



Technische
Universität
Braunschweig

Institut für Betriebssysteme
und Rechnerverbund



An Introduction to Disruption Tolerant Networking (DTN)

V-Charge Summer School

Julian Timpner, Stephan Rottmann, IBR, TU Braunschweig
July 10, 2014

DTN vs. Classical (Wireless) Networking



Argh, those dreaded mobility! Always a big headache with nodes moving around getting out of range and stuff!

What a wonderful thing mobility is. It helps me spreading data.



DTN vs. Classical (Wireless) Networking



Links and routes can fail anytime. Luckily I am prepared for this horrible incidents by using sophisticated route recovery and repair mechanisms!

You are a “horrible incident”. Links will fail all the time. That’s just the way wireless networks are.



DTN vs. Classical (Wireless) Networking



I have so many powerful ways of routing. I can find shortest paths, most reliable paths, use location, interest and much more to make your data arrive!

I can do all that. And if, even with all those measures, I can't deliver your packet directly, I will travel into the future to deliver it!



DTN vs. Classical (Wireless) Networking



I am the standard. The whole internet builds on my principles. After all, most networks, even wireless ones, are more or less reliable and connected, and this is my domain!

Networks for sissies? I am just as good as you in those networks!



DTN vs. Classical (Wireless) Networking



Ordinary Networking	DTN Networking
Mobility is a challenge	Mobility is exploited
Link failures are handled as accidents	Failing links are considered normal
Routing through space	Routing through space and time
The standard	Superset of ordinary networking



In the beginning...

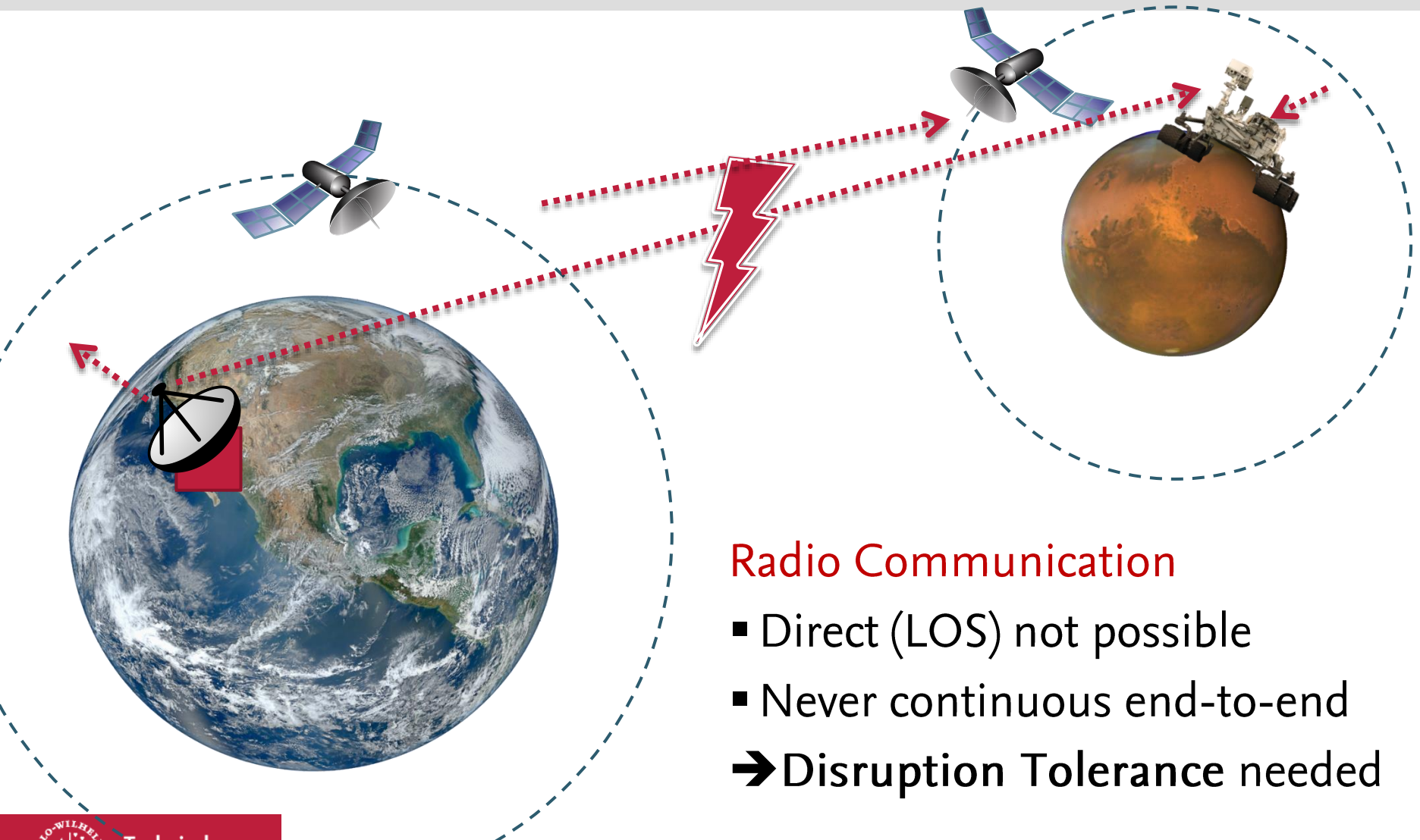
A network for Interplanetary Networks

- Originating in NASA surroundings
- Small problem: Since currently FTL transmission are not possible, there are long delays communicating in space
 - It already takes 8 min to reach the sun...
 - Common protocols do not work (well) with such high delays
- Other effects, like orbiting spacecraft disappearing behind the horizon of stellar objects, yield even longer delays
 - Can not always send data when they are available

Interplanetary Networks

- Long delays
- Highly deterministic
- Develop general protocols and paradigms for networks with large delays and temporary disruptions
- Story-Carry-Forward paradigm: Store your data until the next contact comes into radio range

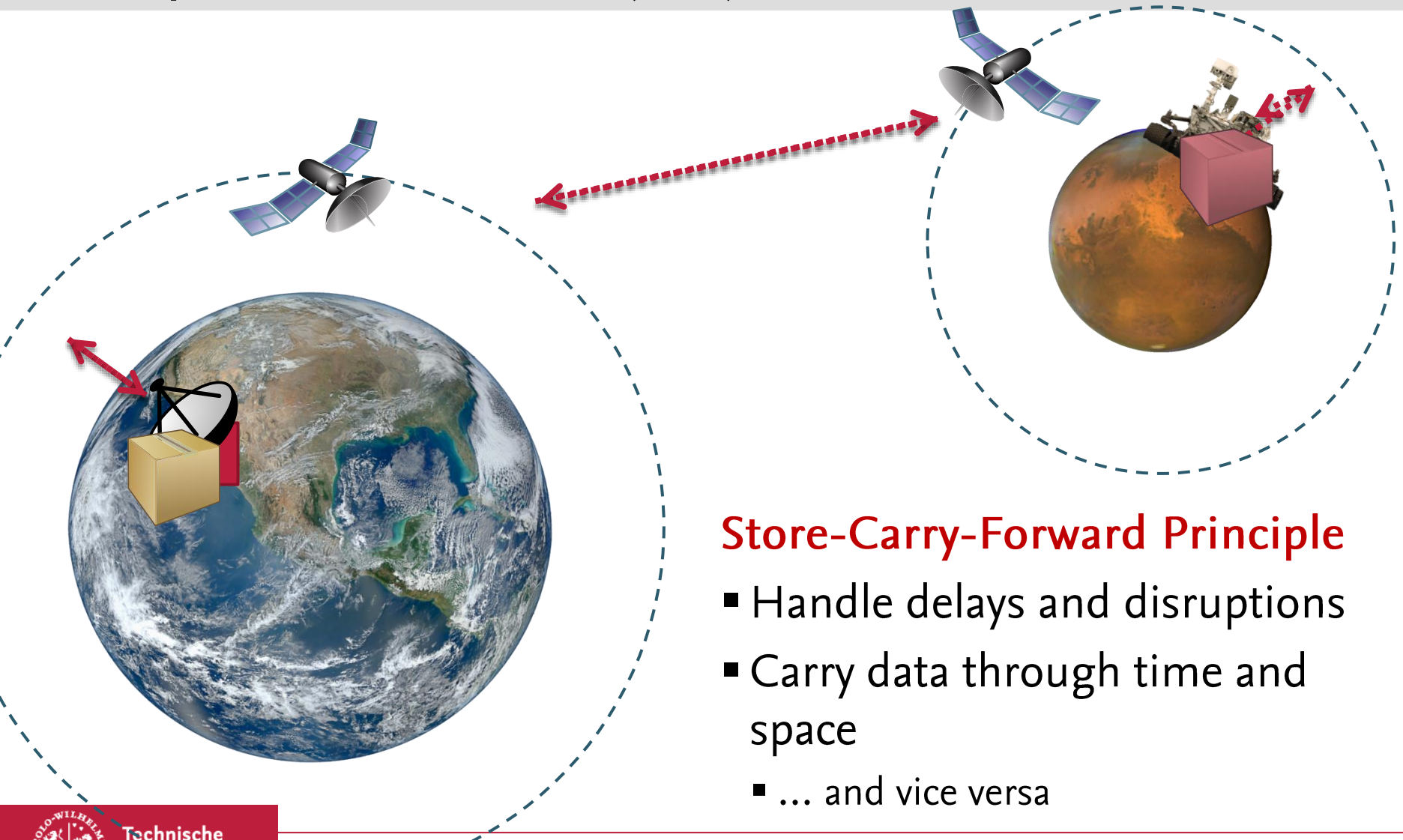
Interstellar Communication



Radio Communication

- Direct (LOS) not possible
 - Never continuous end-to-end
- ➔ **Disruption Tolerance needed**

Disruption Tolerant Networks (DTN)



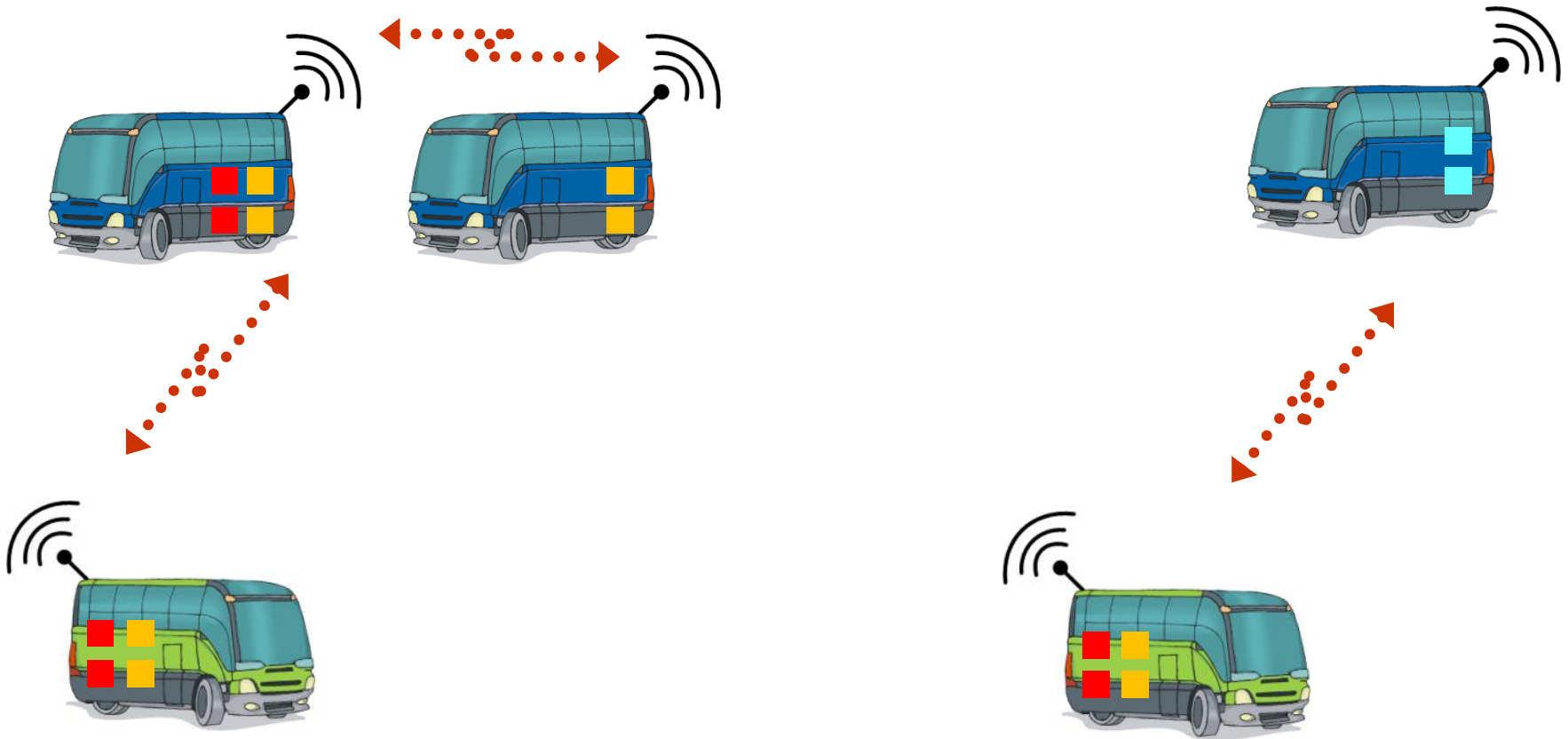
Store-Carry-Forward Principle

- Handle delays and disruptions
- Carry data through time and space
 - ... and vice versa

