

# *Towards the Future Internet*

*An Overview of Challenges and Solutions  
in Research and Standardization*

---

Dr. Thomas Dreibholz

Networking Technology Group

of the Institute for Experimental Mathematics  
at the University of Duisburg-Essen

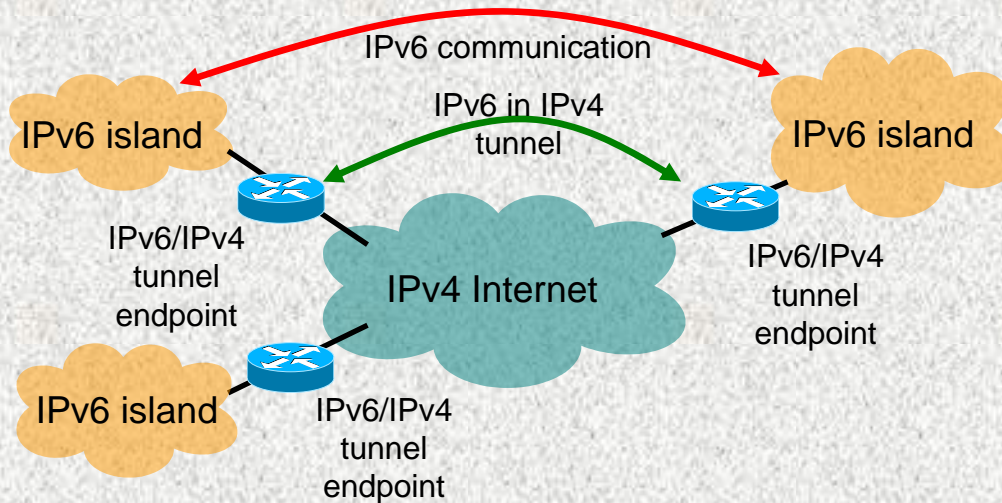
# Table of Contents

- How will the Internet look in 2020?
- IP Version 6 (IPv6)
- Stream Control Transmission Protocol (SCTP)
- Reliable Server Pooling (RSerPool)
- Conclusion

# How will the Internet look in 2020?

- **Revolutionary approaches**
  - Clean slate approaches
  
- **Evolutionary approaches**
  - Extending the existing Internet
  - Slow migration
  
- **Goal of this talk: an outline of the evolutionary track**
  - Standardization activities of the IETF
  - Our related research at the University of Duisburg-Essen

# Migration to IP Version 6



In the (not so far) future:

- IPv4 over IPv6 tunnels
- IPv4 islands

## ■ Multi-homed sites:

- IPv4 and IPv6 in parallel
- Multiple IPv6 providers (e.g. provider and tunnel broker)

## ■ Challenge:

- Copying with this redundancy
- TCP establishes connection between two IP addresses => one path
- Solution: SCTP

# From TCP to SCTP

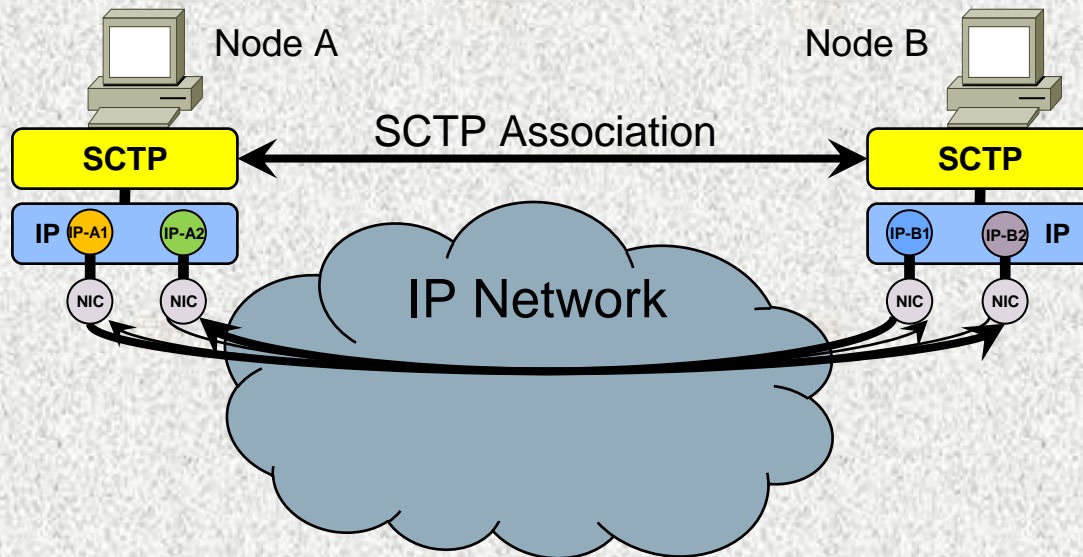
- SCTP – Stream Control Transmission Protocol
- “Next Generation of TCP”
- Main motivation: telephone signalling transport (SS7) over IP

