtnbone* [...] denote[s] our effort to establish a worldwide collection of nodes running DTN bundle agents and applications.”

The Result: DTNBone “WNS”

“ *dtnbone* [...] denote[s] our effort to establish a worldwide collection of nodes running DTN bundle agents and applications.”



Wiki **N**ame **S**ystem!

What is Needed?

Problem Statement

- We should be able to contact a Bundle Protocol DTN node by its *name*: the Endpoint Identifier (EID)
- This should work across the Internet

What is Needed?

Problem Statement

- We should be able to contact a Bundle Protocol DTN node by its *name*: the Endpoint Identifier (EID)
- This should work across the Internet

Requirements

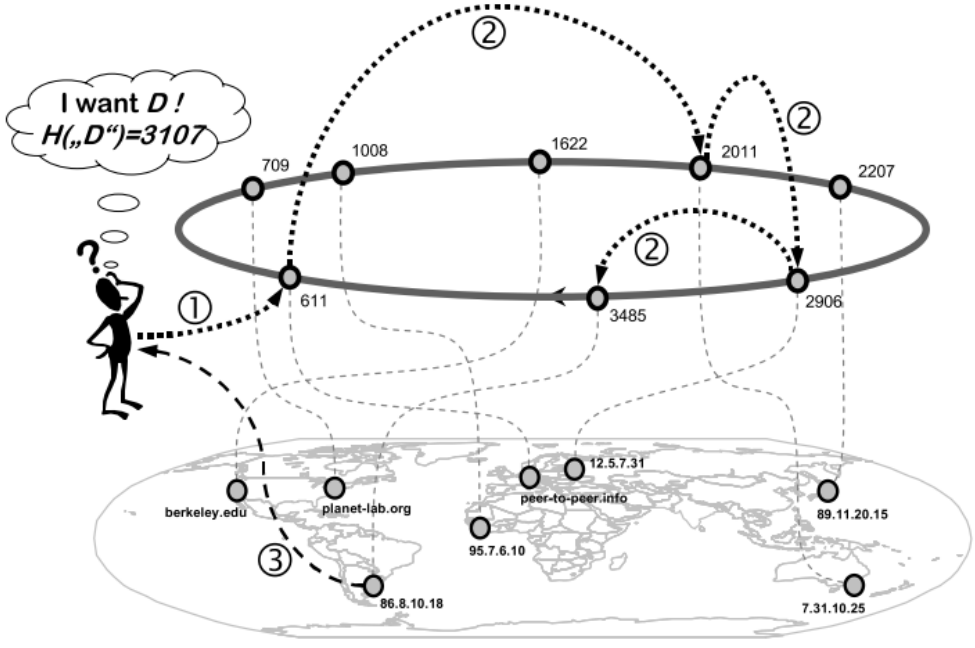
- No additional restrictions: EIDs are URIs offering a flat namespace, as the BP specification imposes no further semantics
- No additional infrastructure: Nobody wants to operate or pay for a DNS-like infrastructure

Solution Outline

Use a Distributed Hash Table (DHT) to store naming information.

Solution Outline

Use a Distributed Hash Table (DHT) to store naming information.



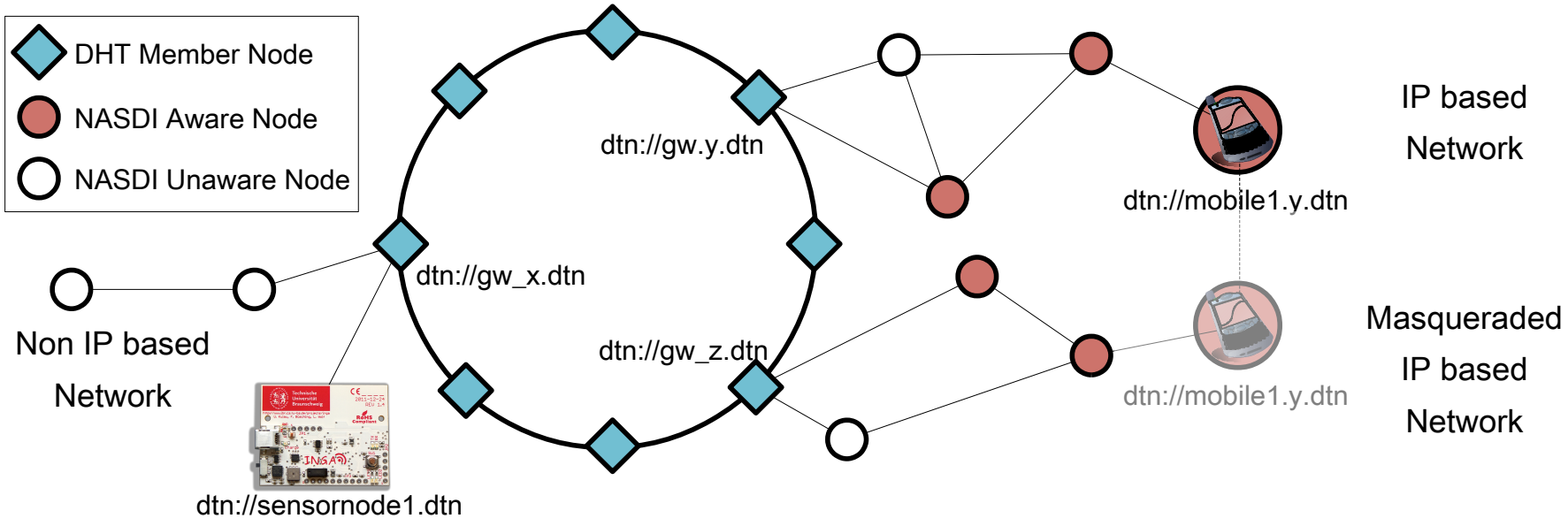
Steinmetz, R., & Wehrle, K. (2005). Peer-to-peer systems and Applications