Extending the concept to dynamic wireless networks

- Disrupted networks also quite common in earth-bound scenarios
 - Lots of wireless networks with mobile participants (WiFi, WSNs, vehicular networks, etc.)
 - Mostly unstructured with no static topology
- Adapting the store-carry-and-forward methodology can make networks useful, that couldn't be used before
 - More opportunities to communicate
 - Much higher delays, end-to-end acknowledgements hard
- Applications need to be aware of this

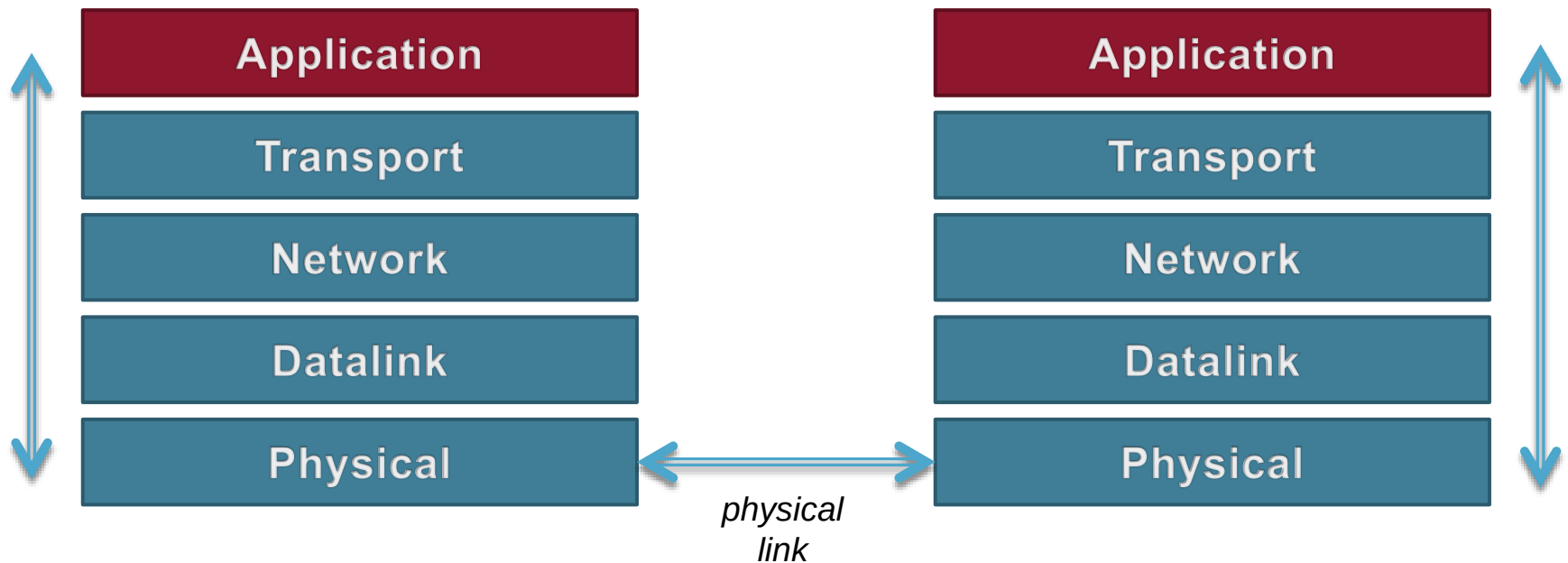
Store-Carry-And-Forward



Architecture

Overlay Network

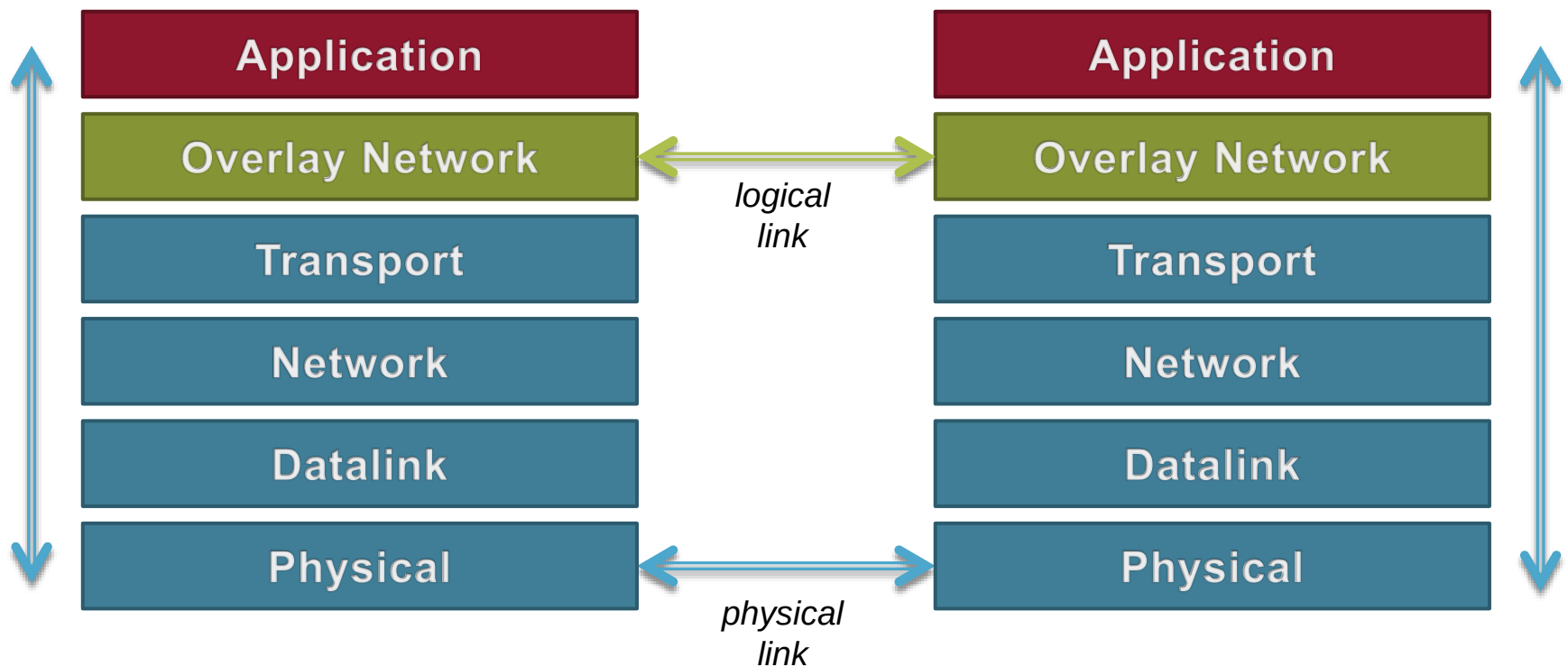
„An overlay network is a computer network which is built on the top of another network.“



Architecture

Overlay Network

„An overlay network is a computer network which is built on the top of another network.“

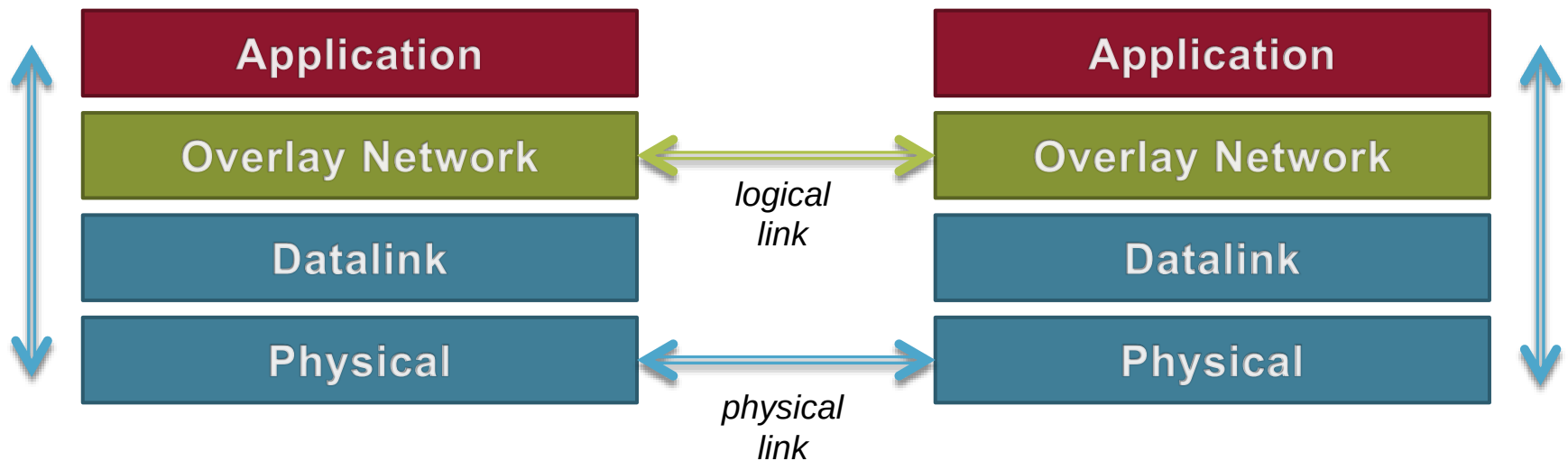


Architecture

Overlay Network

„An overlay network is a computer network which is built on the top of another network.“

The transport- and network layer is not necessary if the overlay network already implements the required features.



