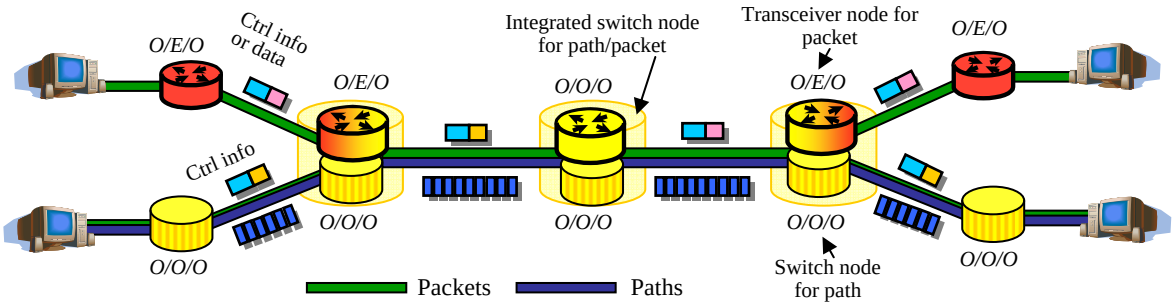
Infrastructure Technologies for Optical Packet & Circuit Integrated Networks

New Generation Network Research Center, NICT

Optical Packet & Circuit Integration toward New Generation Network (NWGN)

Novel Architecture that WDM-based optical packet and circuit(=path) switched networks are integrated based on common physical resources (Wavelength, Fiber, etc) and unified control mechanisms

- Hop-by-hop optical packet switching can provide a bandwidth-sharing and best-effort service.
- End-to-end optical path transmissions can provide an occupied bandwidth and end-to-end QoS guaranteed service at large capacity, low-cost and high-security.



Feature and advantage of Optical Packet & Circuit Integrated Network

・Providing diverse service such as best-effort or QoS guaranteed service by request from users with reduced cost
- Optical packet and optical path are switched on the same network infrastructure-

・Low power consumption
- Optical switching can decrease O/E/O conversion and increase throughput without extra power consumption because of transparency for various format, compared with electronic router -
- Optical circuit switching can decrease extra packet processing because of established connection -

・Simplification of network by unifying control mechanisms
- Control mechanisms of optical packet switching and optical circuit switching are unified by multiplexing path-control-packet on packet switched links (decreasing of extra interface) -

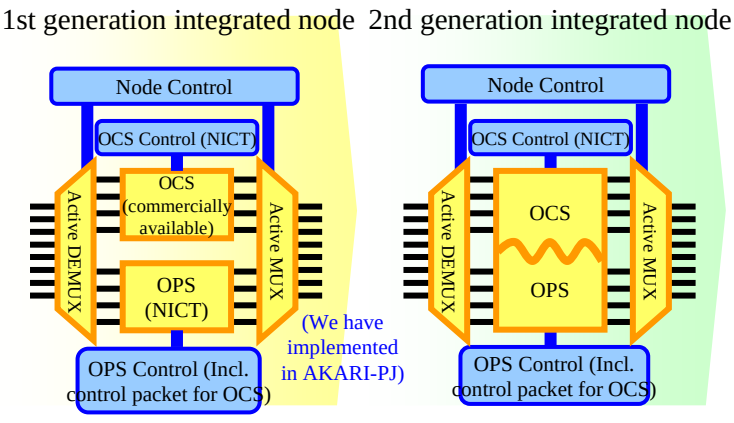
・Supporting new, unexpected or urgent service by sharing physical resources on backbone network
- Wavelength resources are dynamically assigned to optical path/packet links -

Development of Optical Packet & Circuit Integrated Node

Consisted of optical packet switch (=OPS), optical circuit switch (=OCS), control system (routing, signaling, resource control, etc)

