Network-virtualization Nodes that Support Mutually Independent Development and Evolution of Node Components

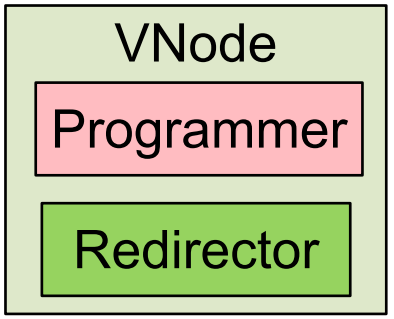
Yasusi Kanada & Kei Shiraishi, Hitachi Ltd.
Akihiro Nakao, University of Tokyo

Introduction: Previous Work

- **We are developing VNode — *deeply-programmable* physical node for network-virtualization platforms — in a collaborative project.**

  - **Deeply-programmable**: packet data processing, such as new L3 protocol processing, can be programmable.

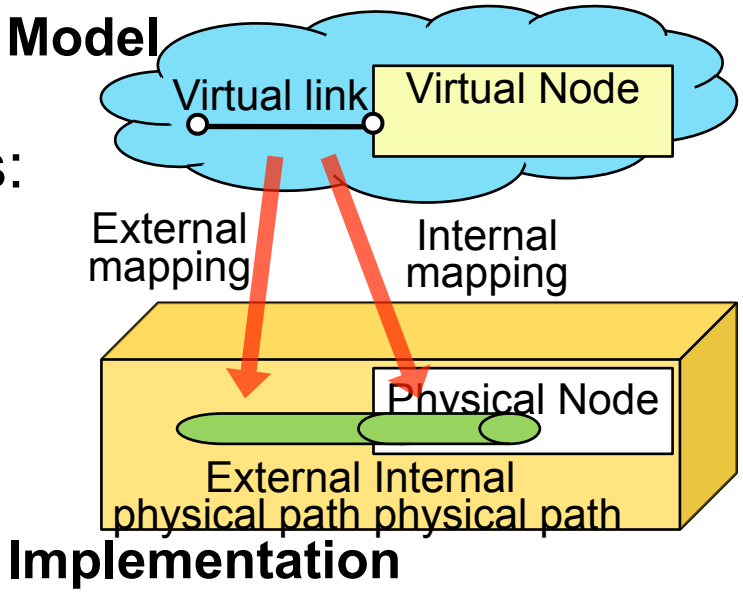
- **A VNode consists of two types of components.**
  - Deeply-programmable computational component called "**programmer**".
  - Networking component called "**redirector**".



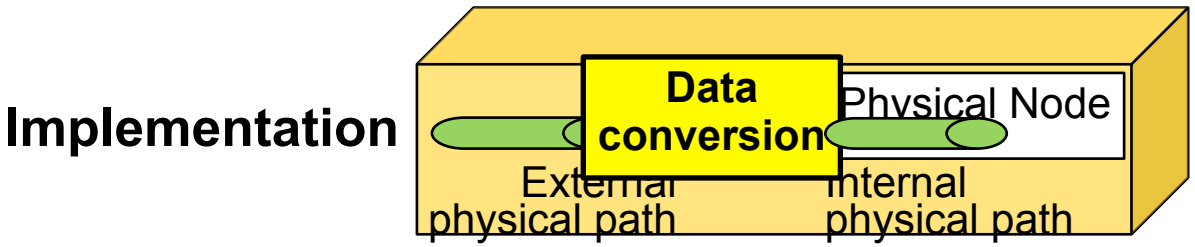
- **Programmer and Redirector are developed and evolve independently in the VNode architecture proposed by Nakao** [Nak 12b].

Introduction: Today's Topic

- **Redirector plays the central role in implementing two functions of VNodes, which enables the independence.**
  - Creation of external and internal model-implementation mappings:
    - Mapping between virtual links to external physical paths
    - Mapping between virtual links to internal physical paths of VNodes