- **1998** Signalling Transport (SIGTRAN) Working Group
- Improvements and extensions of the SCTP drafts
- First prototype implementations, e.g. *SCTPLIB*
- Interoperability tests



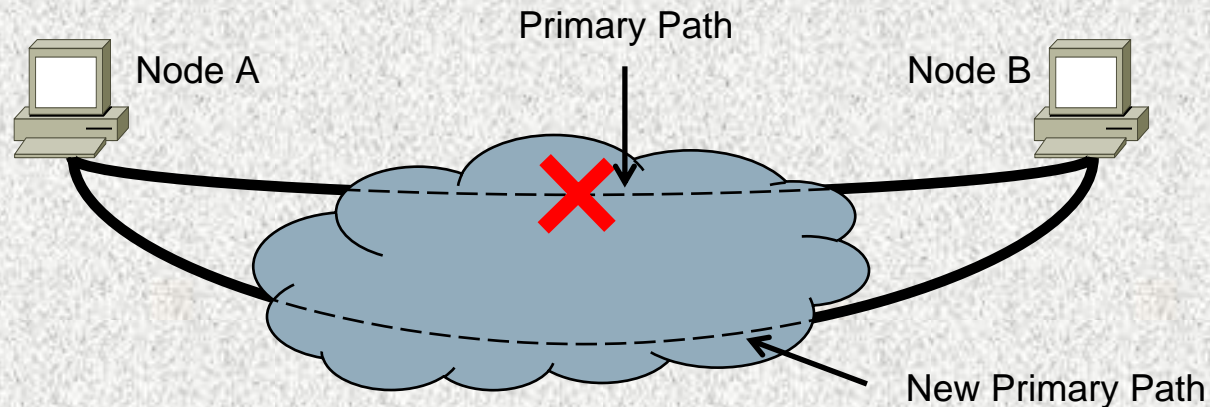
- October 2000: RFC 2960 (official SCTP standard)
- September 2002: RFC 3309 (change of checksum algorithm)
- April 2007: RFC 4460 (corrections and clarifications)
- **September 2007:** **RFC 4960** (updated official SCTP standard)

Almost 10 years



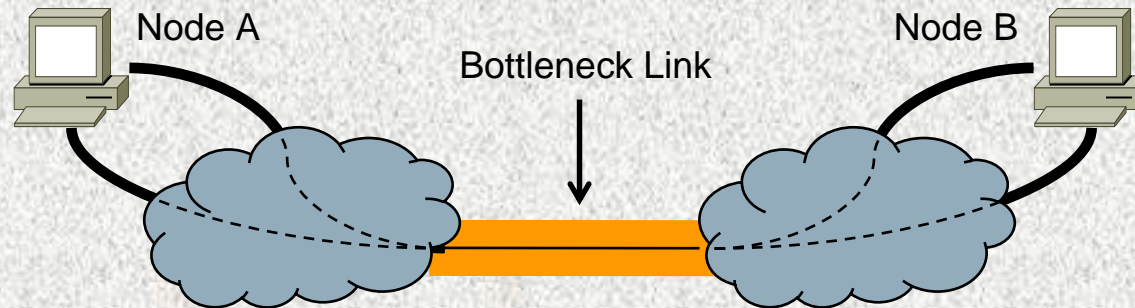
- **Association** = connection between two endpoints
- **Path** defines unidirectional relation to one of remote peer's addresses
- **Primary path** = selected path used for data transmission
- Break of primary path => switching to another path
  - Alternative path is also used for retransmissions