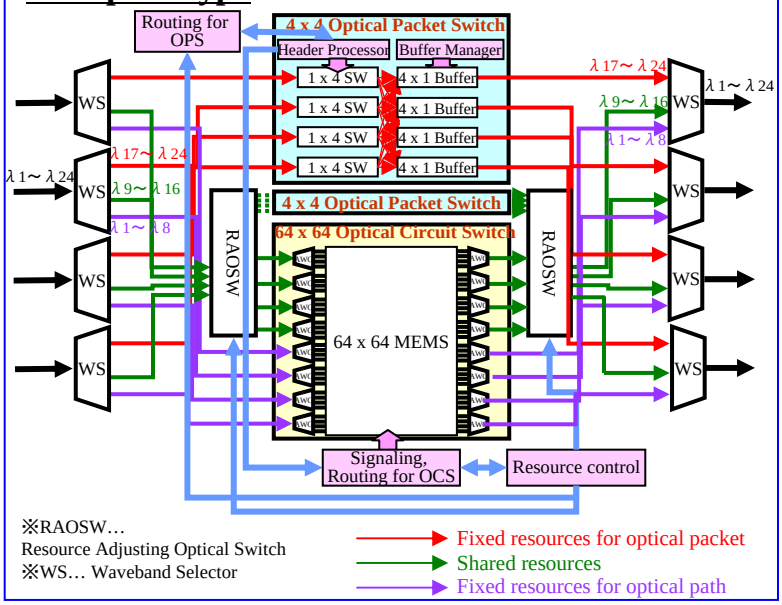
(1st Generation) Integrated node prototype designed in *AKARI architecture design project*. Combination of OPS and OCS.

(2nd Generation) Integration of OPS and OCS by common merged hardware to achieve more flexibility, power-saving, and space-saving.

