A New Software Architecture for Core Internet Routers

Robert Broberg

September 16, 2011

Disclaimers and Credits

- This is research and no product plans are implied by any of this work.
- r3.cis.upenn.edu
- Early and continued support from www.vu.nl
- A large team has generated this work and I am just one of many spokespersons for them.
  - any mistakes in this talk are mine.

Agenda

Overview of the evolution of Core router design

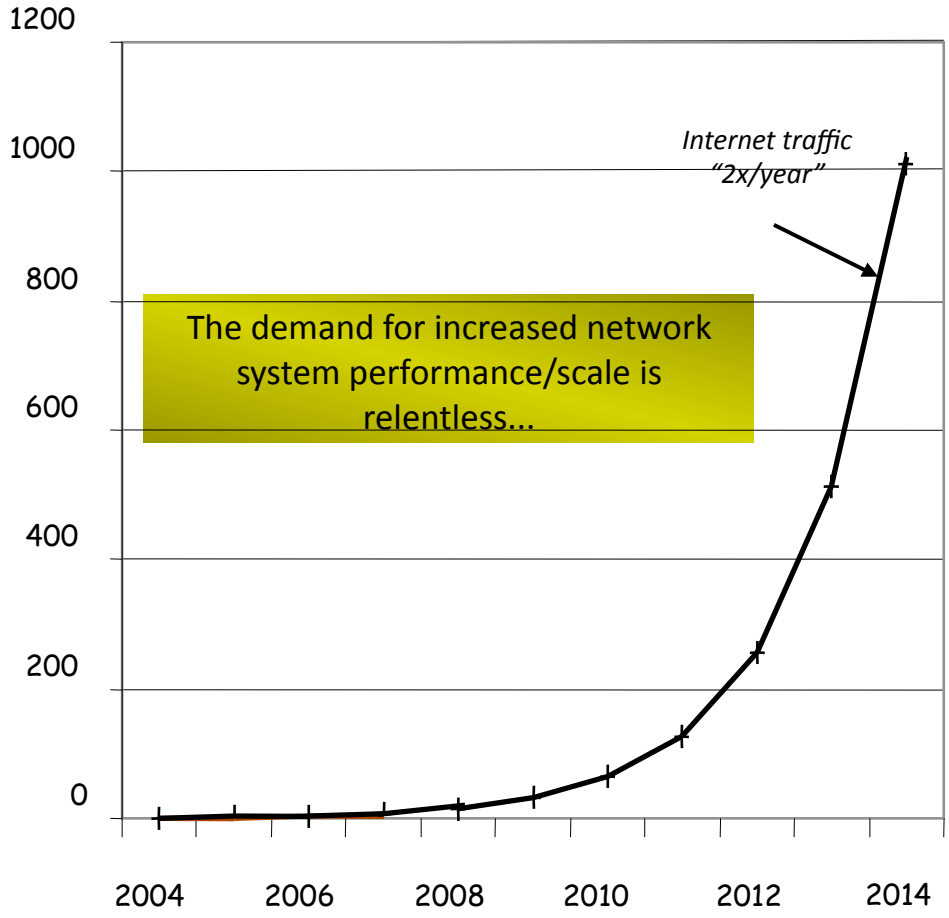
A sampling of SW problems encountered during evolution

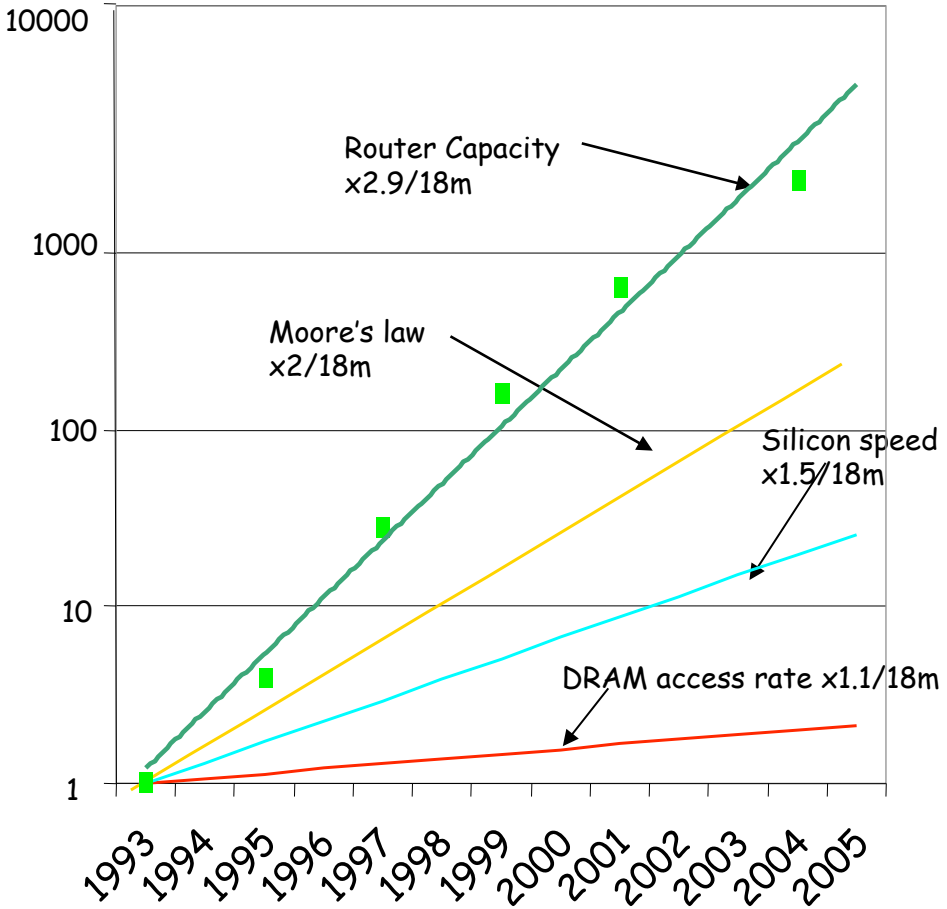
An approach to resolving SW problems and continued evolution