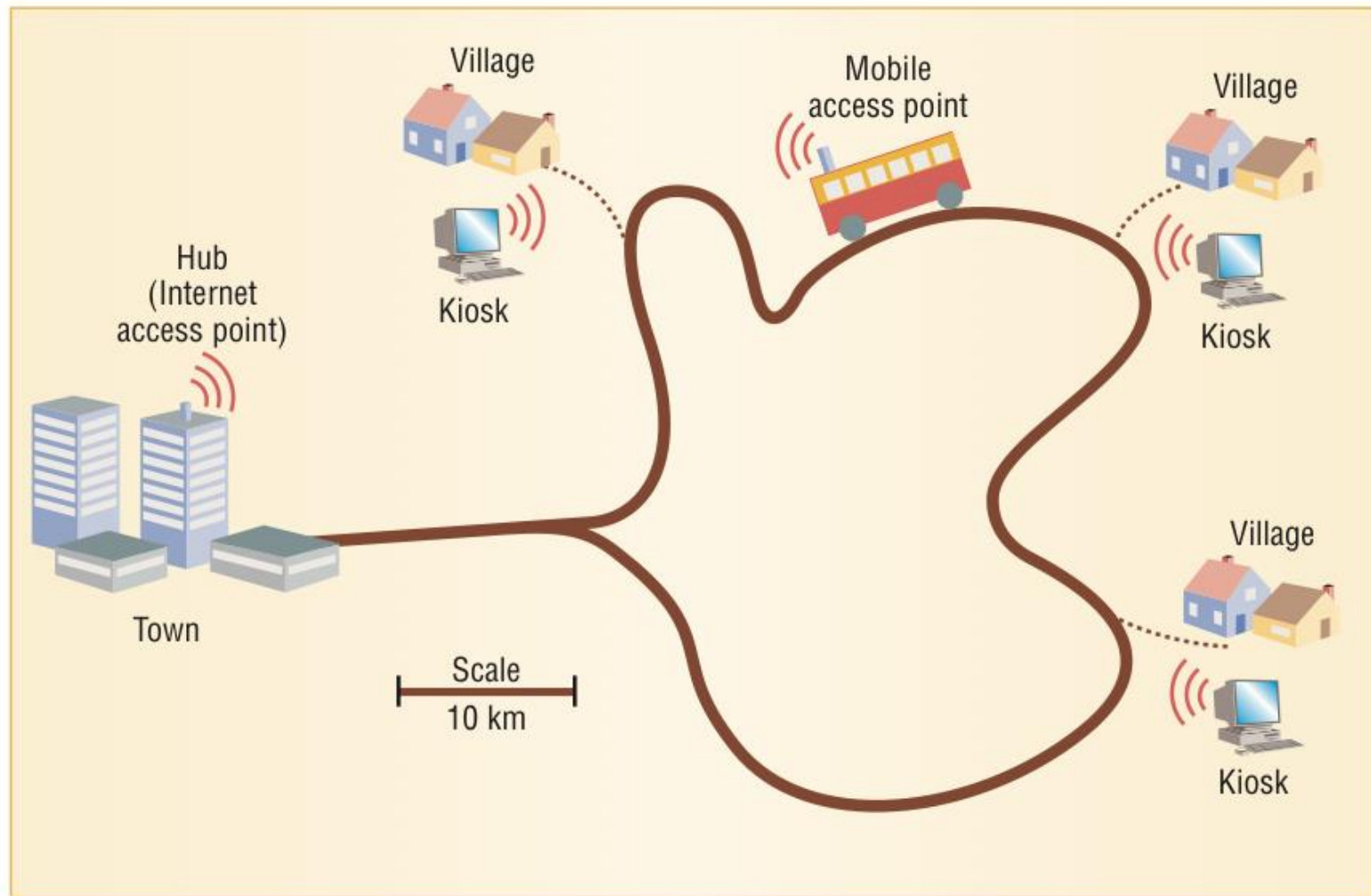
DakNet / FirstMileSolutions (2004)

- Goal: Enable the use of Internet applications in rural areas with no infrastructure
- Can be used if even wireless directional radio is not economically feasible
- Idea: Use vehicles as data-mules
- Developed by the MIT Media Lab
- Now commercialized by FirstMileSolutions in India



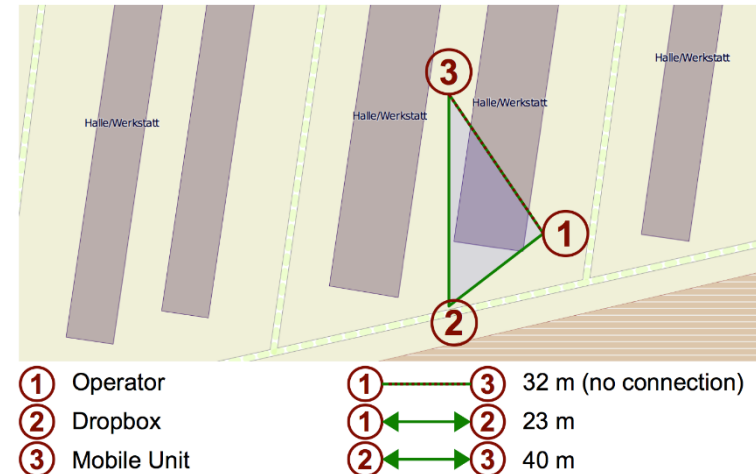
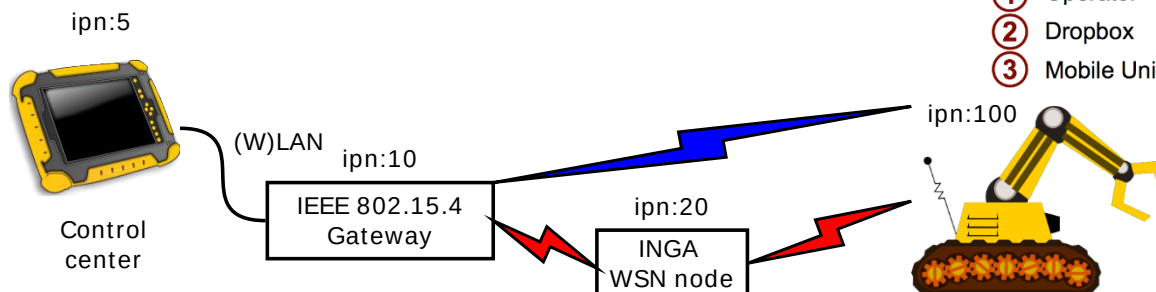
<http://www.firstmilesolutions.com/>

DakNet Architecture

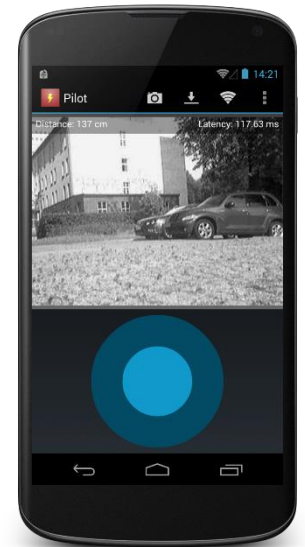
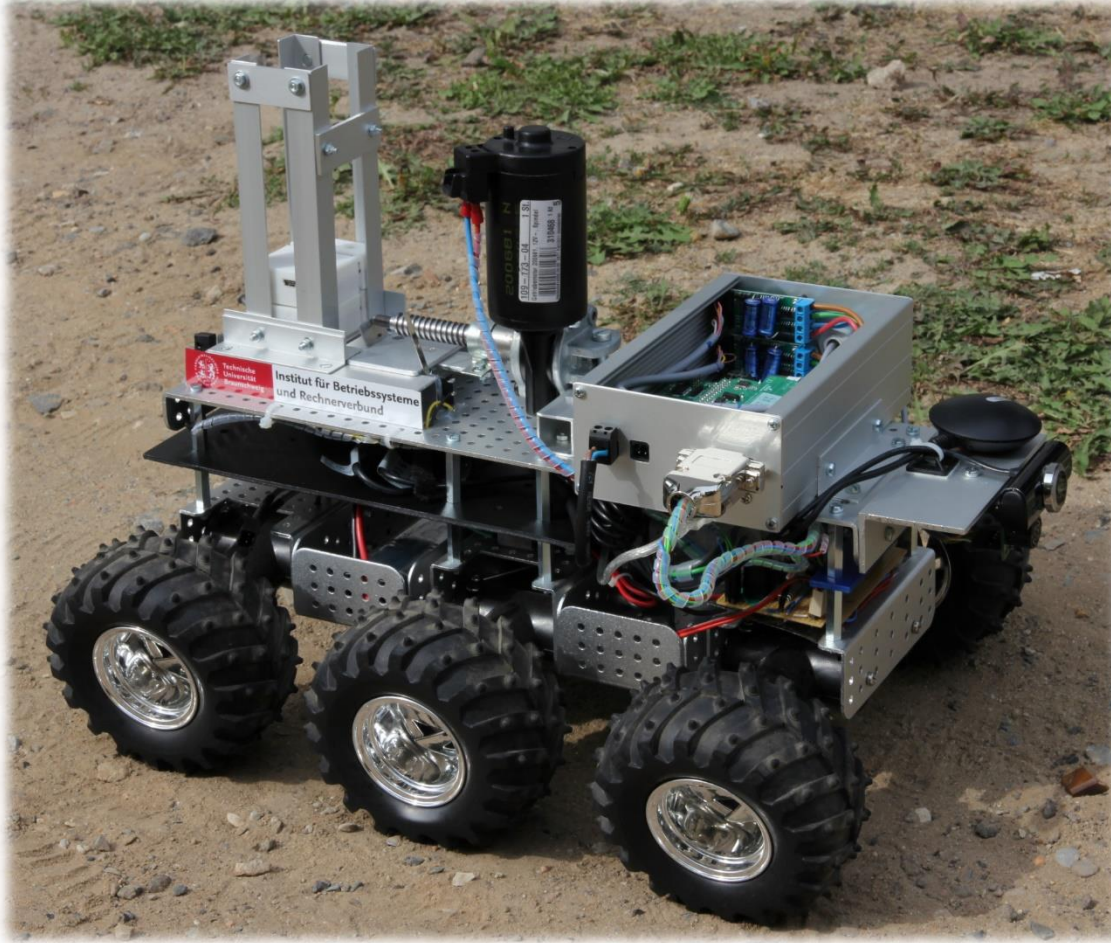


RoboticFireFighters (2013)

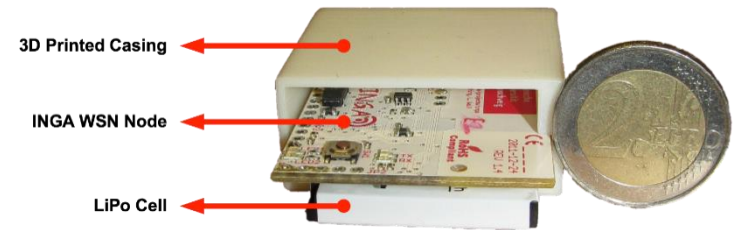
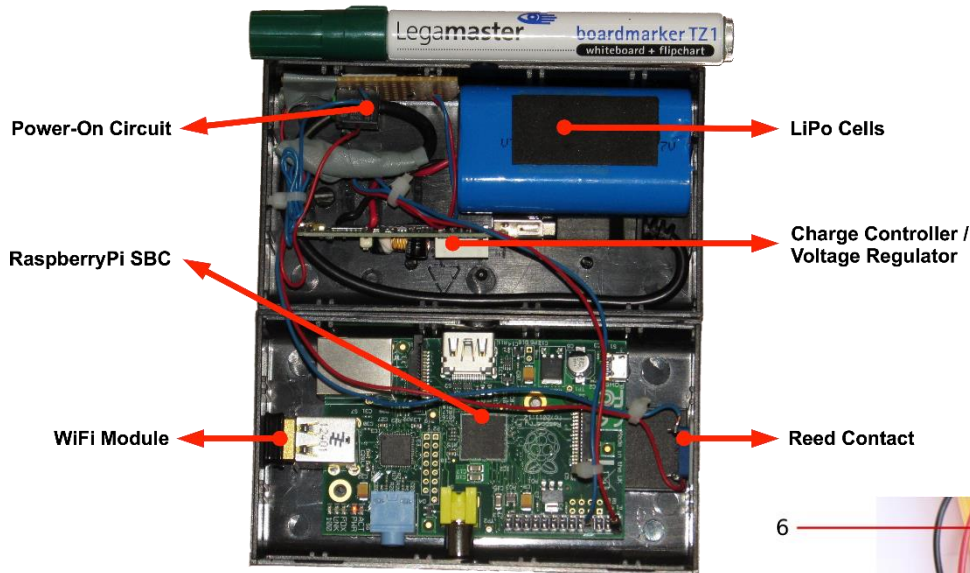
- Robotic SAR missions
- Autonomous behavior
- Deploying WSN nodes or DTN nodes
- Realize links for direct control and telemetry information
- Improve data exchange