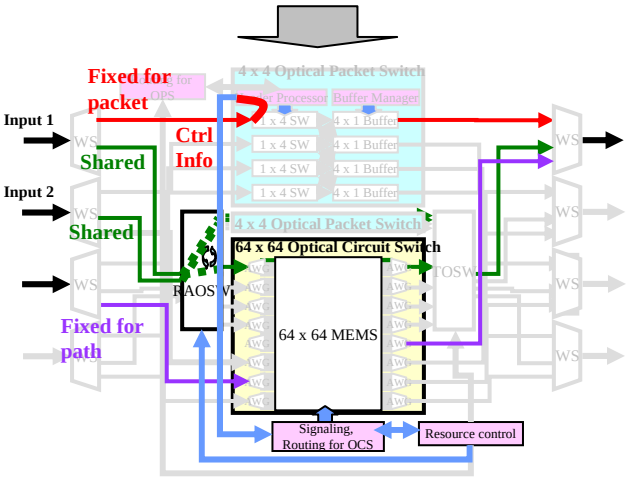


*AKARI architecture design project → http://akari-project.nict.go.jp/

1st generation optical packet & circuit integrated node prototype

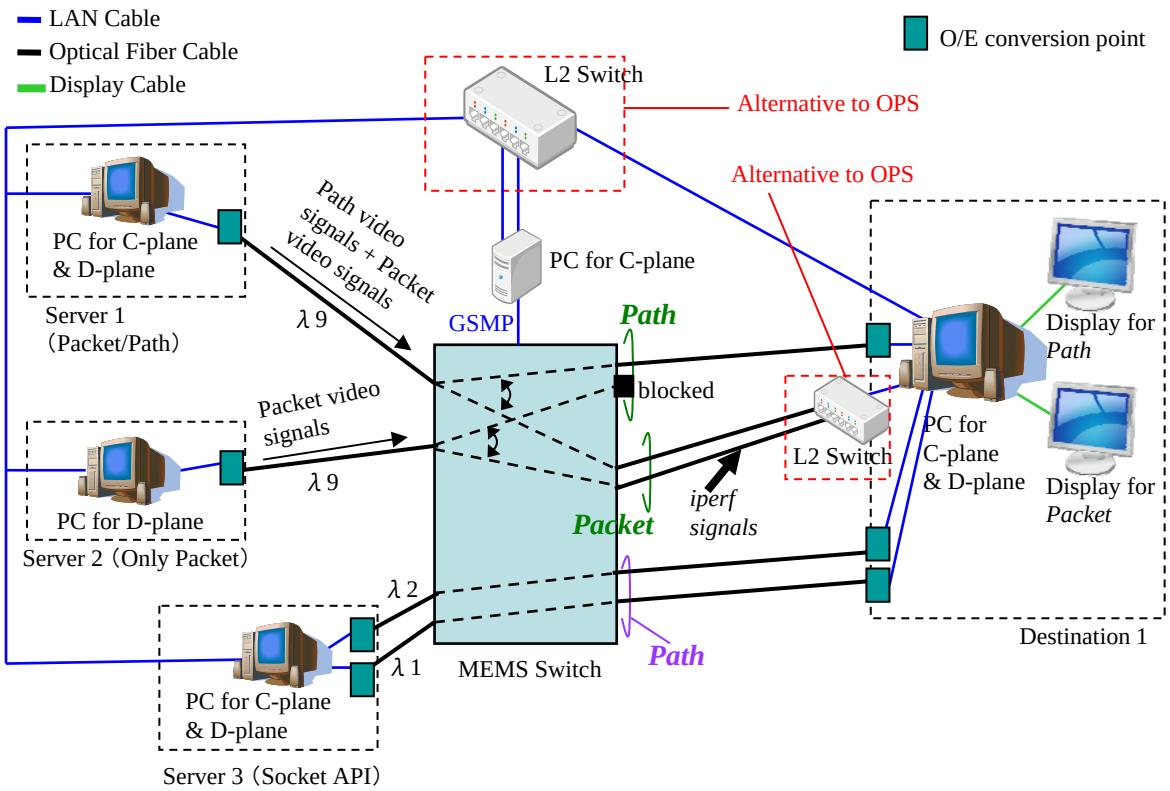


Partial Implementation of integrated node



* In this demonstration, RAOSW and MEMS are aggregated into one MEMS.

Brief schematic view of the experimental demonstration of video transmissions



*Packet switching: The bandwidth is shared by video and *iperf* traffic flowed from the two servers.
→ The qualities of two video signals drastically degrade.
*Path transmission: *Server 1 occupies the bandwidth* when Server 1 switches the packet transmissions to the path transmissions. → High quality video can be seen on the Display for *Path*.
In Server 3, the Socket API automatically selects and sets up a lightpath (i.e. wavelength) depending on each application or content.

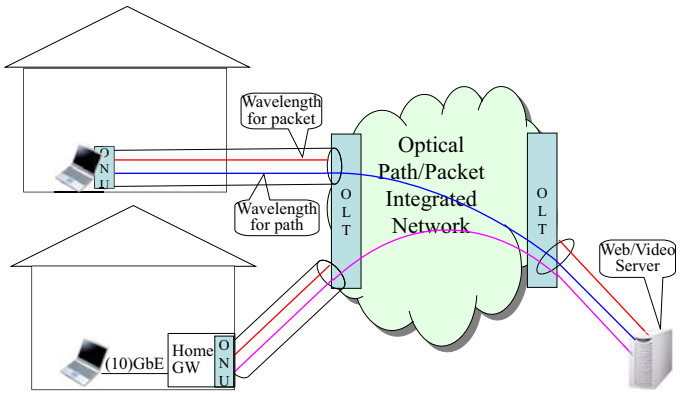
Contact: Takaya Miyazawa (takaya@nict.go.jp), Hideaki Furukawa (furukawa@nict.go.jp)

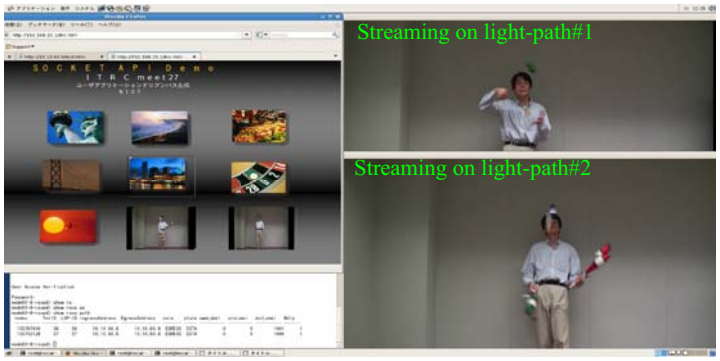
Socket API Assigning Optical Paths to L4 Connections for Optical Path/Packet Integrated Networks

(Presentation #5-2 2010/06/11 17:00-)

Objectives:
- Implementing an optical-path/packet-integrated network system
- Providing end users with end-to-end QoS-guaranteed optical paths

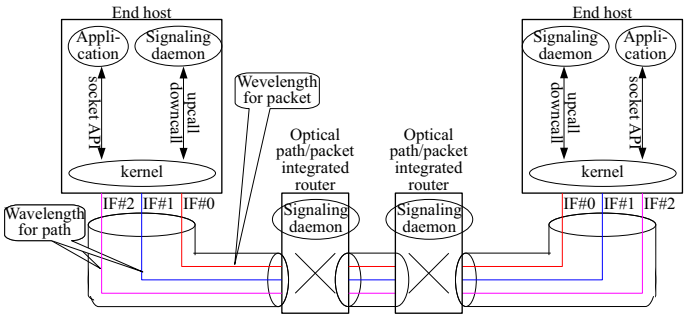
Usage of End-to-end Optical Paths: PC/HomeGateway Directly Use Wavelengths