Core Router Evolution

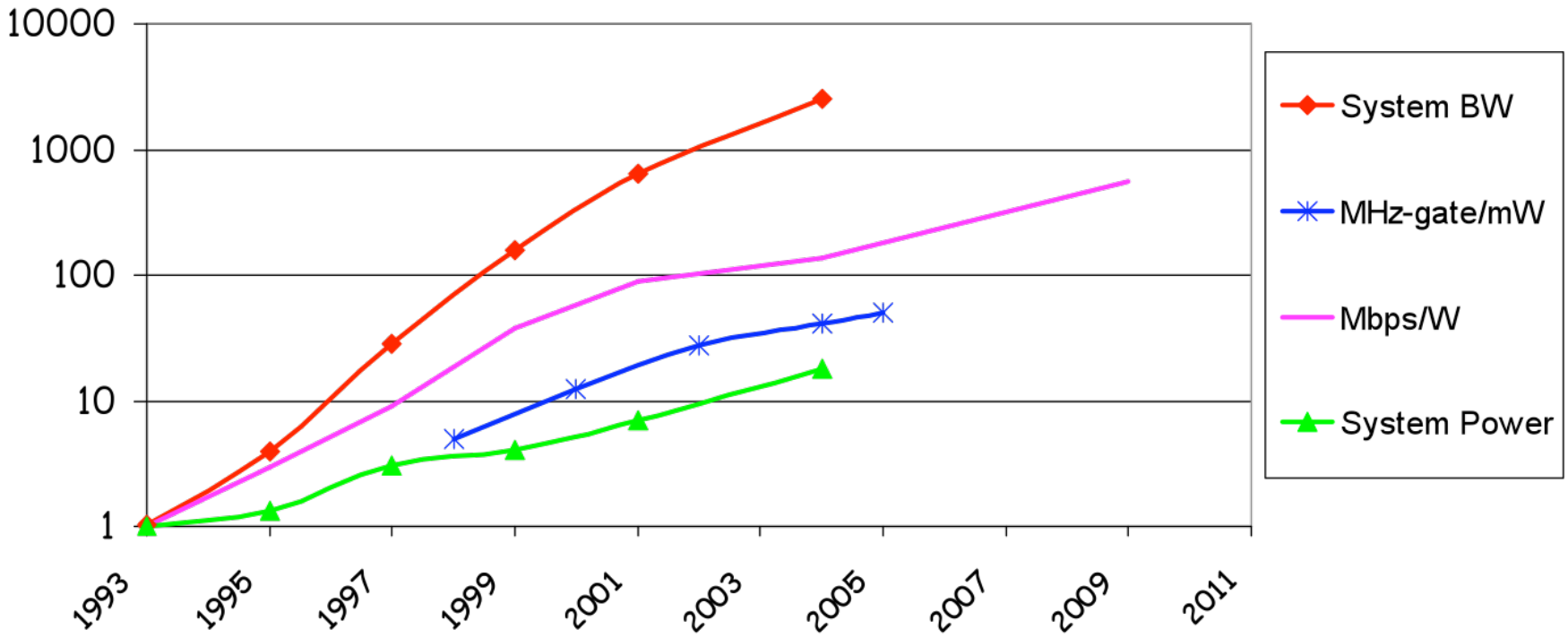
- WAN interconnects of Mainframes over telecommunication infrastructure
- LAN/WAN interconnects
  - CORE routers(1+1 architectures)
  - Leased telco lines for customers
  - Dialup aggregation
- As CORE routers evolved the old migrated to support edge connects
- Telco becomes a client of the IP network

Growth driven by increased user demand





System Scaling Problems



Product example

- **Largest Routing System available today**
  - **Each Linecard Chassis: 1.28Tbps, 13.6kW**
  - **Switch Fabric Chassis: 8kW**

Some of the reasons SW problems were encountered

- Routers started as tightly coupled embedded systems
  - speeds and feeds were the game with features
    - CPUs + NPUs + very aware programmers led the game
- Evolution was very fast
  - Business customers
    - leased lines and frame relay
  - Mid 1990s 64kbit dialup starts
  - Core bandwidth doubling every year
- As IP customer populations grew feature demands increased
- Model of SW delivery not conducive to resilience of rapid feature deployment

Intent/Goals

- build an application unaware fault tolerant distributed system for routers
- always on(200msec failover of apps)
- allow for insertion of new features with no impact to existing operations
- support +/- 1 versioning of key applications with zero packet loss
- versioning to allow for live feature testing

Fault Tolerant Routing

Motivations

- We must be able to do better than 1+1
  - Low confidence in 1+1 as only tested when actually upgrading/downgrading/crashing
- Want 100% confidence in new code
  - Despite lab time, rollout often uncovers showstoppers
  - Rollback can be very disruptive
- Aiming for sub-200ms ‘outages’
  - Want to be able to recover before VOIP calls notice

Core Routers are built as Clusters but act as a single virtual machine

- Multiple line cards with potentially various types of interfaces use NPUs to route/switch amongst themselves via a data-plane (switch fabric)

- A separate control plane controls all NPUs programming switching tables and managing interface state along, routing protocols along with environmental conditions
  - Control plane CPUs are typically generic and ride the commodity curve

- The Systems are heterogeneous and large
  - Current Cisco CRS3 deployments switch 128tb, have ~150 x86 CPUs for the control plane along with ~1terabyte of memory and scale higher

`

Virtualization/Voting/BGP

- BGP state is tied to TCP connection state
  - loopback interfaces
- Process Placement
- Versioning
- Leader election
- HW virtualization
  - e.g. NPU virtualization???