# Changing the Primary Path



- Upon failure of the primary path ...
  - ... SCTP selects another path
- Upper layer has to do nothing ...
  - ... but can be notified about the path change (by option)
- Upper layer can explicitly request a change of the primary path:
  - Locally
  - From the remote node to itself (i.e. for the reverse direction)
- Current research topic: which path to choose?

# Why is there only one Primary Path?



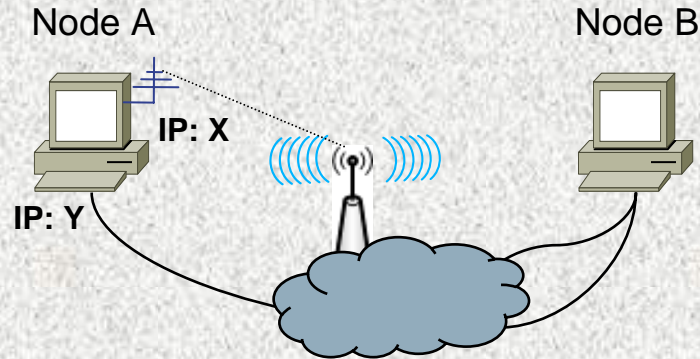
## ■ Example:

- Two paths between nodes A and B
- Both paths share the same bottleneck link
- If both paths would be used for data transmission:
  - Behaviour would be similar to two TCP flows
  - In case of congestion: unfair to other TCP/SCTP flows running over the same bottleneck!

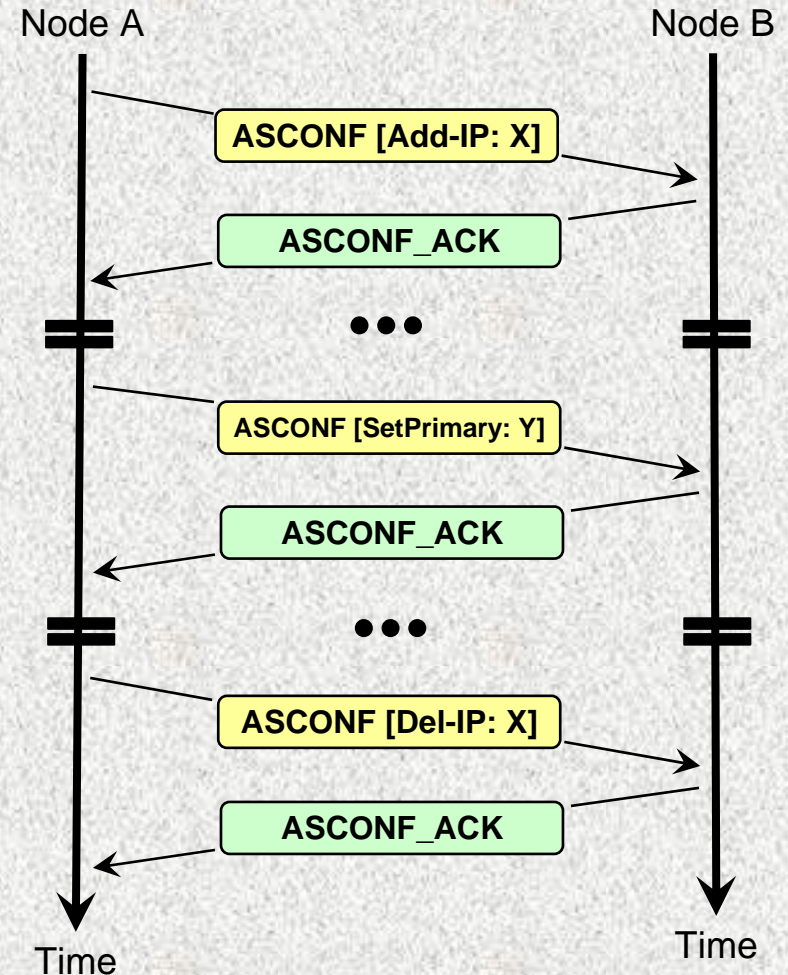
■ Network topology is unknown – there may be non-disjoint paths