An Optical Path/Packet Integrated System from the Viewpoint of Applications

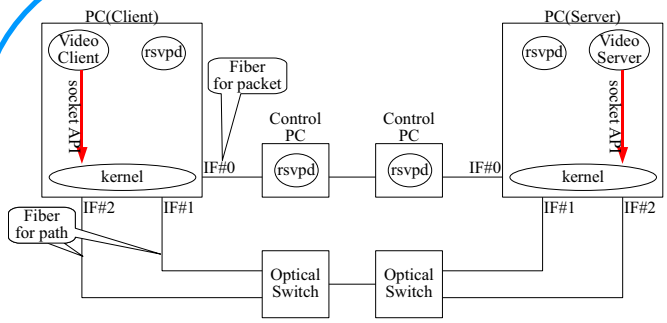
- Assigns an optical path(wavelength path) to a socket of an application



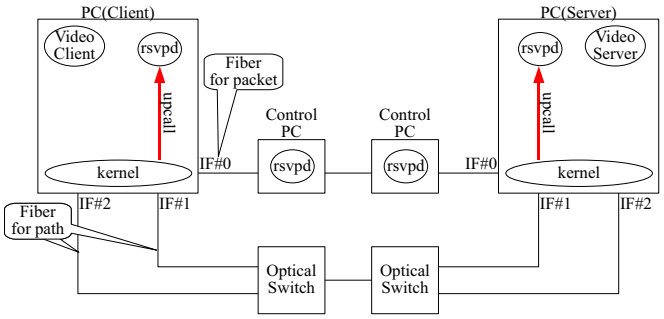
An Socket API (setsockopt) Programing Example

```
struct wl_resv /* wavelength reservation */
{
#def ne WL_RESV_FLAG_SEND (1 << 0)
#def ne WL_RESV_FLAG_RECV (1 << 1)
      int wlr_f ags;
      int wlr_lambda_no;
};
struct wl_resv resv;
memset (&resv, 0, sizeof (resv));
resv.wlr_f ags = WL_RESV_FLAG_SEND | WL_RESV_FLAG_RECV;
resv.wlr_labmda_no = 1;
setsockopt (s, IPPROTO_IP, WL_RESV_ADD, &resv, sizeof (resv));
/* bind, connect or accept/listen */
```

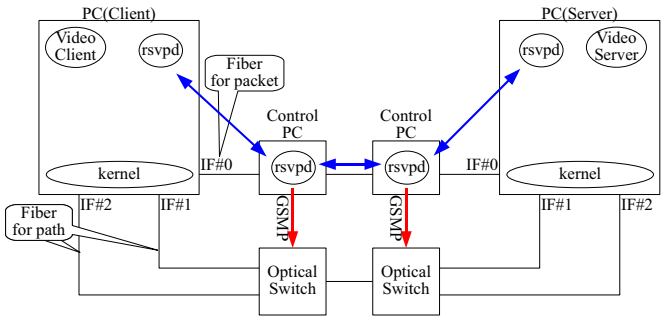
The Procedure of Establishing a Light Path (implemented on Linux)



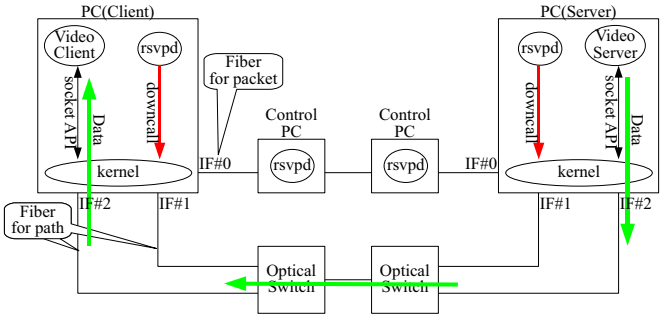
1. Applications on PCs(client/server) create a TCP connection or a UDP f ow, and make a request for an optical path by socket API (setsockopt)



2. The kernel receives the request for establishing an optical path, and forwards the request to an rsvpd daemon by upcall



3. The rsvpd daemons start RSVP-TE signaling and establish an optical path



4. The rsvpd daemon on server redirect the data packet from the ordinary interface to another interface(IF#2 in the f g.) that is dedicated to optical paths by means of setting the kernel routing table

FUJIKAWA Kenji <hudikaha@nict.go.jp>

National Institute of Information and Communications Technology

AKARI