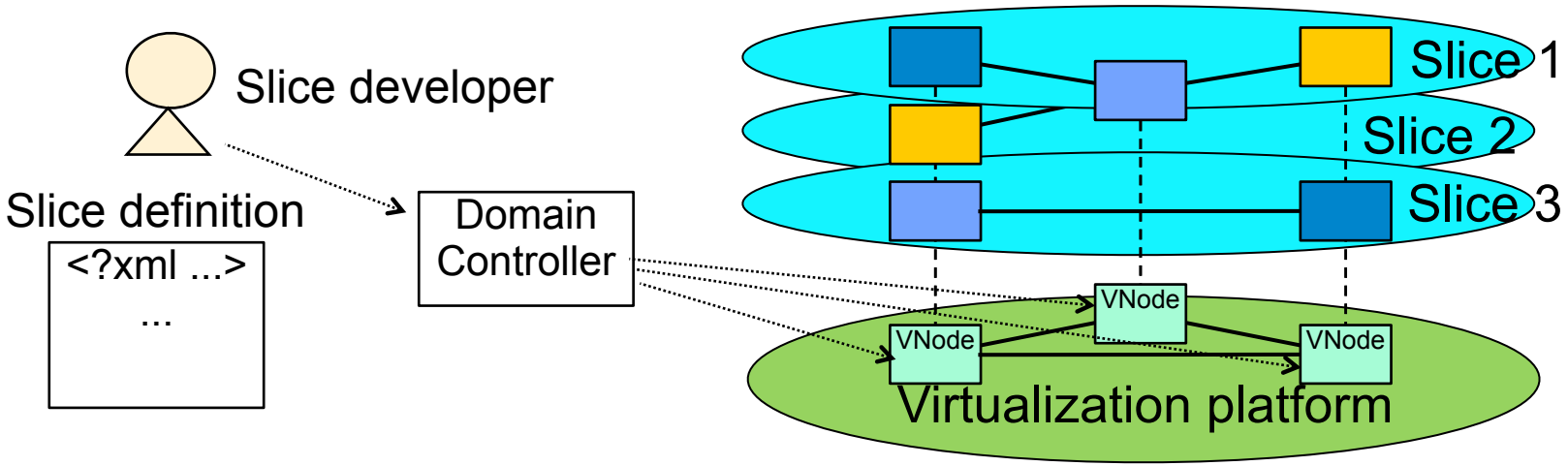


  - High-performance data conversion, which connects the external and internal data formats.



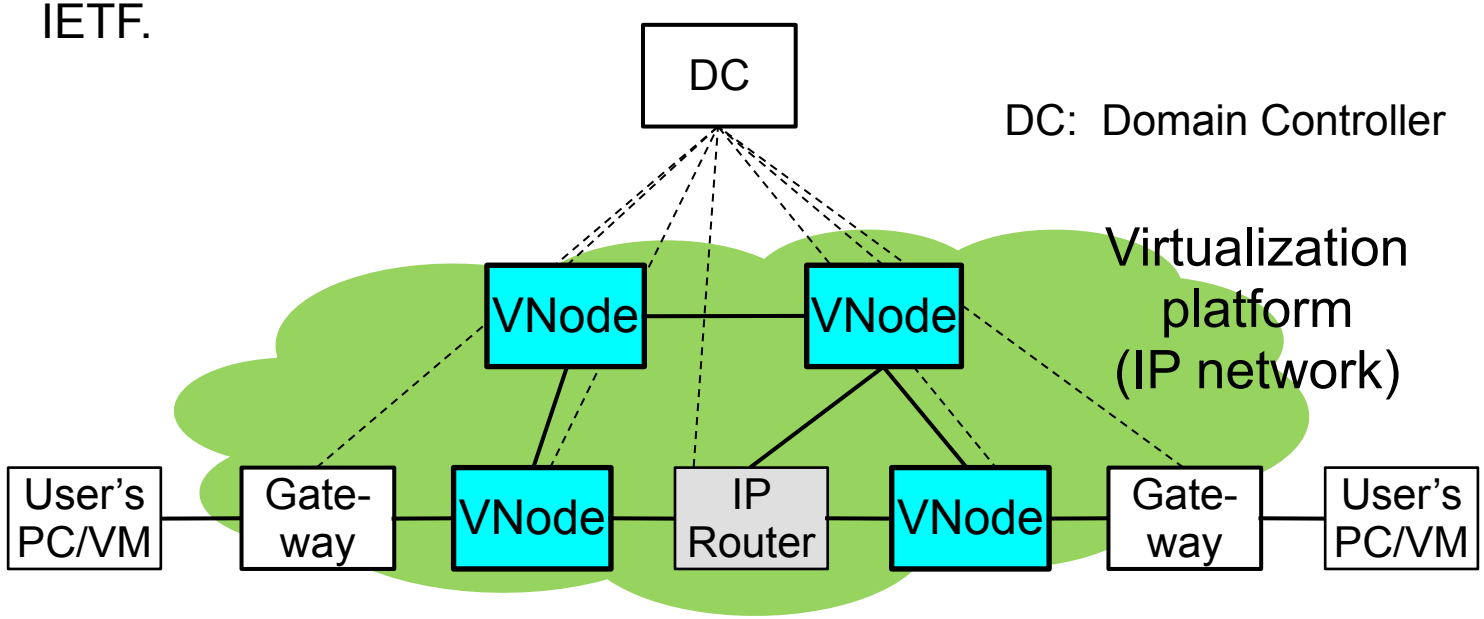
Network Virtualization Architecture and Platform