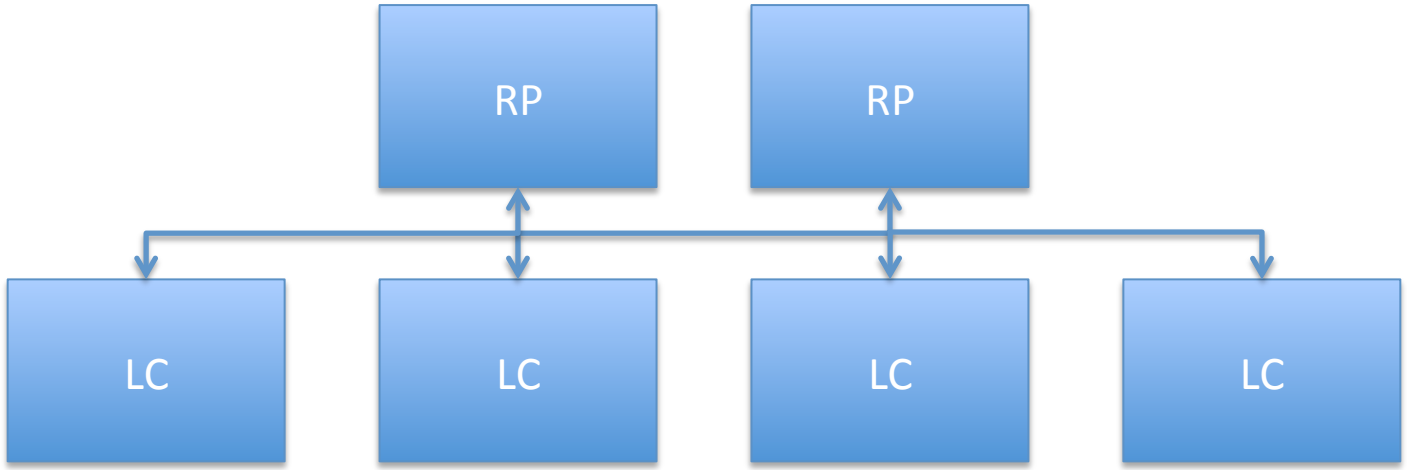
Approach taken

- Abstraction layers chosen to isolate applications
  - applications (e.g. protocols) isolated with wrappers
    - application transparent check pointing!!!!
- FTSS used to store state
- SHIM used as wrapper
  - model to allow for voting
- Optimize, optimize, optimize
  - experiment and prototype
- ORCM used for process placement
- Protocols isolated by a shim layer
  - multiple versions called siblings
- 2 levels of operation chosen
  - no use seen for hypervisor
  - user mode for apps; kernel; abstraction layer via SHIM + FTSS

Protocol Virtualization

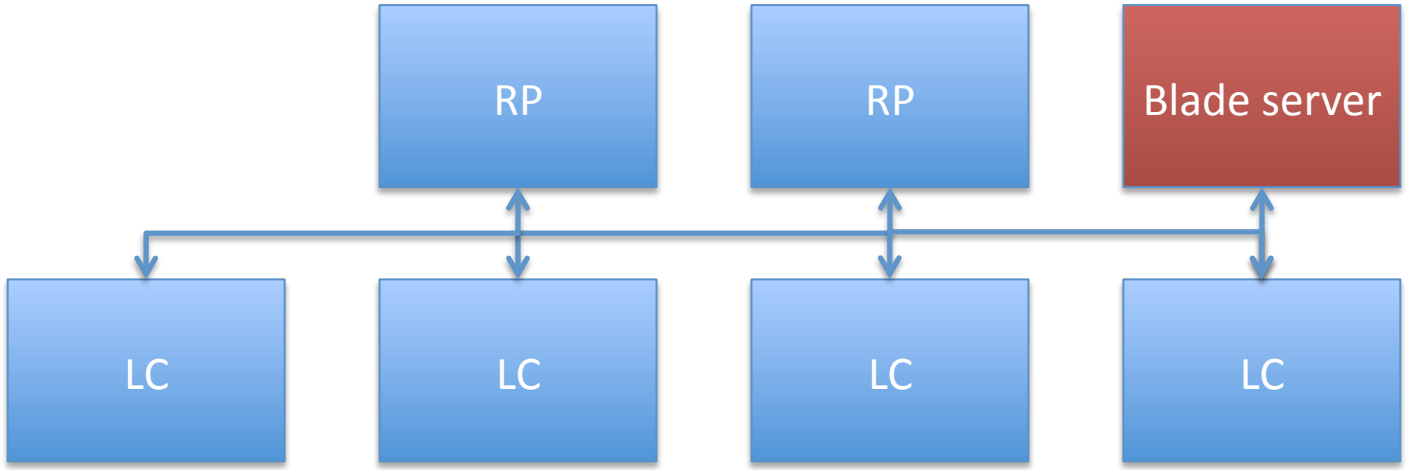
- Existing protocol code largely untouched
- Can run N siblings
  - Can be different versions – the protocol being virtualized
  - Allows full testing of new code – with seamless switchover and switch back
- Currently we run one virtualization wrapper
  - Protected by storing state into FTSS
  - Can be restarted thus upgradeable
  - Designed to know as little about protocol as possible
    - Treats most of it as a 'bag of bits'
- 'Run anywhere' – no RP/LC assumptions
  - We don't care what you call the compute resources

CRS utilisation



- The CRS contains many CPUs which we treat as compute nodes in a cluster
- If a node fails the others take up its workload
- No data is lost on a failure, and the software adapts to re-establish redundancy