DHT=P2P, decentralized, scalable ($\sim O(log(n))$ node state and communication steps for retrieval)

NASDI Basics

- DHT solution based on Kademlia
  - MaidSafe DHT `BSDlicensedKademliaimplementation`
  - Advanced NAT traversal
- Sophisticated timeout mechanisms
- Beneficial for nodes not implementing NASDI
- High-speed asynchronous change notifications

NASDI Overview



NASDI Record

val_{stored}

Key	ABBC2134	
Value	time_to_live	1000
	time_since_last_seen	100
	time_refresh_passive	200
	type_entry	SINGLE
	type_information_list	TCP, UDP

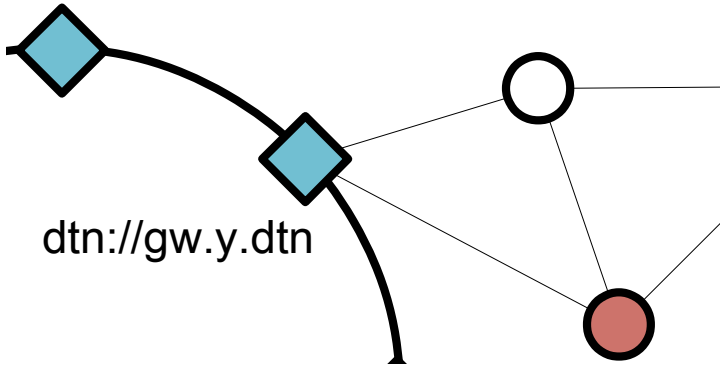
- key: Hashed EID (H(`dtn://node1.dtn`))
- type_entry: Entry for a node (not a group)
- type_information_list: TCP or UDP connectivity information

NASDI Timeouts

- time_to_live: Expected minimum validity for this entry in seconds
- time_since_last_seen: Seconds since the last time this entry was updated
- time_refresh_passive: Timeout after which the publisher promises to refresh the entry

time_since_last_seen > *time_refresh_passive* indicates connectivity problems

NASDI Proxied Nodes



A node can

- Request being proxied by another node
- Opt-out from being proxied via IPND (i.e. if it does not want to be discoverable by outisde nodes)
- Be proxied automatically if a neighbor's NASDI implementation is set to publish neighbors (which do not opt-out) automatically

NASDI Asynchronous Notification

Situation

In a DTN it is to be expected that nodes will not be available at any given moment

Consequences

- A node wants to connect a certain EID, but cannot find any connectivity information in the DHT
- A node has connectivity information but it is outdated (i.e. the node cannot be connected)

The node wants to be notified as soon as possible, once the target becomes available (no polling)

NASDI Asynchronous Notification

1. Store a notification request using the key belonging to the EID we are interested in
   - Include our own EID
   - Describe which events we are interested in
   - Specify whether this is a one-shot or continuous notification