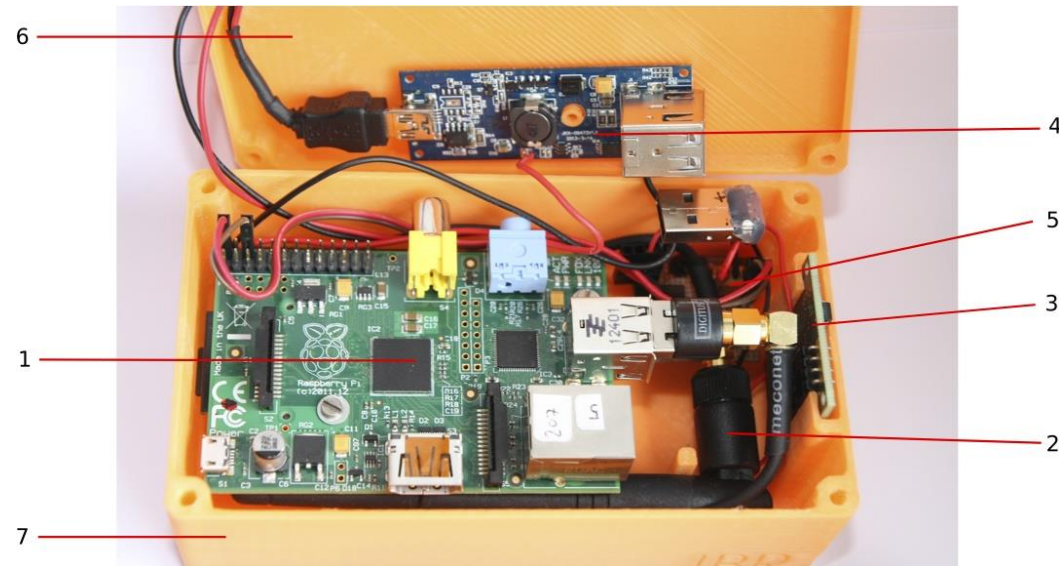
RoboticFireFighters Test Platform



RoboticFireFighters – DTN Dropbox and WSN Dropbox



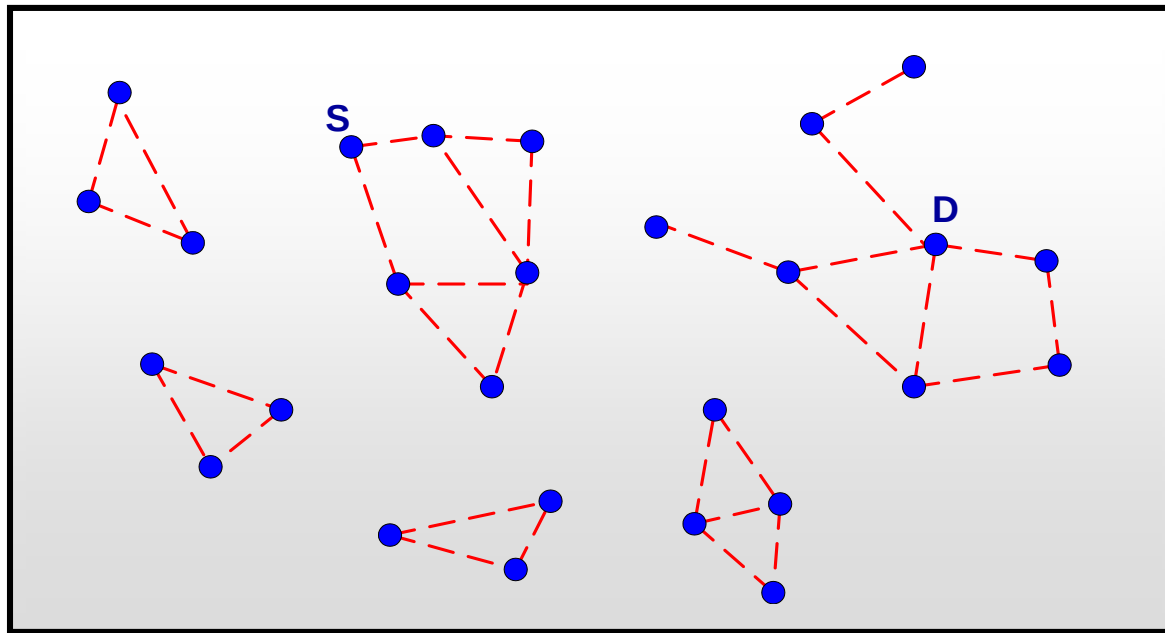
1. Raspberry Pi SBC
2. WiFi Antenna
3. Inductive Charging Receiver
4. Lilo Charge controller
5. Power-On circuit
6. Custom Case – top
7. Custom Case – bottom



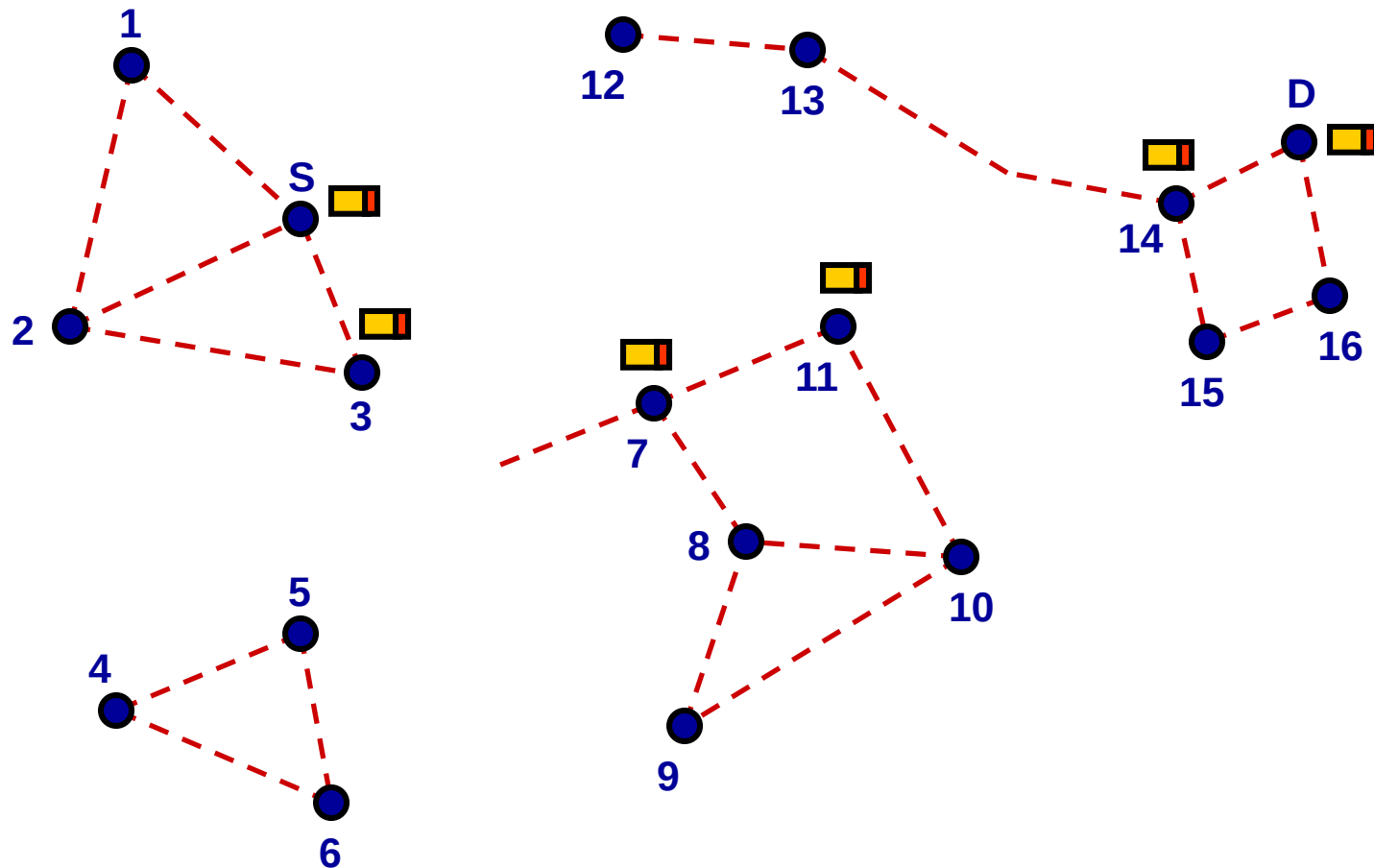
- General Routing Properties and Challenges
- Simple Routing Approaches
(Direct Delivery , First Contact, Epidemic)
- Spreading based Routing
(Spray & Wait, Spray & Focus)
- Probabilistic based Routing
(ProPHET)
- Knowledge based Routing
(RUTS)

Characteristics of Delay and Disruption Tolerant Networks

- Intermittent connectivity
 - Node mobility
- Traditional Internet or ad-hoc protocols would fail



Example of Store-and-Forward Routing



Main Issue: What is an “appropriate” next hop?

Routing Paradigm

- Exploit node mobility to deliver bundles
- **Store-and-forward** approach
 - A node stores a bundle until an appropriate communication opportunity is available
 - A series of independent forwarding decisions {time + next hop} that will eventually bring the bundle to its destination

Routing Challenges

Performance

- Bundle delivery ratio ↑
 - The fraction of generated bundles that are correctly delivered to the final destination within a given time period
- Transmission delay ↓
 - The time from a bundle is generated to it is received at the destination
- Number of transmissions ↓
 - The number of bundle exchanges between any two nodes

Other Parameter

- Energy consumption, storage space, ...

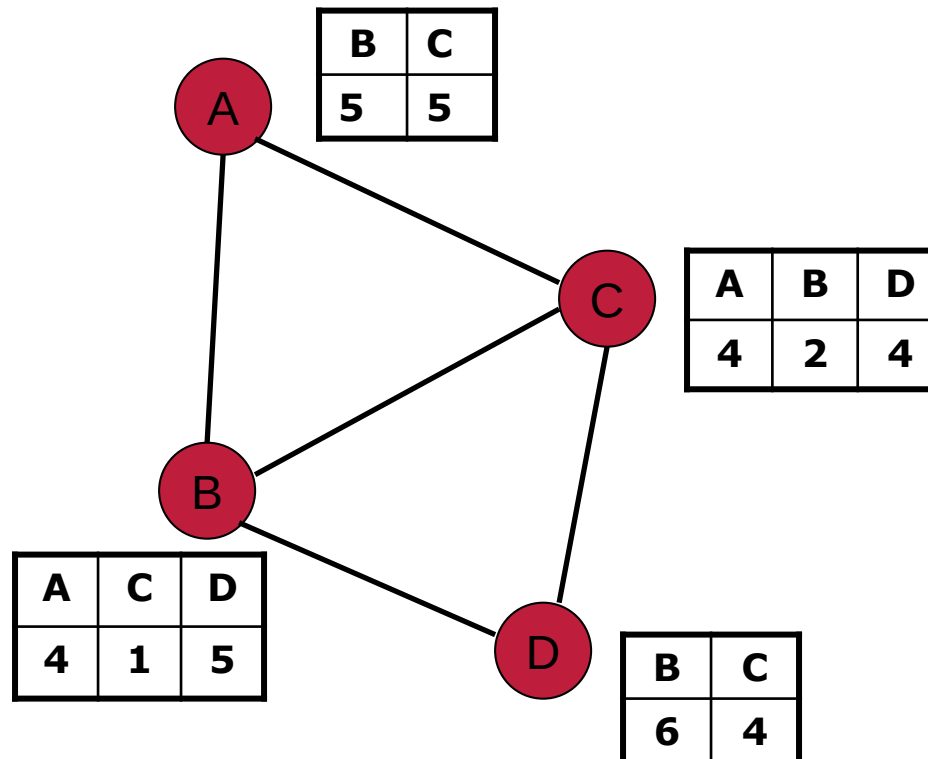
Bundle Forwarding Strategies

General Simple Approaches

- **Flooding:** bundle distribution to all neighbors
- **Direct delivery:** delivers bundle directly from source to destination
- **Hot-Potato:** forward bundles to neighbors as quickly as possible

Bundle Forwarding Strategies – Probabilistic based

Forwarding based on delivery likelihood



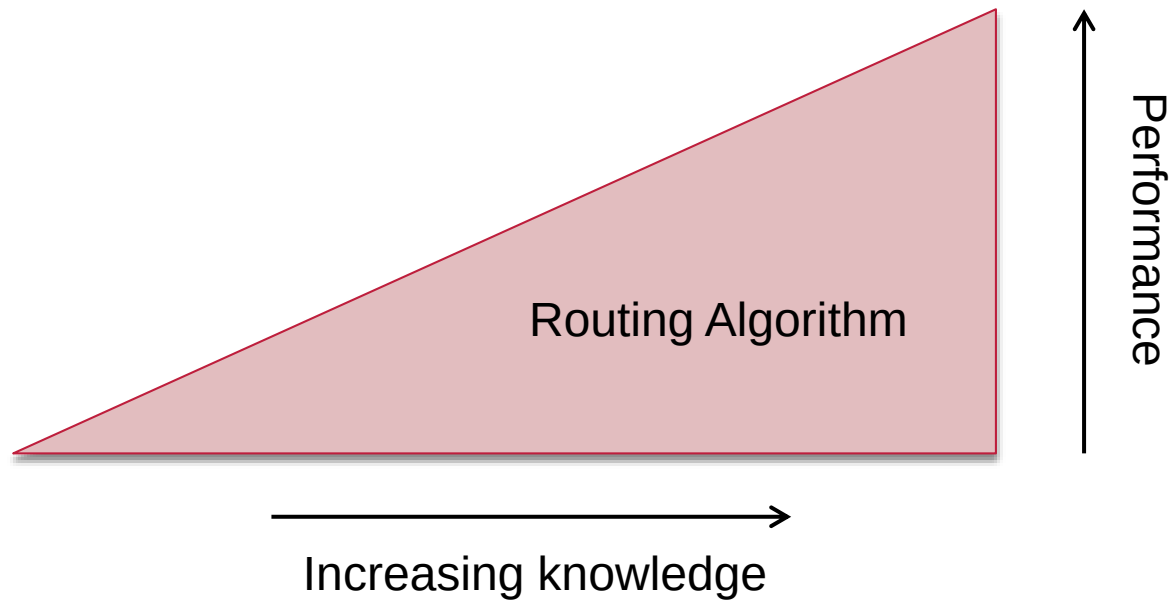
Bundle Forwarding Strategies – Knowledge based

Oracle uses topology information to predict future topology, e.g.