CRS utilisation

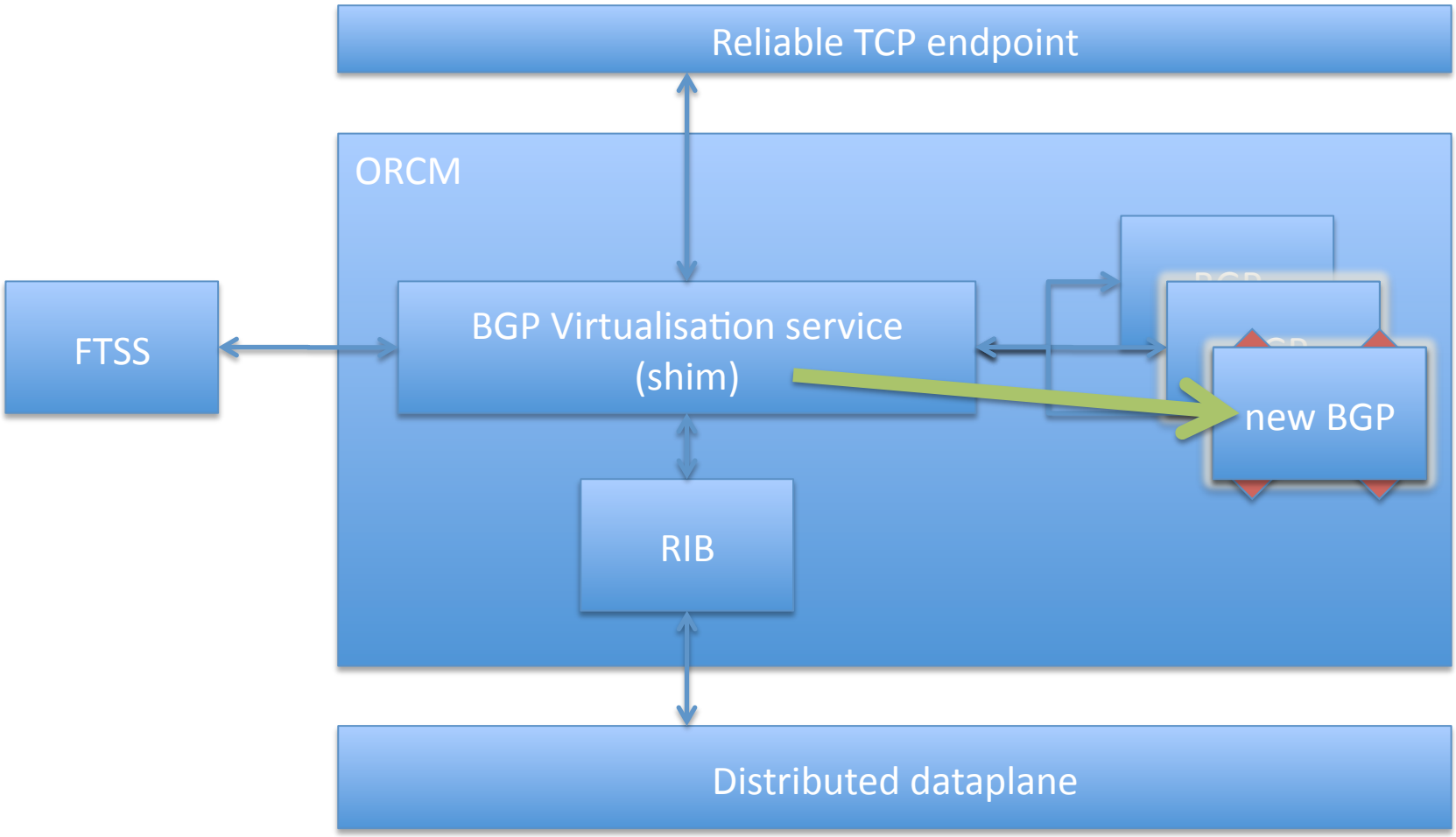


•External resources can be added to the system to add redundancy or compute power

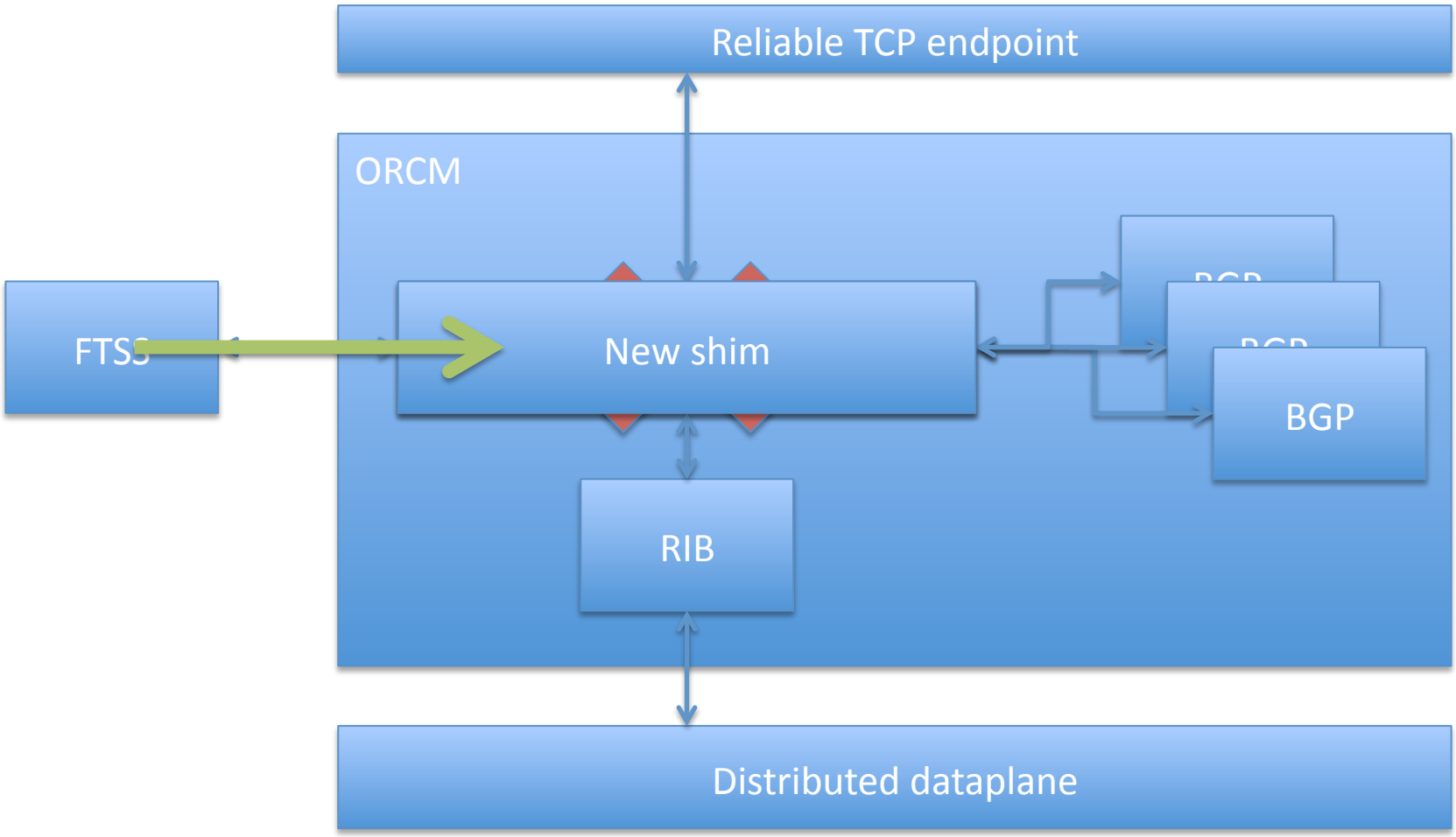
Placement of Components

- Each compute node runs FTSS and ORCM – both are started by ‘qn’ (system process monitor)
- FTSS stores routing data redundantly across all the systems in the router
- ORCM manages routing processes and distributes them around the router – constraints can be applied via configuration
- FTSS can run on other nodes to make use of memory if desired.