- **Network Virtualization Architecture and Platform are developed in the collaborative project.**
- **Multiple slices can be created on a physical network in this architecture.**
  - Slices means virtual networks.



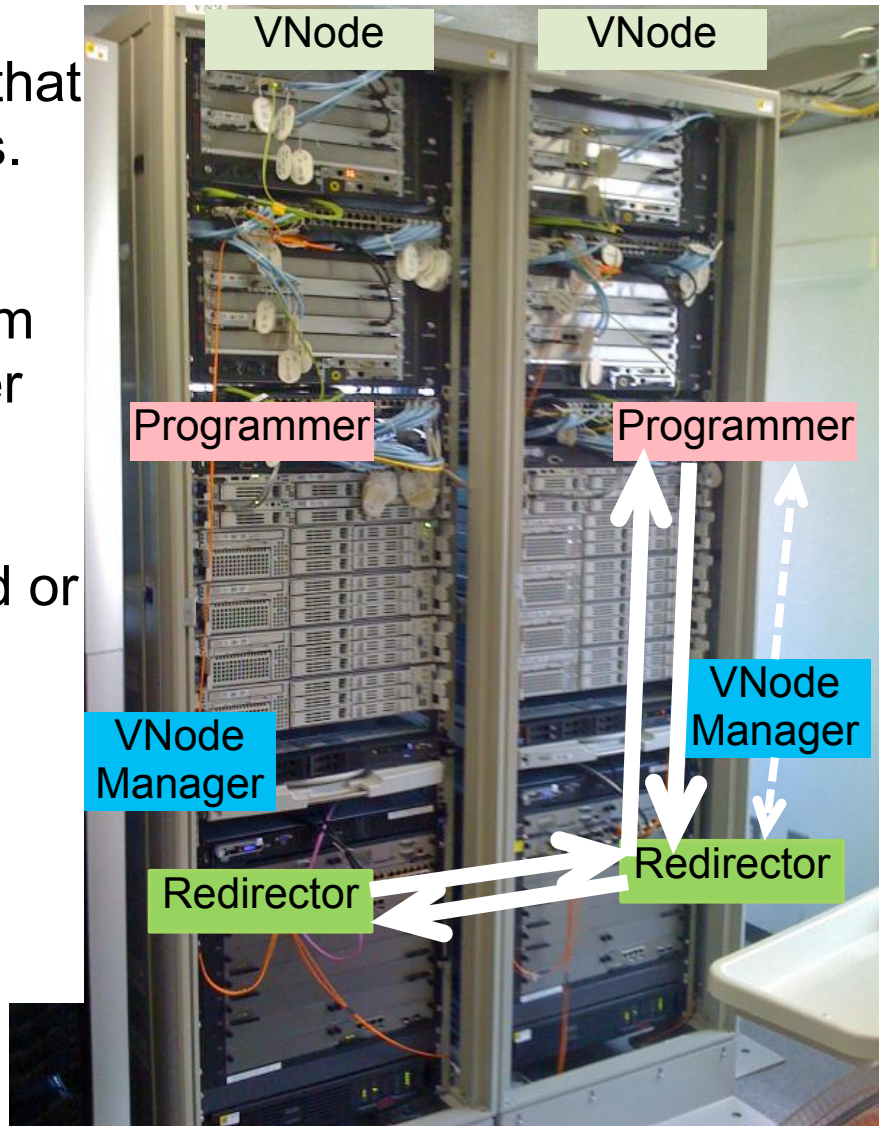
VNode

- **VNode (virtualization node) is a component of the network virtualization platform.**
  - VNode is a physical node.
  - VNode forwards packets on the platform as a router.
  - Slices are implemented as overlay networks on the virtualization platform.
  - VNodes are connected by tunnels using GRE/IP.
    - GRE (Generic Routing Encapsulation) is a protocol standardized by IETF.