■ Current research topic: Concurrent Multipath Transfer (CMT)

# Dynamic Address Reconfiguration ("Add-IP")



- Dynamic Address Reconfiguration
- Defined in RFC 5061
- Application:
  - Adding new IP addresses (Add-IP):
    - Example: reachable W-LAN
  - Can be combined with explicit Primary Path setting
  - Removing IP addresses (Del-IP):
    - Example: unreachable W-LAN



- **Simple case #1: provider change**
  - Add addresses with advertised new prefix
  - Remove addresses with old prefix
- **Simple case #2: adding redundancy**
  - Add addresses with prefix of another provider
  - Connections to multiple providers, for redundancy
- **Simple case #3: mobility**
  - Adding W-LAN when reachable ...
  - ... while also being connected over Ethernet
- **Extreme case: seamless migration from IPv4 to IPv6**
  - Possible without interrupting associations
  - Dual-stack endpoints required, of course
- **Current research topics:**
  - Use which path when? Handover optimizations ...

- SCTP provides features for many network challenges ...
- ... but cannot protect against a server failure
- Classic Internet applications:
  - Mail, FTP, HTTP, ...
  - In case of server failure: service unavailable, until repaired
- **New applications:**
  - E-commerce, e-health, telephony over Internet, ...
  - Even short service **outages are critical**, e.g.:
    - For e-shops, customers simply go to competitor (and never come back again)
    - Handling emergency calls with Internet telephony
- **Required:**
  - **Service availability by component redundancy**

# Reliable Server Pooling (RSerPool)

## ■ Realizing service availability:

- Formerly, each critical application realized its own mechanisms

## ■ Observation:

- Large parts of these implementations essentially do the same ...
- ... and developers reinvent the wheel again and again ☹

## ■ The IETF's solution:

- “A unified, application-independent and lightweight framework for pool and session management”