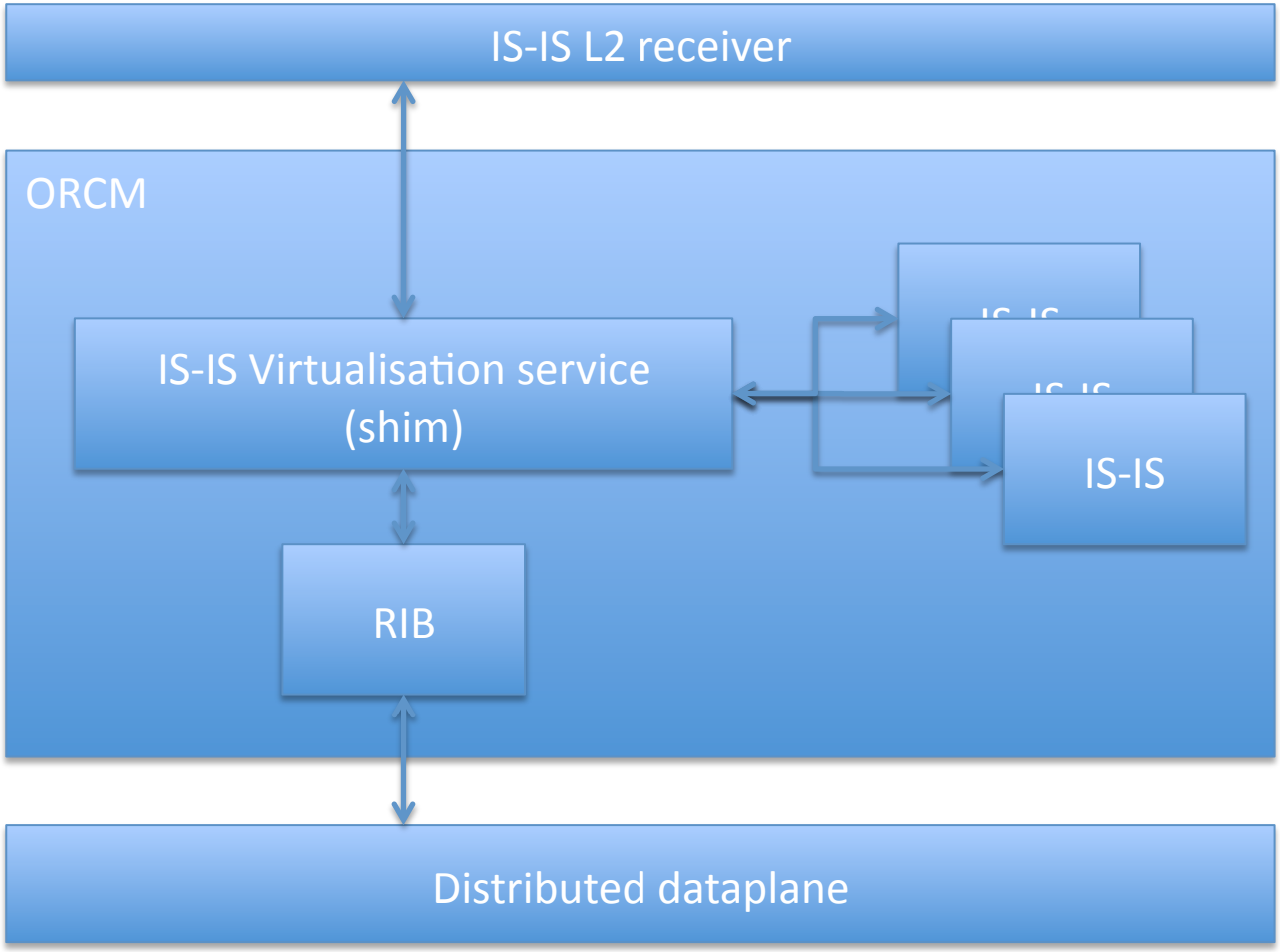
BGP Virtualisation



Virtualisation Layer recovery



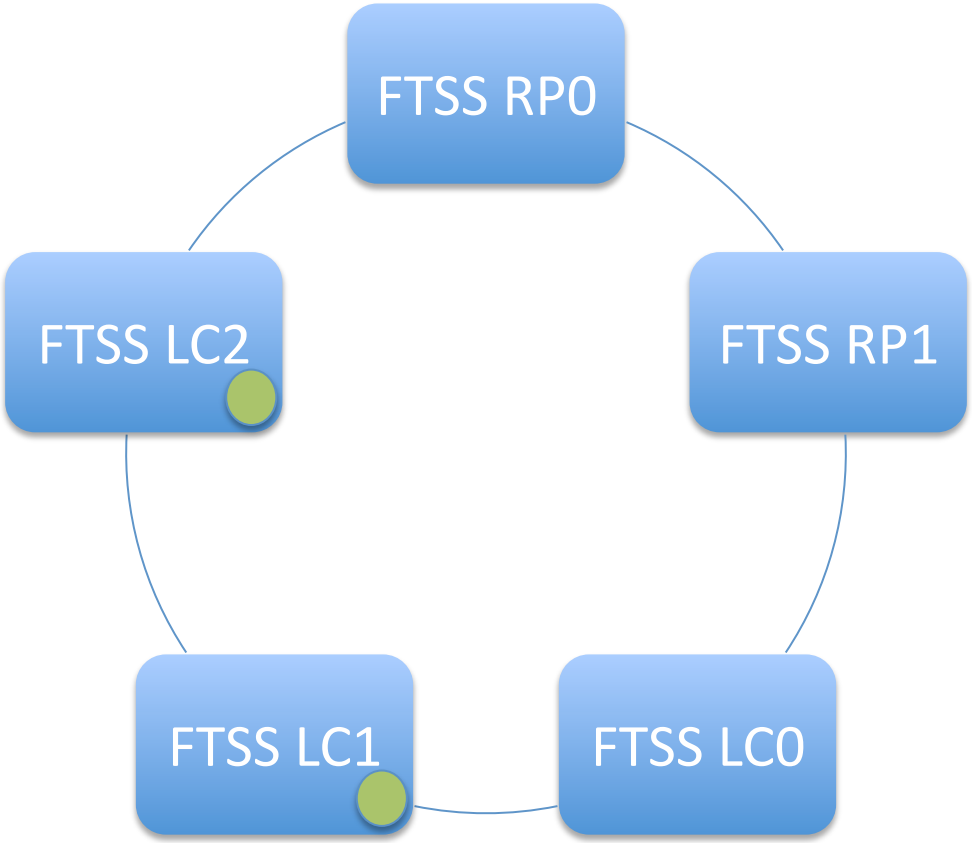
IS-IS Virtualisation



Fault Tolerant State Storage

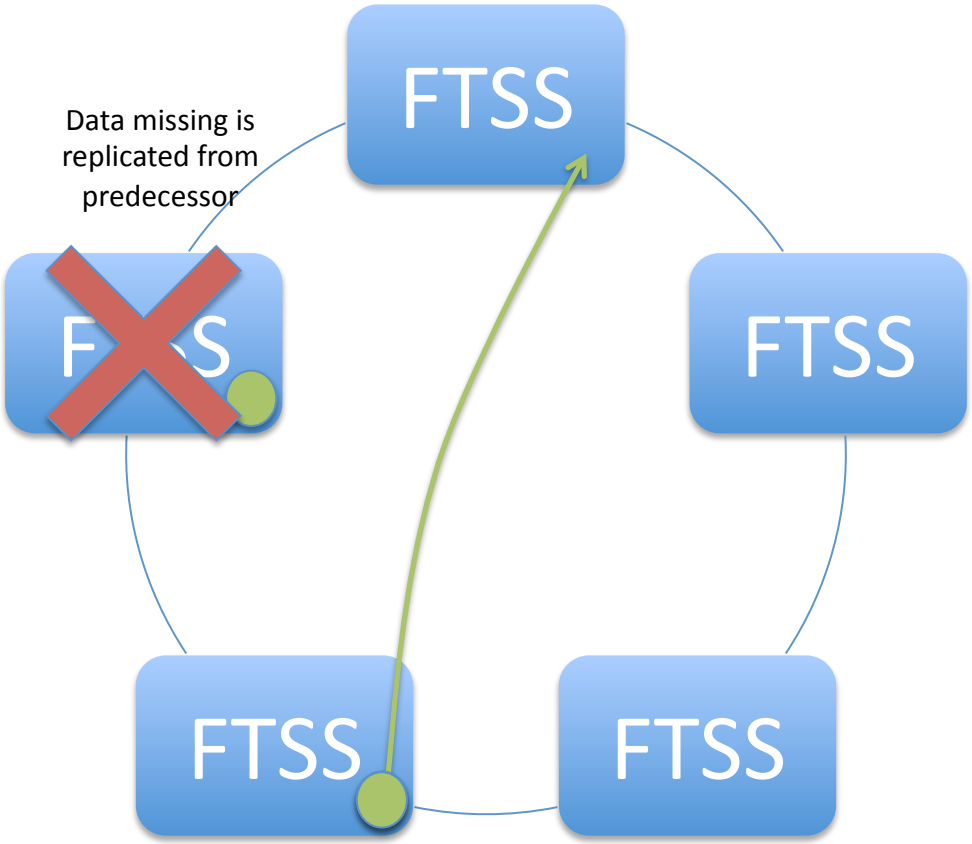
- Distributed Hash Table with intelligent placement of data
- You can decide how much replication
  - 2,3,4,N copies.
    - More copies - more memory & slower write times.
    - Fewer copies – less simultaneous failures
- Virtual Nodes – able to balance memory usage to space on compute node