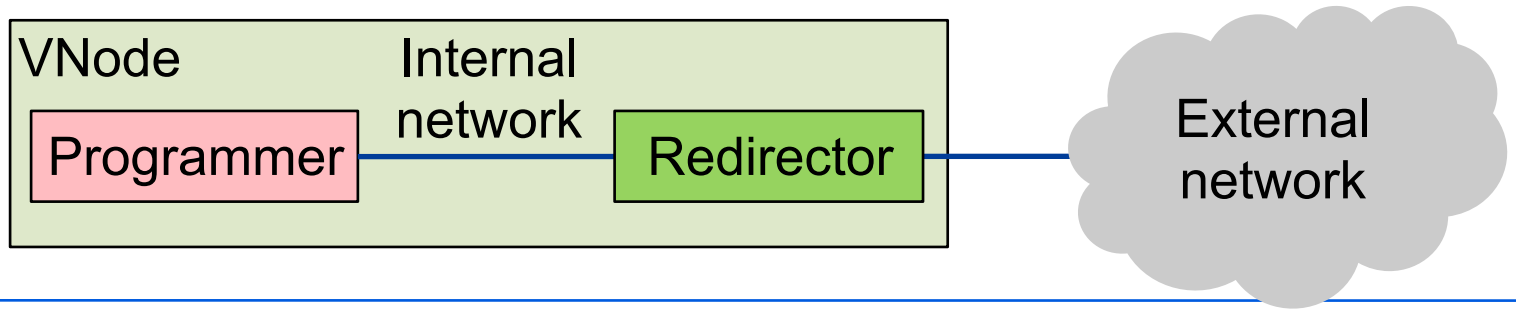
Components of VNode

- **Programmer**
  - is a programmable component that processes packets on the slices.
- **Redirector**
  - forwards (redirects) packets from another VNode to a programmer and forwards packets from a programmer to another VNode.
  - is a component that can forward or route packets on the platform.
- **VNode Manager**
  - is a software component that manages the VNode.