## Reliable Server Pooling – RSerPool

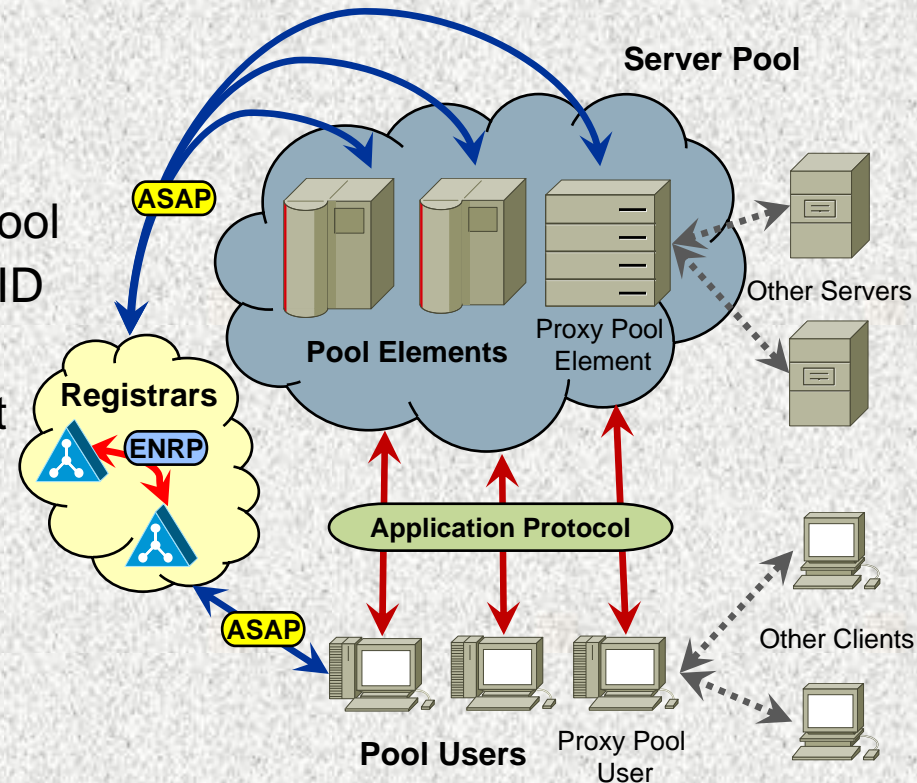
About  
9 years  
↓

- **1999** Foundation of the IETF RSerPool WG
- **September 2008** Publication of the core protocols as RFCs

# RSerPool Overview (RFC 5351)

## Terminology:

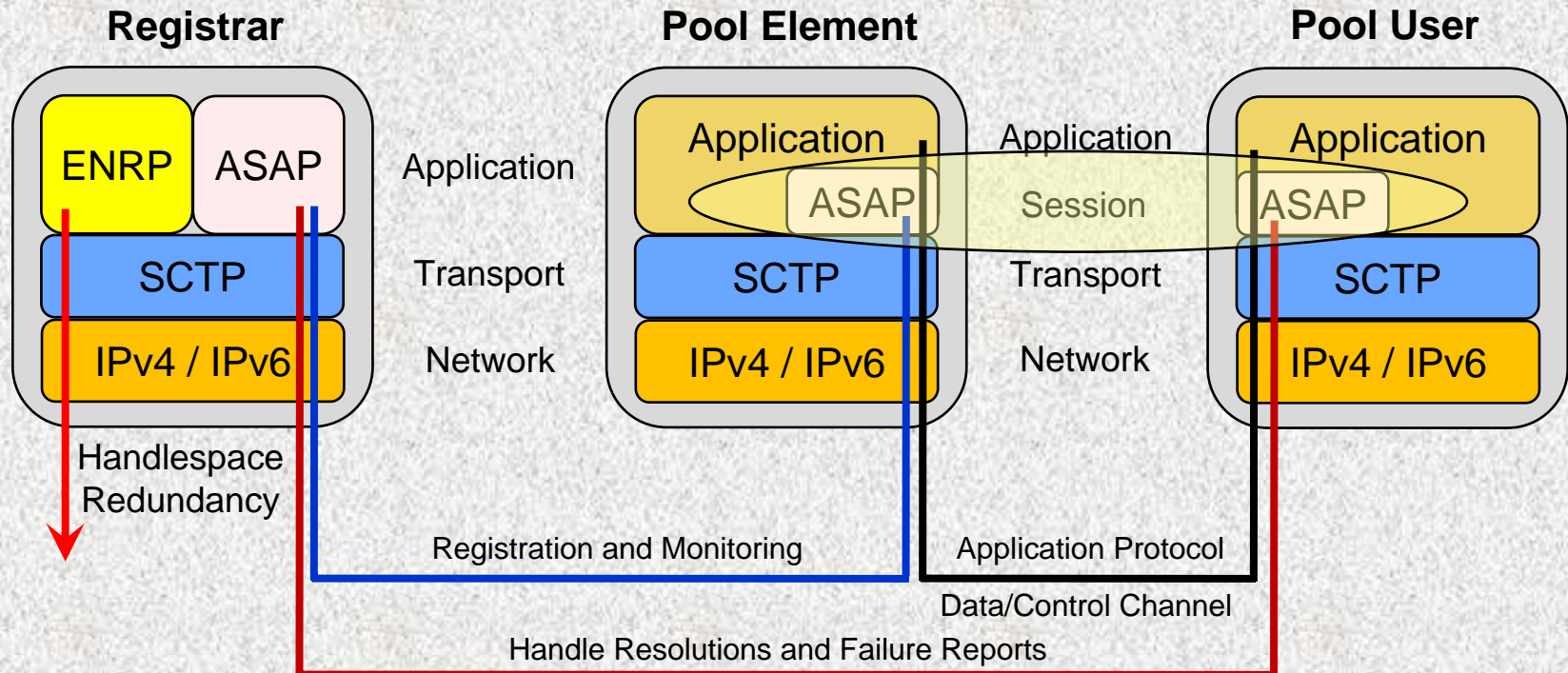
- **Pool Element (PE):** Server
- **Pool:** Set of PEs
- **PE ID:** ID of PE in pool
- **Pool Handle:** Unique pool ID
- **Handlespace:** Set of pools
- **Pool Registrar (PR)** Management
- **Pool User (PU):** Client
  