- Mobility pattern
- Contact probabilities
- Contact history
- Context based

Knowledge versus Performance

- Increasing performance with more acquired knowledge
- Routing based on node mobility and available knowledge



Bundle Copies

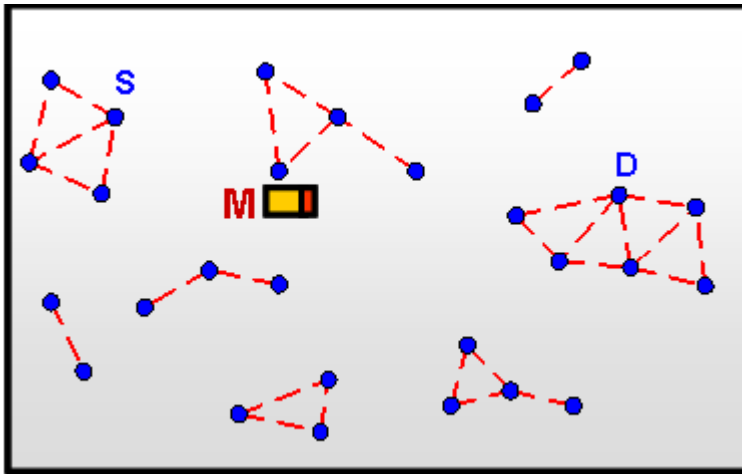
Single-Copy

Only a single copy of each message exists in the network at any time

Multiple-Copy

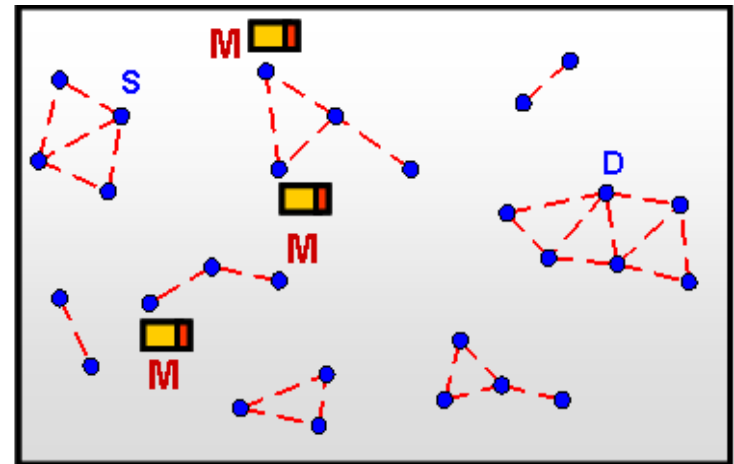
Multiple copies of a message may exist concurrently in the network

Single Copy



- + lower number of transmission
- + lower contention for shared resources

Multiple Copy



- + lower delivery delay
- + higher robustness

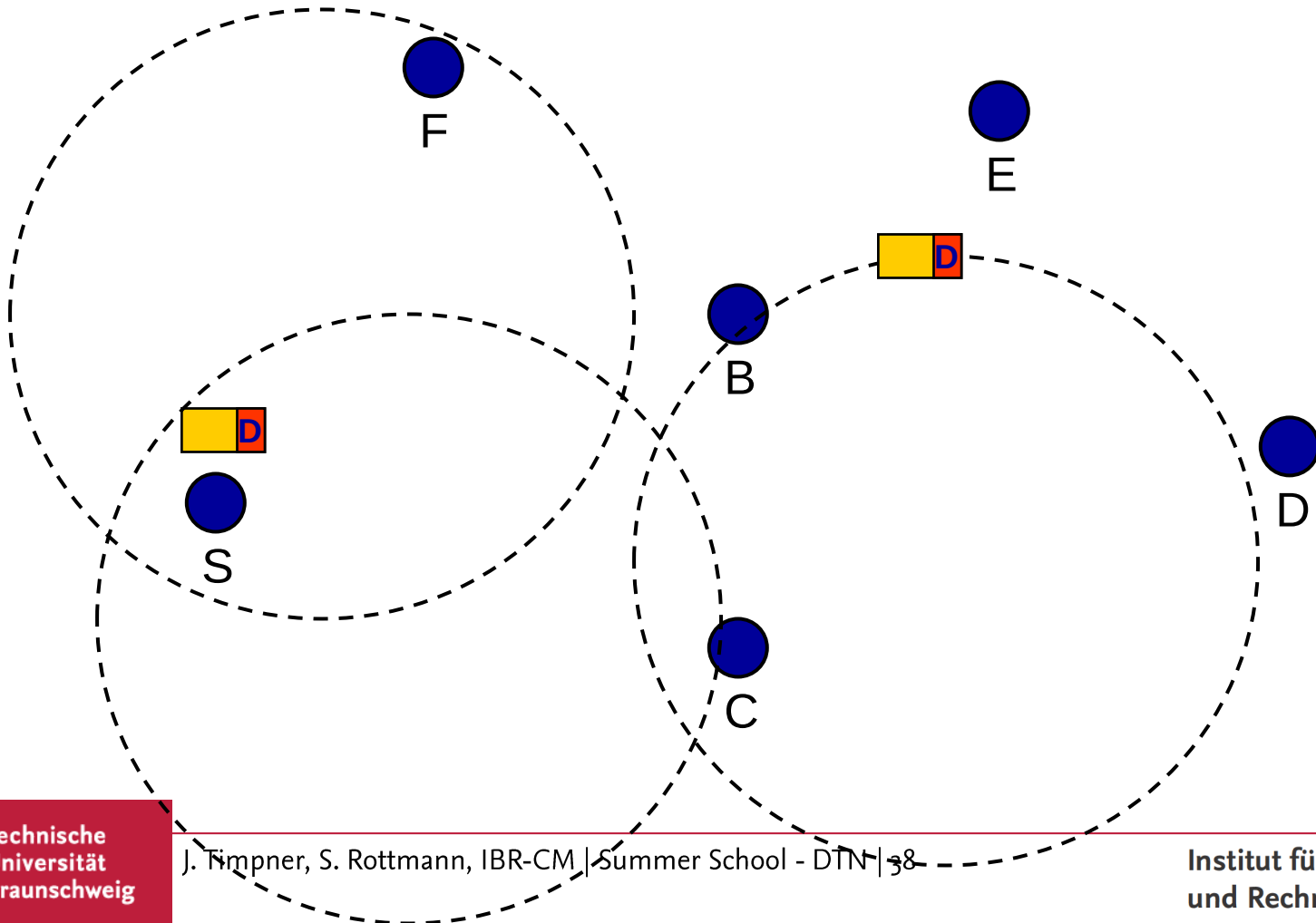
Part II Routing - Overview

- General Routing Properties and Challenges
- Simple Routing Approaches
(Direct Delivery, First Contact, Epidemic)
- Spreading based Routing
(Spray & Wait, Spray & Focus)
- Probabilistic based Routing
(ProPHET)
- Knowledge based Routing
(RUTS)

Direct Delivery

Forward bundles only to its destination

- Simplest strategy and minimizes transmissions



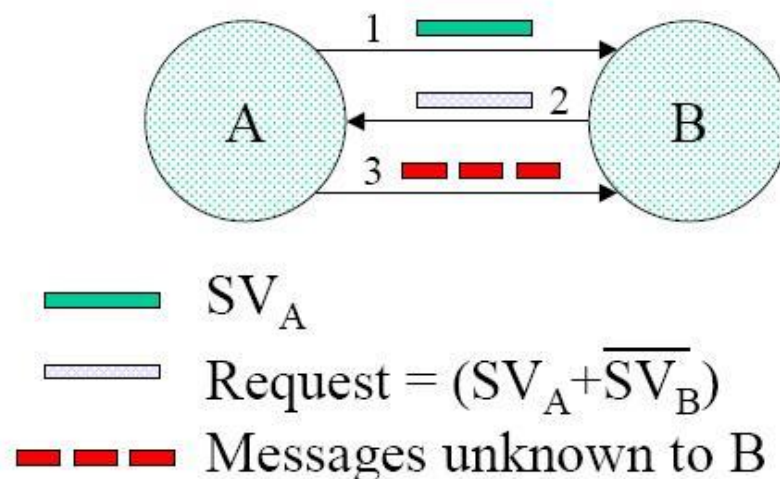
First Contact

- Node forwards the message to next available node, like a hot-potato algorithm
- Next hop is chosen randomly from the available neighbors
- Bundles can circulate among a group of two or more peers for a long time until the one having the message leaves the group
 - Improvement: Use path vector to prevent loops

Epidemic Routing

The overall goals of Epidemic Routing

- Maximize message delivery rate
- Minimize message delivery latency
- Minimizing the consumed system resources



Epidemic Routing (2)

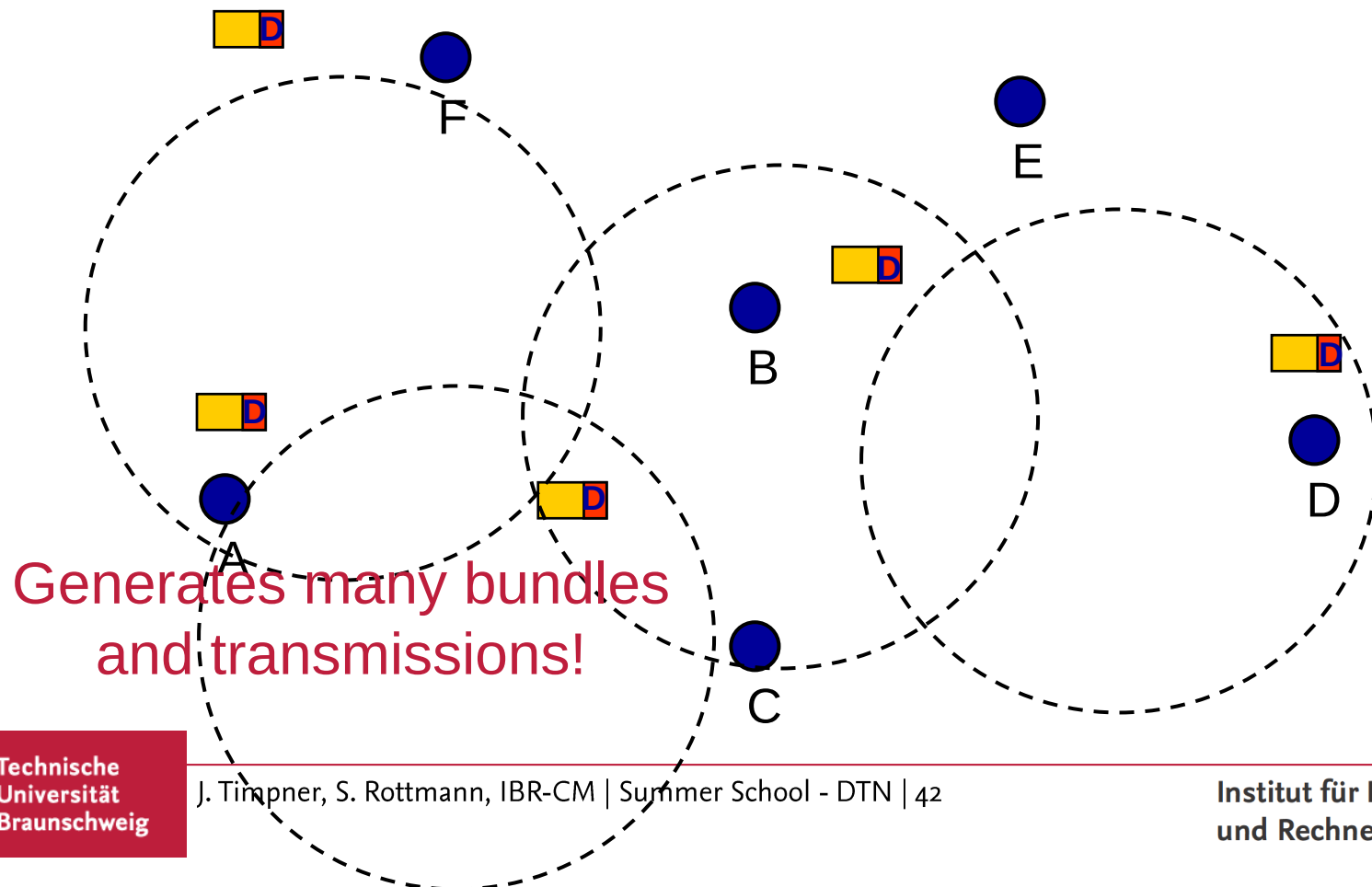
Routing Approach

- Host with smaller identifier initiates connection
- Hash table indexes bundle list
- Summary Vector indicates entries in local hash table
- Exchange of the summary vectors
- Negation of summary vector represents the needed messages
- Hosts request messages
- Cache to avoid redundant connections