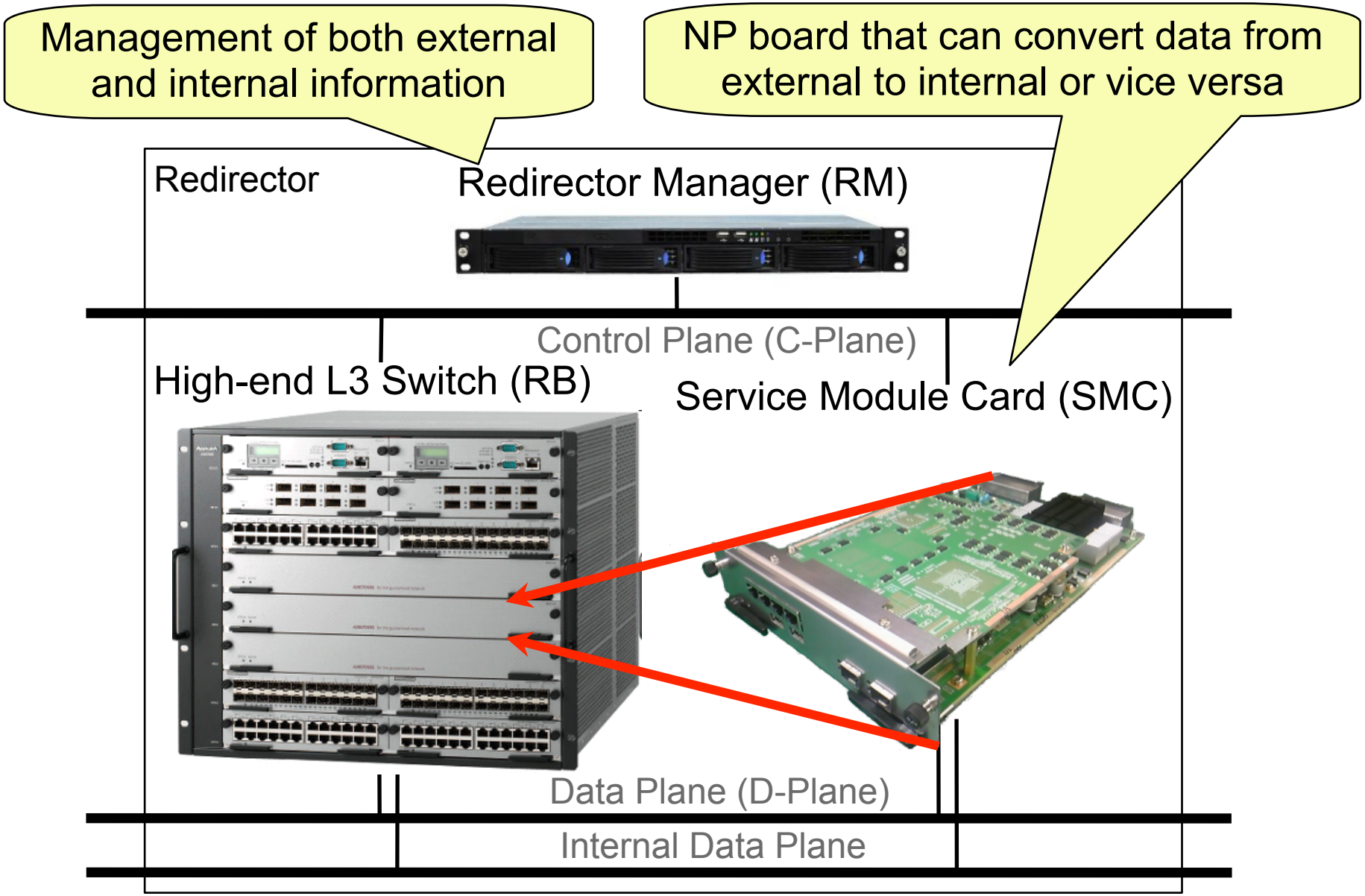


Independently Evolvable VNode Architecture

- **Programmer and Redirector are "separated" in this architecture.**
- **This separation means the protocols of internal and external networks can be independently defined.**
  - Internal network – between Programmer and Redirector.
  - External network – between VNodes.
- **This separation enables various evolution of VNode.**
  - Various types of Programmers can be connected to a Redirector.
  - Various types of Redirectors can be connected to a Programmer.
- **Our challenge is to implement this architecture and to obtain high performance.**



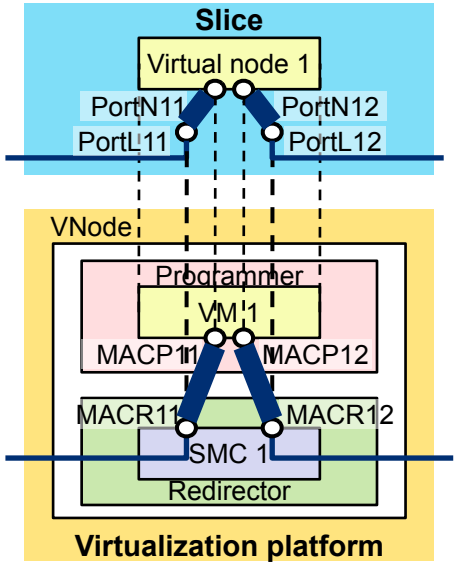
Structure of Redirector



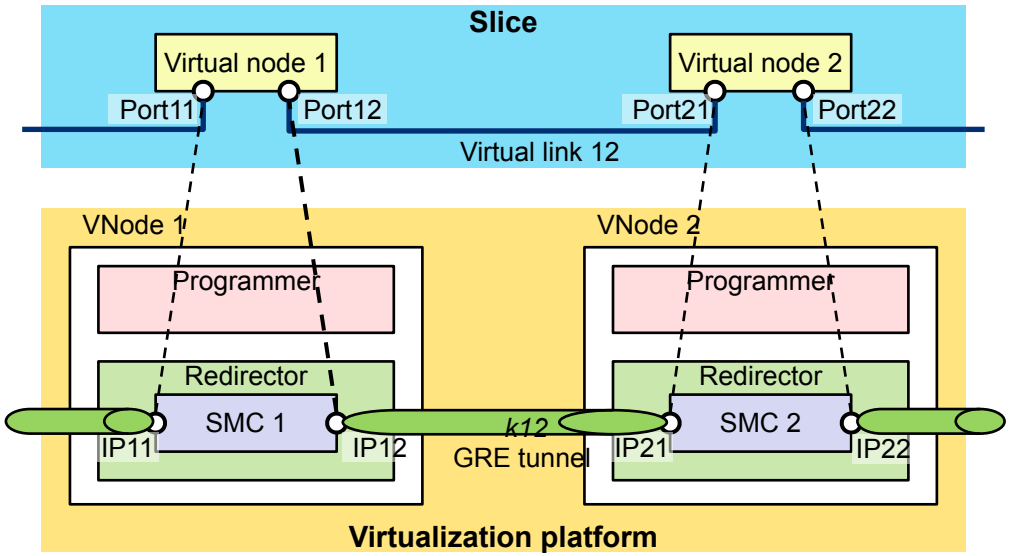
Model-Mappings and Separation

- In a VNode, the model (slice part) is mapped to the internal representation.
- Out of a VNode, the model (slice part) is mapped to the external representation.
- These mappings must be separated for the sake of independent evolution.