- Support for existing applications:
  - Proxy Pool User (PPU)
  - Proxy Pool Element (PPE)



■ ASAP = Aggregate Server Access Protocol (RFC 5352/RFC 5356)

■ ENRP = Endpoint Handlespace Redundancy Protocol (RFC 5353)

# RSerPool Protocol Stack



## ■ Aggregate Server Access Protocol (ASAP)

- PR  $\Leftrightarrow$  PE: registration, deregistration and monitoring by Home-PR (PR-H)
- PR  $\Leftrightarrow$  PU: server selection, failure reports
- PU  $\Leftrightarrow$  PE: session management

## ■ Endpoint Handlespace Redundancy Protocol (ENRP)

- PR  $\Leftrightarrow$  PR: handlespace synchronisation

## ■ Applications scenarios:

- SS7 telephone signalling over IP, also VoIP with SIP
- IP Flow Information Export (IPFIX)
- Video on demand services
- E-commerce
- Battlefield networks
- **Load distribution and balancing**

## ■ Fine-tuning of parameters for certain scenarios:

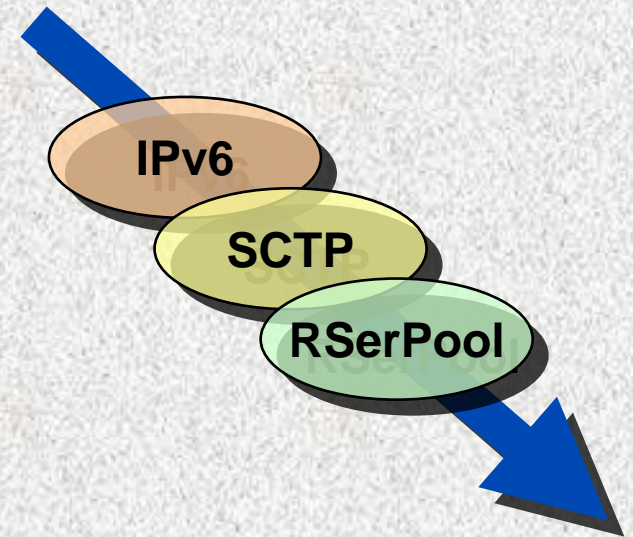
- Further reduction of management overhead (CPU, memory)
- High-latency setups
- Disaster scenarios

## ■ Security:

- Robustness against Denial of Service (DoS) attacks
- Attack countermeasures

# Conclusion: The Internet in 2020

- Internet development is slow and more evolutionary than revolutionary ...
- Deployment today:
  - IPv6 support widespread
    - All relevant operating systems
    - Some providers
  - SCTP
    - Supported by major operating systems
  - RSerPool:
    - Start of deployment
- Deployment in 2020:
  - IPv6: absolutely sure
  - SCTP: quite sure
  - RSerPool: probably
- Beyond 2020:
  - Evolution to approaches which are considered revolutionary today?



Thank You for Your Attention!

# Questions?

Visit our Project Homepage:

<http://www.iem.uni-due.de/dreibholz/rserpool/>