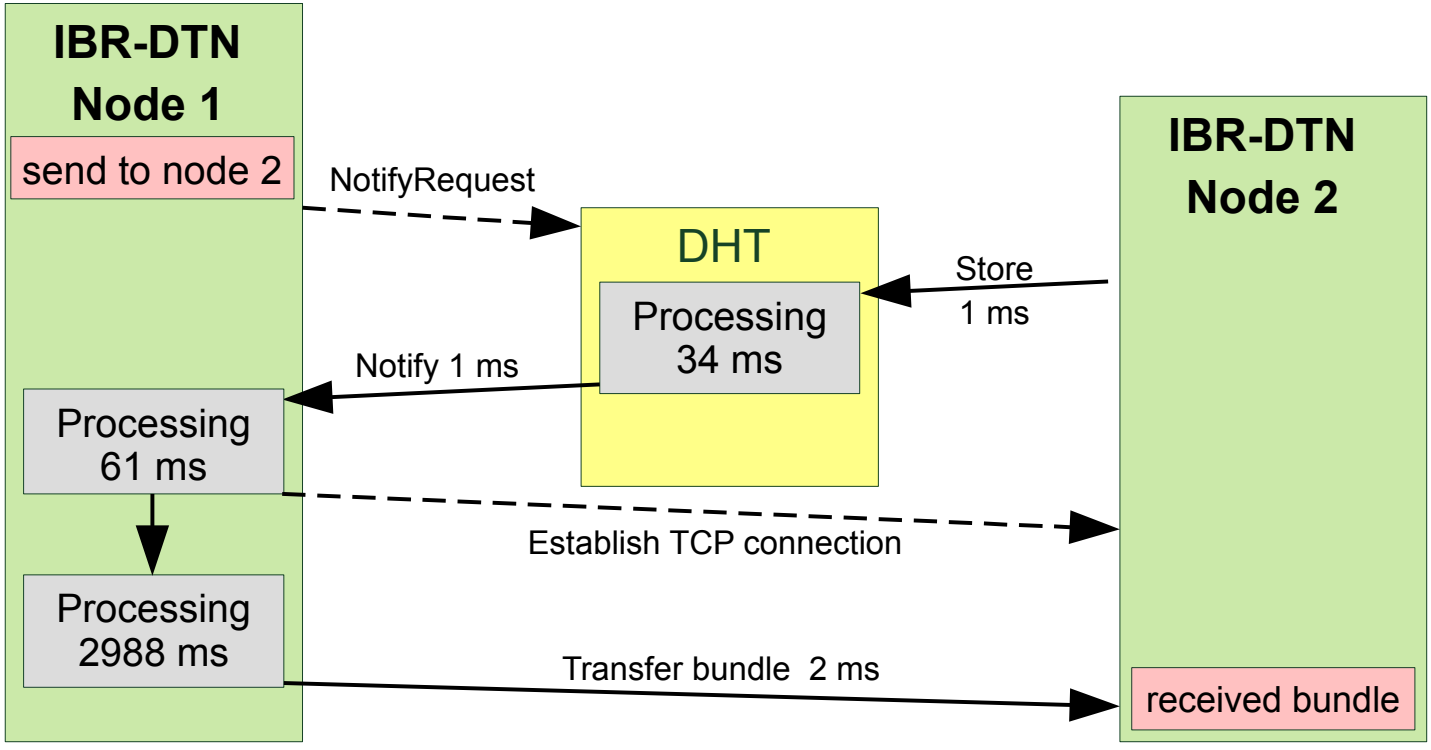
NASDI Asynchronous Notification

1. Store a notification request using the key belonging to the EID we are interested in
   - Include our own EID
   - Describe which events we are interested in
   - Specify whether this is a one-shot or continuous notification
2. Once the node rejoins the network or its entry is updated, this request will be routed to the same DHT member that stored the EID

NASDI Asynchronous Notification

1. Store a notification request using the key belonging to the EID we are interested in
   - Include our own EID
   - Describe which events we are interested in
   - Specify whether this is a one-shot or continuous notification
2. Once the node rejoins the network or its entry is updated, this request will be routed to the same DHT member that stored the EID
3. Upon a store request each DHT member checks, whether there are any pending notification requests for that key. If available, the interested node will be contacted immediately

NASDI Notification Latency



Conclusions

Benefits

- Versatile **N**aming **a**nd **S**ervice **Di**scovery for BP DTNs
- Based on a fully featured, stable DHT implementation
- Fast notification mechanism

Conclusions

Benefits

- Versatile **N**aming **a**nd **S**ervice **Di**scovery for BP DTNs
- Based on a fully featured, stable DHT implementation
- Fast notification mechanism

Challenges

- Lots of of dependencies for the DHT library (embedded devices)
- Chicken/Egg problem in deployment: How to reach critical mass?

Conclusions

Benefits

- Versatile **N**aming **a**nd **S**ervice **Di**scovery for BP DTNs
- Based on a fully featured, stable DHT implementation
- Fast notification mechanism

Challenges

- Lots of of dependencies for the DHT library (embedded devices)
- Chicken/Egg problem in deployment: How to reach critical mass?

Current State

- Already reimplemented main parts of the NASDI concept in top of the BitTorrent DHT (currently shipped with IBR-DTN)

Conclusions

Benefits

- Versatile **N**aming **a**nd **S**ervice **Di**scovery for BP DTNs
- Based on a fully featured, stable DHT implementation
- Fast notification mechanism

Thank you!

Current State

- Already reimplemented main parts of the NASDI concept in top of the BitTorrent DHT (currently shipped with IBR-DTN)