Internal mapping

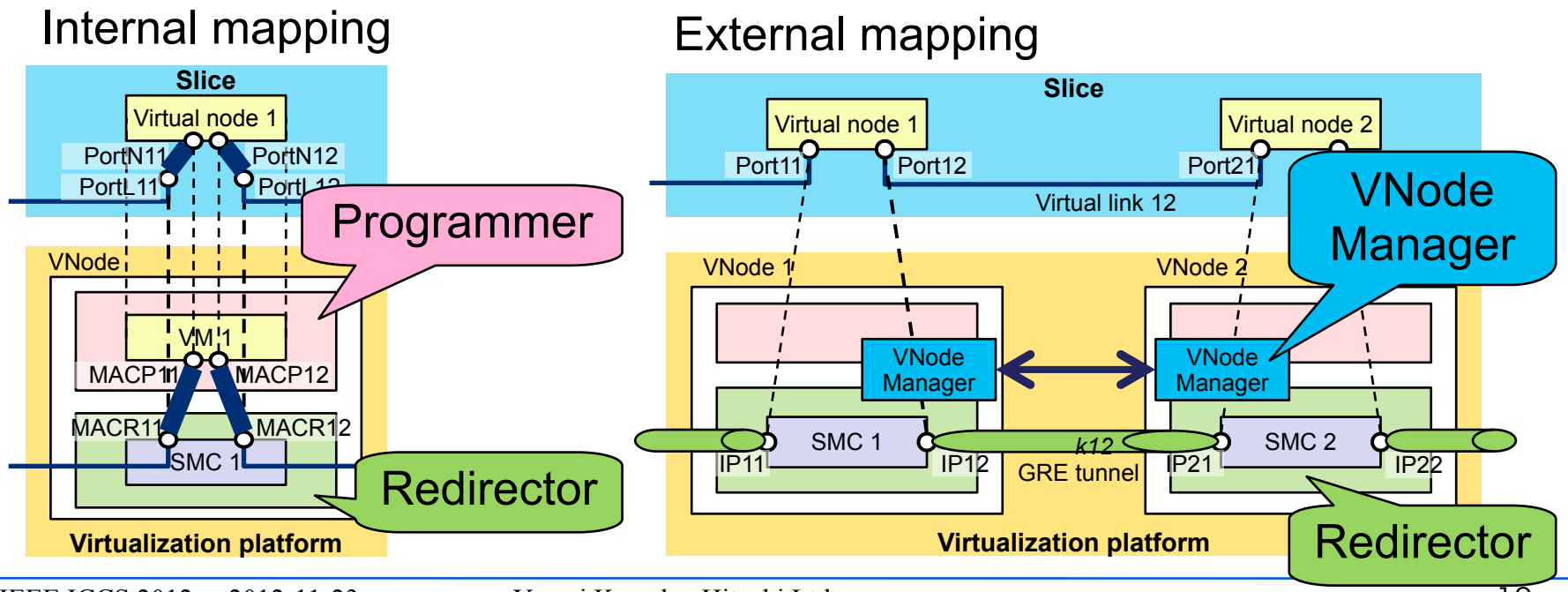


External mapping



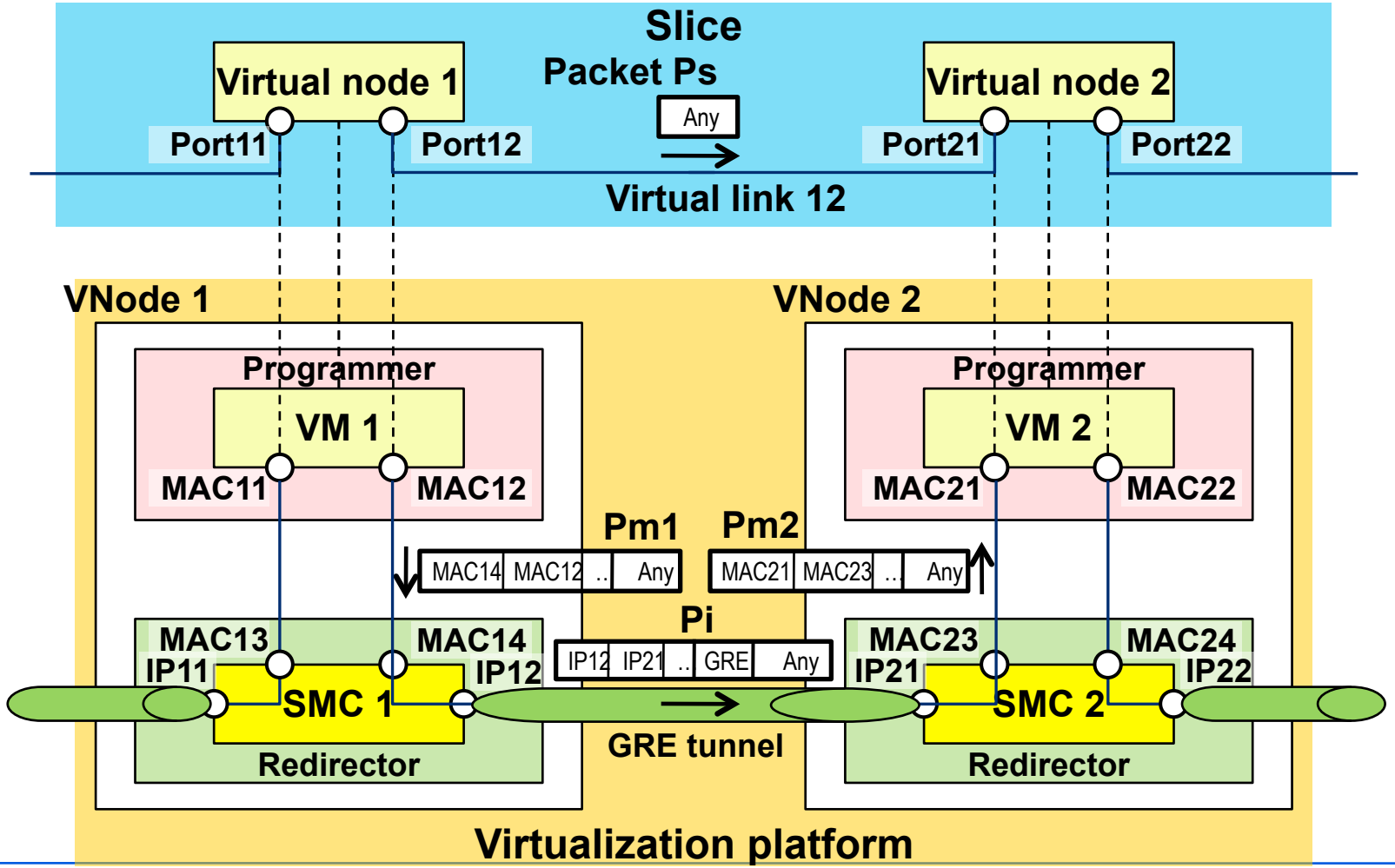
Management of Model Mappings

- **Internal mapping is collaboratively managed by Redirector and Programmer.**
- **External mapping is collaboratively managed by Redirector and VNode Manager.**
- **Therefore, Redirector plays the central role in separating and connecting the model mappings.**



Conversion between External and Internal Representations

- **Redirector (SMC) converts data packets between external and internal representations.**
  - The throughput is 10 Gbps (when the packet size is around 1000B).