Epidemic Routing (3)

Give a message copy to every node encountered

- Essentially: flooding in a disconnected context



Part II Routing - Overview

- General Routing Properties and Challenges
- Simple Routing Approaches
(Direct Delivery , First Contact, Epidemic)
- Spreading based Routing
(Spray & Wait, Spray & Focus)
- Probabilistic based Routing
(ProPHET)
- Knowledge based Routing
(RUTS)

Spray & Wait

Performance goals and properties

- Significantly fewer transmissions than epidemic routing
- “Delivery delay that is better than existing single/multi-copy schemes”
- Generate low contention, especially under high traffic loads

Two phases

- **Spray phase:** spread L bundle copies to L distinct relays
- **Wait phase:** wait until one of the L relays finds the destination

Spray & Focus

Disadvantage Spray & Wait

- Spread the bundle copies quickly to the neighbor nodes
- Nodes in Wait Phase potentially never see the destination

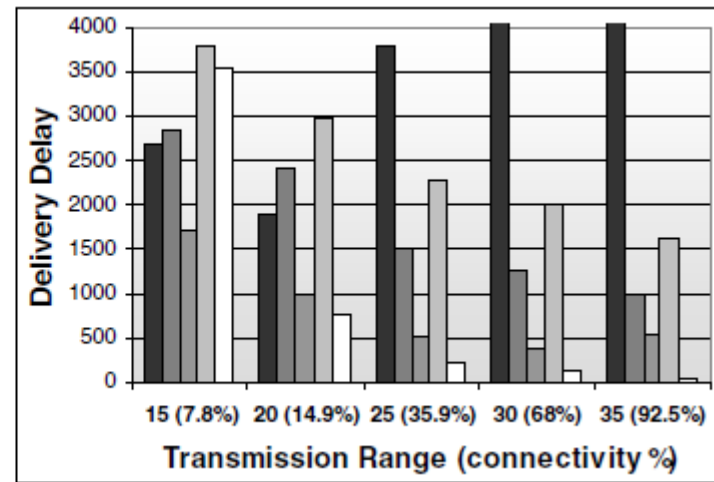
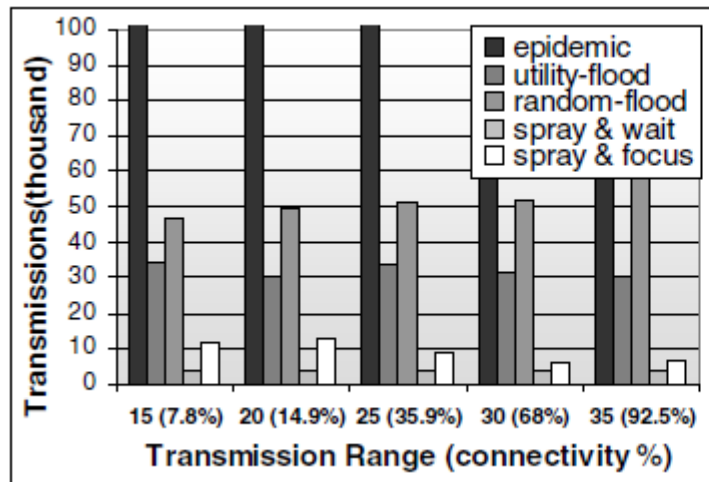
Two phases

- **Spray phase:** binary spraying
- **Focus phase:** forwarding based on utility function, e.g., node encounters, positions, speeds, ...

Spray & Focus - 2

Focus phase

- In general: node A forwards to B if utility value of B greater than the utility value (+ threshold) of node A
- Forwarding decision is based on a set of timers that record the time since two nodes last saw each other



Part II Routing - Overview

- General Routing Properties and Challenges
- Simple Routing Approaches
(Direct Delivery , First Contact, Epidemic)
- Spreading based Routing
(Spray & Wait, Spray & Focus)
- Probabilistic based Routing
(ProPHET)
- Knowledge based Routing
(RUTS)

- Probabilistic Routing Protocol using history of encounters and transitivity
- Exploit the non-randomness of real-world encounters
- Delivery Predictability $P(A,B)$ is calculated at each node A for all known destinations B (otherwise 0)
- Nodes exchange and update delivery predictability tables
- Update DP for the encountered node

$$P(A, B) = P(A, B)_{old} + (1 - P(A, B)_{old}) \times P_{enc}, \text{ where } P_{enc} \in [0,1]$$

- General Routing Properties and Challenges
- Simple Routing Approaches
(Direct Delivery , First Contact, Epidemic)
- Spreading based Routing
(Spray & Wait, Spray & Focus)
- Probabilistic based Routing
(ProPHET)
- Knowledge based Routing
(RUTS)

Knowledge based Routing

Application Scenarios

- Using vehicles for collecting environmental data
- Update displays



OPTraCom: <http://www.optracom.de>

RUTS

- RUTS = Routing in Urban Transport Systems
- Take advantages of the characteristics of urban transport system

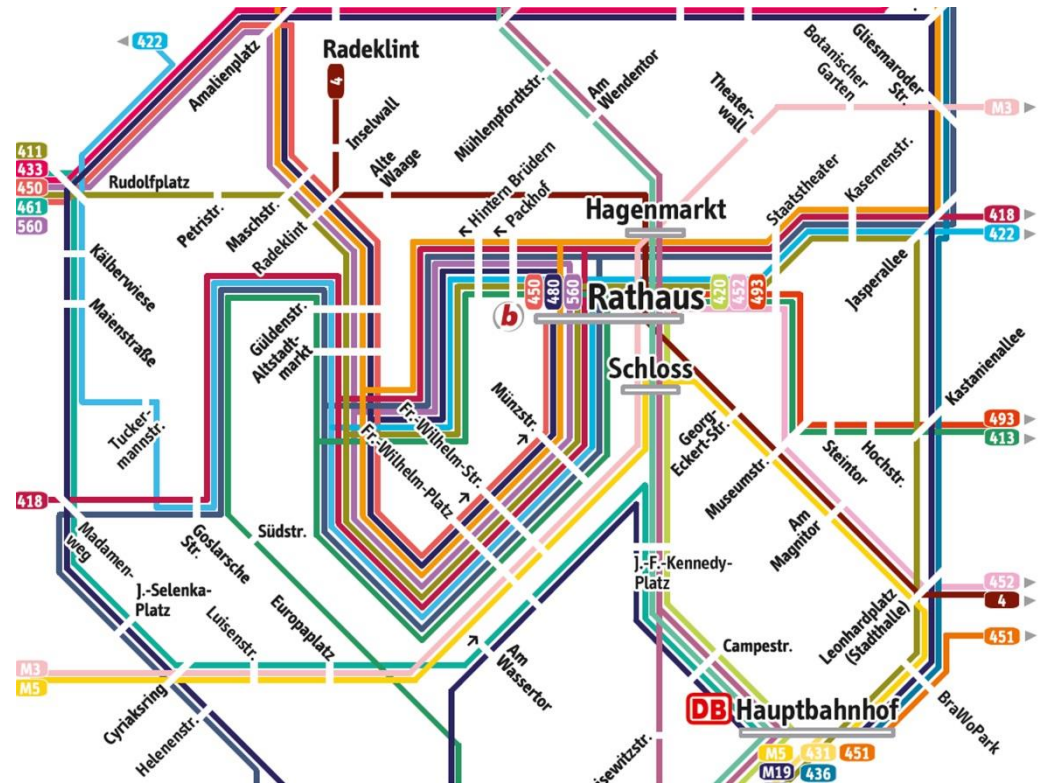
RUTS Routing

Goals

- Low latency
- High reliability
- Save resources
- Consider disturbances

Assumptions

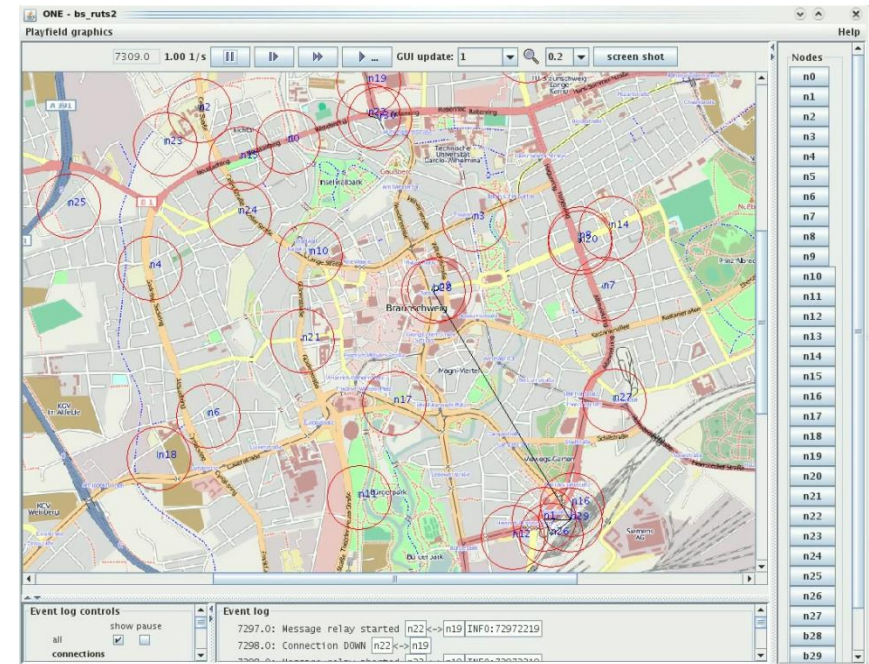
- Vehicles know ...
 - Position
 - Timetables
 - Network Map