FTSS distributed storage

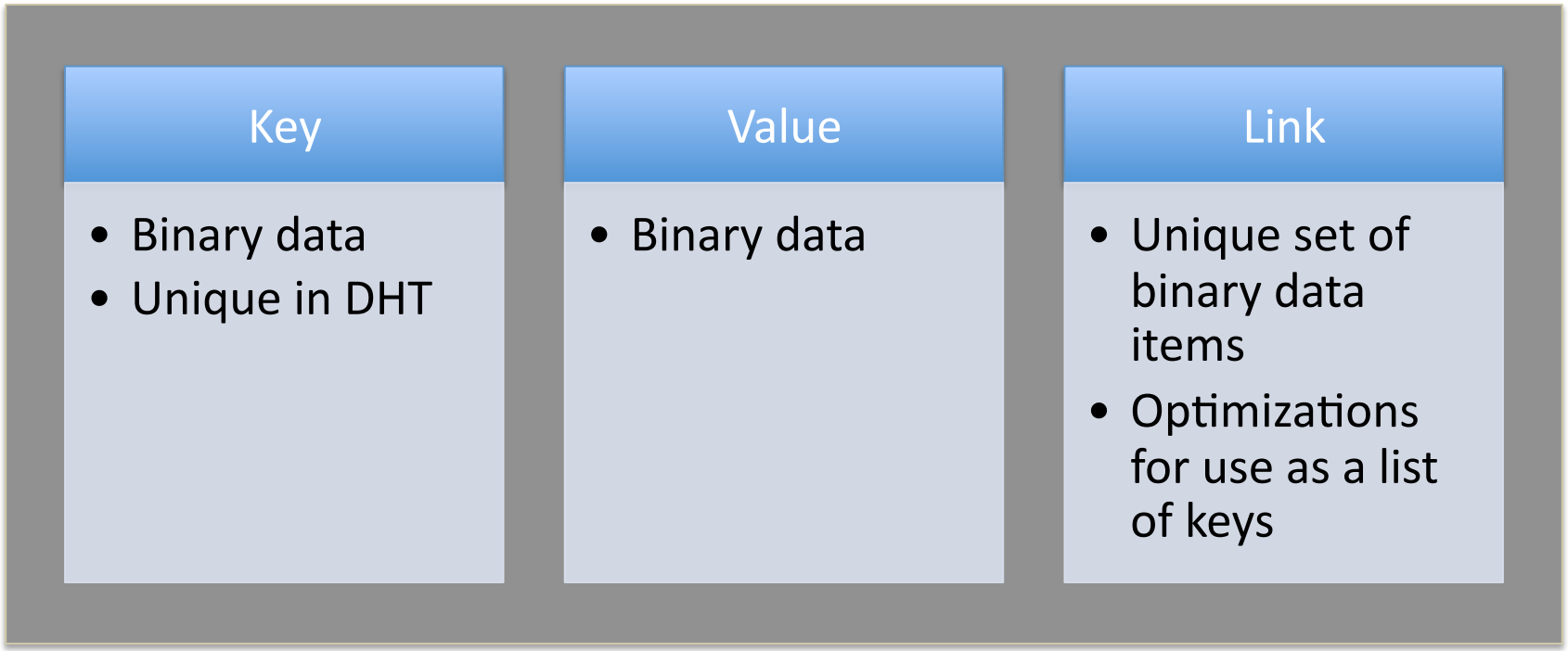


Some data – stored redundantly in 2 places

FTSS: losing a node



DHT tuples

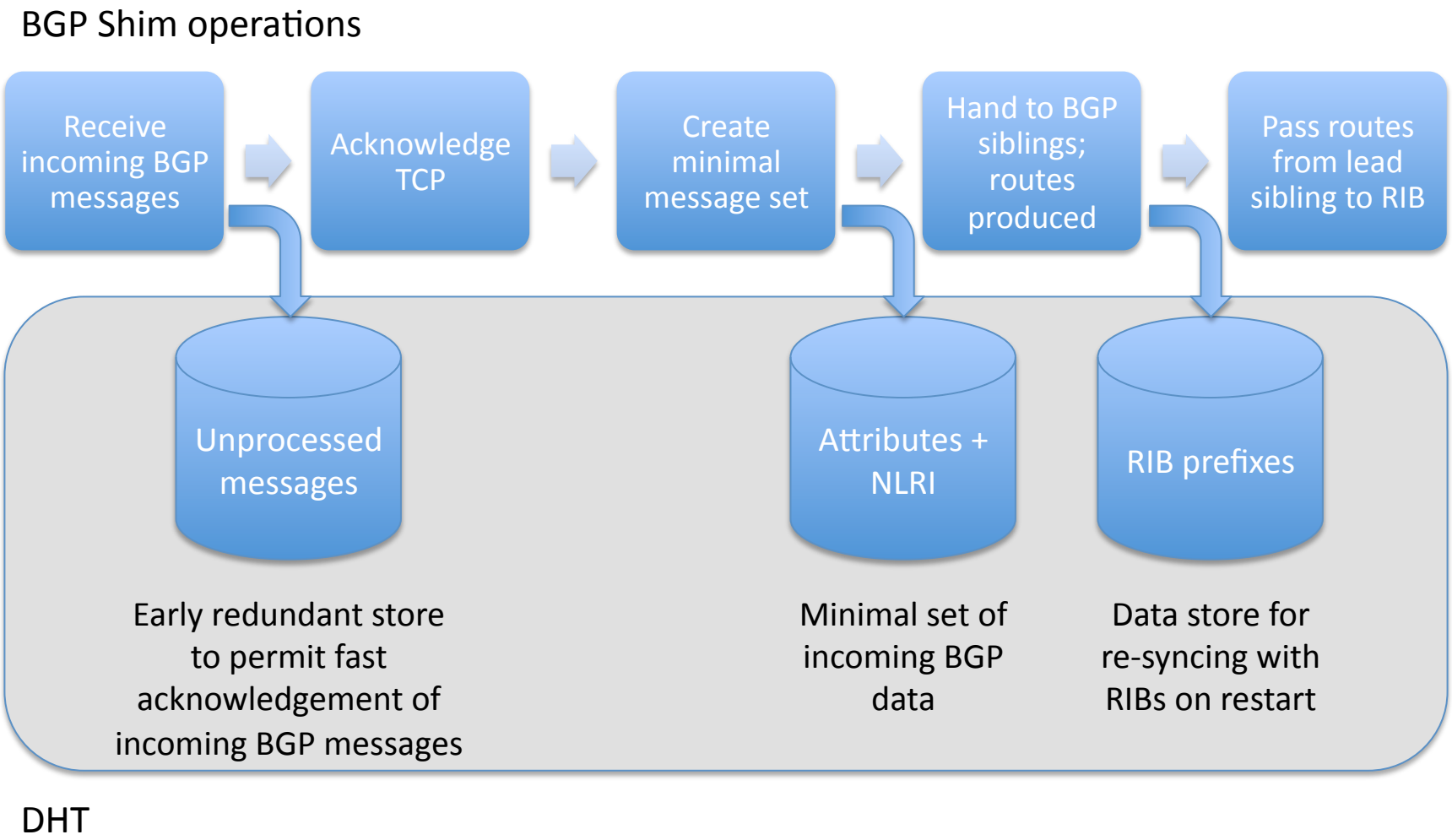


DHT provides optimised routines for:

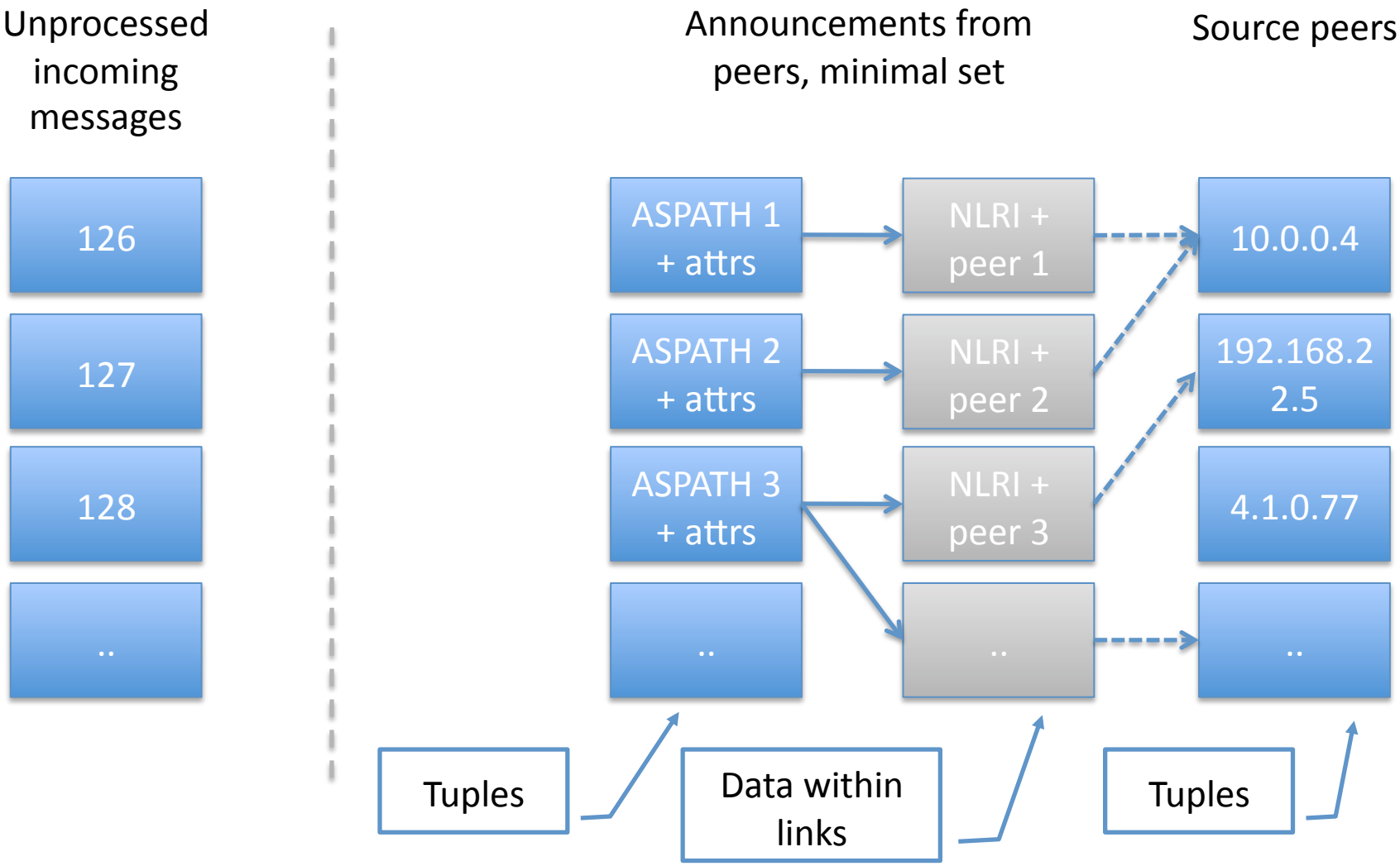
- fast parallel store and deletion of multiple tuples
- fast update of multiple links within a tuple
- Operations directly using the link list for storing related data
- fast parallel recovery of multiple, possibly inter-linked, KVL tuples