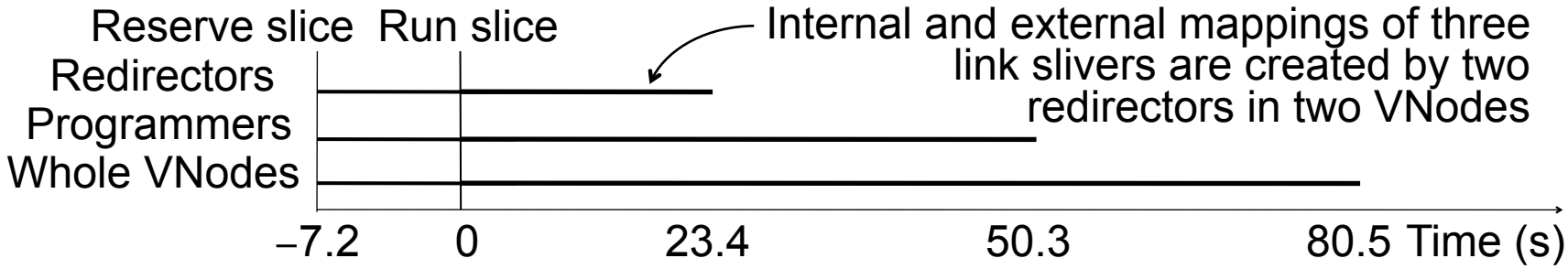


Evaluation: Mapping creation and deletion

- **The creation and deletion of a slice were measured.**
  - Slice structure

- **Performance evaluation of the internal and external mappings**
  - The mappings are concurrently created by the redirector when creating link slivers.

VNode parts	Time for "run" (s)		Time for "shutdown" (s)	
	Average	Std dev	Average	Std dev
Redirector	23.4	1.6	23.5	1.5
Programmer	50.3	3.0	0.007	0.002
Whole VNode	80.5	3.0	47.9	2.3

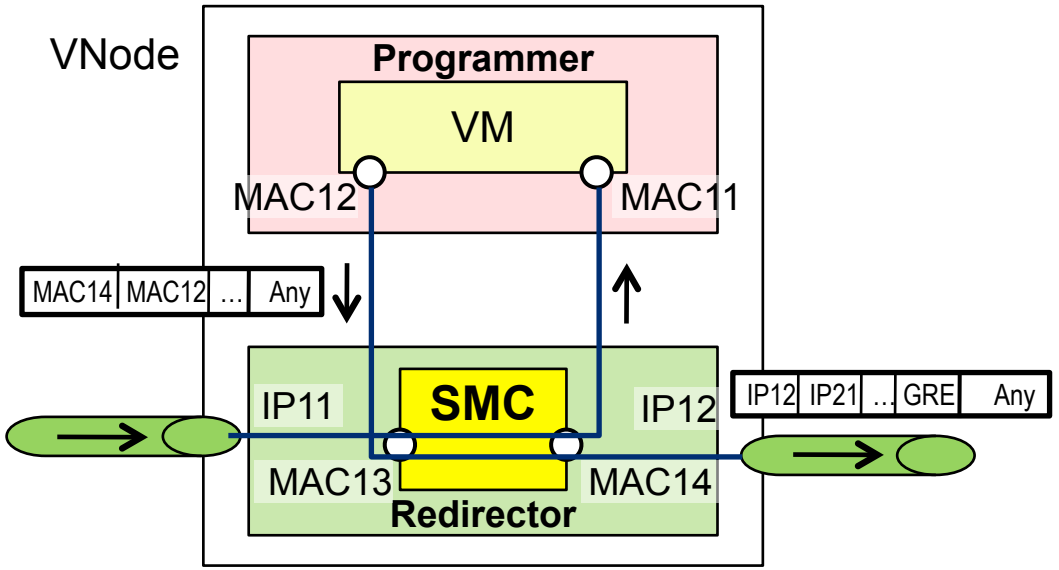


Evaluation: Mapping creation and deletion (cont'd)

- **Observation**
  - The overhead caused by link creation is mostly hidden if the number of virtual links per virtual node is five or less (i.e., in normal conditions).
  - The overhead can still be seen in link deletion because programmers reply to "shutdown" command immediately, but it is less critical.
    - It is less critical because new links can be created while deleting old links.

Evaluation: Data Conversion

- **Data conversion rate is 5-Gbps at maximum when input and output data rates of VNodes are equal.**
  - VNodes are connected by 10-Gbps physical links.
  - The data conversion is performed on the SMC.
  - Each VNode has only one SMC that is used for both direction (internal ⇔ external).



- **In future, this performance can be improved by adding SMCs.**

Conclusion

- **To enable independent evolution of Programmer and Redirector, Redirector is designed ...**
  - Redirector creates two separate mappings between virtual links to external and internal physical-paths.
  - Redirector implements conversion between the external and internal data formats using SMCs.
- **Performance of mappings and data conversion are evaluated, and Redirector mostly satisfies the requirements.**
  - Time for virtual-link creation is usually shorter than other set-up times.
  - The data conversion can be performed at a rate of 5 Gbps.
- **Future work**
  - To enhance mapping and data-conversion methods for various programmers and external networks and for performance improvements.
  - To reduce time for virtual-link deletion.