Braunschweig network map

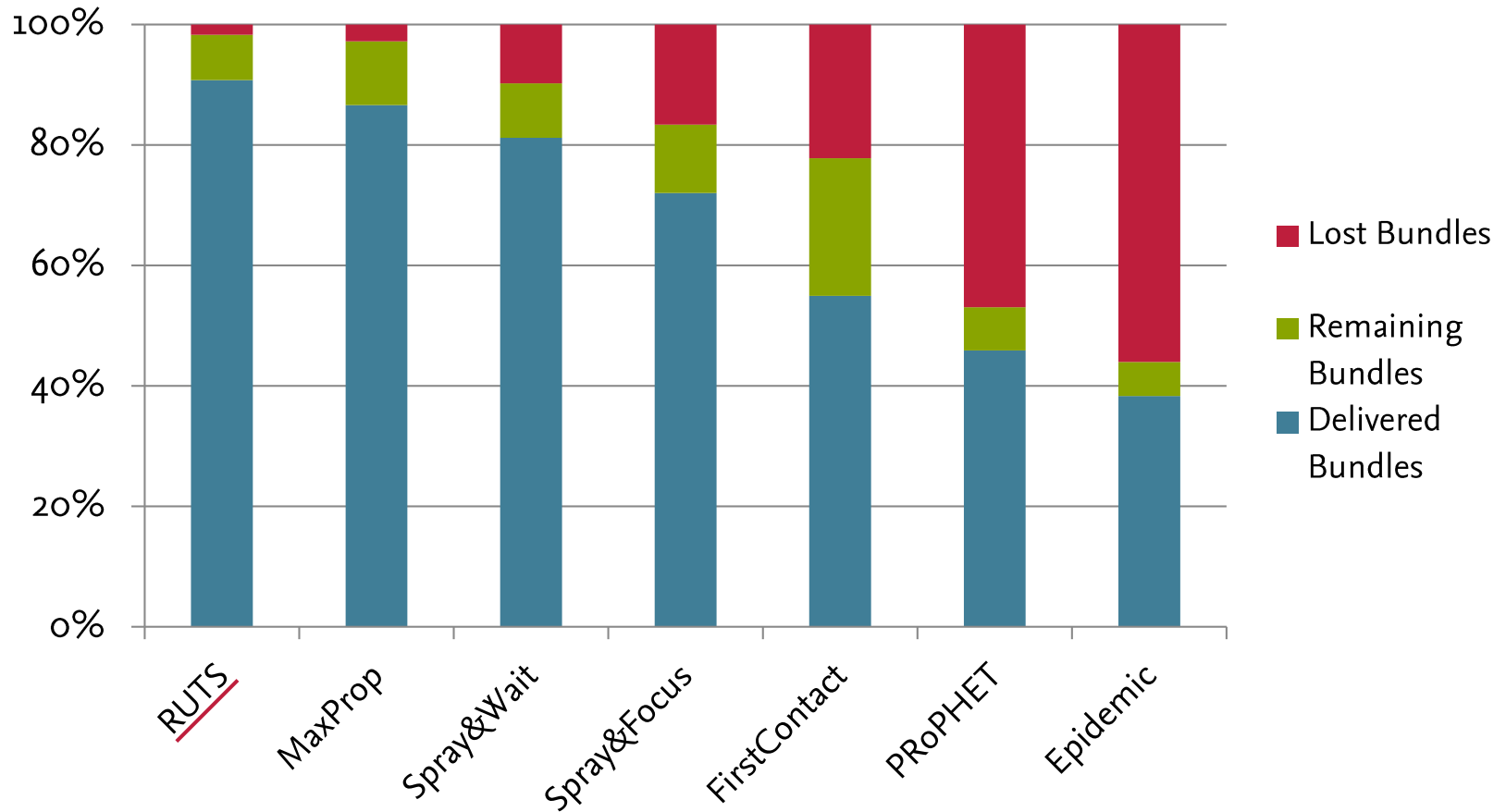
Braunschweig Scenario

- “ONE” DTN simulator
- Realistic movement patterns (13 bus/tram lines, 28 vehicles)
- 5 hours
- 1 random bundle each 5 seconds
- 2 bundle copies, 5 routing paths

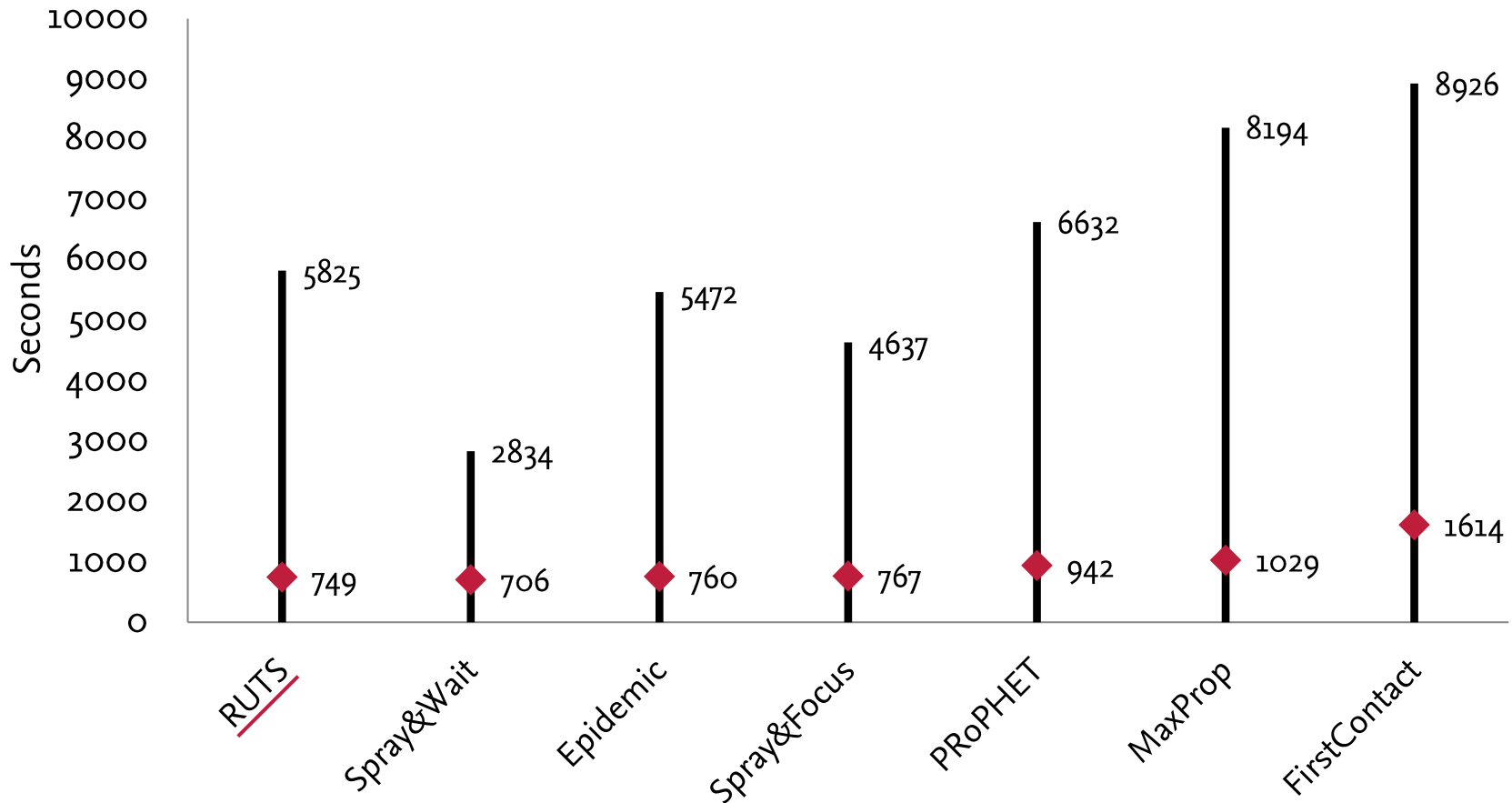


„ONE“ - DTN simulator

Results - Lost, Remaining and Delivered Bundles



Results - Transmission Time (min/avg/max)



References

- *Sushant Jain, Kevin Fall, Rabin Patra; Routing in a delay tolerant network; SIGCOMM Comput. Commun. Rev. 34; October 2004*
- *A. Vahdat, D. Becker; Epidemic Routing for Partially Connected Ad Hoc Networks; Duke University; Technical Report; April 2000*
- *Thrasyvoulos Spyropoulos, Konstantinos Psounis, Cauligi S. Raghavendra; Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility; PERCOMW '07; Washington, USA, 2007*
- *Thrasyvoulos Spyropoulos, Konstantinos Psounis, Cauligi S. Raghavendra; Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks; Workshop on Delay-tolerant networking; WDTN '05; Philadelphia, USA, 2005*
- *Anders Lindgren, Avri Doria, Olov Schelén; Probabilistic routing in intermittently connected networks; SIGMOBILE Mob. Comput. Commun. Journal; 2003*
- *Samo Grasic, Elwyn Davies, Anders Lindgren, Avri Doria; The Evolution of a DTN Routing Protocol - PROPHETv2; Workshop on Challenged Networks, Las Vegas, USA, 2011*
- *Michael Doering, Tobias Pögel and Lars C Wolf; DTN Routing in Urban Public Transport Systems; Workshop on Challenged Networks, Chicago, USA, 2010*

- Delay-Tolerant Networking Architecture
 - RFC 4838 since April 2007
 - Addressing, routing, quality of service, reliability, ...
- Bundle Protocol Specification
 - RFC 5050 since November 2007
 - Terminology, data specification, fragmentation, ...
- Well-known DTN implementations

Introduction

Literature

- **Delay-Tolerant Networking Architecture (RFC 4838)**
V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss, April 2007
- **Bundle Protocol Specification (RFC 5050)**
K. Scott, S. Burleigh, November 2007
- **Bundle Security Protocol Specification (RFC 6257)**
S. Symington, S. Farrell, H. Weiss, P. Lovell, March 2011

Architecture

Tasks of a DTN

- Handle data-sets as bundles
- Store bundles for later delivery or forwarding
- Addressing within the network (Unicast / Multicast / Anycast)
- Routing

Optional Tasks

- Quality of service
- Fragmentation
- Error-Detection and Correction
- Reliability (Custody)
- Detection of duplicates
- Efficiency
- Security / Encryption
- Endpoint to Endpoint

Architecture – Addressing

- Semantic

- DTN address = Endpoint Identifier (EID)
- Describes an application or only a single node
- EID also identify groups (Multicast)

- Specification

- RFC4838: Uniform Resource Identifier (URI) as EID
 - Variable length
(Instead of fixed length with limited number of addresses)
 - E.g. dtn://node-a21/filetransfer, <http://www.ibr.cs.tu-bs.de/>

Architecture – Quality of Service

DTN is like a postal service

- Traffic is generally not interactive and is often one-way
- No strong guarantees of timely delivery
- Priorities
 - Bulk (least effort)
 - Normal
 - Expedited (prior to other bundles)
- 100% reliable transmissions?
 - Not possible in opportunistic networks
 - Best effort delivery



Architecture – Reliability

- Custody Transfers
 - Bundles are marked with custody requested
 - Node can accept or reject custody for a received bundle
- Delivery Options
 - Custody Transfer Requested
 - Source Node Custody Acceptance Required
 - Report When Bundle Delivered
 - Report When Bundle Acknowledged by Application
 - Report When Bundle Received
 - Report When Bundle Custody Accepted
 - Report When Bundle Forwarded
 - Report When Bundle Deleted

Architecture – Fragmentation

- If a bundle is **too large for a single transmission**, then use fragmentation
 - **Proactive** fragmentation
 - Split up a large bundle into several smaller pieces
 - Based on a predefined value
 - **Reactive** fragmentation
 - Try to transmit the large bundle and split it into two parts, if the connection gets interrupted
 - Only possible, if the convergence layer know how much data was successfully transmitted
- Challenge with replicating routing schemes
 - Each interrupted contact produces a new copy of the bundle

Architecture – Application Programming Interface

- Basics
 - Send / receive bundles
 - Register as endpoint
 - Join several groups
- Management
 - List neighbors
 - Initiate connection
- Registration
 - Response of another endpoint can last longer than the runtime of the application

(More or less) precise specification for implementations

- Bundle data definition
 - Primary Block, Payload, Administrative Records, Extension Blocks, etc.
 - Endpoint Identifier, Numeric values, Strings, etc.
- **Mechanisms**
 - Bundle Processing (transmission, dispatching, forwarding)
 - Bundle Expiration
 - Custody Transfer
 - Generation of Administrative Records
 - Processing of Administrative Records
 - Fragmentation / Reassembly