Copies of the tuples are stored on multiple nodes for redundancy

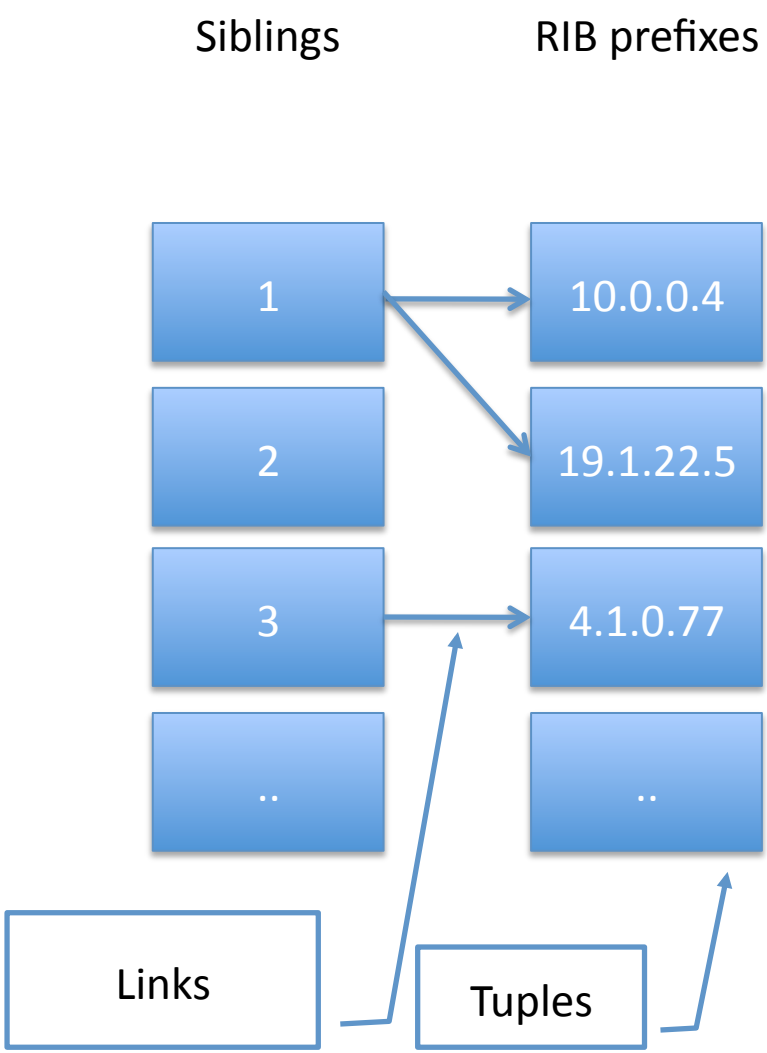
DHT use in BGP processing



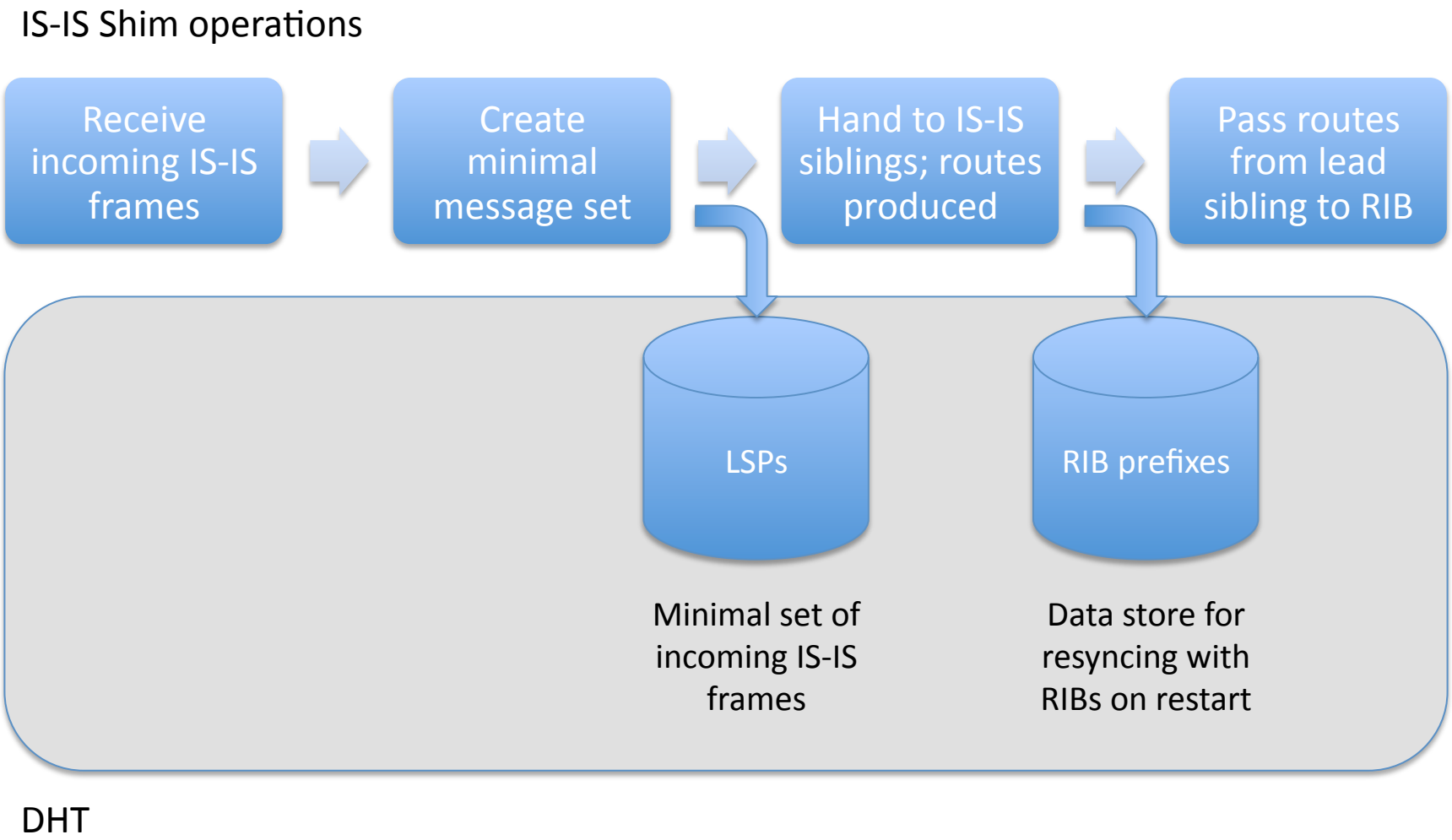
BGP data in DHT (I)



BGP data in DHT (II)



DHT use in IS-IS processing



Multipath IGP/EGP demo

Ludd Project
Demonstration


CISCO