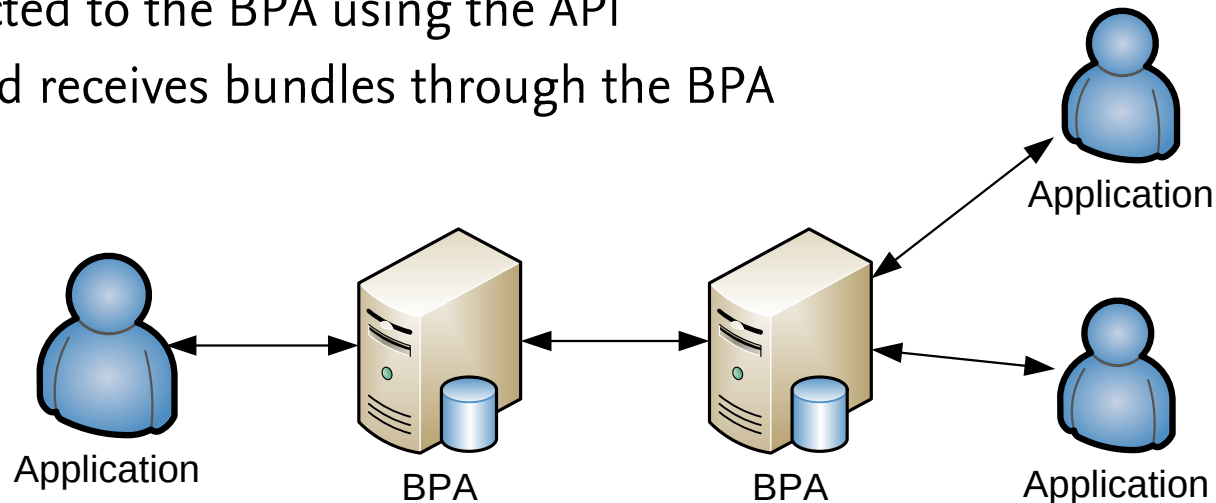
Bundle Protocol – Terminology

- **Bundle Protocol Agent (BPA)**

- Manages the DTN according to RFC5050
- Talks to other BPAs
- Integrated bundle storage

- **Application**

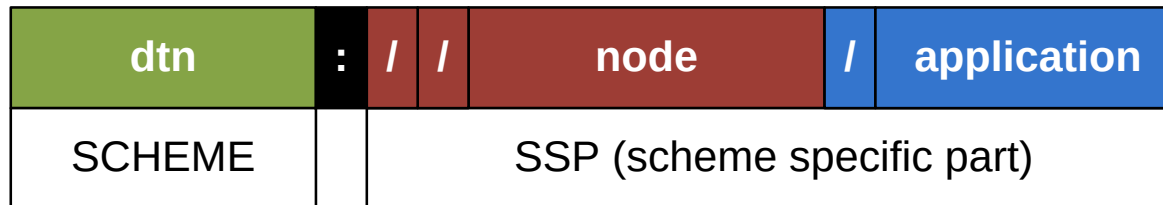
- Is connected to the BPA using the API
- Sends and receives bundles through the BPA



Bundle Protocol – Datatypes

- **Endpoint Identifier**

- Uniform Resource Identifier (URI)
- Split up into scheme and scheme-specific part (SSP)
- Examples
 - dtn://host/application
 - http://www.ibr.cs.tu-bs.de/projects/ibr-dtn
 - mailto:rottmann@ibr.cs.tu-bs.de



Bundle Protocol – Blocks

- **Bundle Blocks**

- Each bundle consists of exact one **primary block** and one **payload block**
- Additionally, each bundle carries an unlimited number of extension blocks
- Extension blocks can be in front of or behind the payload block

- **Bundle Block Format**

- RFC5050 only defines a payload block and an abstract extension block
- Generic block layout
 - Block type
 - Block processing control flags
 - Optional: EID Reference List
 - Block length + body data

Block type	Block proc. ctrl. Flags (SDNV)
Optional EID References	
Block length (SDNV)	
Block body data (variable)	

Bundle Protocol – Primary Block

- One primary block per bundle
- Version Field
 - current value = 6
- Processing flags
- Contains unique identifier
 - Source EID
 - Creation timestamp
 - Sequence number
- Optional (fragmentation):
 - Fragmentation offset
 - Total application data unit length

Version	Proc. Flags (*)
Block length (*)	
Destination scheme offset (*)	Destination SSP offset (*)
Source scheme offset (*)	Source SSP offset (*)
Report-To scheme offset (*)	Report-To SSP offset (*)
Custodian scheme offset (*)	Custodian SSP offset (*)
Creation Timestamp time (*)	
Creation Timestamp sequence number (*)	
Lifetime (*)	
Dictionary length (*)	
Dictionary byte array	
Optional: Fragment offset (*)	
Optional: Total application data unit length (*)	

Bundle Protocol – Primary Block

- **Bundle Processing Control Flags**

- Encoded as SDNV

- Bit 0: Bundle is a fragment
- Bit 1: Application data unit is an administrative record
- Bit 2: Bundle must not be fragmented
- Bit 3: Custody transfer is requested
- Bit 4: Destination endpoint is a singleton
- Bit 5: Acknowledgement by application is requested
- Bit 6: Reserved for future use

General

- Bit 7+8: Priorities; bulk, normal, expedited
- Bits 9 – 13: Reserved for future use

Class of Service

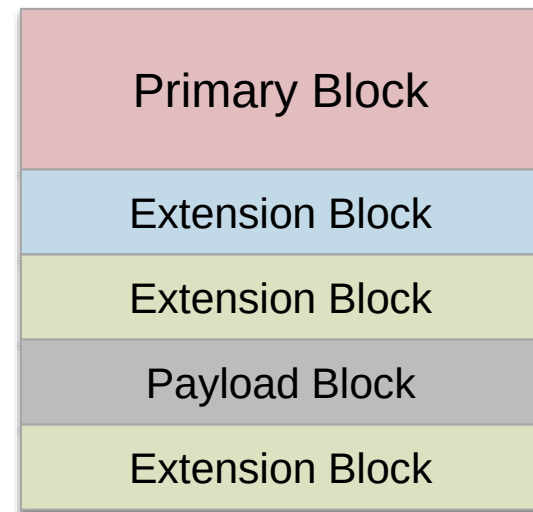
- Bit 14: Request reporting of bundle reception
- Bit 15: Request reporting of custody acceptance
- Bit 16: Request reporting of bundle forwarding
- Bit 17: Request reporting of bundle delivery
- Bit 18: Request reporting of bundle deletion
- Bit 19: Reserved for future use

Status Report

Bundle Protocol – Extension Blocks

- **Extension Blocks**

- Extends the Bundle Protocol with new functionality
- Can be placed in front of or behind the payload block
- Generic block format allows forwarding of unsupported extension blocks
- Examples
 - Delay-Tolerant Networking
Previous-Hop Insertion Block [RFC6259]
 - DTN Scope Control using Hop
Limits (SCHL) [expired draft]
 - Bundle Security Protocol [RFC6257]
 - DTN Bundle Age Block for Expiration
without UTC [expired draft]



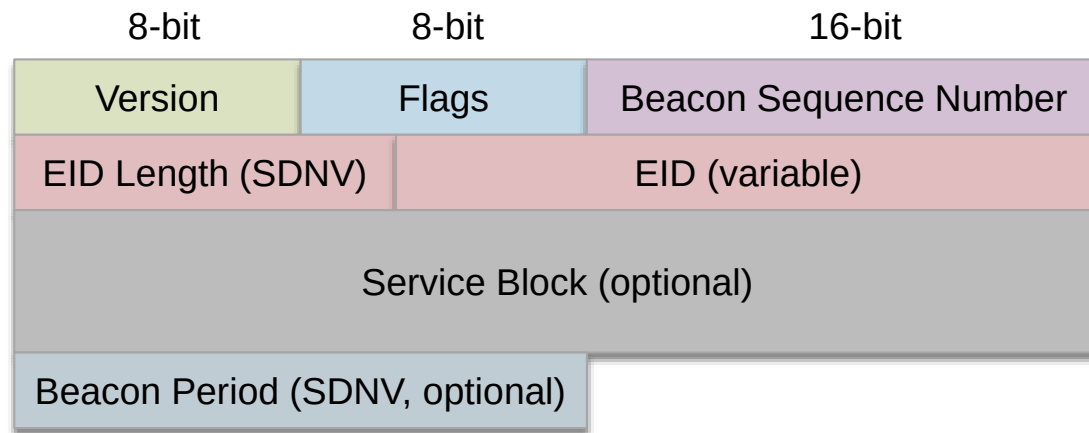
Extensions – TCP Convergence Layer

- **TCP convergence layer**
 - Defines how to inter-connect between BPAs and exchange bundles
 - Experimental Draft, Version 07 (13.09.2013)
 - Initiate a handshake
 - Version
 - Node EID
 - Keepalive interval
 - Flags: ACK request, enable reactive fragmentation, NACK support, Length request
 - Bundle is encoded as defined in RFC5050 and may split up into segments
 - Each segment contains ...
 - the message type code (4-bit): DATA, ACK, REFUSE, LENGTH, KEEPALIVE or SHUTDOWN
 - the flags (4-bit): mark the begin and the end of a bundle
 - the payload length (SDNV) followed by the payload
 - A connection gets closed on keep-alive timeout, error or low resources

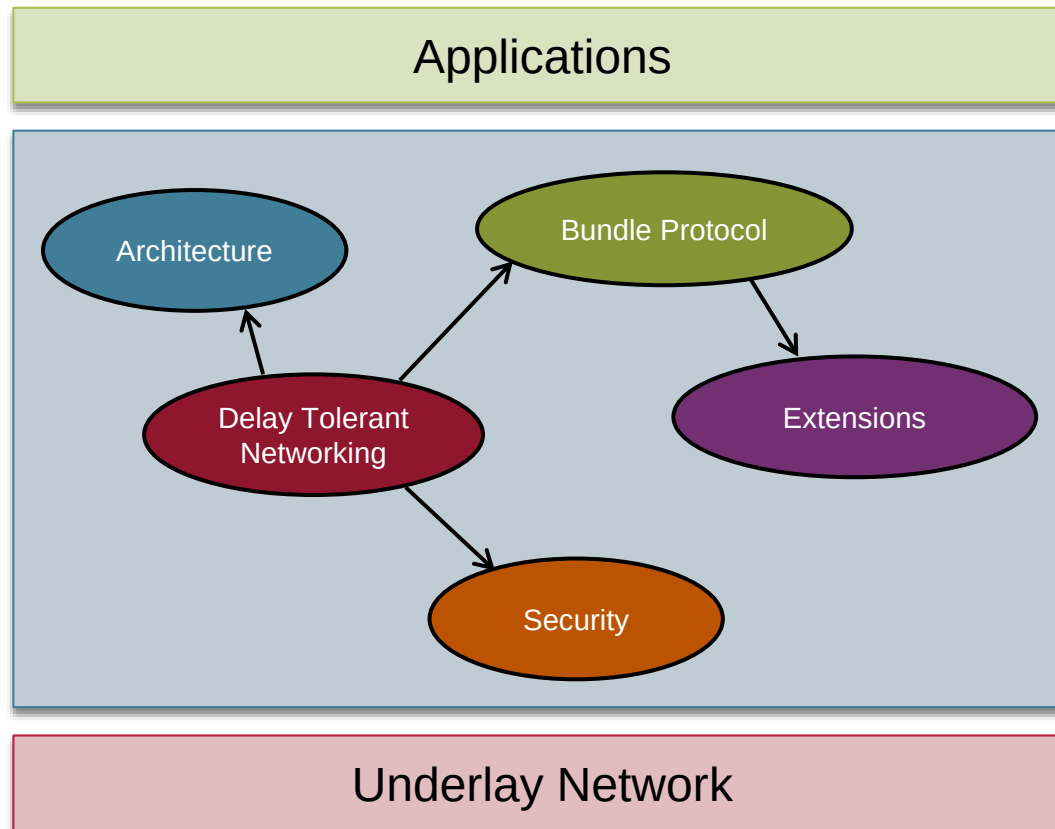
Extensions – IP Neighbor Discovery

- **IP Neighbor Discovery**

- Experimental Draft, Version 02 (08.11.2012)
- Opportunistic detection of neighbors
- Based on beacons (periodical broadcast)
- Service descriptions



Well-Known Implementations



Well-Known Implementations

DTN2 Reference Implementation

- All components of the DTN architecture
- Strict implemented according to RFC5050
- Several convergence layers (TCP, UDP, LTP, Ethernet, ...)
- Bundle Security Protocol
- Written in C++
- Extensible / External interfaces for routing and storage
- Basis for scientific experiments

Well-Known Implementations

Interplanetary Overlay Network (ION)

- Developed by NASA Jet Propulsion Laboratory
- Basic components of the DTN architecture
- Includes special protocols like LTP for space missions
- Designed for embedded systems (robotic spacecraft)
- Written in C
- Portable to POSIX systems

Well-Known Implementations

IBR-DTN

- In-House Implementation
- Most of the components of the DTN architecture
- Bundle Security Protocol
- Several convergence layers
TCP, UDP, HTTP, IEEE802.15.4, USB storage, Serial Line
- Designed for embedded systems
- Event driven modular design / Written in C++
- Portability to POSIX systems
Debian / Ubuntu, OpenWRT, MacOS X, Android, iMote 2 (sensor node), Windows